Interpreting, Evaluating, and Controlling Features Learned and Leveraged by Foundation Models

by

Adam Davies

Dissertation

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois Urbana-Champaign, 2026

Urbana, Illinois

Doctoral Committee:

Professor Julia Hockenmaier, Chair
Professor ChengXiang Zhai, Director of Research
Assistant Professor Han Zhao
Assistant Professor Aaron Mueller, Boston University

Abstract

With the rise of modern foundation models that are capable of using language, creating imagery, and acting in digital environments like humans do, there has been substantial debate regarding how much these models genuinely "understand", how "generally intelligent" they really are, or how their internal "cognition" may or may not be similar to that of humans. Such debates are compounded by the fact that these models are generally perceived as "black boxes": by default, the internal representations and processing that models learn from data are not human-interpretable; so we generally study relevant notions of cognition like understanding or intelligence via benchmark performance. However, this paradigm is seriously challenged by concerns such as contamination, safety, and out-of-distribution generalization – how do we know when/if models are simply regurgitating memorized answers, or whether we can trust them to generalize appropriately to complex, real-world scenarios (particularly in safety-critical contexts)?

This dissertation aims to address such concerns by interpreting, controlling, and evaluating the latent features and internal mechanisms learned and leveraged by foundation models, working to define key desiderata, theoretical frameworks, and empirical methodologies appropriate to the task of understanding machine cognition. I begin my empirical investigation by studying what latent feature representations are learned by foundation models, how they are encoded, and how models make use of these representations in the course of performing various tasks. Next, I examine how models generalize to new tasks and data distributions by controlling and evaluating their use of task-causal versus spurious features, introducing new pre- and post-training methods to increase robustness and reduce bias by enforcing invariance with respect to spurious features and increasing reliance on causal features, and creating challenging new out-of-distribution (OOD) benchmarks to test the extent to which models rely on causal vs. spurious features OOD. Finally, I explore how interpreting models' representations of latent features and internal mechanisms can predict and explain when and why models fail to generalize OOD, and conclude by highlighting key directions for future research to better understand model generalization by further cognitive interpretation and analysis of model internals.

Acknowledgements

First and foremost, I owe a sincere debt of gratitude to my co-advisors, ChengXiang ("Cheng") Zhai and Julia Hockenmaier, for all their guidance and support over these years. Cheng has stuck with me through all the paper rejections, dead projects, failed grant proposals, and every other manner of rejection a grad student is likely to face; but through it all, he has always remained positive, never judged me personally, and found a way to spin each piece of news in a positive light. He also never fails to spot a promising potential new research direction, and his creativity has been a guiding light throughout my studies. I hope to emulate Cheng's kindness, creativity, and optimistic spirit with my own future advisees. Likewise, Julia has always been there when I was confused, lost, or discouraged, and has never failed to offer sage counsel in times of need. It can be difficult to balance research or career guidance with personal advice, as the two can so often conflict; but I have always been able to completely trust Julia to take my needs, and those of her other students, fully into account when providing advice or making decisions – which has been truly invaluable as a PhD student navigating the manifold twists and turns of a field so in flux as AI/NLP. Academically, Julia has a remarkable capacity to rigorously deconstruct any new research direction or result with laser-focused precision, discerning hype or "fluff" from real progress – again, truly essential in this period – and always finding something that could be improved (even if only in future work)! I have learned a tremendous amount from her: both on the hard, scientific side of what constitutes high-quality, rigorous research; and on the human side of how to foster compassionate, productive collaborations.

I also want to express my deepest gratitude to Francesco Pinto, my "informal advisor" and longtime collaborator. Francesco taught me what it actually is to be a PhD student, how to take failure and rejection in stride without giving up on myself, and how to turn a "good idea on paper" all the way to being an *actually good paper*. He has also introduced me to many of my closest collaborators and brought me onto many of my most productive, interesting projects; and I have learned volumes for having experienced, in so many cases, his ability to take a project experiencing a difficult moment, problematic result, or unfortunately-timed novelty-killing preprint, and find a way to put it all back together even better than before. All in all, I am a far better researcher, with a much brighter future (and many fewer gray hairs), for having worked with and learned from Francesco over these years.

There are so many other people who have also supported me personally throughout this long, hard process. I am so grateful for the love, compassion, and unconditional moral support from my family: Kathy, Mark, and Joseph Davies, and Celeste Burnette. I know it hasn't always been clear what I've been working on, but your kind words (and the occasional

choice words for a problematic Reviewer 2) really have helped carry me through all the hardest moments. Likewise, my close friends have offered such valuable companionship, understanding, and many a good laugh when I really needed it – so, a heartfelt thank you to Linda Derhak, Francesco Pinto, Cecilia Molina, Kevin Ros, Marc E. Canby, Elisa Nguyen, Martin Gubri, and Amogh Mannekote. Without a sense of community, it's impossible not to burn out with the workload of a PhD student, and I really couldn't have made it without you.

I also want to express my deep appreciation to the many wonderful collaborators I've had who have contributed so much to the work in this dissertation. Thank you to Ashkan Khakzar, Jianhao Yuan, Arshia Hemmat, Tom A. Lamb, Philip Torr, Alasdair Paren, Marc E. Canby, Chirag Rastogi, Jize Jiang, Paul Smolensky, Mattia Opper, Roland Fernandez, and Jianfeng Gao – and of course, to my exceptional advisors ChengXiang Zhai, Julia Hockenmaier, and (informally) Francesco Pinto, as well as all my other collaborators over the years with whom I worked on projects not included in this dissertation. There are also several other individuals to whom I am indebted for their excellent advice in developing the ideas and research presented in this dissertation: thank you to Atticus Geiger, Aaron Mueller, Han Zhao, Adel Bibi, Prashant Jayannavar, Fabio Pizzati, Arindam Banerjee, and Bo Li. Likewise, I sincerely appreciate Heng Ji, Alan B. Craig, H. Chad Lane, and Philip Torr for their career advice and help in securing funding to support my work toward this dissertation. Finally, a special thanks to my PhD thesis committee – Julia Hockenmaier, ChengXiang Zhai, Han Zhao, and Aaron Mueller – who have all contributed invaluable ideas, feedback, and advice that have helped me put together a far better dissertation than I ever could have otherwise.

Funding and Compute Resources I am very lucky in that my work toward this dissertation has been supported by a variety of funding and compute resources, and I am very grateful to each of them:

- Thank you to Microsoft Research for supporting me as a PhD Research Intern in the Deep Learning Group. My work in Chapter 8 would not have been possible without the support of my managers Paul Smolensky, Roland Fernandez, and Jianfeng Gao, as well as the compute resources made available by Microsoft for this work.

- My work has been supported in part by the National Science Foundation and the Institute of Education Sciences, U.S. Department of Education, through Award #2229612 (National AI Institute for Inclusive Intelligent Technologies for Education). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of National Science Foundation or the U.S. Department of Education.

- My work has been supported in part by U.S. DARPA AIDA Program No. FA8750-18-2-0014 . The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of DARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

- My work has utilized the Delta advanced computing and data resource which is supported by the National Science Foundation (award OAC 2005572) and the State of Illinois. Delta is a joint effort of the University of Illinois Urbana-Champaign and its National Center for Supercomputing Applications.

- My work has utilized resources supported by the National Science Foundation's Major Research Instrumentation program, grant #1725729, as well as the University of Illinois at Urbana-Champaign. These resources are made available through HAL (Kindratenko et al., 2020).

Contents

Chapter 1 Introduction

1.1 Motivation

How can we understand, interpret, and explain the behavior of complex, intelligent biological systems like humans, chimpanzees, or dolphins – or even "artificially" intelligent systems like large language models (LLMs)? This question lies at the heart of cognitive science, and different answers have defined entire paradigms in member disciplines such as psychology, neuroscience, and linguistics. For instance, in the 1950s and early 1960s, the dominant paradigm in academic psychology was behaviorism (Mandler, 2002), resting on the fundamental assumption that human behavior can be fully understood and explained in terms of stimulus-and-response mechanisms (Skinner, 1957; Chomsky, 1980), with no need to consider humans' internal mental states, representations, or processing. However, behaviorism failed to account for many key facets of human psychology, including language acquisition (Chomsky, 1980, 2002), concept learning and problem-solving (Bruner, 2017), and working memory (Miller, 1956), leading to the so-called "cognitive revolution" of the 1960s that birthed the modern cognitive sciences built around studying human cognition on the basis of mental representation and information processing (Sperry, 1993; Miller, 2003).

Most contemporary deep learning research also rests upon the behaviorist assumption – i.e., that it is only necessary to study models' behaviors (and not their internal representations, processing, etc.) in order to understand and explain their capabilities, limitations, benefits, and risks (Raji et al., 2021; Mahowald et al., 2024). Indeed, it has long been said that such models are "black boxes": while we may design their neural architectures, develop learning algorithms used to train them, curate their training data, etc., we do not know precisely what these models learn from the data used to train them, nor how they internally carry out various tasks we may provide them with (Adadi and Berrada, 2018; Lipton, 2018). However, recent developments in mechanistic interpretability have challenged the "black box" paradigm, opening the door for a potential *cognitive revolution in deep learning*. The goal of such work is not (only) to explain specific model *outputs*, but more broadly to interpret the latent representations and internal algorithms that are learned from pre- and post-training, and to explain how these representations and mechanisms result in observed model behaviors (see Chapter 2).

This dissertation will make the case that this "cognitive revolution" enables the direct empirical study of longstanding, foundational questions about the nature of neural networks. For instance, many argue that foundation models do not learn to genuinely *understand* their training data, functioning instead via memorization, shortcut learning, and shallow pattern-matching (Raji et al., 2021; Yuan et al., 2024; Shojaee et al., 2025; Fodor, 2025) –

e.g., perhaps most famously, that they are so-called *stochastic parrots* (Bender et al., 2021) – but these arguments are complicated by the fact that such models present nontrivial capabilities in generalizing to novel tasks and data distributions (Mahowald et al., 2024; Wang et al., 2025; Chu et al., 2025). And yet, such generalization can be brittle, with models often showing a degree of generalization in simple contexts while still failing to exhibit human-level generalization to sufficiently complex tasks or substantially out-of-distribution (OOD) samples (Yuan et al., 2023; Wang et al., 2023a; Yang et al., 2023a,b; Smolensky et al., 2025; Huang et al., 2025b; Alazraki et al., 2025). How can we explain such behavior? What do foundation models actually learn from their training data, and how is it applied in practice? And how can we improve the pre- and post-training of foundation models to teach them more generalizable, human-like "understanding"?

This dissertation works to help answer these questions in three thrusts: *mechanistic interpretability*, *transfer learning*, and *integrating mechanistic interpretability for transfer learning*.

1.2 Dissertation Structure and Contributions

- **Part I** contains my work on **mechanistic interpretability**, studying *what latent feature representations are learned* by foundation models, how they are *encoded*, and how models *make use* of these representations in the course of performing various tasks.

  - **Chapter 2** is based on *The Cognitive Revolution in Interpretability: From Explaining Behavior to Interpreting Representations and Algorithms* (Davies and Khakzar, 2024)

    We survey the field of mechanistic interpretability, arguing that it heralds a *cognitive revolution* in deep learning akin to that which occurred in 1960s psychology and birthed the modern cognitive sciences. We propose a taxonomy of mechanistic interpretability methods according to Marr's levels of analysis (Marr, 1982), highlighting key challenges, underlying assumptions, and divergent goals and objects of study across the mechanistic interpretability literature.

  - **Chapter 3** is based on *Competence-Based Analysis of Language Models* (Davies et al., 2024)

    We introduce a framework for analyzing models' *competence* on a given task as the extent to which the features they leverage to perform the task (as measured via *causal probing*) match the causal features of the task's underlying data-generating process. We also develop the first nonlinear counterfactual

method for causal probing, *gradient-based interventions* (GBIs), which we use to measure models' competence across lexical inference tasks.

- **Chapter 4** is based on *How Reliable are Causal Probing Interventions?* (Canby*, Davies*, et al., 2025)

  We define a framework to evaluate the *reliability* of causal probing methods in terms of their *completeness* (how thoroughly the representation of the target feature has been transformed) and *selectivity* (how little non-targeted properties have been impacted). Across a range of LLMs and causal probing methods, we find that: (1) all methods showed a clear tradeoff between completeness and selectivity; (2) more complete and reliable methods had a greater impact on LLM behavior; and (3) nonlinear interventions (such as the GBIs introduced in Chapter 3) are consistently more reliable than linear interventions.

- **Part II** contains my work on **transfer learning**, studying how models generalize to new tasks and data distributions by *controlling* and *evaluating* their use of task-causal vs spurious features.

  - **Chapter 5** is based on *Not Just Pretty Pictures: Toward Interventional Data Augmentation Using Text-to-Image Generators* (Yuan*, Pinto*, Davies*, et al., 2024)

    To control task-specific, supervised models' reliance on causal vs spurious features, we leverage generative models to perform *interventional data augmentation* – i.e., to augment existing training datasets by synthesizing samples across distribution shifts to break spurious correlations from models' training datasets, making spurious features non-predictive of task labels. We show that this approach yields state-of-the-art results on domain generalization and bias reduction, and ablate across key dimensions of data synthesis to determine how each component contributes to the robustness of downstream models.

  - **Chapter 6** is based on *Focus On This, Not That! Steering LLMs with Adaptive Feature Specification* (Lamb, Davies, et al., 2025)

    To control task-general models' (e.g., LLMs') reliance on causal vs spurious features, we introduce a post-training procedure (Focus Instruction Tuning, or FIT) that trains models to dynamically condition their responses based on specified features on which to focus or ignore, leading to different behaviors based on what features are specified. We show that FIT enables dynamic inference-time steering of which features models leverage to respond to any given task instance,

improving generalization by suppressing spurious correlations and focusing on causal features; and, unlike more traditional data augmentation or steering approaches, FIT steerability generalizes to new features and domains unseen during FIT training.

  - **Chapter 7** is based on *Hidden in Plain Sight: Evaluating Abstract Shape Recognition in Vision-Language Models* (Hemmat, Davies, et al., 2024)

    To evaluate models' use of causal vs spurious features, we introduce a challenging new out-of-distribution (OOD) benchmark, leveraging generative models to synthesize data to systematically perturb the relationship between spurious features and ground-truth outputs. Specifically, we extend traditional "shape vs texture" image-classifier robustness benchmarking to the context of abstract shape information embedded in scene elements, synthesizing three OOD shape-recognition benchmarks that are easily solved by human annotators but cannot be performed by even leading frontier vision-language models (VLMs).

- **Part III** contains my work at the intersection of Parts I and II, leveraging mechanistic interpretability techniques to predict and explain when and why models fail to generalize out-of-distribution.

  - **Chapter 8** is based on *How Do LLMs Represent and Process Symbols In-Context?* (Davies et al., 2026)

    We leverage a variety of mechanistic interpretability techniques to study how language models perform a general symbolic reasoning task in-context, examining how models trained on a given distribution of this task compositionally generalize to increasingly difficult distributions. We find that the symbol-processing mechanisms models learn to perform the task in-distribution are programmatically valid, but that these mechanisms are brittle with respect to compositional distribution shifts, indicating that the failure of such models to generalize compositionally is not primarily due to the lack of programmatic internal mechanisms but rather the robustness of these mechanisms to more complex inputs.

- **Part IV** (**Chapter 9**) concludes by providing a retrospective summary of the work in Parts I to III and indicating actionable directions for future research to build on this work.

Part I
Mechanistic Interpretability

Chapter 2 The Cognitive Revolution in Interpretability

Preface

In **Chapter 2**, I present a substantially revised and updated version of our preprint *The Cognitive Revolution in Interpretability: From Explaining Behavior to Interpreting Representations and Algorithms* (Davies and Khakzar, 2024).[1] A major goal of this work is to help bridge the fields of cognitive science and mechanistic interpretability by grounding mechanistic interpretability in the context of cognitive science – which has long struggled with analogous questions in studying and explaining the behavior of "black box" intelligent systems like the human brain (as discussed in Chapter 1) – in order to better understand how associated cognitive disciplines have approached the question of understanding, interpreting, and explaining the behavior of intelligent systems. We build on important ideas and frameworks from the history of cognitive science to disentangle divergent objectives in mechanistic interpretability and indicate a clear path forward, highlighting key historical parallels and analogous problems, perspectives, and frameworks for thinking about both fields, and survey a variety of mechanistic interpretability methods and results through this cognitive lens.

Fundamentally, we believe that mechanistic interpretability *is* (or at least, has the potential to be) the *cognitive science of deep learning* – and of artificial intelligence more broadly, at least to the considerable extent to which the field is dominated by deep learning models – and as such, we argue that it would greatly benefit both cognitive science and mechanistic interpretability to understand and build on the deep parallels between these fields of study to accelerate and improve research by incorporating key insights across fields, avoid repeating mistakes in the history of either field, and perhaps ultimately yield a more unified study of cognition encompassing both biologically- and artificially-intelligent systems.

Chapter 2: *The Cognitive Revolution in Interpretability*

Preprint: Adam Davies and Ashkan Khakzar (2024). "The Cognitive Revolution in Interpretability: From Explaining Behavior to Interpreting Representations and Algorithms". In: *arXiv preprint arXiv:2408.05859*. URL: https://arxiv.org/abs/2408.05859

[1] Note that a nontrivial portion of the original preprint discussing the proposed *cognitive revolution in deep learning* is used in the general dissertation introduction in Chapter 1. As such, I do not repeat it here in Chapter 2; but it remains essential context for this chapter.

2.1 Introduction

Artificial neural networks have long been understood as "black boxes": though we know their computation graphs and learned parameters, the knowledge encoded by these weights and functions they perform are not inherently interpretable. As such, from the early days of deep learning, there have been efforts to explain these models' behavior and understand them internally; and more recently, *mechanistic interpretability* has emerged as a distinct research area studying the features and implicit algorithms learned by foundation models such as large language models. Chapter 1 argued that current mechanistic interpretability research is ripe to facilitate a transition in deep learning interpretation echoing the "cognitive revolution" in 20th-century psychology that shifted the study of human psychology from pure behaviorism toward mental representations and processing; and in this work, we sketch out the potential structure of this revolution, proposing a taxonomy of mechanistic interpretability work mirroring key parallels in computational neuroscience, breaking mechanistic interpretability work into three broad categories:[2]

- *Algorithmic interpretation*: what *operations* and *algorithms* are implicitly implemented by models to carry out a given behavior? (See Section 2.5.)
- *Semantic interpretation*: what latent *feature representations* are learned and leveraged by models and manipulated in higher-level operations and algorithms? How do they contribute to model behaviors? (See Section 2.4.)
- *Implementational interpretation:* how are higher-level representations, operations, and algorithms *implemented* by models? For instance, how are representations encoded in embedding spaces; or how are operations/algorithms performed using model weights? (See Section 2.3.)

Our goal is to explore the relationship between these levels of interpretation, elaborating the different assumptions, parallels, and distinctions between approaches in each category, and how they each conceive of "interpretation" in different ways. We further discuss the challenges faced by each category, analyze the respective strengths and weaknesses of representative works, clarify underlying assumptions, and outline key challenges faced by each mode of interpretation.

[2]In this categorization, we substitute the more common term "interpretability" with "interpretation", as each category of work is concerned with *interpreting* the inner structure of an otherwise "black box" model, rather than the state of being inherently interpretable (as studied in the area of interpretable machine learning; see Appendix A.1.1).

2.2 Background

2.2.1 A Brief History of Deep Learning Interpretation

Since their introduction, neural networks have been considered "black boxes," meaning they are not inherently interpretable. The research on deep learning interpretation is concerned with casting light on the "black box" problem in some form or another. In the era of deep learning preceding the development of foundation models (i.e., neural networks pre-trained on large-scale, self-supervised tasks such as LLMs; Bommasani et al., 2021), the terms *interpretation* and *explanation* most often referred to saliency (feature attribution) methods.[3] However, following the paradigm shift toward foundation models, these terms now more often refer to mechanistic interpretation. In this section, we provide a broad overview of both periods and associated paradigms in deep learning interpretation.

Behaviorist Interpretation: The Rise of Post-hoc Explanations The paradigms of behaviorist and mechanistic interpretation of neural networks emerged concurrently, shortly following the modern study of deep learning. For instance, early work in deep learning interpretation discussed explaining the network behavior by identifying the importance of input features for the output, coining the term "saliency maps" (Simonyan et al., 2013); and contemporaneous work studied representations by analyzing different neuron activations within convolutional neural networks (Zeiler and Fergus, 2014). However, the focus on post-hoc behavioral explanations – i.e., explaining why a given model produced a specific output (Lipton, 2018; Doran et al., 2017; Arrieta et al., 2020) – gained more traction in the context of the then-dominant paradigm of classification and supervised learning, particularly in the forms of feature attribution methods and counterfactual explanations. Specifically, feature attribution (Lundberg and Lee, 2017; Sundararajan et al., 2017; Sundararajan and Najmi, 2020; Ribeiro et al., 2016; Fong et al., 2019; Simonyan et al., 2013; Selvaraju et al., 2017; Khakzar et al., 2022; Bach et al., 2015) analyzes the behavior of the network by identifying input features that are relevant for that output; and counterfactual explanations (Wachter et al., 2017; Lang et al., 2021; Ribeiro et al., 2020) understand the behavior of the network by answering what needs to change in the input for the network to have a particular output. Note that these methods are analogous to the behaviorist paradigm in psychology (discussed in Chapter 1), which studied cognition only in terms of observable stimulus/response mappings, not internal mental representations or cognitive processing.

[3] See Appendix A.1.1 for further discussion on earlier definitions of interpretability outside the context of modern foundation models.

Mechanistic Interpretation: The Cognitive Revolution in Interpretability With the rise of generative and self-supervised learning paradigms that have enabled increasingly powerful and task-general foundation models, the focus has shifted toward questions regarding what these models actually learn from pre- and post training and what implicit algorithms they perform internally, rather than why a model generated a particular output in a downstream task. Such work is popularly referred to as *mechanistic interpretability*, which is generally defined as the subfield of interpretability research concerned with "reverse engineer[ing] neural networks, similar to how one might reverse engineer a compiled binary computer program" (Olah, 2022; see Saphra and Wiegreffe, 2024 for further discussion of the term "mechanistic" in this context, as well as a broad overview of the longstanding line of research conducted in this area before the term "mechanistic interpretability" came into popular usage.) This description clearly covers implicit algorithms and individual constituent operations, as studied in circuit discovery (see Section 2.5); but it is less clear precisely how mechanistic interpretability relates to the study of latent representations of human-interpretable concepts, particularly when specific concepts of interest are provided in advance of empirical analysis (as in probing; see Sections 2.4.2 and 2.4.3), rather than dynamically discovered from embedding representations (as in dictionary learning; see Section 2.4.4). In order to avoid conflating these related but fundamentally distinct notions of interpretability and clarify the specific object of analysis for each family of work, we define a taxonomy of questions, goals, methods, and challenges according to the study of latent representations (*semantic interpretation*; see Section 2.4), operations/algorithms (*algorithmic interpretation*; see Section 2.5), and the manner in which representations, operations, and algorithms are implemented (encoded/processed) by models (*implementational interpretation*; see Section 2.3).

Complementary Paradigms It is important to recognize that mechanistic interpretation and behavioral explanation are complementary rather than competing. For instance, understanding why a given deep neural network produced a certain output remains crucial, especially in safety-critical domains such as healthcare (Reddy, 2022; Petch et al., 2022; Thirunavukarasu et al., 2023); and monitoring the latent knowledge and capabilities of frontier models has key implications for AI safety (Hendrycks et al., 2021b; Zou et al., 2023). Such directions are deeply intertwined and should continue to inform each other, as they have throughout the history of interpretability. For instance, internal representations have been leveraged for feature attribution explanations in Class Activation Mapping (CAM) (Zhou et al., 2016), where neuron activations are used to explain which input features are relevant to the output; and subnetwork analysis (which is closely related to circuits; see

Section 2.5) has worked to explain behaviors in terms of input features by finding and ablating sparse internal pathways through the network (Bach et al., 2015; Khakzar et al., 2021). More recently, approaches such as causal probing (Davies et al., 2024; Canby* et al., 2025; see Chapter 3 and Chapter 4, respectively) and dictionary learning (Bricken et al., 2023; Templeton et al., 2024) have been leveraged for explaining the behavior of LLMs in terms of interpretable latent features (see Sections 2.4.3 and 2.4.4, respectively), and circuit discovery has been applied to uncover the interpretable subnetworks of LLMs that correspond to certain behaviors (Wang et al., 2023b; Conmy et al., 2023) (see Section 2.5). This interplay mirrors parallel approaches to understanding human intelligence as observed between behavioral and cognitive modes of analysis: just as cognitive neuropsychology can be invaluable in explaining real-world behaviors and pathologies, mechanistic interpretations can inform our understanding of model outputs, and vice versa. Both perspectives are essential for a holistic understanding of neural networks.

2.2.2 Levels of Analysis

Marr's levels of analysis (Marr, 1982) have long been a workhorse foundational framework for scientific inquiry in neuroscience and other disciplines of cognitive science (Niv and Langdon, 2016). They are as follows:

- The **functional level** provides a mathematical description of the function computed by an information-processing system in terms of how it maps inputs to outputs (behaviors).
- The level of **representations and algorithms** is concerned with...
    - the system's **representation** of inputs, intermediate states, and outputs;
    - and the overall **algorithm** that the system employs over representations to perform the mapping described at the functional level.
- The level of **implementation** is concerned with how representations and algorithms are realized in a physical or software medium.

Recent developments in semantic, algorithmic, and implementational interpretation have enabled the study of these models at the level of *internal representations and algorithms* they learn and leverage to implement the input/output mappings described at the functional level, where studying representations is a matter of *semantic interpretation*, studying algorithms is a matter of *algorithmic interpretation*, and studying how these mechanisms are implemented

by models is a matter of *implementational interpretation* (discussed in Sections 2.3 to 2.5, respectively).[4]

2.3 Implementation Level

2.3.1 Implementing Representations

Neuron-level Representation A natural starting point for feature implementation (encoding) in neural networks might be to ask whether individual neurons code for distinct features of interest. An analogous hypothesis in neuroscience is the notion of "gnostic cells", which are cells (neurons) that fire only in the context of very specific concepts, such as one's grandmother (hence the name) (Gross, 2002; Quiroga, 2013). Do we find such neurons in contemporary deep learning systems? At face value, there are several reasons that neural networks may not be expected to encode primitive representations using individual neurons. First, neural networks represent inputs using *dense embeddings*, which are *distributed* representations wherein features can be encoded by arbitrary linear combinations of any number of neurons, meaning that we might not expect models to exhibit *privileged bases* (Elhage et al., 2021, 2022) – i.e., there may be no particular bias toward representing features using the standard basis (encoding features at the neuron level) rather than an arbitrary rotation of the standard basis where each feature utilizes all dimensions of the embedding space. Additionally, there is robust, longstanding evidence that, across a variety of architectures, data modalities, and other key axes of variation for neural networks, these models tend to be highly *polysemantic*, meaning that individual neurons are involved in encoding multiple features (Nguyen et al., 2016b; Arora et al., 2018; Mu and Andreas, 2020; Elhage et al., 2022; Marshall and Kirchner, 2024). As such, the underlying assumption made in interpreting the semantics of only single, isolated neurons – i.e., that there is a one-to-one mapping from neuron activations to values taken by latent features (analogous to the "gnostic cell" hypothesis) – is in no way guaranteed to hold. However, in practice, there is some limited evidence that privileged bases can sometimes emerge during normal training (Elhage et al., 2022, 2023; Brown et al., 2023), suggesting that neuron-level representation may indeed be a more reasonable starting point for feature interpretation than random, non-standard bases; and some works have further relaxed the "one-to-one neuron-to-feature" mapping assumption by considering sparse combinations of neurons instead of only

[4]Note that Hamrick and Mohamed (2020) and Sawant and Singh (2020) have also discussed the role of Marr's levels of analysis in the context of machine learning, but have instead approached these levels only from the perspective of the model training process – e.g., where algorithms refer to the learning algorithms used to train models, or implementations refer to network architecture/hyperparameters – rather than actual models that are the output of this process, or the internal algorithms, representations, and implementations of such that these models learn during training.

individual neurons (Geva et al., 2021; Dai et al., 2022).

Linear Representation Hypothesis The *linear representation hypothesis* (LRH) states that models implement different features using distinct (and usually low-dimensional) linear subspaces of their embedding spaces (Bolukbasi et al., 2016; Alain and Bengio, 2017; Vargas and Cotterell, 2020; Geiger et al., 2024). Under a more specific version of the LRH, it is further asserted that these subspaces are 1-dimensional and mutually orthogonal (Elhage et al., 2022; Park et al., 2024b; Engels et al., 2025a; Lee et al., 2025), meaning that each feature essentially acts as a basis vector for the model's representation in a given layer, with compositions of features simply represented as linear combinations of corresponding basis vectors (see Olah and Jermyn, 2024). One key argument in favor of the LRH is that neural networks must make class-discriminative information linearly separable in their final layer, so we might expect them to prefer linear encoding of features which are important for classification, particularly in later layers (Alain and Bengio, 2017). The LRH has been extensively studied in the context of probing and dictionary learning, as discussed in further detail in Sections 2.4.2 to 2.4.4.

Superposition Hypothesis Under the strictest interpretation of the LRH discussed above, wherein all features $\{f_i\}_1^d$ are implemented as corresponding orthogonal basis vectors $\{v_i\}_1^d$ in a d-dimensional embedding space, it would only be possible for models to represent as many features as the number of neurons (d), as any additional $(d+1)$-th feature vector could be represented as a linear combination of basis vectors $\{v_i\}_1^d$. The *superposition hypothesis* (Elhage et al., 2022) posits that models learn to represent more features than they have neurons in a given layer by encoding them via *almost-orthogonal* directions in the embedding space, exponentially increasing the number of features that can be represented in the embedding space at the cost of some noise due to interference between non-orthogonal features.[5] According to this hypothesis, models leverage superposition because the benefits of representing (potentially many) more features with fewer neurons outweighs the cost of filtering out this noise (via nonlinear activation functions) so long as features are sufficiently sparse (Elhage et al., 2022; Scherlis et al., 2022; Lee et al., 2025). The superposition hypothesis is an important foundation for modern sparse auto-encoders, which are discussed in detail in Section 2.4.4.

[5]Specifically, with d number of neurons, embeddings can encode $D \sim \exp(d)$ features $\{f_i\}_1^D$ with no greater than ε interference – i.e., $|f_i \cdot f_j| \leq \varepsilon$, for $\varepsilon = \sqrt{\frac{20 \ln D}{d}}$ (see Lee et al., 2025).

Nonlinear Representation As neural networks are, by design, highly nonlinear mathematical objects, it is natural to expect that they may implement features nonlinearly (White et al., 2021). This is particularly true of earlier or intermediate layers, where – unlike the final layer (as noted above; cf. Alain and Bengio, 2017) – class-discriminative information in these layers does not need to be linearly separable for models to correctly classify outputs. Another argument in favor of nonlinear implementation is that interpretability methods should mirror the architecture of the model being probed, as this more directly reflects the information that can be extracted by the model from its own representations (Pimentel et al., 2022). There is also a growing empirical literature suggesting that many key features learned and leveraged by a variety of neural networks are indeed nonlinearly implemented (Canby* et al., 2025; Csordás et al., 2024; Engels et al., 2025b; Sutter et al., 2025).[6]

Note that the study of nonlinear implementation of representations via causal probing (see Section 2.4.3) is a major focus of the work in this dissertation: in Chapter 3, we introduce a method for causal probing that allows one to flexibly specify arbitrary (e.g., nonlinear) implementations of target features; and in Chapter 4, we leverage this method alongside other causal probing techniques to validate which implementation class is most reliable for interpreting model behaviors, finding that nonlinear methods are generally more reliable than linear ones.

2.3.2 Implementing Algorithms

Architectural Targets When interpreting how models implement various algorithms architecturally, a key question is how different components of model architecture should be taken into account for algorithmic analysis. For example, in the context of Transformer-based models (Vaswani et al., 2017), some work investigates the algorithmic role played at the coarse-grained level of entire Transformer-block layers (Sajjad et al., 2023; Lad et al., 2025), while other work operates at the more fine-grained level of components/sub-layers such as individual attention heads (Voita et al., 2019; Vig and Belinkov, 2019; Clark et al., 2019; Conmy et al., 2023; Wang et al., 2023b) or MLPs (Geva et al., 2021; Dai et al., 2022; Meng et al., 2022b, 2023), or even specific combinations of model parameters (Savarese et al., 2020; De Cao et al., 2020; Cao et al., 2021; Csordás et al., 2021b; Zhang et al., 2021a; Lepori et al., 2023; He et al., 2025b).[7]

[6] Note that such findings are equally relevant to the context of linear features in superposition, as studied in SAEs (see Section 2.4.4).

[7] See Section 5.1 of Mueller et al., 2024 for a more detailed discussion on coarse- versus fine-grained (sub-)layer analysis.

2.4 Semantic Interpretation

We may define *semantic interpretation* as a matter of answering the following question: what latent features are learned and represented by neural networks, and how do they contribute to observed model behaviors? For example, do vision models learn representations of semantically-related object categories or spatial relations, or do LLMs learn representations of syntactic dependencies or lexical relations? And how do models leverage these features to perform various tasks of interest?

Implementation Classes All approaches to semantic interpretation must be empirically defined with respect to a given class of implementations (e.g., neuron-level, linear, or nonlinear; see Section 2.3.1) before it is possible to semantically interpret features represented and used by a given model. Some approaches specifically "bake in" a particular implementation class – for instance, most traditional optimization and search methods (discussed in Section 2.4.1) operate specifically at the level of individual neurons, and would be ill-defined in the context of linear feature vectors, etc. However, many approaches can in principle be defined generally across arbitrary classes of implementations, only requiring a specific class to be selected in order to run a given experiment with respect to that implementation class, potentially allowing for empirical comparison across arbitrary classes of implementation – for instance, probing (discussed in Section 2.4.2) can be defined at any implementation class expressible via neural network classifiers (probes), and some approaches to causal probing (discussed in Section 2.4.3) even allow for interventions and direct empirical comparisons across arbitrary such implementation classes (Davies et al., 2024; Canby* et al., 2025; see Chapter 3 and Chapter 4 for our work in this area).

2.4.1 Optimization and Search

What inputs maximize the activation of a given neuron? The goal of optimization and search methods is to interpret features at a given level of implementation (e.g., individual neurons, sets of such, or feature directions) by finding which inputs maximally activate the feature. The most prominent approach to *neuron-level* semantic interpretation involves selecting an individual neuron and either searching through a pre-defined query set or generating an input (such as an image) that maximally activates the target neuron, and inspecting these optimized inputs for common features. Early work in this area focused on image recognition networks (Erhan et al., 2009; Zeiler and Fergus, 2014; Simonyan et al., 2013; Zhou et al., 2014), where some approaches operate by searching for patterns that maximize the output of neurons via image optimization (Simonyan et al., 2013; Zeiler and

Fergus, 2014; Nguyen et al., 2016a; Olah et al., 2017), and others searched through a pre-defined query set to find inputs that maximized neuron activations (Erhan et al., 2009; Fong and Vedaldi, 2018; Bau et al., 2017, 2019). For instance, Net2Vec (Fong and Vedaldi, 2018) and Network Dissection (Bau et al., 2017, 2019) systematically analyze the relationship between concepts and neurons by analyzing the activations of neurons in response to images in the dataset clustered by human-annotated visual concepts. Later work has investigated how the same paradigm can be applied to interpret individual neurons in language models (Karpathy et al., 2015; Dalvi et al., 2019; Geva et al., 2021; Dai et al., 2022) and multimodal vision-language models (Hernandez et al., 2021; Schwettmann et al., 2023).

2.4.1.1 Assumptions and Challenges

Needle in the Haystack A central concern with analyses centered around the activations of individual neurons (or small sets of such) is that billion-parameter scale models have millions of functional neurons to interpret; and naturally, as the number of neurons scales with model size, this problem becomes increasingly severe with ever-larger models. Given that it would be computationally intractable to perform detailed analysis with respect to each individual neuron or feature direction, how can one determine which are meaningful and merit individual analysis, and which can be ignored? While a number of heuristic approaches have been proposed for targeting individual neurons (Bau et al., 2019; Meng et al., 2022b; Schwettmann et al., 2023) or feature directions of interest, there is currently no generally accepted methodology for determining which are most important for any given analysis, meaning that there is no way to guarantee (or even provide bounds on the probability) that one has correctly targeted those neurons/directions whose activations/magnitude are most important in studying any given research question.

2.4.2 Probing

Framework One of the most popular and well-studied approaches to semantic interpretation has been *probing* (Belinkov, 2022; Rogers et al., 2021; Belinkov et al., 2020). The goal of probing is to analyze which features (e.g., part-of-speech, sentiment labels, etc.) are represented by a deep learning model (e.g., LLM) by training auxiliary, supervised neural classifiers to predict such features from latent embedding representations (neural activations) using a dataset of embeddings labeled with respect to target features (Belinkov, 2022). For example, one may train a probe to predict parts-of-speech from LLM token embeddings, so when given each token in the sequence (`The`, `cat`, `meows`, `for`, `dinner`), the probe predicts the corresponding parts of speech (`determiner`, `noun`, `verb`, `preposition`, `noun`),

respectively. Formally: given a foundation model M, input $\mathbf{x}$, and embeddings $\mathbf{h}^l = M_l(\mathbf{x})$ of input $\mathbf{x}$ at layer l of M, suppose Z is a latent feature of interest that takes a discrete value $Z = z$ for input $\mathbf{x}$; and define $\mathbf{X}$ as the input space of M, with $\mathbf{H}^l = M_l(\mathbf{X})$ as the image of $M_l(\mathbf{X})$. Here, the goal of probing is to train a classifier $g_Z^l : \mathbf{H}^l \mapsto Z$ on the probe task of predicting $P(Z \mid \mathbf{H}^l)$ from some labeled training dataset $D_{\text{train}} = \{\mathbf{h}_i^l, z_i\}_{i=1}^n$.

A strong form of the underlying assumption made by many probing studies is that a model M is "representing" a feature Z if and only if this feature can be consistently predicted from embeddings – i.e., if probes g_Z^l can be trained such that they achieve high test-set performance on the probe task $P(Z \mid \mathbf{h}^l)$ for some unseen test set D_{test}. For example, an early and influential hypothesis in probing, known as the *pipeline hypothesis*, uses probe accuracies over linguistic tasks across BERT (Devlin et al., 2019) layers to argue that BERT processes linguistic features in the same order as the "classical NLP pipeline", with surface-level features recognized first, followed by syntactic features, and semantic features recognized last (Tenney et al., 2019a; Jawahar et al., 2019; Rogers et al., 2021; Niu et al., 2022).

2.4.2.1 Assumptions and Challenges

Implementation Class In order to carry out probing research, one must first define the class of implementations to be probed – or equivalently, the architecture of the probe being trained (e.g., for a linear probe $g_Z^l(\mathbf{h}^l) = W(\mathbf{h}^l)$ that learns a linear transformation parameterized by W, this probe is naturally only able to consistently detect features that are linearly-encoded). The question of which probing architecture should be utilized is a contentious one (Belinkov, 2022; Hewitt and Liang, 2019; Pimentel et al., 2022); and below, we outline several choices of probing architecture as utilized in the literature, supporting arguments in favor of each architecture, and corresponding limitations.

- **Linear Probes** Perhaps the most well-studied probing architecture is the *linear probe* (Alain and Bengio, 2017; Kim et al., 2018; Liu et al., 2019a; Ravfogel et al., 2020; Elazar et al., 2021b; Schwettmann et al., 2021; Park et al., 2024b; Marks and Tegmark, 2023; Tigges et al., 2023; Nanda et al., 2023b). Use of such probes assumes the *linear subspace hypothesis*, as discussed above in Section 2.3.1. For instance, Concept Activation Vectors (CAV) (Kim et al., 2018) are vectors in the representation space perpendicular to a linear classification boundary that classifies the representations of a concept versus other input representations. One argument in favor of linear probing is the intuition that neural classifiers must make class-discriminative information linearly separable in their final embedding layer, so probes (particularly over final-layer

embeddings) should also be linear (Alain and Bengio, 2017). Another motivation is that, given enough training data, sufficiently expressive probes can memorize arbitrary probe tasks irrespective of the model being probed (Hewitt and Liang, 2019), so the accuracy of a simpler (i.e., less expressive) probe may better reflect the actual content of embeddings rather than the expressiveness of the probe.

- **Nonlinear Probes** An alternative approach to probing involves training arbitrarily expressive (nonlinear) probes in order to learn any representation that may be encoded by the model (Pimentel et al., 2020; White et al., 2021; Pimentel et al., 2022; Davies et al., 2024). However, as noted above, there have been concerns that highly expressive probes may memorize the mapping from embeddings to features irrespective of the model being probed (Hewitt and Liang, 2019); so for more complex probes, there is an increased risk that high probing accuracy is more a reflection of the probe itself than it is of the model being probed. For instance, in an extreme case, consider generative vision-language models that use embeddings from a frozen image encoder and fine-tune an LLM to generate corresponding image captions (as in, e.g., Zhang et al., 2021b; Zhai et al., 2022). In this case, the LLM could be understood as a highly complex and expressive probe in the sense that the LLM is an auxiliary network (cf. probe) stacked on top of a frozen vision encoder (cf. model being probed) and trained to generate text captions (cf. probe task). Here, it is clear that the generative "probe task" is, in fact, being learned by the probe (LLM) and not the original model (vision encoder), given that the vision encoder is never trained to generate text.

Probing for what? Another important assumption is the choice of features for which to probe. It is impossible to know *a priori* which features a model is representing or leveraging in any given context (Rogers et al., 2021; Yun et al., 2021; Belinkov, 2022); so for any given probing experiment, it is always possible that one has simply failed to capture whatever features are most important to the model, leading to potentially misleading results. This is a serious concern for semantic interpretation, given that we cannot reasonably presume to know ahead of time what complete set of features may be represented and leveraged by models, or whether they happen to correspond to key features in human cognition; and there is a long-documented trend toward anthropomorphizing intelligent-seeming models (especially in the context of linguistic systems such as LLMs; Weizenbaum, 1966; Turkle, 2007; Switzky, 2020). For instance, while substantial early work in probing studied "classic NLP pipeline" features (Tenney et al., 2019a; Jawahar et al., 2019; Rogers et al., 2021; Niu et al., 2022; as discussed above), there is no particular reason to believe that such features are the most important features for interpreting the internal representation or explaining the behavior of

any given LLM. Additionally, there are many ways to interpret probing results (Belinkov, 2022): naively, one might simply consult probing accuracies and compare them between features or across layers; but various works have argued for the necessity of comparing probe predictions against randomized baselines (Tenney et al., 2019b), control tasks (Hewitt and Liang, 2019), or to compute information gain using control functions (Pimentel et al., 2020).

Correlation does not imply causation. Perhaps the single most important concern with probing is that, under this paradigm, it is only possible to measure *what features can be predicted* from a representation, not *whether (or how) they are actually used* by the model (Belinkov, 2022; Elazar et al., 2021b; Hewitt and Liang, 2019; Davies et al., 2024). We discuss a few proposed solutions to this problem in the following sections.

2.4.3 Causal Probing

Framework One proposed solution to traditional probing's inability to distinguish correlation from causation problem is *causal probing*, which uses latent-space interventions over learned probes $g_Z^l : \mathbf{H}^l \mapsto Z$ to remove or alter the predicted feature Z in the representation $\mathbf{H}^l$ in layer l of model M, and analyzes the impact of such interventions on the model's predictions to study how it makes use of the feature (Elazar et al., 2021b; Tucker et al., 2021; Lasri et al., 2022; Davies et al., 2024; see Chapter 3 and Chapter 4 for our work in this area). Specifically, causal probing performs interventions $\text{do}(Z)$ that modify M's representation of Z in embeddings $\mathbf{H}^l$, producing $\hat{\mathbf{H}}^l$, where interventions are variously intended to encode a counterfactual value $Z = z'$ (denoted $\text{do}(Z = z')$ where $z \neq z'$), or remove the representation of Z entirely (denoted $\text{do}(Z = 0)$). (Note that comparing counterfactual and removal interventions is a major theme of our work in Chapter 4; Canby* et al., 2025.) Following the intervention, modified embeddings $\hat{\mathbf{h}}^l$ are fed back into M beginning at layer $l + 1$ to complete the forward pass, yielding intervened predictions $P_M(\cdot \mid \mathbf{x}, \text{do}(Z))$. Comparison with the original predictions $P_M(\cdot \mid \mathbf{x})$ allows one to measure the extent to which M uses its representation of Z in predicting/generating outputs.

2.4.3.1 Assumptions and Challenges

Inherited from Probing As in traditional probing, causal probing requires one to pre-define the level of representation (neuron-level, linear, or nonlinear) and set of features for which to probe before any probing experiment can begin. Most intervention methodologies are built to operate at only a single level of representation – typically linear (Ravfogel et al., 2020; Elazar et al., 2021b; Ravfogel et al., 2021, 2022a), with some work

exploring kernelized linear representations (Ravfogel et al., 2022b; Shao et al., 2022) – meaning that methods and results from one level cannot be directly adapted or compared to those from other levels. In Chapter 3 (Davies et al., 2024), we address this limitation by introducing *gradient-based interventions,* a more flexible class of causal probing interventions that use adversarial attacks against arbitrary probing architectures, allowing interventions to target whatever implementation class is defined by the probe.

Intervention Reliability An important observation made by Elazar et al. (2021b) is that the original values of features targeted by causal probing interventions can be *recoverable* by later ones – e.g., when INLP is used to remove information about some target feature from the embeddings of a given upstream layer, it is often still possible to train (linear) probes to predict the feature from embeddings of a later downstream layer, meaning that these linear interventions are *incomplete*. For such features, this finding may be taken as evidence against the linear subspace hypothesis discussed above: INLP removes *all* information that is linearly predictive of the target feature, so if BERT only encoded these features linearly, it would not be possible to recover them following an INLP intervention. Later works in causal probing (including our work in Chapter 3) have investigated nonlinear interventions (Ravfogel et al., 2022b; Shao et al., 2022; Davies et al., 2024); but as in probing, there is a tradeoff associated with expressivity: theoretically, the more powerful (and potentially more *complete*) an intervention is, the more "collateral damage" it may also cause to representations more generally (Zhao et al., 2022; Kumar et al., 2022), in which case the intervention is less *selective* in restricting damage to only the target feature.[8] Thus, for the most expressive intervention methodologies, it is difficult to determine whether any observed changes to model behavior in the presence of interventions is attributable to the model's representation of the target feature or simply to collateral damage (i.e., low *selectivity*).

In Chapter 4, we study the tradeoff between completeness and selectivity of causal probing interventions in detail, finding that the most expressive nonlinear causal probing methods (GBIs, as introduced in Chapter 3; Davies et al., 2024) exhibit the most favorable tradeoff between these two criteria.

Rashomon Effect Finally, as in most studies of causality, causal probing introduces the *Rashomon effect* (Breiman, 2001; Hancox-Li, 2020): when there are multiple explanations with equal causal efficacy, then they would be considered equally "correct" (even if contradictory). For instance, if intervening on the representation of feature A in layer L and

[8]Here, we use *selectivity* in the sense described by Elazar et al. (2021b) and Canby* et al. (2025), and not other probing work such as Hewitt and Liang (2019), where it instead refers to the gap in performance between probes trained to predict real features versus randomized "nonsense" features.

feature B in layer $L + k$ from the model leads to the same impact on its behavior, then one cannot definitely attribute the model's behavior to its use of either A or B, which is particularly problematic when considering that there may always be features on which we simply have not yet tested but have equal (or greater) causal efficacy to those that we have tested (Mueller, 2024; Sutter et al., 2025).

2.4.4 Dictionary Learning

As noted in Sections 2.4.2 and 2.4.3, one of the key limitations of both traditional and causal probing is that they are fully supervised, meaning that one must pre-define a set of latent features for which to probe. This means that, even for a perfect (causal) probing methodology, it is always possible that the most important features leveraged by the model in the course of performing a particular task could be completely missed if one simply does not probe for these particular features (Rogers et al., 2021; Yun et al., 2021; Belinkov, 2022). This is a serious concern for semantic interpretation, given that we cannot reasonably presume to know ahead of time what complete set of features may be represented and leveraged by a model in any given context.

Dictionary Learning An alternative paradigm in semantic interpretation, *dictionary learning*, removes this presumption by inverting the traditional probing process: instead of directly training a *supervised* probe to predict some latent feature of interest from a model's intermediate embedding representations, the goal of dictionary learning is to train an *unsupervised* probe to decompose embeddings into a sparse combination of features and use them to reconstruct the original embeddings, yielding a *dictionary* of features that are useful for sparsely representing embeddings (Lewicki and Sejnowski, 2000; Lee et al., 2006; Faruqui et al., 2015).

Sparse Auto-Encoders Sparse Auto-Encoders (SAEs) (Subramanian et al., 2018; Yun et al., 2021) have recently emerged as a powerful and scalable approach to unsupervised probing via dictionary learning (Cunningham et al., 2023; Bricken et al., 2023; Templeton et al., 2024), performing a nonlinear transformation of input embeddings onto an "overcomplete linear basis",[9] allowing them to learn (potentially exponentially many) more features than the dimensionality of the embeddings they are trained on. (Note that this formulation explicitly relies on the *superposition hypothesis*, as discussed in Section 2.3.1.) Where early dictionary learning methods in signal processing were based on sparse coding

[9]While "overcomplete basis" is the standard term of art in SAE work, note that this term violates standard nomenclature in linear algebra (Preetham, 2024).

methods involving Bayesian modeling (Lewicki and Sejnowski, 2000) or convex optimization (Lee et al., 2006), SAEs carry out dictionary learning using neural networks, improving scalability while maintaining the same goal of learning sparse features for decomposition and reconstruction.

2.4.4.1 Assumptions and Challenges

Unsupervised Feature Interpretation While SAEs (and dictionary learning more broadly) remove the need for labeled training data for supervised probes, they effectively transfer the burden of annotation from probe training data to unsupervised feature interpretation, as each feature vector in the dictionary cannot be directly interpreted any more easily than dense embeddings themselves. Rather, each feature must be retroactively interpreted, usually by asking an annotator (either a human or an LLM; Bills et al., 2023) to inspect the input samples that maximally activate the feature and annotating it according to whatever feature these samples intuitively appear to have in common (Cunningham et al., 2023; Bricken et al., 2023; Templeton et al., 2024). For instance, in the largest SAE study to date (including up to 34 million SAE features learned from embeddings of a frontier LLM), Templeton et al. (2024) find that one feature learned by an unsupervised probe over a large multilingual, multimodal vision-language model corresponds to the *Golden Gate Bridge*, and that this feature is strongly activated in contexts discussing the bridge in many different languages or for images depicting the bridge, while being weakly activated in contexts discussing (or images depicting) other tourist landmarks in San Francisco or other famous bridges. While such features are indeed interesting, it is not clear what proportion of features have such clear interpretations – how many of these millions of features are as easily interpreted as "Golden Gate Bridge"? At such a large scale, it is not feasible for human annotators to manually interpret features, requiring automated interpretation – e.g., prompting another LLM to explain the relationship between multiple passages that highly activate the same feature, a scalable approach that has been shown to be reasonably well-aligned with human judgments (Bills et al., 2023). However, it is important to note that, in contrast to supervised probing methods (where target features are labeled in advance), unsupervised feature interpretations (whether human- or LLM-annotated) are not *transferable* in that this process must be performed *each time* one analyzes a new model, layer, or trains a new SAE – that is, where probing datasets only require that each input (or parts of inputs, such as individual tokens) be labeled once, dictionary learning requires that one (re-)interpret learned features every time a new dictionary is learned, significantly increasing the burden of annotation and preventing direct comparisons between different models, layers, or dictionary learning methods.

Evaluating Interpretations More broadly, even for the most intuitive features, it is not clear precisely how one should evaluate whether any given feature interpretation is correct. At the scale of frontier models and large SAEs, beyond requiring automating feature interpretation, it is also necessary to automate *evaluation* of these interpretations. Thus, even for scalable and high-quality automated techniques, stacking multiple layers of LLM automation for interpretation and evaluation (which may even be carried out by the same type of model that is the object of analysis) could lead to compounding errors. Current research has addressed the problems inherent in this paradigm by leveraging *causal interventions* over discovered features (Bricken et al., 2023; Templeton et al., 2024) (analogous to, e.g., those performed in causal probing over supervised features) in order to observe whether model behavior changes consistently with the hypothesized interpretation of each feature – e.g., when the Golden Gate Bridge feature discussed above is significantly strengthened, the associated LLM responds to many queries with references to the famous bridge where it normally would not. However, as in causal probing, such interventions can only tell us whether a given feature contributes to a particular output, not whether there are other features that contribute just as strongly to the prediction (Mueller, 2024), nor the extent to which these interventions are *complete* or *selective* – i.e., whether they fully control the model's representation of the target feature without also damaging non-targeted features, respectively (Canby* et al., 2025).

2.5 Algorithmic Interpretation

We may define *algorithmic interpretation* as a matter of answering the following questions: what operations is a given neural network implicitly performing (over representations); what role do these operations play in observed behaviors; and, when these operations are taken in aggregate, what algorithm is being implicitly implemented by the model? Such operations and algorithms are typically understood in terms of *circuits*: "sub-graphs of the network" that implement a given operation or algorithm, which may be themselves composed to form larger circuits (Olah et al., 2020). As noted in Section 2.3.2, various levels of implementation have been considered for algorithmic interpretation, such as entire Transformer block layers, layer components such as attention heads or MLPs, or specific subsets of model parameters.

2.5.1 Circuit Discovery

Circuit Discovery The task of finding circuits that faithfully describe a given category of model behaviors is often referred to as *circuit discovery* (Wang et al., 2023b; Conmy et al., 2023). The simplest circuits to understand are *end-to-end circuits*, which describe a full path

(composed of sub-circuits) from input to output, implementing a complete algorithm. For instance, one of the first circuits identified in a Transformer language model is the *induction circuit* (Elhage et al., 2021; Olsson et al., 2022): an end-to-end circuit which, given an input of the form "[A] [B] ... [A]" (where [A] and [B] are arbitrary token sequences), determines whether or not to predict that [B] once again follows the second [A] (as it did earlier in the sequence). For instance, given the input "Vernon Dursley and Petunia Durs" (tokenized into [Vern, #on, Durs, #ley, and, Petunia, Durs], where # denotes a token that has been created by splitting an existing word into multiple word pieces), an induction circuit would be tasked with predicting whether or not to follow the second "Durs" token with "#ley". (See Conmy et al., 2023 for several additional examples of end-to-end circuits that have been discovered in LLMs.)

Causal Interventions As in causal probing and dictionary learning above, a popular strategy in circuit discovery is to explain a circuit's contribution to a model's behavior by intervening in its forward pass and study the effect on the model's predictions. There are two popular strategies for doing so, *knockouts* and *patching*, which we discuss in turn below.

- **Knockout** Knockouts "turn off" a (sub-)circuit by nullifying it in order to determine its contribution to LLM behaviors. Given a circuit, a few approaches to performing knockout have been proposed: *zero ablation* sets the output of the analyzed circuit to zero (Olsson et al., 2022); whereas *mean ablation* instead sets its output to the mean value of a given reference distribution (Wang et al., 2023b). The former approach is simpler, as it does not require one to define a reference distribution; but as subsequent circuits may "rely on activation value[s] as an implicit bias term" (Wang et al., 2023b), it is theoretically more sound (and empirically less noisy) to perform knockout by setting outputs to the mean value of a reference distribution where possible.
- **Activation Patching** (Activation) patching replaces circuit outputs in the course of computing a *target sequence* with activations from computing a *source sequence*, allowing one to measure whether LLM outputs for the patched target sequence change consistently with the hypothesized function of the circuit in the context of the source sequence (Meng et al., 2022b; Wang et al., 2023b; Conmy et al., 2023; Meng et al., 2023). For instance, in the earlier example of an induction circuit given a source sequence of "Vernon Dursley and Petunia Durs", where the next token predicted by the LLM will be "#ley", one may patch the circuit outputs from this source sequence into the LLM in its forward pass while processing the target sequence "Thus, the argument must be val" in order to swap its prediction for the next token from "#id" to

"#ley". (Note that patching is an essential part of the algorithmic interpretation we carry out in Chapter 8.)

2.5.1.1 Assumptions and Challenges

Circuit Architecture As in semantic interpretation, circuit discovery requires one to target a specific level at which to interpret models – for instance, in probing, this is determined by the expressivity of the selected probe architecture; but in circuit discovery, one must decide on the *atomic* (smallest-scale) sub-graphs considered as possible features or (sub-)circuits. For example, Olah et al. (2020) starts at the level of individual neuron activations; but given the intractability of considering all neurons in larger models, more recent work targeting LLMs has begun at the level of attention heads (Conmy et al., 2023; Wang et al., 2023b). While such tradeoffs are necessary in order to study models at scale, it is important to remember that any operations or algorithms implemented by sub-graphs below or outside the scope of the atomic level – e.g., any analysis of Transformer models built exclusively on attention heads will miss sub-circuits implemented by MLPs (Geva et al., 2021; Meng et al., 2022b).

One-to-One Mapping Another important assumption in circuit discovery is that atomic sub-graphs are understood as performing one and only one operation. Given that even the simplest of models are known to often represent multiple features using the same neuron (Elhage et al., 2022), the assumption that single neurons of LLM architectures can be neatly discretized into individual, atomic operations is suspect; and larger sub-graphs can be difficult to precisely localize and may contain redundant elements (Shi et al., 2024). This assumption can be somewhat attenuated with the interventions discussed above, as they can be used to test the extent to which knocking out or patching a given sub-graph leads to the behavior predicted by the hypothesized circuit, evaluating whether or not the sub-graph actually performs the indicated operation. However, this process only solves part of the problem: while it can test whether or not the sub-graph is indeed involved in implementing the observed behavior, it cannot determine whether there are other sub-graphs that may also have a similar effect (Zhong et al., 2023; Mueller, 2024); and in some cases, knocking out randomized circuits can have a similar effect as knocking out circuits that are precisely calibrated to the target behavior (Shi et al., 2024).

Pre-Specification Another important limitation of current circuit discovery work is that, to our knowledge, all circuits that have so far been identified in real-world LLMs[10] have required pre-specifying a full algorithmic description before they can be found. While this significantly reduces computational complexity and thus allows discovery to scale to LLMs (Conmy et al., 2023), it also means that such discovery can generally only move "top-down": in this case, one cannot discover a circuit that implements an algorithm which has not been explicitly specified ahead of time, and will be less likely to discover intermediate sub-graphs that do not already fit into pre-specified algorithms (a more "bottom-up" approach that has been employed in smaller-scale "toy model" investigations; Elhage et al., 2021; Nanda et al., 2023a; Zhong et al., 2023).

Rashomon Effect Finally, perhaps the greatest challenge in circuit discovery is that it is possible to yield multiple distinct circuit descriptions for the same LLM behavior depending on how circuit analysis is carried out (Zhong et al., 2023; Mueller, 2024) – another instance of the Rashomon effect discussed above in Section 2.4.3.1. For instance, Zhong et al. (2023) explore an "algorithmic phase transition" experiment where a form of patching is used to interpolate between source and target circuits, allowing them to characterize how correctly each circuit describes the internal representation as the balance is shifted between the source and target, finding that, in some cases, it appears that a single model may actually be performing the algorithms associated with each circuit simultaneously.

2.6 Conclusion

In this work, we discussed the parallels between contemporary mechanistic interpretability and research trends in the history of cognitive science, including Marr's levels of analysis and the cognitive revolution that birthed the modern cognitive sciences. We explored the relationship between three broad categories of deep learning interpretability research – *semantic interpretation* (what latent features are represented), *algorithmic interpretation* (what operations are performed over representations), and *implementational interpretation* (how representations, operations, and algorithms are encoded in model activations or parameters) – highlighting the importance of causal analysis across categories in delivering faithful explanations of model behavior. For each category, we surveyed salient works, analyzed their comparative strengths and weaknesses, clarified underlying assumptions, and outlined key challenges; and we discussed the relationships between these categories of work, focusing on the parallel challenges they each face and how they can complement each other.

[10] I.e., LLMs that are not small-scale "toy models" trained specifically for the purpose of algorithmic interpretation studies, as in, e.g., Elhage et al. (2021), Olsson et al. (2022), and Elhage et al. (2022).

Our goal is to facilitate a more open, productive conversation regarding deep learning interpretation by providing a common lexicon of goals, assumptions, and challenges associated with different modes of interpretation, stimulating further research toward more rigorous neural network interpretation, and informing current discourse by consulting lessons from the cognitive sciences.

Chapter 3 Competence-Based Analysis of Language Models

Preface

Chapter 3 is based on *Competence-Based Analysis of Language Models* (CALM; Davies et al., 2024), which forms the intellectual backbone of this dissertation in two important ways:

1. CALM formalizes a key desideratum of semantic interpretability – determining which features a foundation model uses to perform a given task (see Section 2.4) – via causal probing, defining a language model's *linguistic competence* (a useful but somewhat contentious concept borrowed from cognitive science; Chomsky, 1965; Lyons, 1977; Newmeyer, 2001; Sag and Wasow, 2011; Marconi, 1997; Marconi, 2020) in terms of whether it performs the task by leveraging some representation of linguistic structure that actually corresponds to the ground-truth data-generating process of the task, or whether it simply resorts to shortcut learning (e.g., by exploiting spurious correlations).

2. CALM formalizes the connection between (a) using mechanistic interpretability methods to determine what features a model uses to perform a given task, and (b) the model's OOD robustness. As such, it is a key bridge between the mechanistic interpretability work in Part I and the transfer learning work in Part II, paving the way for the integration between these two areas in Part III.

Regarding (1), one of our primary goals in originally defining and measuring this notion of *linguistic competence* was to provide an empirical alternative to the theoretical debate at the time regarding whether language models could be meaningfully said to "understand" language, or were simply "stochastic parrots" that only *seemed* to understand due to shortcut learning at scale (see, e.g., Carlini et al., 2020; Bender and Koller, 2020; Bender et al., 2021; Shojaee et al., 2025). That is, rather than focusing on the precise philosophical nature and definition of what it means to "understand" language – a worthwhile intellectual goal, but also a millennia-old question in the philosophy of language that is unlikely to reach a strong consensus anytime soon (Blackburn, 1995) – we hope to introduce an empirical alternative that could be directly applied to a given language model to quantify the degree to which the *internal latent representations* driving the model's *performance of a given task* were representing the same linguistic features known to drive the *linguistic performance of fluent human speakers* for that task. Given that linguistic competence is already a well-studied concept in linguistics capturing precisely this notion (i.e., a fluent speaker's underlying representation of linguistic structure that enables their linguistic performance),

we feel it was appropriate to adopt this term to stand in for our intended empirical substitute for "understanding".

Finally, in order to study models' representations of *relational* features like the lexical relations analyzed in this work, it was necessary to develop a new method for causal probing interventions (as none of the methods available at the time were directly applicable to features implemented relationally across multiple embedding representations). To address this problem and expand the scope of causal probing to arbitrary feature implementations, we introduce *gradient-based interventions* (GBIs), which use white-box adversarial attacks against arbitrary (e.g., nonlinear) probes to modify embedding representations to encode counterfactual feature values. (In followup work presented in Chapter 4, we show that GBIs are generally much more reliable than a variety of other leading causal probing intervention methods.)

Chapter 3: *Competence-Based Analysis of Language Models* (CALM)

Publication: Adam Davies, Jize Jiang, and ChengXiang Zhai (2024). "Competence-Based Analysis of Language Models". In: *NeurIPS 2024 Workshop on Interpretable AI*. URL: https://openreview.net/forum?id=x6ZM5Is2Po

3.1 Introduction

The rise of large, pretrained neural language models (LLMs) has led to rapid progress in a wide variety of natural language processing tasks (Brown et al., 2020; Chowdhery et al., 2022; Dubey et al., 2024). However, these models can also be quite sensitive to minor changes in input prompts (Elazar et al., 2021a; Moradi and Samwald, 2021; Mizrahi et al., 2024) and fail to generalize outside their training or fine-tuning distribution (Wang et al., 2023a; Yang et al., 2023b). It is usually unclear where these limitations come from, as LLM task performance is typically studied using only "black box" behavioral analysis where limitations can only be detected if they are adequately represented in evaluation datasets, which cannot cover every potentially relevant limitation using a finite dataset (Raji et al., 2021; Siska et al., 2024). Thus, a deeper understanding of how these models can perform as well as they usually do while exhibiting unexpected limitations is critical for ensuring robust, trustworthy, and socially-responsible LLM-enabled applications (Shin, 2021; Liao and Vaughan, 2023; Zou et al., 2023; Bereska and Gavves, 2024), and constitutes a key question in the basic science of LLM interpretation and analysis (Bereska and Gavves, 2024; Anwar et al., 2024).

We approach this question in terms of *competence*, drawing on the traditional competence-performance distinction in linguistic theory to motivate the study of LLMs in

terms of their underlying representation of language. We define LLM competence in the context of a given linguistic task as the alignment between the ground-truth causal structure of the task and the LLM's latent representation of the task's structure, measured by intervening on the LLM's representation of task-causal versus spurious features and observing how its behavior changes in response. Models leveraging causal representations to perform a task generalize better under distribution shift than those that do not (Peters et al., 2016; Arjovsky et al., 2019; Bühlmann, 2020), meaning that more competent LLMs are also expected to exhibit greater robustness to distribution shift.

While the representations of causal or spurious features are not directly observable, we take inspiration from *causal probing*, which intervenes on LLMs' representations of latent features using causal interventions to study how these representations contribute to their behavior (Elazar et al., 2021b; Lasri et al., 2022). We introduce a general model interpretation and analysis framework, CALM (for *Competence-based Analysis of Language Models*), to study and measure LLM competence using causal probing. While CALM can be instantiated using a variety of existing causal probing interventions (e.g., Ravfogel et al., 2020, 2022b,a; Shao et al., 2022; Belrose et al., 2024), we develop a new methodology for intervening on LLM representations, *Gradient-Based Interventions* (GBIs), which use white-box adversarial attacks against supervised probes to modify LLM embedding representations. GBIs are the first causal probing technique that can study arbitrary feature implementations (e.g., linear vs nonlinear; see Section 2.3.1), enabling the first comparative studies of feature implementation across different implementation classes (see Chapter 4). We carry out a case study of CALM using GBIs to intervene on two well-studied LLMs in order to measure and compare their competence across 14 lexical inference tasks, showing that CALM can indeed explain important patterns in behavior across these tasks by distinguishing between models' use of causal versus spurious features.

3.2 Competence-based Analysis of Language Models

3.2.1 Linguistic Competence

Linguistic competence is generally understood as the ability to utilize one's knowledge of a language in producing and understanding utterances in that language, and is typically defined in contrast with linguistic performance, which is speakers' use of language in practice considered independently of the underlying knowledge that supports it (Marconi, 2020).[11]

[11]There has been significant debate in linguistics and the philosophy of language regarding the precise definition and nature of competence (Lyons, 1977; Newmeyer, 2001; Sag and Wasow, 2011; Marconi, 2020). However, the formalization of competence provided in this work is sufficiently general to incorporate most notions of competence, which may be flexibly specified by instantiating CALM in different ways.

Given a linguistic task, we may understand competence in terms of the underlying linguistic knowledge that one draws upon to perform the task. If fluent human speakers rely on (implicit or explicit) knowledge of the same set of linguistic features to perform a given task, then we may understand their performance of this task as being causally determined by these features, and invariant to other features. For example, if we consider the two utterances "the chicken crosses the road" and "the chickens cross the road", the grammatical number of the subject (i.e., singular and plural, respectively) determines whether the verb "(to) cross" should be conjugated as "crosses" or "cross". As English (root) verb conjugation always depends on the grammatical number of the subject, grammatical number may be regarded as having a causal role in the task of English verb conjugation, so we may understand fluent English speakers' mental representation of grammatical number as having a causal role in their behavior.

In this work, we focus on *lexicosemantic competence*, the ability to utilize knowledge of word meaning relationships in performing tasks such as lexical inference (Marconi, 1997; Marconi, 2020). While the study of human competence has a rich history in linguistics (Chomsky, 1965; Lyons, 1977; Newmeyer, 2001; Sag and Wasow, 2011; Marconi, 1997; Marconi, 2020), there is currently no generally accepted framework for studying LLM competence (Mahowald et al., 2024; Pavlick, 2023), a gap which we aim to address in this work.

3.2.2 CALM Framework

In order to make the study of competence tractable in the context of LLMs, we introduce the CALM (Competence-based Analysis of Language Models) framework, which describes an LLM's competence with respect to a given linguistic task in terms of its latent representation of the causal structure of the task.

Task Structure Formally, given supervised task $\mathcal{T} \sim P(\mathcal{X}, \mathcal{Y})$ where the goal is to correctly predict $\mathbf{y} \in \mathcal{Y}$ given $\mathbf{x} \in \mathcal{X}$, and a collection of (discrete) latent features $\mathbf{Z} = \{Z_j\}_{j=1}^{m}$ that are (potentially) involved in generating $\mathbf{x}$, we formulate the causal structure of $\mathcal{T}$ in terms of the data-generating process

$$\mathbf{x} \sim P(\mathbf{x} \mid \mathbf{Z}_c, \mathbf{Z}_e), \quad \mathbf{y} \sim P(\mathbf{y} \mid \mathbf{Z}_c) \tag{1}$$

where $\mathbf{Z}$ may be decomposed into $\mathbf{Z} = \mathbf{Z}_c \cup \mathbf{Z}_e, \mathbf{Z}_c \cap \mathbf{Z}_e = \varnothing$, where $\mathbf{Z}_c$ contains all *causal* features that determine $\mathbf{y}$, and $\mathbf{Z}_e$ are the remaining (*environmental*) features that may be involved in generating $\mathbf{x}$ (cf. Ilse et al., 2021). However, there may be an unobserved

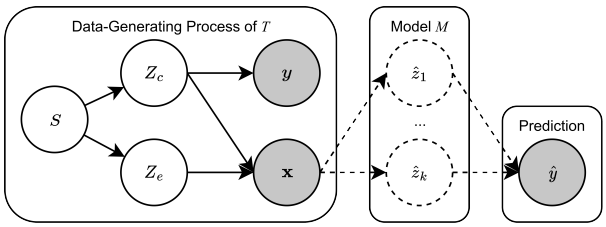


Figure 1: Structural causal model (SCM) of task $\mathcal{T}$'s data-generating process and how it may be performed by model M. Shaded and white nodes denote observed and unobserved variables, respectively. In CALM, the goal is to determine which representations $Z_j = z_j$ are causally implicated in M's predictions $\hat{\mathbf{y}}$.

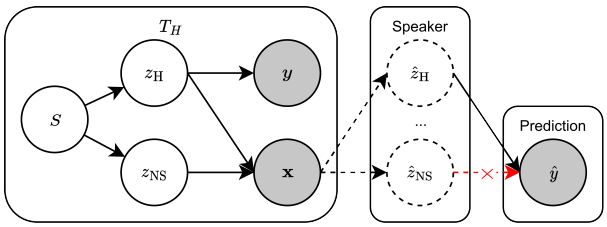


Figure 2: SCM of a competent English speaker on the hypernym prediction task. Shaded and white nodes denote observed and unobserved variables, respectively.

confounder S that produces spurious correlations between $\mathbf{y}$ and $\mathbf{Z}_e$, which, if leveraged by language model M in the course of predicting $\hat{\mathbf{y}}$, can lead to unexpected failures on $\mathcal{T}$ when the spurious association is broken (Pearl, 2009). The structural causal model (SCM) of this data-generating process is visualized on the left side of Figure 1, where each node is a random variable, directed edges denote causal dependencies from *parent* nodes to *child* nodes – e.g., $\mathbf{x}$ is dependent on both parents $\mathbf{Z}_c$ and $\mathbf{Z}_e$; whereas $\mathbf{y}$ is dependent only on $\mathbf{Z}_c$ – and any two nodes which share a common ancestor are *marginally dependent*, meaning that they may be spuriously correlated due to a common ancestor (here, the unobserved "confounder" node, denoted S). For further background on SCMs, we refer the reader to Bongers et al. (2021).

As an example, suppose a speaker wants to communicate that orangutans are a genus of primate. She might say "orangutans are primates" or "orangutans, a genus of apes, are primates". In both cases, the conjugation of the root verb would be "are" because the task of English verb conjugation (denoted $\mathcal{T}_{\text{VC}}$) is invariant to whether the subject is complemented by an appositive phrase like "a genus of apes", and this phrase does not change the grammatical number of the subject "orangutans". Thus, if we define Z_{NS} as the grammatical number of the subject, and Z_{AP} as the presence of an appositive phrase modifying the subject, then $Z_{\text{NS}} \in \mathbf{Z}_c$ and $Z_{\text{AP}} \in \mathbf{Z}_e$ for $\mathcal{T}_{\text{VC}}$. However, if we instead consider the task $\mathcal{T}_{\text{H}}$ of predicting hypernyms – for example, predicting y in "orangutans are $\mathbf{y}$s", where $\mathbf{y}$ = "primate" and $\mathbf{y}$ = "ape" would both be correct answers – then $Z_{\text{NS}} \in \mathbf{Z}_e$ (e.g., the same answers will be correct if the question is instead posed as "an orangutan is a $\mathbf{y}$"), and the hypernymy relation Z_{H} is the causal feature $Z_{\text{H}} \in \mathbf{Z}_c$, with the corresponding SCM visualized in Figure 2.

For other examples of task-causal vs. spurious features, see Part II of this dissertation (Chapter 5 to Chapter 7), in which we study models use of such features at the level of

functional (behavioral) interpretation (as defined in Chapter 2), rather than the level of *semantic* (representational) interpretation (as studied in this chapter, as well as in our followup work in Chapter 4).

Internal Representation Our main concern is measuring how attributable an LLM M's behavior in a given task $\mathcal{T}$ is to its representation of various features $\mathbf{Z} = \{Z_1, ..., Z_m\}$, and how these features correspond to the causal structure of the task. If M respects the data-generating process of $\mathcal{T}$, then its behavior should be attributable only to causal features $Z \in \mathbf{Z}_c$ (and not to environmental features $Z \in \mathbf{Z}_e$), in which case we say that M is *competent* with respect to $\mathcal{T}$ (see Figure 2). We study model M's use of each feature $Z_j \in \mathbf{Z}$ by performing causal interventions $\text{do}(Z_j)$ on its representation of Z_j in the course of performing task $\mathcal{T}$, and measure the impact that these interventions have on its predictions.

Measuring Competence We evaluate the competence of M with respect to task $\mathcal{T} \sim P(\mathcal{X}, \mathcal{Y})$ in terms of its relationship with a *competence graph* $\mathcal{G}_\mathcal{T}$, which we define as a structural causal model (SCM) of $\mathcal{T}$ with nodes corresponding to each latent variables $Z_j \in \mathbf{Z}$ and an additional node for outputs $\mathbf{y} \in \mathcal{Y}$ and directed edges denoting causal dependencies between these variables. That is, the set of causal features $\mathbf{Z}_c$ defined by $\mathcal{G}_\mathcal{T}$ is the set of all features $Z_j \in \mathbf{Z}$ such that there is an edge or path from Z_j to $\mathbf{y}$.

To study the extent to which M's behavior is determined by causal versus spurious dependencies in $\mathcal{G}_\mathcal{T}$, we examine whether M and $\mathcal{G}_\mathcal{T}$ make the same predictions under interventions $\text{do}(\mathbf{z})$, where setting $\mathbf{z} = \{z_j\}_{j=1}^m$ is the set of values $Z_j = z_j$ taken by each corresponding latent variable $Z_j \in \mathbf{Z}$. For example, consider an instance $(\mathbf{x}, \mathbf{y}) \sim \mathcal{T}_H$ of the hypernym prediction task $\mathcal{T}_H$ where input $\mathbf{x}$ = "orangutans are **y**s" and ground-truth output $\mathbf{y}$ = "primate". Here, the values taken by $\mathbf{z}$ would be $Z_\text{H} = 1, Z_\text{NS} = 1$ (where 1 indicates the presence of hypernymy and a plural noun subject, respectively), and we might define an alternative $\mathbf{z}'$ where $Z_\text{H} = 0, Z_\text{NS} = 1$, under which a competent model's prediction would be expected to change with the causal variable Z_H (i.e., $M(\mathbf{x} \mid \text{do}(\mathbf{z}')) \neq M(\mathbf{x})$).

The alignment of M with $\mathcal{G}_\mathcal{T}$ is measured in terms of the similarity S of their predictions under interventions $\text{do}(\mathbf{z})$ given input $\mathbf{x} \sim P(\mathcal{X})$, and can be computed using a given similarity metric $S : \mathcal{Y} \times \mathcal{Y} \to [0, 1]$ (e.g., equality, n-gram overlap, cosine similarity, etc.) depending on the SCM $\mathcal{G}_\mathcal{T}$ and output space $\mathcal{Y}$. That is, we define $\mathcal{C}_\mathcal{T}(M \mid \mathcal{G}_\mathcal{T})$ as M's competence with respect to task $\mathcal{T}$ as a function of its alignment with corresponding task SCM $\mathcal{G}_\mathcal{T}$ under interventions $\text{do}(\mathbf{z})$ measured by similarity metric S, as follows:

$$\mathcal{C}_\mathcal{T}(M \mid \mathcal{G}_\mathcal{T}) = \mathbb{E}_{\mathbf{x},\mathbf{z} \sim P(\mathcal{X}, \text{val}(\mathbf{Z}))} S\big(M(\mathbf{x} \mid \text{do}(\mathbf{z})), \mathcal{G}_\mathcal{T}(\mathbf{x} \mid \text{do}(\mathbf{z}))\big) \tag{2}$$

This $\mathcal{C}_{\mathcal{T}}(M \mid \mathcal{G}_{\mathcal{T}})$ metric (bounded by $[0, 1]$) is an adaptation of the Interchange Intervention Accuracy (IIA) metric (Geiger et al., 2022, 2023) to the context of causal probing, where instance-level interventions are replaced with concept-level interventions enabled by the gradient-based intervention methodology we introduce in Section 3.3. (See Section B.3.1 for a detailed comparison of our competence metric with IIA.)

Causal Probing A key technical challenge in implementing CALM (and causal probing more generally) is designing an algorithm to perform causal interventions $\text{do}(Z)$ that maximally damage the representation of a feature Z while otherwise minimally damaging representations of other features Z' (Canby* et al., 2025; see Chapter 4). For example, *amnesic probing* (Elazar et al., 2021b) uses the INLP algorithm (Ravfogel et al., 2020) to produce interventions g_Z that remove all information that is linearly predictive of feature Z from a pre-computed set of embedding representations $\mathbf{H}$, showing that BERT makes variable use of parts-of-speech, syntactic dependencies, and named-entity types in performing masked language modeling. However, Elazar et al. (2021b) also found that, when INLP is used to remove BERT's representation of these features in early layers, it is often able to "recover" this representation in later layers, which is likely due to BERT encoding these features nonlinearly; and later work has found that the same "recoverability" problem persists even when linear information removal methods like INLP are kernelized (Ravfogel et al., 2022b). Thus, it is necessary to develop interventions that do not require restrictive assumptions about the structure of LLMs' representations (e.g., linearity), a problem which we aim to address in the following subsection.

3.3 Gradient-Based Interventions

Our goal in developing gradient-based interventions (GBIs) as a causal probing technique is to enable interventions over arbitrarily-encoded LLM representations. GBIs allow users to flexibly specify the class of representations they wish to target, expanding the scope of causal probing to arbitrarily-encoded features. We take inspiration from Kos et al. (2018), who developed a technique to perturb latent representations using gradient-based adversarial attacks.[12] They begin by training probe $g_Z : \mathbf{h} \mapsto z$ to predict image class $z \in Z$ from latent representations $\mathbf{h} = f_{\text{enc}}(\mathbf{x})$ of images $\mathbf{x}$, where f_{enc} is the encoder of a VAE-GAN (Larsen et al., 2016) trained on an unsupervised image reconstruction task (i.e., $f_{\text{dec}}(f_{\text{enc}}(\mathbf{x})) = \hat{\mathbf{x}} \approx \mathbf{x}$, for decoder f_{dec} and reconstructed image $\hat{\mathbf{x}}$ approximating $\mathbf{x}$). Next, gradient-based attacks like FGSM (Goodfellow et al., 2015) and PGD (Madry et al., 2017)

[12]Notably, Tucker et al. (2021) developed a similar methodology without explicit use of such attacks (see Section 3.6).

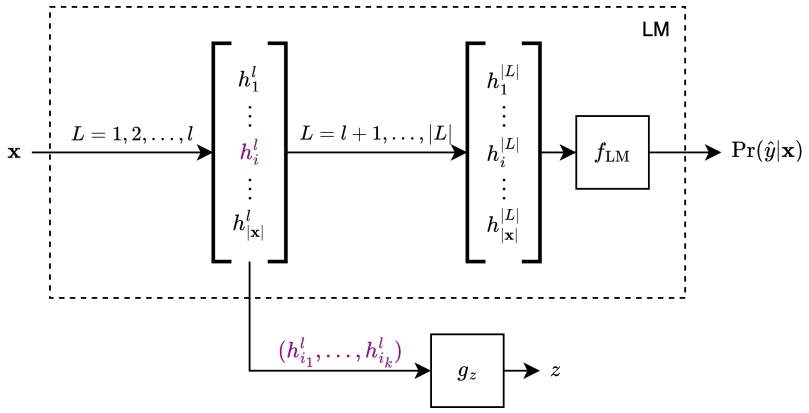


Figure 3: **Gradient-Based Interventions.** Input tokens $\mathbf{x} = (x_1, ..., x_{|\mathbf{x}|})$ are passed through layers $L = 1, ..., l$, where embedding $\mathbf{h}_i^l$ (encoding the value $Z = z$) is extracted from layer l and given to g_Z as input. Next, the embedding is modified by gradient-based attacks on g_Z to encode the counterfactual value $Z = z'$, then fed back into subsequent layers $L = l + 1, ..., |L|$ and language modeling head f_{LM} to obtain the intervened predictions $M(\mathbf{x} \mid \text{do}(Z = z'))$.

are performed against g_Z in order to minimally manipulate $\mathbf{h}$ such that it resembles encoded representations of target image class $Z = z'$ (where $z' \neq z$, the original image class), yielding perturbed representation $\mathbf{h}'$. Finally, $\mathbf{h}$ and $\mathbf{h}'$ are each fed into the VAE decoder to reconstruct corresponding output images $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}'$ (respectively), where $\hat{\mathbf{x}}$ resembles input image class $Z = z$ and $\hat{\mathbf{x}}'$ resembles target class $Z = z'$.

We reformulate this approach in the context of causal probing as visualized in Figure 3, treating layers $L = 1, ..., l$ as the encoder and layers $L = l + 1, ..., |L|$ (composed with language modeling head f_{LM}) as the decoder, allowing us to target representations of feature Z across embeddings $\mathbf{h}_i^l$ of token $x_i \in \mathbf{x}$ in layer l. We train g_Z to predict Z from a set of such $\mathbf{h}_i^l$, then attack g_Z using FGSM and PGD to intervene on $\mathbf{h}_i^l$ (representing the original value $Z = z$), producing $\mathbf{h}_i^{l'}$ (representing the counterfactual value $Z = z'$). Finally, we replace $\mathbf{h}_i^l$ with $\mathbf{h}_i^{l'}$ in the LLMs' forward pass from layers $L = l + 1, ..., |L|$, simulating the intervention $\text{do}(Z = z')$, and observe the impact on its word predictions $M(\mathbf{x} \mid \text{do}(Z = z'))$. (See Section B.2.3 for further details.)

Benefits and Drawbacks The key advantage of gradient-based interventions (GBIs) as a causal probing methodology is that they may be applied to any differentiable probe. For example, if we are investigating the hypothesis that M's representation of Z is captured by a linear subspace of representations in a given layer (cf. Vargas and Cotterell, 2020), then we may train a linear probe and various nonlinear probes on representations and observe whether GBIs against the linear probe have a comparable impact to those against the

nonlinear probes. Alternatively, if we believe that a probe's architecture should mirror the architecture of the model it is probing (as argued by Pimentel et al., 2022), we may implement probes as such. Finally, where previous intervention methodologies for causal probing have focused on *nullifying* interventions that remove the representation of the target feature Z (Ravfogel et al., 2020, 2022b,a; Shao et al., 2022; Belrose et al., 2024), GBIs allow one to perform targeted interventions that set LLMs' representations to counterfactual values $\text{do}(Z = z')$, effectively simulating the model's behavior under counterfactual inputs, which may be useful for predicting behaviors under various distribution shifts (see Section B.3.1). However, the benefits associated with GBIs do come with some important limitations, as we discuss in Section B.1.1.

3.4 Experiments

Our primary goal in the following experiments is to develop and test an experimental implementation of CALM using GBIs in the context of comparatively small, well-studied models and tasks in order to validate whether CALM can explain behavioral findings of earlier work in this simplified environment. (We motivate this choice in greater detail in Section B.1.2.) Thus, we begin by examining BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019b),[13] two language models which have already been extensively studied in the context of probing (Rogers et al., 2021; Ravfogel et al., 2020; Liu et al., 2021; Elazar et al., 2021b; Lasri et al., 2022).

Tasks We use the collection of 14 lexical inference tasks included in the ConceptNet (Speer et al., 2017) subset of LAMA (Petroni et al., 2019), which are each formulated as a collection of cloze prompts (Liu et al., 2023b). For example, the LAMA "IsA" task contains ∼2K hypernym prompts corresponding to the "IsA" ConceptNet relation (including, e.g., "A laser is a `[MASK]` which creates coherent light", where the task is to predict that the `[MASK]` token should be replaced with "device", a hypernym of "laser"), with the remaining 13 LAMA ConceptNet tasks corresponding to other lexical relations such as "PartOf", "Hasfeature", and "CapableOf". (See Section B.2.1 for additional details.)

Using these task datasets allow us to test how the representation of each relation is used across all other tasks. In the context of a single task $\mathcal{T}_j$, intervening on a model's representation of the task-causal relation Z_j allows us to measure the extent to which its predictions are attributable to its representation of the causal feature $\mathbf{Z}_c = \{Z_j\}$ (where a large impact indicates competence). On the other hand, intervening on the representations of the other 13 lexical relations $Z_k \in \mathbf{Z}_e$ allows us (in the aggregate) to measure how much

[13] Specifically, `BERT-base-uncased` and `RoBERTa-base` (Wolf et al., 2019).

the model is performing task $\mathcal{T}_j$ by leveraging representations of general, non-causal lexical information (where a large impact indicates incompetence).[14]

Experimentally Measuring Competence Given LLM M and task $\mathcal{T}$, measuring the empirical competence $\hat{\mathcal{C}_{\mathcal{T}}}(M \mid \mathcal{G}_{\mathcal{T}})$ of M given $\mathcal{G}_{\mathcal{T}}$ requires us to specify an experimental model $E = (\mathbf{Z}, \mathcal{G}_{\mathcal{T}}, S)$, where $\mathbf{Z}$ is a set of features, $\mathcal{G}_{\mathcal{T}}$ is a competence graph for task $\mathcal{T}$, and S is a scoring function that compares the predictions of M and $\mathcal{G}_{\mathcal{T}}$. Given that each task $\mathcal{T}_i$ is defined by a single causal lexical relation Z_i (i.e., $\mathbf{Z}_{c_i} = \{Z_i\}$), we model settings $\mathbf{z}$ as a collection of values $Z_j = z_j$ taken by each feature Z_j in the context of a specific task instance $(\mathbf{x}, \mathbf{y}) \sim \mathcal{T}_i$, where $Z_j = 1$ if $i = j$ (i.e., where the feature Z_j is the causal feature for the task $\mathcal{T}_i$) or $Z_j = 0$ otherwise. That is, for each instance $(\mathbf{x}, \mathbf{y}) \sim \mathcal{T}_i$, the corresponding setting $\mathbf{z}$ is a one-hot vector whose i-th element $z_i = 1$. We may specify $\mathcal{G}_{\mathcal{T}_i}$ in a similar manner: for task $\mathcal{T}_i \sim P(\mathcal{X}, \mathcal{Y})$, outputs $\mathbf{y} \in \mathcal{Y}$ are causally dependent on the feature Z_i and invariant to other concepts $Z_j, j \neq i.$, meaning that the only direct parent node of $\mathbf{y}$ in $\mathcal{G}_{\mathcal{T}_i}$ is Z_i. Finally, as we are dealing with masked language models whose output space $\mathcal{Y}$ for each task consists only of single tokens in M's vocabulary V_M, our experimental model can define the scoring function S as the overlap $\text{overlap}(\mathbf{y}_i, \mathbf{y}_j)$ for top-k token predictions $\mathbf{y}_i = \{y_1, ..., y_k\} \subset V_M$, where $\text{overlap}(\cdot, \cdot)$ is the size of the intersubsection of each set of predictions divided by the total number of predictions $\text{overlap}(\mathbf{y}_i, \mathbf{y}_j) = \frac{|\mathbf{y}_i \cap \mathbf{y}_j|}{k}$, and $\hat{\mathcal{C}_{\mathcal{T}k}}$ denotes the empirical competence $\hat{\mathcal{C}_{\mathcal{T}}}$ as measured using only the top-k token predictions $\mathbf{y}_i$. (See Section B.3.2 for additional details on how we compute competence in each experiment.)

Probes We implement probes g_Z as a 2-layer MLP over each language model's final hidden layer, and train the probe on the task of classifying whether there is a particular relation Z between a final-layer `[MASK]` token in the context of a cloze prompt and the final-layer object token from the "unmasked" version of the same prompt. All reported figures are the average of 10 runs of our experiment, using different randomly-initialized g_Z each time. (See Section B.2.2 for further details.)

Interventions We implement GBIs against g_Z using two gradient attack strategies, FGSM (Goodfellow et al., 2015) and PGD (Madry et al., 2017). We bound the magnitude of each intervention as follows: where h is the input to g_Z and h' is the intervened representation following a GBI, $||h - h'||_\infty \leq \epsilon$. For all experiments reported in our main paper, we use FGSM with $\epsilon = 0.1$. (See Section B.2.3 for more details and PGD results.)

[14]Note that this experimental formulation makes the simplifying assumption that each environmental feature is equally (un)related to the target feature, which is not necessarily true; see Section B.1.3.

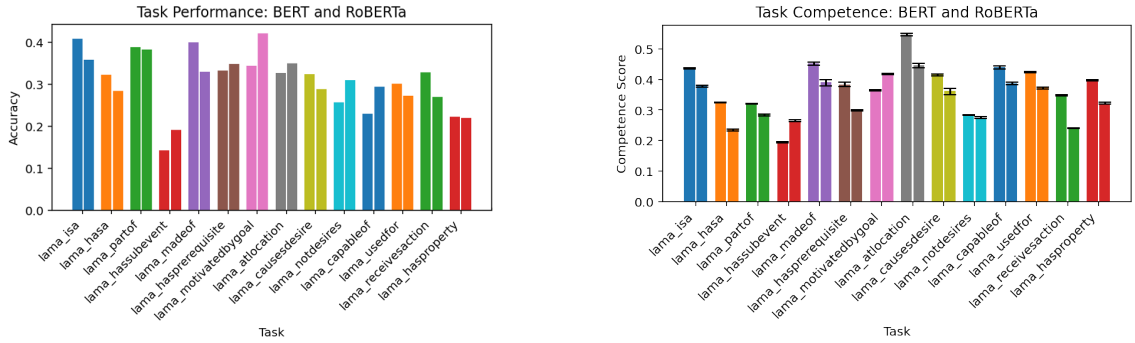


Figure 4: Performance (left) and competence (right) of BERT (left bars) and RoBERTa (right bars) for all tasks, using FGSM with $\epsilon = 0.1$. In the competence plot, y-values are the average competence score and error bars are the maximum and minimum competence score, as measured over 10 experimental iterations (each with a different randomly-initialized probe g_Z).

3.5 Results

In Figure 4, we visualize the performance and competence of BERT and RoBERTa across the test set of each LAMA ConceptNet task. Performance is measured using $(0, 1)$-accuracy, competence is measured using the experimental competence metric in Equation (12), and both metrics are averaged across the top-k predictions of each model for $k \in [1, 10]$. That is, for ground truth $(\mathbf{x}, y)$ and $n = 10$, we compute accuracy and competence as follows:

$$\text{acc}(M) = \frac{1}{n}\sum_{k=1}^{n} \mathbb{1}[y \in \underset{\hat{y}}{\text{top-}k}\, P_M(\hat{y} \mid \mathbf{x})] \qquad \text{and} \qquad \hat{\mathcal{C}_{\mathcal{T}}}(M \mid \mathcal{G}_{\mathcal{T}}) = \frac{1}{n}\sum_{k=1}^{n} \hat{\mathcal{C}_{\mathcal{T}k}}(M \mid \mathcal{G}_{\mathcal{T}})$$

To account for stochasticity in initializing and training probes g_Z, scores are also averaged over 10 randomized experiments for each target task where the probe is randomly re-initialized each time (resulting in different GBIs).

3.5.1 Analysis

Performance While their accuracies on individual tasks vary, BERT and RoBERTa have quite similar aggregate performance: BERT outperforms RoBERTa on just over half (8/14) of the tasks, achieving essentially equivalent performance when averaged across all tasks (0.3099 versus 0.3094).

Competence Given our experimental model E with $m = 14$ tasks, consider a random baseline language model R whose predictions always change in response to each intervention, making equal use of all features in each task. R would yield a competence score of

$\mathcal{C}(R \mid \mathcal{G}_{\mathcal{T}}) = \frac{1}{m} \approx 0.0714$ for each task. Both BERT and RoBERTa score well above this threshold for all tasks, meaning that they are much more competent than a model that does not distinguish between causal and environmental features. RoBERTa is consistently less competent than BERT (on 12/14 tasks), and also has lower competence scores averaged across all tasks (0.381 vs. 0.334); but for the two tasks where RoBERTa is more competent than BERT, it also achieves substantially higher performance. More generally, relative performance and competence are correlated: the Spearman's Rank correlation coefficient between the average difference in accuracy and average difference in performance is a moderately strong positive correlation $\rho = 0.508$ with significance $p = 0.064$.

Explananda The performance of BERT and RoBERTa on lexical inference tasks such as hypernym prediction has been shown to be highly variable under small changes to prompts (Hanna and Mareček, 2021; Ravichander et al., 2020; Ettinger, 2020; Elazar et al., 2021a). Our findings offer a potential explanation for such brittle performance: BERT and RoBERTa's partial competence in hypernym prediction indicates that it should be possible to prompt these models in a way that will yield high performance, but that its reliance on spurious lexical associations may lead it to fail when these correlations are broken – e.g., by substituting singular terms for plurals (Ravichander et al., 2020) or paraphrasing a prompt (Elazar et al., 2021a).

3.6 Related Work

Causal Probing Most related to our work is amnesic probing (Elazar et al., 2021b), as discussed in Section 3.2.2. Lasri et al. (2022) applied amnesic probing to study the use of grammatical number representations in performing an English verb conjugation prompt task. As this experiment involves intervening on the representation of a feature which is causal with respect to the prompt task, it may be understood as an informal instantiation of CALM (albeit without considering environmental features or measuring competence).

Gradient-based Interventions Tucker et al. (2021) developed an approach similar to GBIs without explicit use of gradient-based adversarial attacks. Their methodology is equivalent to performing a targeted, unconstrained attack, where gradient updates are continually applied to embeddings until the target probe loss saturates (irrespective of perturbation magnitude). In such attacks, it is standard practice to constrain the magnitude of resulting perturbations (Goodfellow et al., 2015; Madry et al., 2017; Kos et al., 2018), which we do here in order to minimize the effect of "collateral damage" done by such attacks (see Section B.2.3). In our followup work (Chapter 4; Canby* et al., 2025), we study the

collateral damage (low selectivity) of a variety of causal probing methods, finding that it is indeed necessary to constrain highly expressive and general methods such as GBIs in order to limit such damage. However, when appropriately constrained, we find that GBIs are generally more reliable than all other evaluated causal probing methods, including leading linear methods such as INLP (Ravfogel et al., 2020), RLACE (Ravfogel et al., 2022a), and AlterRep (Ravfogel et al., 2021).

Unsupervised Probing Instead of training supervised probes to predict a pre-specified feature of interest (as we do here), an alternative approach is to train *unsupervised* probes such as Sparse Auto-Encoders (SAEs; Subramanian et al., 2018; Yun et al., 2021; Cunningham et al., 2023), which learn a "dictionary" of features that can be used to sparsely represent embeddings, and can also be used to control models' use of these learned features (Bricken et al., 2023; Templeton et al., 2024; see Section 2.4.4). Unlike in supervised approaches to semantic interpretation like (causal) probing, unsupervised dictionary features must be retroactively interpreted in order to determine their relationship to a given task (see Section 2.4); but given a suitable approach to interpreting such features (see, e.g., Bills et al., 2023; Paulo et al., 2025) and a sufficiently reliable method for intervening on them (cf. Chapter 4), it would also be possible to implement CALM using unsupervised probes like SAEs.

3.7 Conclusion

In this work, we introduced CALM, a general analysis framework that enables the study of LLMs' linguistic competence using causal probing, including the first quantitative measure of linguistic competence. We developed the gradient-based intervention (GBI) methodology, the first causal probing method that can target arbitrarily implemented features, expanding the scope of causal probing to new questions in LLM interpretability and analysis. Finally, we carried out a case study of CALM using GBIs, analyzing BERT and RoBERTa's competence across a collection of lexical inference tasks, finding that even a simple experimental model is sufficient to explain their behavior across a variety of lexical inference tasks.

Chapter 4 How Reliable are Causal Probing Interventions?

Preface

Causal probing aims to analyze foundation models by examining how intervening on their representation of various latent features impacts their outputs. In CALM (Chapter 3), we introduced *gradient-based interventions* (GBIs), a flexible and powerful class of causal probing interventions that can modify model representations under arbitrary implementation classes (e.g., linear vs nonlinear representation; see Section 2.3.1) by applying gradient-based adversarial attacks against different classes of probes. Furthermore, in contrast to prior methods, GBIs allow for precise modulation of the resulting perturbation magnitude in embedding spaces. Notably, prior work has found that, in certain scenarios, linear concept-removal interventions used for causal probing (e.g., ILNP; Ravfogel et al., 2020) can be unreliable for interpreting the features used by LLMs, either because the intended intervention is not fully carried out in the latent space (Elazar et al., 2021b; Ravfogel et al., 2022b, 2023), or because interventions inadvertently damage non-targeted features (Kumar et al., 2022; Belrose et al., 2024; Dobrzeniecka et al., 2025). In this chapter, we aim to more precisely characterize and study such concerns, and examine how they apply to more flexible and powerful causal probing methods such as GBIs.

That is, in **Chapter 4**, which is based on *How Reliable are Causal Probing Interventions?* (Canby*, Davies*, et al., 2025), we formalize the problem of causal probing in terms of *completeness* (how thoroughly the representation of the target feature has been transformed), *selectivity* (how little non-targeted features have been impacted), and *reliability* (the harmonic mean of completeness and selectivity, capturing the inherent tradeoff between these two desiderata). We introduce an empirical analysis framework to measure and evaluate these quantities, allowing us to make the first direct comparisons between different families of leading causal probing methods (e.g., linear vs. nonlinear, or concept removal vs. counterfactual interventions). We find that all tested methods do indeed show a clear tradeoff between completeness and selectivity, that more complete and reliable methods have a greater impact on LLM behavior, and that nonlinear interventions like GBIs are almost always more reliable than linear interventions. In particular, across all models and nearly all layers, we find that GBIs have the greatest "ceiling reliability" – that is, when precisely calibrated via the reliability validation procedure introduced below, they are capable of achieving substantially higher reliability scores than simpler, more traditional alternatives – but such calibration is much more important for expressive and flexible methods like GBIs than it is for methods that assume linearity or explicitly minimize perturbation magnitude, as their reliability is much more variable due to the inherent

flexibility among the many ways they can be configured (e.g., selecting probe architecture, gradient-based adversarial attack, perturbation magnitude, or other intervention hyperparameters). These findings suggest that LLMs do indeed represent features nonlinearly; but that reliable interpretation of nonlinear feature representations requires careful calibration in order to avoid causing excessive "collateral damage".

Chapter 4: *How Reliable are Causal Probing Interventions?*

Publication: Marc Canby*, Adam Davies*, Chirag Rastogi, and Julia Hockenmaier (2025). "How Reliable are Causal Probing Interventions?" In: *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics. URL: https://aclanthology.org/2025.ijcnlp-long.47/

(denotes co-primary authorship.)*

4.1 Introduction

What latent features do large language models (LLMs) learn to represent, and how do they leverage such representations? Causal probing (as studied in Chapter 3) aims to answer this question by intervening on a model's embedding representations of some feature of interest (e.g., parts-of-speech), feeding the altered embeddings back into the LLM, and assessing how the model's behavior on downstream tasks changes (Geiger et al., 2020; Ravfogel et al., 2020; Elazar et al., 2021b; Tucker et al., 2021; Lasri et al., 2022; Zou et al., 2023; Davies et al., 2024). However, it is only possible to draw meaningful conclusions about the model's use of the latent feature if we are confident that interventions have fully and precisely carried out the intended transformation; and several prior works have raised serious doubts about causal probing, finding that many intervention methods may have an unintended impact on non-targeted features (Kumar et al., 2022; Belrose et al., 2024; Dobrzeniecka et al., 2025), or that the original value of the feature may still be recoverable (Elazar et al., 2021b; Ravfogel et al., 2022b; Belrose et al., 2024). So far, it has been unclear how these doubts generalize to other types of interventions or how serious they are in practice, as there is no generally accepted approach for evaluating or comparing different methods.

Thus, our main goal in this study is to work toward a systematic understanding of the effectiveness and limitations of current causal probing methodologies. Specifically, we propose an empirical analysis framework to evaluate the *reliability* of causal probing according to two key desiderata:

1. *Completeness*: interventions should fully transform the representation of targeted features.

2. *Selectivity*: interventions should not impact non-targeted features.

We define completeness and selectivity using "validation probes" that enable measuring the impact of an intervention on both targeted and non-targeted features. We apply our framework to several intervention methods and LLMs, observing that each method exhibits a clear tradeoff between these criteria. We also show that the most complete and reliable interventions lead to the largest and most consistent impact in LLM task performance. Finally, we find a substantial difference between the reliability of linear versus nonlinear interventions, where nonlinear methods are almost universally more reliable than linear methods across LLMs and between different layers. This suggests that interventions assuming linear implementation (e.g., those which assume the linear representation hypothesis; see Vargas and Cotterell, 2020; Ravfogel et al., 2020, 2022a; Tigges et al., 2023; Burns et al., 2023; Jiang et al., 2024; Park et al., 2024b,a) may yield inaccurate interpretations of model internals and behaviors. Finally, our framework also provides the first concrete basis for calibrating intervention hyperparameters to balance completeness and selectivity, allowing for more reliable interpretation of LLMs using existing methods.

4.2 Background and Related Work

(Causal) Probing Traditional probing aims to analyze which features (e.g., part-of-speech, sentiment labels, etc.) are represented by a deep learning model (e.g., LLM) by training auxiliary neural classifiers to predict such features from latent embeddings – see Section 2.4.2 for technical details and further discussion. An important criticism of such claims is that *correlation does not imply causation*: that is, the mere fact that a given feature can be predicted from embedding representations does not imply that the model is actually using that feature (Hewitt and Liang, 2019; Elazar et al., 2021b; Belinkov, 2022; Davies et al., 2024). A prominent response to this concern has been *causal probing*, which uses interventions over trained probes to remove or alter the model's representation of the probed feature, then measures the impact of such interventions on the model's predictions to analyze how the model makes use of the targeted feature, as visualized in the top-left gray region of Figure 5 – see Section 2.4.3 for technical details and further discussion.

Causal Probing: Limitations Prior works studying causal probing interventions have indicated that information about the target feature that should have been completely removed by interventions may still be recoverable by the model (Elazar et al., 2021b;

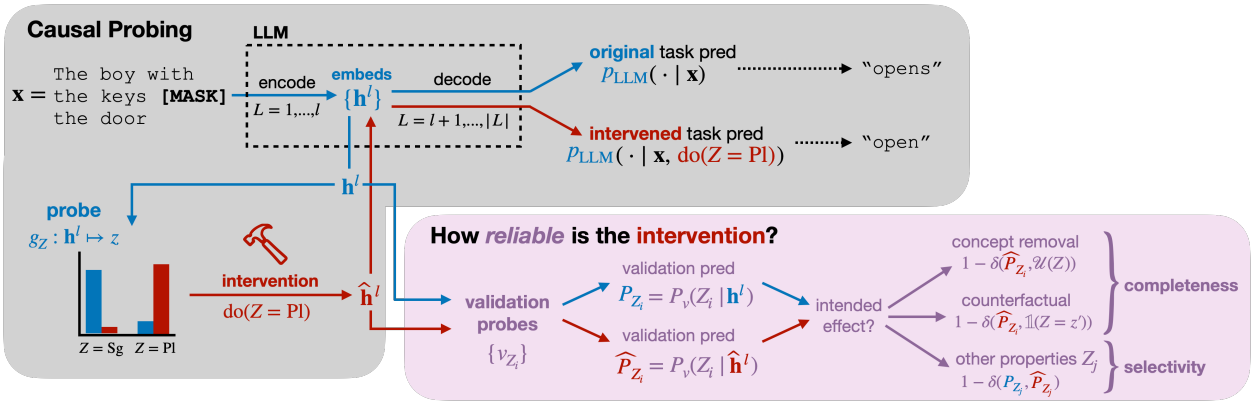


Figure 5: **Causal Probing and Our Reliability Framework.** The process of causal probing is shown in the gray box, with our reliability framework in the purple box.

- *Causal Probing:* embeddings $\mathbf{h}^l$ are extracted from layer $L = l$ of a model and used to train a probe g_Z to predict the value $Z = z$ of feature Z from embeddings (e.g., the number of the subject, `boy`, is $Z = \texttt{Sg}$ for singular). A causal probing intervention $\text{do}(Z = \texttt{Pl})$ uses the probe g_Z to modify the representation encoded by $\mathbf{h}^l$ to encode plural instead. The resulting intervened embedding $\hat{\mathbf{h}}^l$ is fed back into the model at layer $L = l + 1$ and the forward pass is completed, changing the original prediction `opens` to the intervened prediction `open`.

- *Reliability Framework:* instead of feeding the intervened embedding $\hat{\mathbf{h}}^l$ back into the model, it is passed alongside $\mathbf{h}^l$ to validation probes $\{v_{Z_i}\}$ that independently test whether the intervention has had the intended effect. Completeness is measured as the similarity between the validation probe prediction and the target distribution for the intervention (e.g., a perfectly *complete* counterfactual intervention $\text{do}(Z = \texttt{Pl})$ would lead validation probe v_Z to predict plural with probability $P_v(Z = \texttt{Pl} \mid \hat{\mathbf{h}}^l) = 1$), and selectivity is the similarity between the validation probe distribution for non-targeted features before and after the intervention (which, for a perfectly *selective* intervention, should not change).

Ravfogel et al., 2022b, 2023), in which case interventions are not *complete*; or that most of the impact of interventions may actually be the result of collateral damage to non-targeted features (Kumar et al., 2022; Belrose et al., 2024; Dobrzeniecka et al., 2025), in which case interventions are not *selective*. How seriously should we take such critiques? We observe several important shortcomings in each of these prior studies on the limitations of causal probing interventions:

1. These limitations have only been empirically demonstrated for the task of removing information about a target feature from embeddings such that the model *cannot be fine-tuned to use the feature for downstream tasks* (Kumar et al., 2022; Ravfogel et al., 2022b, 2023). But considering that the goal of causal probing is to interpret the behavior of an existing pre-trained model, the question is not whether models *can* be fine-tuned to use the feature; it is whether models *already* use the feature without task-specific fine-tuning, which has not been addressed in prior work. Do we observe the same limitations in this context?

2. These limitations have only been studied for linear concept removal interventions (e.g., Ravfogel et al., 2020, 2022a), despite the recent proliferation of other causal probing methodologies, including nonlinear (Tucker et al., 2021; Ravfogel et al., 2022b; Shao et al., 2022; Davies et al., 2024) and counterfactual interventions (Ravfogel et al., 2021; Tucker et al., 2021; Davies et al., 2024) (see Section 4.4). Do we observe the same limitations for, e.g., nonlinear counterfactual interventions?

In this work, we answer both questions by providing a precise, quantifiable, and sufficiently general definition of completeness and selectivity that it is applicable to *all* such causal probing interventions, and carry out extensive experiments to evaluate representative methods from each category of interventions when applied to a pre-trained LLM as it performs a zero-shot prompt task.

Causal Probing: Evaluation Note that, while we are the first to define and measure the completeness and selectivity of *causal probing interventions*, RAVEL (Huang et al., 2024) provides a broadly analogous evaluation framework and dataset with respect to *interchange interventions*. Methods for performing interchange interventions over embedding representations of a given feature are trained on counterfactual minimal pairs of the feature (i.e., two inputs which are identical in all respects except the input feature; Geiger et al., 2020; Vig et al., 2020a; Geiger et al., 2024). In contrast, causal probing does not require minimal pairs for training probes or performing interventions, allowing our empirical analysis to be carried out without access to such data.

4.3 Evaluating Causal Probing Reliability

Recall that our main goal in this work is to evaluate intervention reliability in terms of completeness (completely transforming M's representation of some target feature Z_i) and selectivity (minimally impacting M's representation of other features $Z_j \neq Z_i$).[15] Given that we cannot directly inspect what value M encodes for any given feature Z_i, it is necessary to introduce the notion of *validation probes*, which we use to measure the extent to which interventions have fulfilled either criterion. Our complete reliability framework is visualized in Figure 5.

Validation Probes Given model M, input $\mathbf{x}$, and embedding representations[16] $\mathbf{h} = M_l(\mathbf{x})$ extracted from some layer l of M, we may define validation probe v as a probe (trained independently from interventional probes; see Section 4.4) that returns a distribution $P_v(Z \mid \mathbf{h})$ over the values of feature Z given embedding $\mathbf{h}$, and we interpret $P_v(Z = z \mid \mathbf{h})$ as the degree to which the model's embedding representations $\mathbf{h}$ given natural-language input $\mathbf{x}$ encode the belief that $\mathbf{x}$ has the feature $Z = z$. Thus, if $\mathbf{h}$ encodes value $Z = \hat{z}$ with complete certainty, v should return a degenerate distribution $P_v(Z \mid \mathbf{h}) = \mathbb{1}(Z = \hat{z})$, whereas we would expect a uniform distribution $P_v(Z \mid \mathbf{h}) = \mathcal{U}(Z)$ if $\mathbf{h}$ does not encode feature Z at all.[17] Thus, validation probes enable us to estimate how well various intervention methods carry out the target transformation. (See Section 4.4 for details on validation probe training.)

Completeness If a counterfactual intervention $\text{do}(Z = z')$ is perfectly *complete*, then it would produce a perfectly-intervened $\mathbf{h}^*_{Z=z'}$ that fully transforms $\mathbf{h}$ from encoding value $Z = z$ to encoding counterfactual value $Z = z' \neq z$. Thus, after performing the intervention, validation probe v should emit $P_v(Z = z' \mid \mathbf{h}^*_{Z=z'}) = P^*_Z(Z = z') = 1$. For concept removal interventions $\text{do}(Z = 0)$, a perfectly complete representation $\mathbf{h}^*_{Z=0}$ should not encode Z at all: $P_v(Z \mid \mathbf{h}^*_{Z=0}) = P^*_Z(Z) = \mathcal{U}(Z)$.[18]

[15] In this work, we use selectivity in the sense described by Elazar et al. (2021b), and not other probing work such as Hewitt and Liang (2019), where it instead refers to the gap in performance between probes trained to predict real features versus nonsense features.

[16] For simplicity, we omit the superscript l from the technical description of causal probing provided in Section 2.4.3 that denotes the layer l from which embeddings $\mathbf{h}^l$ are extracted. However, our framework can be applied to study interventions over embeddings from any layer – see, e.g., Section 4.5.2.

[17] A validation probe's prediction is subtly different from the prediction an arbitrary classifier should make in the absence of any evidence about Z: such a classifier should revert to the empirical distribution $\hat{P}(Z)$.

[18] This is only expected when using concept removal interventions *for causal probing* – i.e., when intervening on a model's representation and feeding it back into the model to observe how the intervention modifies its behavior. When considering concept removal interventions *for the explicit task of concept removal* (which is naturally a more common setting), a more appropriate "goal" distribution P^*_Z would be $P(Z)$, the label distribution – see Appendix C.1 for further discussion.

We can use any distributional distance metric $\delta(\cdot, \cdot)$ bounded by [0, 1] to determine how far the observed distribution $\hat{P}_Z = P_v(Z \mid \hat{\mathbf{h}}_Z)$ is from the "goal" distribution P^*_Z. Throughout this work, we use total variation (TV) distance, which allows us to directly compare counterfactual and concept removal distributions: in both cases, $0 \leq c(\hat{\mathbf{h}}_Z) \leq 1$, where attaining 1 means the intervention had its intended effect in transforming the encoding of Z. Finally, for a given set of test embeddings $\mathbf{H} = \{\mathbf{h}^k\}_{k=1}^n$, the aggregate completeness over this test set $C(\mathbf{H}_Z)$ is the average $c(\hat{\mathbf{h}}^i_Z)$ across all $\mathbf{h}^k \in \mathbf{H}$.

For **counterfactual interventions**, we measure completeness as:

$$c(\hat{\mathbf{h}}_Z) = 1 - \delta(\hat{P}_Z, P^*_Z) \tag{3}$$

If the intervention is perfectly complete, then $\hat{P}_Z = P^*_Z$ and $c(\hat{\mathbf{h}}_Z) = 1$. On the other hand, if $\hat{P}_Z$ is maximally different from the goal distribution P^*_Z (e.g., $\hat{P}_Z = P_v(Z = z \mid \hat{\mathbf{h}}_{Z=z'}) = 1$), then $c(\hat{\mathbf{h}}_Z) = 0$. For features with more than two possible values, completeness is computed by averaging over each possible counterfactual value $z'_1, ..., z'_k \neq z$, yielding $c(\hat{\mathbf{h}}_Z) = \frac{1}{k}\sum_{i=1}^{k} \hat{c}(\mathbf{h}_{Z=z'_i})$.

For **concept removal interventions**, we measure completeness as:

$$c(\hat{\mathbf{h}}_Z) = 1 - \frac{k}{k-1} \cdot \delta(\hat{P}_Z, P^*_Z) \tag{4}$$

where k is the number of values Z can take. The normalizing factor is needed because P^*_Z is the uniform distribution over k values and hence $0 \leq \delta(\hat{P}_Z, P^*_Z) \leq 1 - \frac{1}{k}$.

Selectivity If an intervention on feature Z_i is *selective*, the intervention should not impact M's representation of any non-targeted feature $Z_j \neq Z_i$. Thus, for both counterfactual and concept removal interventions, validation probe v's prediction for any such Z_j should not change after the intervention.

To measure the selectivity of a modified representation $\hat{\mathbf{h}}_{Z_i}$ with respect to Z_j, denoted $s_j(\hat{\mathbf{h}}_{Z_i})$, we can again measure the distance between the observed distribution $\hat{P}_{Z_j} = P_v(Z_j \mid \hat{\mathbf{h}}_{Z_i})$ and the original (non-intervened) distribution $P_{Z_j} = P_v(Z_j \mid \mathbf{h})$:

$$s_j\big(\hat{\mathbf{h}}_{Z_i}\big) = 1 - \frac{1}{m} \cdot \delta\big(\hat{P}_{Z_j}, P_{Z_j}\big) \tag{5}$$
$$\text{where } m = \max\big(1 - \min(P_{Z_j}), \max(P_{Z_j})\big)$$

Since $0 \leq \delta(\hat{P}_{Z_j}, P_{Z_j}) \leq m$, we divide by m to normalize selectivity to $0 \leq s_j(\hat{\mathbf{h}}_{Z_i}) \leq 1$. If multiple non-targeted features $Z_{j_1}, ..., Z_{j_{\max}}$ are being considered, selectivity $s(\hat{\mathbf{h}}_{Z_i})$ is

computed as the average over all such features $s_{j_m}(\hat{\mathbf{h}}_{Z_i})$. Finally, analogous to completeness, the aggregate selectivity over a set of test embeddings $\mathbf{H}_{Z_i} = \{\mathbf{h}^k\}_{k=1}^n$, denoted $S(\mathbf{H}_{Z_i})$, is the average selectivity $s(\hat{\mathbf{h}}^k_{Z_i})$ across all $\mathbf{h}^k_{Z_i} \in \mathbf{H}_{Z_i}$.

Reliability Since completeness and selectivity can be seen as a trade-off, we define the overall reliability of an intervention $R(\hat{\mathbf{H}})$ as the harmonic mean of $C(\hat{\mathbf{H}}^l)$ and $S(\hat{\mathbf{H}}^l)$. This is analogous to the F1-score, which is the harmonic mean of precision and recall: just as a degenerate classifier can achieve perfect recall and low precision by always predicting the positive class, a degenerate intervention can achieve perfect selectivity and low completeness by performing no intervention at all. Using harmonic mean to calculate reliability heavily penalizes such interventions.

4.4 Experimental Setting

LLMs We test our framework in experiments across six language models: BERT (Devlin et al., 2019), GPT-2 (Radford et al., 2019), three Pythia models (160M, 1.4B, and 6.9B; Biderman et al., 2023), and Llama 3.2 (3B, instruction-tuned; Dubey et al., 2024). We include BERT and GPT-2 to test causal probing methods in more traditional settings they were originally designed for – e.g., BERT (an encoder-only masked language model) has been very extensively studied in causal probing (Ravfogel et al., 2020; Rogers et al., 2021; Elazar et al., 2021b; Ravfogel et al., 2021; Lasri et al., 2022; Ravfogel et al., 2022b, 2023; Davies et al., 2024; see Chapter 3), and many methods have been designed specifically with this model in mind. We include the range of Pythia models to study how these methods scale and generalize to the popular GPT-like family of architectures (decoder-only models trained on autoregressive language modeling). Finally, we account for the effect of popular post-training techniques like instruction-tuning and RLHF by studying the selected Llama model.[19]

Task Following several prior causal probing works (Lasri et al., 2022; Ravfogel et al., 2021; Arora et al., 2024a), we select the prompting task of **subject-verb agreement**. (In Section C.3.5, we also repeat some experiments for the IOI task introduced by Wang et al., 2023b.) In subject-verb agreement, each data point takes the form $\langle \mathbf{x}_i, y_i \rangle$ where $\mathbf{x}_i$ is a sentence such as "the boy with the keys `[MASK]` the door," and the task of the LLM is to predict $P_M(y_i \mid \mathbf{x}) > P_M(y_i' \mid \mathbf{x})$ (here, that y_i = "locks" rather than y_i' = "lock") – see the example in Figure 5. The causal variable Z_c is the number of the subject (`Sg` or `Pl`), because (grammatically) this is the only variable that determines the number of the verb in English.

[19] For information on post-training of the `Llama-3.2-3B-Instruct` model used in our experiments, see: `https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct#training-data`

The environmental (non-causal) variable Z_e is the number (`Sg` or `Pl`) of the noun immediately preceding the verb to be conjugated when that noun is not the subject (e.g., "keys" in the phrase "with the keys"). Note that, in this work, we consider a task in the simplest experimental setting (two binary features) that allows us to study interventions using our framework; however, nothing in our methodology precludes the use of more features, or features with more possible values.

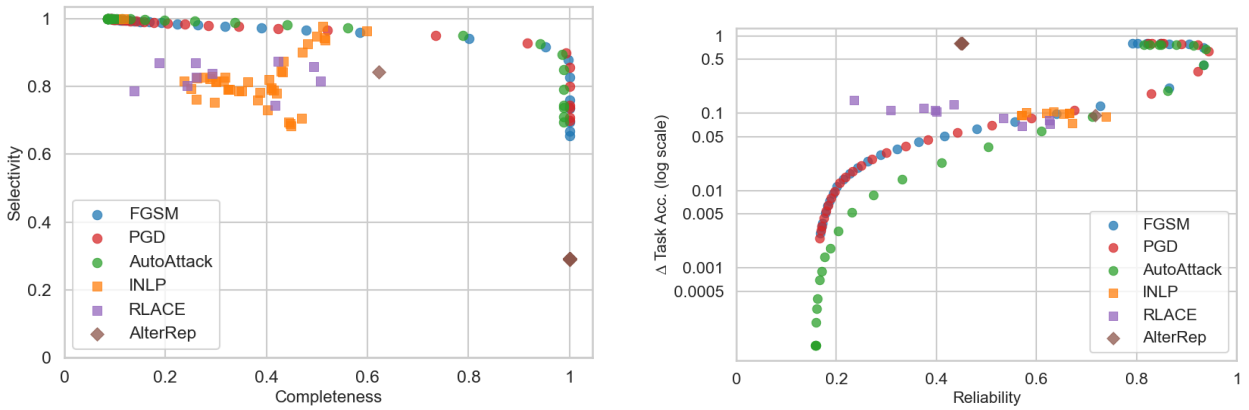
Dataset We use the LGD dataset (Linzen et al., 2016), which consists of >1M naturalistic English sentences from Wikipedia; from this we take only sentences for which both singular and plural forms of the target verb are in LLMs' vocabularies. We use 40% of the examples to train validation probes, 40% to train interventional probes, and 20% as a test set. (More dataset details can be found in Appendix C.2.1.)

Validation Probes For each layer l and probed feature Z, we experiment with several instantiations of validation probes, including linear and MLP probes across a range of hyperparameters (Section C.2.2), observing similar results between them (see Appendix C.3.3). Thus, for all results reported in the main paper, we default to the validation probe architecture and hyperparameters with the highest validation-set accuracy for the probed feature.[20] Validation probes are trained on data that is completely disjoint from that used to train interventional probes, and where Z_c and Z_e are made independent by subsampling the largest (random) subset that preserves label distributions $P(Z_c)$ and $P(Z_e)$. This is important for validation probes to serve as unbiased arbiters of selectivity, as spurious correlations between the variables could lead a probe that is trained on feature Z_c to partially rely on representations of Z_e (Kumar et al., 2022).[21]

Interventions We explore two (linear) concept removal interventions: INLP (Ravfogel et al., 2020), which iteratively trains classifiers on Z and projects embeddings into their nullspaces; and RLACE (Ravfogel et al., 2022a), which identifies a minimal-rank subspace to remove information that is linearly predictive of Z by solving a constrained minimax game. We explore one linear counterfactual method, AlterRep (Ravfogel et al., 2021), which builds on INLP by projecting embeddings along classifiers' rowspaces, placing them on the

[20]Note that this always results in MLP validation probes; see Appendix C.3.3 for results with linear validation probes. Validation sets used for selecting validation probes have the same independence and label-distribution features as their train sets.

[21]We leave these spurious correlations in training data of interventional probes to test the impact that they have on the resulting interventions' completeness and selectivity, as this is a better proxy for their completeness and selectivity in more realistic settings where controlling for spurious correlations may not always be possible.



(a) **Selectivity vs. Completeness** (b) **ΔTask Acc vs. Reliability**

Figure 6: **Completeness, selectivity, reliability, and ΔTask Acc** for all interventions in the **final layer of Pythia-160M.** Each point in both plots corresponds to a different hyperparameter setting. (Appendix C.3.1 contains analogous results for all other models.)

counterfactual side of the separating hyperplanes. Finally, we study three nonlinear counterfactual methods, which are all gradient-based interventions (GBIs, as introduced in Chapter 3): a MLP probe is trained on Z, then gradient-based ("white-box") adversarial attacks are applied to minimize the loss of the probe with respect to the target counterfactual value $Z = z'$ within an L_∞-ball of radius ε around the original embedding. We experiment with three gradient attack methods – FGSM (Goodfellow et al., 2015), PGD (Madry et al., 2017), and AutoAttack (Croce and Hein, 2020) – as described in Section C.2.3. After intervening on Z_c to obtain representations $\hat{\mathbf{h}}_{Z_c}$, we use validation probes $\hat{o}$ to measure completeness, selectivity, and reliability. Due to compute limitations, we restrict our analysis for Pythia-1.4B and -6.9B to three of the six methods: INLP, AlterRep, and FGSM. Note that this includes at least one method from each of the three classes of methods defined above (linear removal, linear counterfactual, and nonlinear counterfactual, respectively).

Impact on Model Behavior The ultimate goal of causal probing is to measure a model M's use of a feature Z by comparing intervened predictions $P_M(\cdot \mid \mathbf{x}, \text{do}(Z))$ to its original predictions $P_M(\cdot \mid \mathbf{x})$. Our framework aims to measure the reliability of the interventions themselves, a prerequisite to making claims about the underlying model. It is nonetheless important to consider how the completeness, selectivity, and reliability of a given intervention relate to its impact on model behavior. Thus, for each intervention, we also feed intervened final-layer embeddings $\hat{\mathbf{h}}^L$ for all test instances back into models immediately before word prediction, measuring task accuracy based on whether they assign the correct verb form a higher probability, and subtract this "intervened" accuracy from the original

	BERT	GPT2	Pythia-160M	Pythia-1.4B	Pythia-6.9B	Llama-3.2-3B
INLP	0.464	0.106	0.739	0.841	0.751	0.668
RLACE	0.429	0.495	0.627			
AlterRep	**0.835**	0.427	0.717	0.597	0.519	0.891
FGSM	0.552	0.958	0.934	**0.920**	**0.951**	**0.960**
PGD	0.509	**0.98**	**0.943**			
AutoAttack	0.514	0.97	0.938			

Table 1: **Intervention scores for maximum-reliability hyperparameters** in the **final layer** of all models. (See Section 4.5.2 for results in earlier layers.) Scores are reported for the hyperparameter x_{opt} that maximizes the reliability of each respective method, and the highest-reliability method for each model is bolded.

task accuracy (98.62%) for each intervention to yield ΔTask Acc.

4.5 Experimental Results

Below, we present results for completeness, selectivity, reliability, and ΔTask Acc of all intervention methods in models' final layer (Section 4.5.1), then examine their reliability in earlier layers (Section 4.5.2). Note that, while we only have space to include plots for Pythia-160M (henceforth referred to as "Pythia") in this section of the main paper, analogous plots for the other models are available in Appendix C.3.

4.5.1 Final-Layer Results

First, we note that both validation probes are able to consistently predict each feature (97.3% and 94.4% accuracy for Z_c and Z_e, respectively), which is a necessary prerequisite to validate any further results.

Completeness, Selectivity, & Reliability Each intervention has a hyperparameter (ε for GBIs, rank r for INLP and RLACE, and α for AlterRep), where increasing its value leads to stronger interventions. Thus, each hyperparameter setting yields a different value of completeness, selectivity, and reliability for a given intervention. Figure 6a plots selectivity against completeness for each method in Pythia's final layer, showing that increasing the hyperparameter values yields higher completeness and lower selectivity. (Analogous results for other models, as well as plots of completeness, selectivity, and reliability broken down by method and hyperparameter value, are available in Appendix C.3.1.)

Table 1 shows these metrics for each method at the hyperparameter that yields the highest reliability. For all models other than BERT, the nonlinear counterfactual methods (GBIs: FGSM, PGD, and AutoAttack) have the highest overall reliability; for BERT only,

AlterRep is most reliable; and otherwise, the linear methods (both removal and counterfactual) tend to show middling reliability, varying from a low of 0.106 for INLP on GPT2 to a high of 0.841 for INLP on Pythia-1.4B.

Task Accuracy Figure 6b shows ΔTask Acc as a function of the reliability for each intervention and hyperparameter setting. For most methods and hyperparameter values, ΔTask Acc increases alongside intervention reliability. Notably, the points at which the GBIs (FGSM, PGD, and AutoAttack) achieve the highest ΔTask Acc are *not* at their highest reliability values, resulting in a backward curve visible at the top of Figure 6b, corresponding to hyperparameter ε being raised past the point of maximum reliability where there is near-perfect completeness but much lower selectivity (see Appendix C.3.1). Finally, RLACE and INLP shows a similar impact on task accuracy even for different completeness and reliability scores due to its "noisy" equilibrium in reliability and completeness for high rank r (see Appendix C.3.1).

4.5.2 Reliability by Layer

As in the final layer, validation probes over earlier layers can consistently predict each feature with high accuracy (see Section C.3.4). Figure 7 shows the reliability for each method using the hyperparameters that obtain the highest reliability in that layer. Across all layers, each nonlinear counterfactual method (GBIs: FGSM, PGD, and AutoAttack) is more reliable than all linear methods. We also observe this trend for all other models Appendix C.3.2 (with the exception of BERT, for which AlterRep is more reliable than the GBIs in layers 10-12; see Figure 43).

4.6 Discussion

Tradeoff: Completeness vs. Selectivity In Pythia's final layer, no method is able to achieve perfect completeness without sacrificing selectivity (see Figure 6a), a trend which we also see for all other models (see Appendix C.3.1). However, we observe a very favorable tradeoff for GBIs, which incur only a small selectivity cost for increasing completeness, leading to high overall reliability across all models and layers (see Appendix C.3.2). In contrast, linear removal methods (INLP and RLACE) tend to have much higher selectivity than they do completeness, which is likely because these methods are explicitly optimized to minimize collateral damage (Ravfogel et al., 2020, 2022a), despite being linear and potentially incomplete (with respect to putatively nonlinear representations). Finally, the linear counterfactual method (AlterRep) tends to have highly variable behavior between

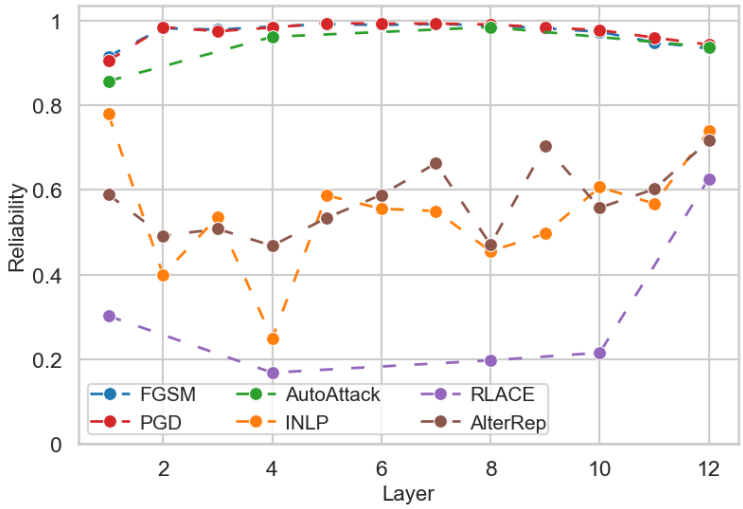


Figure 7: **Maximum reliability by layer** for each intervention across **all layers of Pythia-160M.** (Appendix C.3.2 contains analogous results for all other models.)

different models: for Pythia-160M, Llama 3.2, and GPT2, it shows middling performance across layers; it is the *most* reliable method in BERT's last 3 layers, with high selectivity and near-perfect completeness; and it is the *least* reliable method in the later layers of Pythia-1.4B and Pythia-6.9B.

Reliability and Task Accuracy Figures 6b, 38b and 39b show a clear trend: more complete interventions and hyperparameter values show a greater impact on task performance. In particular, counterfactual methods (GBIs and AlterRep), which show consistently higher completeness than removal methods (INLP and RLACE), also show (near-)total ΔTask Acc. This is highly intuitive: in the case where models perform the subject-verb agreement task by leveraging its representation of Z_c, then more complete interventions would have a greater effect on the model's task performance. We do not claim that this is necessarily the case – e.g., our results might have looked different if we had intervened in earlier layers; and the primary object of our study is the completeness, selectivity, and reliability of the causal probing methods we have experimented with, not the representations used by LLMs to perform a simple grammatical task. Rather, we take the clear relationship between intervention completeness and ΔTask Acc to be a strong indicator that more complete methods indeed yield stronger results, reinforcing the utility of our framework in evaluating causal probing interventions as tools for studying models' use of latent representations. In particular, our framework provides the first concrete approach for calibrating intervention hyperparameters in the latent space (i.e., max-reliability hyperparameter search using validation probes), allowing researchers to adaptively balance

the priorities of completeness and selectivity and examine the corresponding effect on model behaviors, rather than simply resorting to maximum-strength (Tucker et al., 2021) or minimum-collateral damage (Ravfogel et al., 2020, 2022a) interventions.

Linearity by Layer Overall, the nonlinear GBI methods are more reliable than the linear methods across all models and layers, (with the sole exception of BERT in layers 10-12; see Appendix C.3.2). Without adequate controls, this might simply be the result of using MLP validation probes in all results reported in the main paper, which could bias our analysis in favor of nonlinear methods, as validation probes and nonlinear interventional probes might be relying on similarly-encoded information and neglecting linearly-encoded information. We account for this possibility by repeating layerwise reliability experiments using linear validation probes in Appendix C.3.3, finding that they show remarkably similar results to MLP validation probes.

Thus, we briefly consider the more interesting possibility that the reliability gap between linear and nonlinear LLMs *may be due to LLMs encoding task-relevant representations nonlinearly*, particularly in intermediate layers: for instance, in addition to the aforementioned example of BERT's last 3 layers, Pythia-160M also shows that all linear methods are substantially more reliable in the first and last layers than they are in intermediate layers. While this conjecture is not fully supported by all results (e.g., INLP and AlterRep drop substantially in reliability in GPT2's final layer), it is nonetheless intuitive that some models may be more nonlinear in intermediate layers than their final layer, as embeddings in earlier layers will pass through many nonlinearities before word prediction, allowing a high degree of nonlinear representation (White et al., 2021); whereas any output-discriminative information must be made linearly separable in the final embedding layer of neural networks (Alain and Bengio, 2017). There is a long history of work studying the so-called *linear representation hypothesis* (LRH; Mikolov et al., 2013; Pennington et al., 2014; Bolukbasi et al., 2016; Vargas and Cotterell, 2020) – i.e., that neural networks implement most or all features linearly (see Section 2.3.1) – with some recent works suggesting that this hypothesis is true even for modern LLMs (Burns et al., 2023; Tigges et al., 2023; Park et al., 2024b). However, many of these studies often consider embeddings only in the input or final ("unembedding") layer of LLMs (Jiang et al., 2024; Park et al., 2024a,b), neglecting intermediate layers. Our findings provide an important contrast: while they do not directly validate or refute the LRH, the stark difference between the reliability of linear and nonlinear counterfactual methods indicates that it is critical to consider multiple layers throughout models when studying the LRH, as findings of linearity in the final layer may not generalize to earlier layers.

4.7 Conclusion

In this work, we proposed a general empirical evaluation framework for causal probing, defining the reliability of interventions in terms of completeness, selectivity, and reliability. Our framework makes it possible to directly compare different kinds of interventions, such as linear vs. nonlinear or counterfactual vs concept removal methods. We applied our framework to study leading causal probing techniques across a range of LLMs, finding that they all exhibit a tradeoff between completeness and selectivity, that more reliable and complete methods yield a greater impact on LLM task performance, and that nonlinear methods tend to be much more reliable than linear methods. Finally, we explored the implications of these findings for future work in optimizing intervention hyperparameters and studying the linear representation hypothesis.

Part II
Transfer Learning

Preface

Part II abstracts away from mechanistic interpretation at the the implementational, representational, and algorithmic levels to directly study foundation models at the *functional level* (see Chapter 2) – i.e., examining the input/output behaviors of models, and analyzing how these behaviors vary by test distribution. That is, we study how and when models succeed or fail to generalize to new tasks and data distributions by *controlling* and *evaluating* their use of task-causal vs spurious features, providing a functional-level analogue to the internal analysis performed in CALM (Chapter 3).

This part begins with an overview of the *distribution shift* problem in transfer learning (Section 5.0.1) and an outline of three broad categories of work studying this problem (Section 5.0.2) before presenting our work in each category in Chapter 5 to Chapter 7.

5.0.1 The Problem of Distribution Shift

A longstanding problem in deep learning is how best to facilitate robustness to *distribution shift*, where models' training data is not fully i.i.d. with respect to the context in which it will be deployed (Gulrajani and Lopez-Paz, 2021; Wang et al., 2022a); and even for current foundation models trained on a sizable percentage of the modern web, robust generalization to out-of-distribution (OOD) samples remains a key challenge (Yuan et al., 2023; Wang et al., 2023a; Yang et al., 2023a,b; Huang et al., 2025b; Alazraki et al., 2025). To formalize the problem, let us consider any machine learning task with inputs $\mathbf{x} \in \mathbf{X}$, labels/outputs $\mathbf{y} \in \mathbf{Y}$, and training data $\hat{D}_{\text{train}} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^n \sim P_{\mathcal{S}}(\mathbf{X}, \mathbf{Y})$, where the goal is to learn a classifier that robustly predicts $p(\mathbf{y} \mid \mathbf{x})$ given arbitrary $(\mathbf{x}, \mathbf{y}) \sim P_D(\mathbf{X}, \mathbf{Y})$ that could be sampled from any domain $D \in \mathcal{S} \cup \mathcal{T}$, where source domains $\mathcal{S} = \{S_1, ..., S_m\}$ are seen at train-time and target domains $\mathcal{T} = \{T_1, ..., T_n\}$ are not.[22] There are several (often overlapping) kinds of distribution shift between source and target domains (Quiñonero-Candela et al., 2022; Tamang et al., 2025):

- *Label shift* is when the label/output probability varies between source and target domains, $P_S(\mathbf{Y}) \neq P_T(\mathbf{Y})$. For example, a spam detector might be trained in a setting where most emails are benign, then deployed in a setting with a much higher proportion of spam.
- *Covariate shift* is when input (covariate) probabilities vary between source and target probabilities, $P_S(\mathbf{X}) \neq P_T(\mathbf{X})$. For example, a sentiment classifier might be trained on

[22] When there is only a single source domain $\mathcal{S} = \{S_1\}$, this is the setting of *single-domain generalization*; whereas *multi-domain generalization* is any setting with multiple source domains $|\mathcal{S}| > 1$ (Wang et al., 2022a).

movie reviews from critics, then deployed for classifying reviews from non-critic viewers. (This is the type of distribution shift we study in Chapter 5 to Chapter 7.)

- *Concept shift* is when the underlying relationship between task inputs and outputs changes, $P_S(\mathbf{Y} \mid \mathbf{X}) \neq P_T(\mathbf{Y} \mid \mathbf{X})$. For example, a climate model trained on 20th-century climate data may be deployed in 21st-century conditions where the underlying relationship between observable climate conditions and future conditions has changed.

Furthermore, *compositional generalization* is a specific variety of distributional generalization (with respect to *covariate shift*) that is particularly relevant to our work in Chapter 8. Broadly speaking, *compositionality* can be understood as the ability to correctly process complex wholes given the ability to correctly process their constituent parts (Fodor and Pylyshyn, 1988; Partee et al., 1995; McCurdy et al., 2024); and correspondingly, compositional *generalization* involves applying this ability in order to process novel wholes, given experience with their constituent parts (Hupkes et al., 2020; Soulos et al., 2024). For instance, two classic notions of compositionality are *systematicity* and *productivity*, where systematicity is a matter of systematically recombining known concepts and relations in novel configurations (e.g., if one understands the syntactic mapping from "the cat hunts the mouse" to hunts(subject = cat, object = mouse), then they should just as easily understand that "the mouse hunts the cat" maps to hunts(subject = mouse, object = cat) even if they have never heard of such an event before), and productivity is a matter of composing known elements into a more complex whole (e.g., if one understands the sentences "the cat hunts the mouse" and "the mouse scurries away", they should also understand the sentence "the cat hunts the mouse, so the mouse scurries away"). In the context of deep learning models, a conventional way to test for systematicity is to train models on datasets containing, e.g., instances of concepts a, b and relation $R(\cdot, \cdot)$ over concepts – but never $R(a, b)$ – then test them on inputs containing $R(a, b)$ (Kim and Linzen, 2020; Vegner et al., 2025); and productivity is often tested in terms of *length generalization*, where models are trained only on inputs of a given length $l <= L$, then tested on inputs of length $l' > L$ (Hupkes et al., 2020).

In **Chapter 7** (Davies et al., 2026) we examine the compositional generalization of models (including evaluation via OOD splits for both systematicity and productivity) on a symbolic reasoning task, leveraging mechanistic interpretability techniques to study how and why various internal mechanisms for performing the task lead to successes or failures in compositional generalization.

5.0.2 Approaches to Studying Distribution Shift

Invariant Representation Learning The problem of distribution shift is often understood in terms of causal vs spurious features: put simply, models that rely on task-*causal* input features (which hold their relationship with the label in all domains) are expected to generalize to novel target domains much better than for models relying on *spurious* features (whose relationship with the label varies by domain) (Peters et al., 2016; Arjovsky et al., 2019; Bühlmann, 2020). Correspondingly, a major goal in the area of causal machine learning is to teach models to perform a given task by relying on causal features while maintaining invariance with respect to spurious (or "environmental") features (Gong et al., 2016; Schölkopf et al., 2021; Zhao et al., 2022). For instance, the most common approach is to augment data across dimensions to which models are intended to be invariant (e.g., image classifiers should be invariant to lighting conditions or gaussian noise, sentiment analysis models should be invariant to paraphrases, etc.) and train models on a combination of original and augmented data (Sennrich et al., 2016; Hendrycks et al., 2020; Cubuk et al., 2020; Ilse et al., 2021).

In **Chapter 5** (Yuan* et al., 2024) we developed an approach leveraging generative models to perform targeted augmentations over specific spurious features.

Steering Foundation Models Two key limitations of augmentation-based techniques (and supervised invariant representation learning more broadly; see Kaddour et al., 2022) are that they require both: (a) a known, fixed set of invariances that should always be enforced for a given model, and (b) a reliable method to enforce arbitrary invariances with respect to the model. Our work in Chapter 5 – alongside closely related prior and followup works such as Ilse et al. (2021), Bansal and Grover (2023), Wang et al. (2024b), and Parihar et al. (2024) – have made important strides toward (b); but a more fundamental problem is the notion in (a) that the same set of invariances should be enforced across any applications of a given model. That is, while this assumption may be uncontroversial in the context of models trained from scratch on a single fully-supervised task, it is inadequate to the task of task-general foundation models, which are expected to perform multiple tasks with contradictory invariances – for instance, if one queries a vision-language model regarding "what animal is depicted" given either a charcoal sketch or a watercolor painting of a cat, the answer should be invariant to the style (and causally dependent on the type of animal); but if one provides the same images with the query "what is the artistic style of this image", the answer should instead be invariant to the type of animal (and causally dependent on the style).

Instead of assuming invariances at a model-level, such scenarios necessitate the ability to

dynamically enforce arbitrary invariances at inference time, depending on the context in which a model is being deployed. One prominent approach to such problems is *latent-space steering* methods, which conceptually function similarly to causal probing (see Chapter 3 and Chapter 4): auxiliary models like probes or SAEs are trained on LLM representations (see Chapter 2), then used to perform embedding-space interventions that push representations towards intended (e.g., causal) features or away from undesirable (e.g., spurious) ones (Turner et al., 2023; Zou et al., 2023; Li et al., 2023c; Bhattacharjee et al., 2024; Han et al., 2024). While flexible, such methods require white-box access to model representations during inference (to perform interventions), and interventions must be learned for each target feature. Furthermore, there is a gap between this notion of steering and the goal of invariant representation learning, as discussed above – instead of training models to be invariant to certain features at the *distribution-level*, latent steering requires advance *instance-level* knowledge of which spurious features are relevant to any given input, and the values taken by these features for a given input (e.g., to enforce invariance to the "gender" feature, one must first know whether a given input has the value of "male" or "female", then push embeddings into the opposite direction).

To address these limitations, in **Chapter 6** (Lamb et al., 2025), we propose an alternative approach to train models to dynamically condition their responses based on specified features at inference time, allowing us to enforce arbitrary invariances at inference time without requiring white-box access to model internals (after training), and assuming no instance-level knowledge of input features.

Visual Robustness and Shape Recognition One of the best-studied cases of "causal versus spurious feature learning" is the *shape versus texture* problem in visual recognition. That is, human object recognition is largely based on shape recognition (Landau et al., 1988; Biederman and Ju, 1988; Xu et al., 2004; Baker and Kellman, 2018) which is essential to the robustness of human vision due to the invariance of shape to environmental conditions such as translation, rotation, scaling, and changes in illumination, color, and texture (Kendall, 1984; Hummel, 2001; Ommer, 2013; Dryden and Mardia, 2016), which are among the most common image augmentations deployed for improving robustness of visual classifiers (see, e.g., Hendrycks et al., 2020; Cubuk et al., 2020). Thus, a natural target in working toward human-like visual robustness has been to train visual recognition models to rely on shape, rather than spurious environmental conditions. Early deep vision models relied much more on texture than shape in image classification (Geirhos et al., 2018; Islam et al., 2021; Pinto et al., 2022a), which has been argued to be a major factor in their lack of robustness (Geirhos et al., 2020a; Gavrikov et al., 2024). More recently, Gavrikov et al. (2024) showed

that vision-language models (VLMs) can be prompted to more reliably leverage shape in visual recognition and mitigate texture bias.

However, these works have all relied on the Cue Conflict (CC) and/or Stylized-ImageNet (SIN) benchmarks (both introduced in Geirhos et al., 2018) to measure models' reliance on shape versus texture; but these datasets were generated using only simple programmatic transformations or legacy style-transfer methods (specifically, those from Gatys et al., 2016; Huang and Belongie, 2017). Thus, while CC and SIN have been significant contributions and provided longstanding utility in studying shape recognition, we believe them to be insufficient tools for benchmarking modern VLMs, which have advanced several generations since CC and SIN were first created.

In **Chapter 7**, we present IllusionBench (Hemmat et al., 2024), our benchmark leveraging state-of-the-art (as of mid-2024) generative models and conditioning mechanisms to create a far more complex and challenging shape-recognition benchmark, finding that even frontier VLMs are unable to robustly detect shapes in this more difficult setting.

Chapter 5 Not Just Pretty Pictures

Preface

In **Chapter 5**, which is based on *Not Just Pretty Pictures: Toward Interventional Data Augmentation Using Text-to-Image Generators* (Yuan*, Pinto*, Davies*, et al., 2024),[23] we address *invariant representation learning* (see Section 5.0.2) as a matter of *interventional data augmentation:* we augment existing datasets by using generative models to synthesize samples across distribution shifts, thus breaking spurious correlations that models might otherwise learn to perform the task at high accuracy in the train set while failing to generalize to test distributions where the spurious correlation is broken. While this general approach (using generative models to augment data with respect to various distribution shifts) has become fairly widespread in invariant representation learning since the time we completed this work (Alimisis et al., 2025), our initial submission[24] was, to the best of our knowledge, the first work to explicitly address and study the potential of generative Text-to-Image (T2I) models to perform interventional data augmentation (IDA) – i.e., to use these generators to simulate interventions on the underlying data-generating process that produced the available training data. The potential to simulate such interventions in pixel-space had been conjectured (Ilse et al., 2021; Wang et al., 2022b; Wang and Veitch, 2022), but not yet demonstrated using T2I generators. As such, the primary goals (and corresponding contributions) of this work are to:

1. Demonstrate that T2I-enabled IDA is (a) already possible in practice; and (b) substantially outperforms a wide variety of previously state-of-the-art data augmentation approaches across a broad selection of tasks in domain generalization and reducing reliance on spurious features.

2. Precisely characterize how various elements of the data synthesis process (including interventional prompting strategies, conditioning mechanisms, and post-hoc filtering) contribute to the observed performance gains, serving as a guide for future work to target the most important areas of data synthesis for improvements.

[23] Note that this work was a 3-way co-primary author project, with author contributions listed in Appendix D.1.

[24] Available at: `https://arxiv.org/abs/2212.11237v1`

Chapter 5: *Not Just Pretty Pictures*

Publication: Jianhao Yuan*, Francesco Pinto*, Adam Davies*, and Philip Torr (2024). "Not Just Pretty Pictures: Toward Interventional Data Augmentation Using Text-to-Image Generators". In: *Proceedings of the 41st International Conference on Machine Learning*. Ed. by Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp. Vol. 235. Proceedings of Machine Learning Research. PMLR, pp. 57924–57952. URL: https://proceedings.mlr.press/v235/yuan24e.html

(denotes co-primary authorship.)*

5.1 Introduction

The success of deep image classifiers is largely built on the assumption that the train and test data come from the same domain – i.e., that they are independent and identically distributed (i.i.d.) – but in real-world applications, small changes in the environmental conditions under which the image is captured can break this assumption, significantly degrading their performance (Gulrajani and Lopez-Paz, 2021; Wang et al., 2022a; Sakaridis et al., 2020). Since these changes only affect the inputs (i.e., the covariates) in some features without altering the labels, this form of distribution shift is also known as covariate shift (Quiñonero-Candela et al., 2022). In the absence of more sophisticated techniques to simulate the possibility of sampling data coming from different environmental conditions, the employment of complex augmentation pipelines integrating image transformation primitives has been one of the most effective techniques for this purpose.

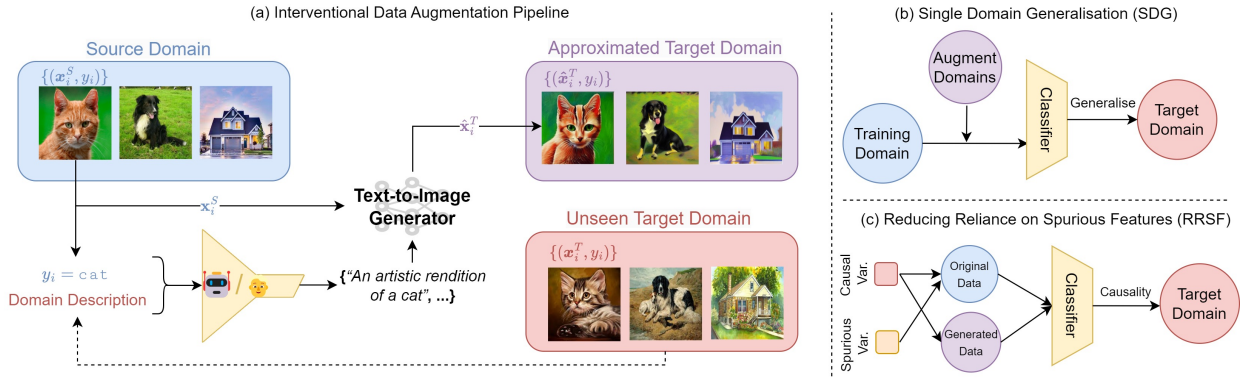


Figure 8: **Using Text-to-Image Generators for Interventional Data Augmentation.** In **(a)**, given an interventional prompt written by a user or LLM (and optionally, an image to edit), Text-to-Image generators simulate the described intervention by synthesizing a new image or edit an existing one to match the prompt. Here, the generator edits the input image to resemble the target domain. The resulting manipulated images can be used to train more robust and generalizable models. In **(b)** (Single Domain Generalization), synthetic data are generated to mimic potential target domains and combined with data from a given source domain to train a downstream classifier. In **(c)** (Reducing Reliance on Spurious Features), synthetic data are generated to break the spurious correlation in a biased dataset and used to train a downstream classifier.

Many types of augmentation primitives can be thought as reproducing (often approximately) a controlled and targeted manipulation of the domain-specific environmental conditions in which the image was captured (e.g., illumination or weather conditions) without affecting the label-related features. As such, augmentations may be understood as an automated, low-cost way of simulating interventions over the environmental factors that are likely to change across domains, turning *observational data* (i.e., with no intentional manipulation of the environment) into approximated *interventional data* (Ilse et al., 2021; Wang et al., 2022b). Motivated by this principle, several works have theoretically conjectured the utility of an augmentation mechanism capable of simulating arbitrary interventions (Ilse et al., 2021; Wang et al., 2022b; Wang and Veitch, 2022; Gowda et al., 2021). However, since it is not possible to target arbitrary interventions in the context of traditional augmentation pipelines (e.g., it is not possible to hard-code a pixel-space intervention to transform images of paintings into realistic photos), prior work has instead focused on leveraging prior knowledge about specific invariances expected to hold in the target domains (Hong et al., 2021; Li et al., 2020; Ilse et al., 2021) or targeting specific downstream applications (Ouyang et al., 2022a; Gowda et al., 2021).

Recently, powerful Text-to-Image (T2I) generative models like Stable Diffusion (Rombach et al., 2022) have emerged that can be used to synthesize new images (or edit existing ones)

using text prompts describing the desired output image (see Figure 8). In this work, our goal is to study how well such models can serve as general-purpose interventional data augmentation (IDA) mechanisms by simulating arbitrary interventions, either by editing existing images or synthesizing new ones using interventional prompts, allowing one to effectively sample from the approximated interventional distribution and augment existing training datasets with the resulting generated images. Unlike previous approaches, these models can be used off-the-shelf without requiring manual hard-coding of individual interventions or training on application-specific data: instead, it is only necessary to describe the desired intervention via language (e.g., simulating interventions over lighting conditions by editing images with prompts like "a photo taken at night" or "it is a cloudy day"). Several recent works have studied the usefulness of synthetic data from T2I generators (see, e.g., Bansal and Grover, 2023; Azizi et al., 2023; He et al., 2023; Trabucco et al., 2023); but so far, their capacity to augment existing datasets by simulating interventions has only been studied in limited contexts (see Section 5.2).

In this work, we systematically analyze the extent to which modern T2I generators can perform general-purpose IDA. We perform extensive experiments across several benchmarks for two key tasks in which the environmental and causal variables can be disentangled and the utility of synthetic interventional data can be precisely measured: **(1)** Single Domain Generalization (SDG) and **(2)** Reducing Reliance on Spurious Features (RRSF). Our investigation spans several key aspects of T2I synthesis and editing, including the use of different interventional prompting strategies, conditioning mechanisms, and post-hoc filtering techniques. Our findings reveal that T2I generators substantially outperform existing state-of-the-art image augmentation methods, regardless of how we configure each of these aspects. Our primary findings and contributions are as follows:

1. We show that T2I-based IDA surpasses previous state-of-the-art augmentation techniques in simulating interventions across a broad range of SDG and RRSF benchmarks representing widely-varying environmental conditions and complexity.

2. We find that the choice of conditioning mechanism has the greatest impact on performance across tasks, followed by the choice of prompting strategy. However, in contrast to prior works, we find that post-hoc filtering is not consistently beneficial.

3. We show that retrieving images directly from the training dataset of the T2I generator can also yield competitive performance in several cases, and explore the comparative strengths and weaknesses of retrieval versus generation.

5.2 Problem Setting and Related Works

The Problem of Out-of-Domain Generalization Given a data distribution $P(\mathbf{x}, y) = P(y \mid \mathbf{x})P(\mathbf{x})$ where $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^d$, $y \in \mathcal{K} = \{1, 2, \ldots, |\mathcal{K}|\}$, learning a classifier amounts to estimating $\hat{f}(\mathbf{x}) \approx P(y \mid \mathbf{x})$ (i.e., predicting the conditional distribution of the label y given a covariate $\mathbf{x}$) using a labeled training set $\mathcal{D}_{\text{train}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$. Given the finite amount of data available in $\mathcal{D}_{\text{train}}$ and the high dimensionality of $\mathcal{X}$, the samples in $\mathcal{D}_{\text{train}}$ are not representative of the whole input space (i.e. $\mathbf{x}_i \in \mathcal{X}_{\text{train}} \subset \mathcal{X}$). When deployed in the wild, the classifier will likely be exposed to inputs sampled from regions of the input space not represented in the training set, even when $\mathcal{K}$ is the same. Specifically, we are in presence of *covariate shift*, a setting in which it has long been observed that neural classifiers' performance significantly degrades relative to i.i.d. data (Gulrajani and Lopez-Paz, 2021; Wang et al., 2022a; Sakaridis et al., 2020; Pinto et al., 2022a; see Section 5.0.1 for further discussion).

Several theoretical frameworks have been developed to make the problem of out-of-domain (OOD) generalization well-posed (Wang et al., 2022b; Wang and Veitch, 2022; Ilse et al., 2021; Quiñonero-Candela et al., 2022) – in this work, we default to the framework proposed by (Ilse et al., 2021). In computer vision, the core principle is that pixel values of image $\mathbf{x}_i \in \mathcal{X}$ are the result of a data generation process that combines (unobserved) features h_{y_i} and $h_{\mathbf{c}_i}$ generated by the label y_i, and conditions described by a vector of *environmental variables* $\mathbf{c}_i \in \mathcal{C}$ (Gowda et al., 2021; Ilse et al., 2021). To make the problem more tractable, it is often assumed it is possible to partition $\mathcal{C}$ into M domains (i.e., $\mathcal{C} = \bigcup_{j=1}^M \mathcal{C}^j, \mathcal{C}^k \cap \mathcal{C}^h = \varnothing, \forall k \neq h$) based on how similarly the environmental conditions impact $\mathbf{x}$, so that the contextual variables values and impact are summarized in the discrete indices j (Arjovsky et al., 2019). For instance, environmental variables could be aggregated to represent similar illumination conditions or backgrounds. Furthermore, an unobserved spurious confounder $\mathbf{s}$ might correlate both y_i and $\mathbf{c}_i$. A high-performing classifier is likely to learn these spurious correlations, as they are predictive of the label y_i; but such correlations will (by definition) not hold under all environmental conditions, damaging classifiers' ability to generalize (Xiao et al., 2020; Geirhos et al., 2018).

Simulating Interventional Data for Out-of-Domain Generalization Prior works (Ilse et al., 2021; Wang et al., 2022b) have proposed that such a problem is solvable by performing interventions on $\mathbf{c}_i$ (i.e., manipulating $\mathbf{c}_i$ to break such spurious correlations without changing y_i). However, direct collection of interventional data is usually quite difficult (e.g., collecting datasets portraying objects of the same class in all environments of interest may be highly impractical). Identifying heuristic methods to disentangle causal from

environmental factors (usually by augmenting original images from the source domain) has been a key component of many leading approaches to domain generalization. For example, CIRL (Lv et al., 2022) and ACVC (Cugu et al., 2022) manipulate the amplitude component of the image frequency spectrum of the Fourier transform (which is presumed to approximately encode environmental information), while others have used style transfer techniques to perturb environmental factors while preserving image content (Hong et al., 2021; Li et al., 2020; Jackson et al., 2019), or trained Cycle-GAN to preserve the causal factors in a cyclic transformation between domains with different styles (Wang et al., 2022b). Beyond methods explicitly attempting to disentangle these two components, (Ilse et al., 2021; Gowda et al., 2021) understand augmentations as simulating alterations of $\mathbf{c}_i$ without affecting y_i – for example, rotations encode the belief that change of viewpoints should preserve the class label. Importantly, these assumptions might not hold in all applications (e.g., in digit classification, rotations of more than 90 degrees can swap the ground-truth labels of 6 and 9), so not all augmentations are valid for any given application. Domain-agnostic data augmentation pipelines (such as those proposed by Hendrycks et al., 2020; Cubuk et al., 2020; DeVries and Taylor, 2017; Hendrycks et al., 2022; Cugu et al., 2022; Pinto et al., 2022b) can be understood as hard-coding interventions over various features that are expected to vary across novel environments; but such assumptions may not hold across all possible domains. For this reason, Ilse et al. (2021) suggests a mechanism to select parametric hand-crafted augmentations that have a greater impact on environmental factors than causal factors.

Text-to-Image Generators With the recent rise of powerful flexible T2I generative models (e.g., Nichol et al., 2021; Rombach et al., 2022; Ramesh et al., 2021), a natural question is whether these models, which are capable of synthesizing images using natural-language prompts, could be used to effectively implement IDA. That is, while some hand-crafted parametric augmentations can be straightforwardly implemented by a programmer to manipulate the image directly in pixel space (e.g., lens distortion, chromatic aberration, vignetting, etc.), such methods many only be able to approximate many augmentations (often with much greater difficulty of implementation; e.g., introducing realistic rain or snow) or may not be able to approximate them at all (e.g., turning a cartoon into a photo, changing the background of a scene, or modifying the material of an object). On the other hand, modern T2I generators that have been training on large amounts of weakly supervised data can be used zero-shot to directly approximate such augmentations (either from existing images or from scratch) using natural language, and have been observed to produce high quality samples (Meng et al., 2022a). Using such models for IDA would be

extremely convenient, as these editing and synthesis abilities can be made available "off-the-shelf" without requiring any task-specific fine-tuning.

Data Augmentation with T2I Generators Contemporary work has investigated how T2I generators can be used to synthesize large-scale pre-training data (He et al., 2023; Sariyildiz et al., 2023; Azizi et al., 2023), compensate for the lack of training data in data-scarce environments (He et al., 2023; Trabucco et al., 2023), and diagnose classifiers' lack of robustness to covariate shift (Vendrow et al., 2023). A related branch of research leverages synthetic data from T2I generators for test-time adaptation, transferring OOD samples to an approximation of the training domain as a form of test-time adaptation (Yu et al., 2023; Gao et al., 2022). Closest to our work, Bansal and Grover (2023) show it is possible to use Stable Diffusion to generate synthetic data that improves the robustness of classifiers trained on ImageNet-1K (Deng et al., 2009) for multiple forms of distribution shift using an ensemble of generative prompts. In this work, we focus on the utility of T2I-generated synthetic data for training downstream classifiers, but depart from standard ImageNet analyzes in order to develop a deeper understanding of how T2I generators can be used for IDA by focusing on SDG and RRSF, allowing us to directly measure the effectiveness of T2I-simulated interventions in these settings across variable conditioning, prompting, and filtering techniques.

5.3 Simulating Interventions with Text-to-Image Editing

5.3.1 Experimental Setting

Given some source training domain $\mathcal{D}^S = \mathcal{X}^S \times \mathcal{K}$ and some target domain $\mathcal{D}^T = \mathcal{X}^T \times \mathcal{K}$, our goal is to assess how well T2I generators can approximately modify environmental features $\mathbf{c}_i$ to simulate $\mathcal{D}^T$ while keeping causal features for y_i constant. As the most common approach to image augmentation (including all baselines we consider) involves editing pre-existing training images and adding them to the training dataset rather than synthesizing new training data from scratch (Shorten and Khoshgoftaar, 2019), we begin our analyzis by studying the analogous setting of T2I-enabled image editing using `SDEdit` (Meng et al., 2022a). For an image $\mathbf{x}_i^S \in \mathcal{X}^S$ of class y_i, we aim to transform it into $\hat{\mathbf{x}}_i^T$ such that it retains y_i but appears to have been sampled from $\mathcal{X}^T$. Each of the editing techniques we consider are conditioned on a natural language prompt, denoted $\mathbf{z}_i^T$. For instance, if $\mathbf{x}_i^S$ represents a cartoon cat and $T = \texttt{painting}$, $\mathbf{z}_i^T$ could be "a painting of a cat". Using the generator G, we transform $\mathbf{x}_i^S$ into $\hat{\mathbf{x}}_i^T = G(\mathbf{x}_i^S, \mathbf{z}_i^T)$. For all experiments, we use Stable

Diffusion v1.5 (Rombach et al., 2022) pre-trained on LAION-Aesthetics.[25] Successful transformations produce $\hat{\mathbf{x}}_i^T \in \hat{\mathcal{X}}^T$ with $\hat{\mathcal{X}}^T \approx \mathcal{X}^T$. The synthetic pairs $\hat{\mathcal{D}}^T = (\hat{\mathbf{x}}_i^T, y_i)$ are then combined with $\mathcal{D}^S$ to perform ERM via cross-entropy minimization to train the neural classifier (ResNet-18 and ResNet-50).

In this work, we focus on *Single-Domain Generalization* (SDG) and *Reducing Reliance on Spurious Features* (RRSF) as representative settings where access to a high-quality approximations of the intervened distributions can measurably affect the performance of classifiers when training on both $\mathcal{D}^S$ and $\hat{\mathcal{D}}^T$. We describe our experimental formulation of both problems below.

Single-Domain Generalization (SDG) Given data $\mathcal{D}^S$ from a source domain accessible at training time, the goal of SDG is to achieve high performance on a set of datasets $\mathcal{D}^{T_j}$ with $j = 1, 2, ..., J$ sampled from different target domains (Qiao et al., 2020). In this setting, the generator uses $\mathcal{D}^S$ and $\mathbf{z}_i^{T_j}$ to generate $\hat{\mathcal{D}}^{T_j} \approx \mathcal{D}^{T_j}$. Following the standard evaluation procedure, we train a classifier on a single domain of each benchmark ($\mathcal{D}^S$) and test it on the others ($\mathcal{D}^{T_j}$), and report the average accuracy over the J target domains. For our experiments, we consider four widely used benchmarks that vary for type of domain shift, number of classes and training samples: (1) **PACS** (Li et al., 2017), containing the domains `art painting`, `cartoon`, `sketch`, and `photo`; (2) **Office-Home** (Venkateswara et al., 2017), containing `Art`, `Clipart`, `Product`, and `Real World`; and (3) **NICO++** (Zhang et al., 2022b), containing `autumn`, `dim`, `outdoor`, `grass`, `water`, and `rock`; and (4) **DomainNet** (Peng et al., 2019), containing `clipart`, `infograph`, `painting`, `quickdraw`, `real`, and `sketch`. We provide a more detailed description of the training procedure and other aspects of the SDG experiments in Section D.2.

Reducing Reliance on Spurious Features (RRSF) Sometimes the training data is collected from a domain $\mathcal{D}^S$ in which spurious features correlate with the labels. If a classifier relies on such spurious features, it will be unable to generalize to unseen test domains in which the spurious feature is no longer predictive of the label (Xiao et al., 2020; Geirhos et al., 2018). In this setting, the prompts $\mathbf{z}_i^T$ intentionally perturb the spurious features to simulate domains in which the spurious correlation is broken. We consider three standard benchmarks: (1) **ImageNet-9** (Xiao et al., 2020) measures the over-reliance on background to predict the foreground (Background Bias), (2) **Cue-Conflict Stimuli (CCS)** (Geirhos et al., 2018) assesses the over-reliance on texture (Texture Bias), and (3) a

[25] LAION-Aesthetics is a subset of the LAION-5B dataset consisting of web-scraped text-image pairs with high "aesthetic scores" (Schuhmann et al., 2022).

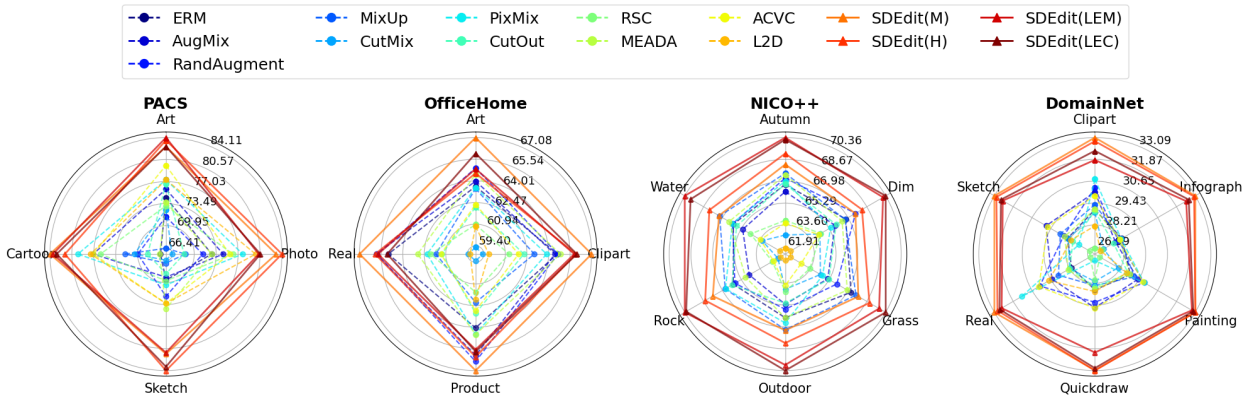


Figure 9: **Single Domain Generalization (SDG) Results.** Average SDG test accuracies on the remaining target domains when training ResNet-50 on each source domain (indicated on each axis) using the respective data augmentation methods. Baseline methods are visualized with dashed lines, and `SDEdit` methods with solid lines.

subset of **CelebA** (Xiao et al., 2020) evaluates over-reliance on spurious demographic features (Demographic Bias – in this case, the spurious correlation between hair color and gender in CelebA). See Section D.3 for further details about each dataset and associated indices measuring each form of bias.

5.3.2 Results

To evaluate the effectiveness of edited data from T2I models, we compare their results with key augmentation baselines broadly representing different approaches in the domain generalization literature (in addition to ERM): (1) **AugMix** (Hendrycks et al., 2020), (2) **RandAugment** (Cubuk et al., 2020), (3) **CutOut** (DeVries and Taylor, 2017), and (4) **PixMix** (Hendrycks et al., 2022), which all combine parametric transformations in complex pipelines to enhance model robustness. We also evaluate (5) **ACVC** (Cugu et al., 2022), which combines parametric transformations and augmentations in the Fourier domain for style mixing; interpolation-based methods like (6) **MixUp** (Zhang et al., 2017) and (7) **CutMix** (Yun et al., 2019); methods that train generators to diversify training data (8) **L2D** (Wang et al., 2021c); and adversarial data augmentation techniques like (9) **MEADA** (Zhao et al., 2020) and (10) **RSC** (Huang et al., 2020).

Single Domain Generalization Considering the fact that Stable Diffusion is built on top of a text encoder that, like many LLMs, can be sensitive to small differences in prompts that are not generally meaningful to humans (see, e.g., Ribeiro et al., 2020; Wang et al., 2021a; Moradi and Samwald, 2021), we experiment with four distinct prompting strategies using `SDEdit` to measure its sensitivity to variation in prompts: (1) ***Minimal (M)***,

sentences including only the domain label, class label, and function words (articles or prepositions) as necessary to make the prompt grammatically correct, like "a `domain` of a `class`" (e.g., "a sketch of an elephant"); (2) ***Domain expert (H)***, a collection of "handcrafted" prompts authored by a human given only metadata descriptions provided by the respective benchmarks, without looking at any samples form the target domain; and (3 & 4) ***Language enhancement (LE)***, a collection of prompts generated by T5 (Raffel et al., 2020), in two variants: one that deterministically selects the highest-probability interventional prompts (LE_C), the other that favors diversity in prompting (LE_M).[26] (See Section D.4 for further details on each prompting strategy.) As shown in Figure 9, `SDEdit` outperforms all baselines regardless of the source domain (when averaging over target domains). Specifically, across all the considered benchmarks, using ResNet-50 with minimal prompt yields a 5% improvement over the strongest baesline, PixMix, which in turn outperforms ERM by just 1.10%. We find that, when considering the performance on individual benchmarks, no single baseline consistently outperforms the others. This reinforces the observation that each of these techniques encodes different assumptions about the types of invariances expected to hold in the test domain. For the largest-scale dataset we consider, `DomainNet`, traditional data augmentation methods fail to demonstrate a substantial performance boost compared to ERM; but `SDEdit` is able to deliver a strong average performance boost of 5.48%. Comparing across all SDG datasets, `SDEdit` is the only method that consistently outperforms ERM across all benchmarks. (For a more detailed breakdown of all results figures, see Section D.2.)

We also find that the most sophisticated prompting strategy does not usually perform best: in `PACS`, `OfficeHome` and `DomainNet`, the Minimal (M) and Handcrafted (H) strategies outperform LE_M and LE_C, indicating that including additional details (e.g., specifying various styles of paintings across multiple prompts) does not yield obvious benefits, and may even degrade performance (e.g., by "injecting noise" into the pipeline). However, in `NICO++`, LE_M and LE_C show superior performance to (M) and (H), which may be explained by the fact the domain labels for NICO are not detailed enough for minimal prompts to be fully descriptive, meaning that the additional details included in prompts can be more beneficial in such contexts.

Precisely describing the target domain is not necessary. As noted above, Stable Diffusion is trained on a massive pre-training corpus of weakly-supervised data scraped from the web, which means it has likely been trained on samples that resemble a number of the

[26] Note that, by design, none of the prompting strategies are optimized to boost the reported metrics: they are generated in a way that is independent from classifiers' performance on downstream tasks or the structure of the generator. See Section D.11 for a complete list of image-generation prompts used in experiments.

	PACS	OfficeHome	NICO++	DomainNet	Average
ERM	61.96	61.94	69.95	25.26	54.78
MixUp	58.17	60.46	<u>70.63</u>	25.49	53.69
CutMix	58.50	57.16	67.03	24.47	51.79
AugMix	64.63	62.60	68.81	26.20	55.56
RandAugment	62.61	<u>63.02</u>	69.88	26.17	55.42
CutOut	60.87	60.03	69.23	24.90	53.76
RSC	64.58	59.10	67.37	23.32	53.59
MEADA	64.04	62.08	69.89	25.26	55.32
PixMix	67.12	61.43	69.48	25.53	<u>55.89</u>
L2D	68.89	58.37	65.19	24.75	54.30
ACVC	<u>67.98</u>	59.92	66.92	<u>26.46</u>	55.32
SDEdit(M)	76.43	**64.66**	71.12	**31.94**	61.04
SDEdit(H)	**77.87**	63.27	71.95	31.82	61.23
SDEdit(LEC)	76.38	63.43	**73.69**	31.44	**61.24**
SDEdit(LEM)	75.65	63.14	73.61	30.94	60.84

Table 2: **Average SDG Performance.** The number reported is the average Single Domain Generalization average of all domains in each dataset, each serving as a single source domain. The best and second-best performing methods are highlighted with bold and underline, respectively.

	Art	Photo	Sketch	Cartoon	Average
ERM	74.44	48.78	50.89	73.74	61.96
MixUp	66.31	42.98	45.64	77.76	58.17
CutMix	72.53	40.03	44.72	76.72	58.50
AugMix	75.80	51.32	49.99	81.42	64.63
RandAugment	71.38	46.80	55.95	76.33	62.61
CutOut	76.67	42.69	48.93	75.2	60.87
RSC	73.15	53.47	51.11	80.58	64.58
MEADA	73.72	48.78	59.81	73.84	64.04
PixMix	77.33	55.58	52.42	83.15	67.12
L2D	77.33	58.41	58.14	81.70	68.89
ACVC	79.63	52.76	58.13	81.40	67.98
SDEdit(M) ×	81.21	57.54	80.60	84.76	76.03
SDEdit(M) ✓	82.67	62.94	73.78	86.33	76.43

Table 3: **SDG PACS result with ResNet-50.** Columns are individual source domains; accuracies are the average test accuracy of the three remaining target domains when training using the indicated source domain. The lower part of the table highlights the comparison between accessing (✓) or not accessing (×) synthetic target domains.

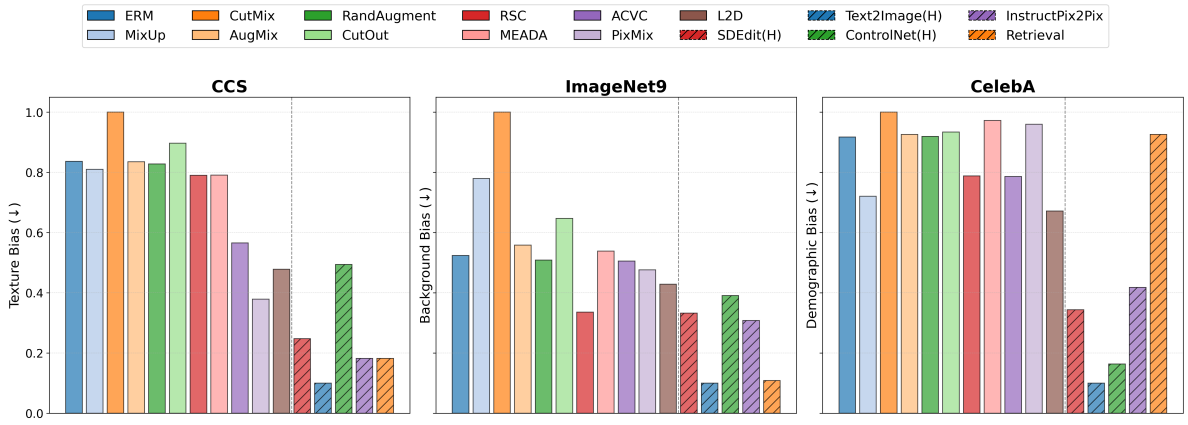


Figure 10: **Performance on Breaking Spurious Correlations.** Reliance on different image attributes in comparison with baselines (solid lines) and OURS (dash lines) using ResNet-18. (Lower scores are better.)

considered test distributions. By comparison, while the baselines we consider do make limited assumptions about the type of interventions they perform (and therefore yield better or worse performance depending on whether those interventions correspond to the covariate shift from the source domain to test domain – see our RRSF analyzis below), they do not have comparable access to approximations of the test domain. For this reason, we perform an experiment to "level the playing field" in order to better assess the usefulness of `SDEdit` as an interventional mechanism by avoiding generating data resembling the test domain. Given a single training domain from the original dataset and a chosen test domain, we use `SDEdit` to transform the training data to all domains *except the test domain* (`SDEdit`(M)×), use it for IDA training, and measure accuracy on the test domain. Fixing a test domain, we repeat this experiment for each possible choice of the training domain, and report the average accuracy on the held-out test domain. In this case, we are measuring `SDEdit`'s capacity to simulate interventions for SDG even when knowledge about the chosen test domain is not used in synthesizing interventional data. We find that the generative model-based methods still substantially outperform the data augmentation baselines, with only a marginal drop in performance with respect to the case in which the target domain is approximated by Stable Diffusion (`SDEdit`(M)✓). This indicates that the interventions simulated by Stable Diffusion are useful even when knowledge about the test domain is not available.

Reducing Reliance on Spurious Features Depending on the type of spurious correlation to be addressed in each experiment, we prompt `SDEdit` in different ways: for ImageNet-9 experiments, we handcraft prompts that describe a wide variety of possible backgrounds and randomize the combination of the object classes and backgrounds; for CCS,

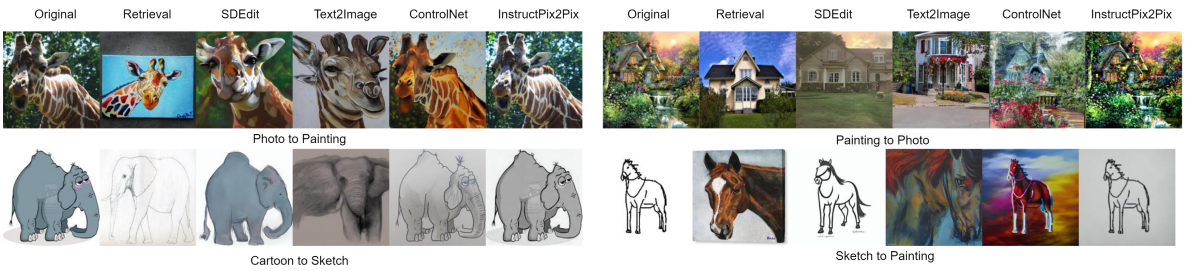


Figure 11: **Visualization of selected samples from PACS.** Recall that `Retrieval` and `Text2Image` do not take the `Original` image into account, but `SDEdit`, `ControlNet`, and `InstructPix2Pix` do.

we use prompts that induce the generator to change the texture of the objects (e.g., turning them into a sculpture of a specific material); and for Celeb-A, we randomize the correlation between gender and hair color. Our results are displayed in Figure 10. We find that, although several techniques are often assumed to perturb spurious features in a way that is agnostic to the target domain, our experiments indicate that this may not the case – instead, baselines are (perhaps unsurprisingly) most effective when their augmentation pipeline implicitly intervenes over the corresponding spurious dependency. For example, PixMix mixes the input images with fractals that alter their texture (and often the background), but yields a worse Demographic Bias than ERM. In contrast, `SDEdit` can perform the desired augmentation based on the relevant spurious dependency by simply describing it using interventional prompts, which enables substantial improvements over ERM in all settings. Such flexibility and ease-of-implementation with respect to interventions of interest are key advantages of using T2I models for IDA.

5.4 Alternative Approaches

In the previous section, we show that `SDEdit`, one of the simplest and most widely-adopted editing techniques, substantially outperforms traditional augmentation pipelines for SDG and RRSF. However, there are several other ways we can simulate interventions with T2I generators: by default, such models can generate images using only text, with no need to provide an input image to edit; and more sophisticated image-editing techniques have also been developed using different conditioning mechanisms. In Section 5.4.1, we investigate the use of these alternative generative approaches for the same tasks. Additionally, in Section 5.4.2, we consider He et al. (2023)'s finding that filtering low-quality image outputs can improve synthetic data from earlier T2I generators, and study whether this is also true for SDG and RRSF using more recent generators.

5.4.1 Conditioning Mechanisms

Despite the impressive performance of `SDEdit`, we observe several cases in which such a simple editing technique is not sufficient to simulate the desired intervention (e.g., see the second row of Figure 11). This might depend on the fact `SDEdit` initializes the diffusion process from an embedding of the image being edited, which may be too constraining to obtain the desired manipulation.[27] However, several other conditioning mechanisms exist. We consider three other forms of conditioning that may be suitable for our goal: `Text2Image`, `ControlNet` and `InstructPix2Pix`. With `Text2Image` we refer to the native ability of Stable Diffusion of generating images by conditioning only on the text: the diffusion process is initialized with random noise, and the prompt embeddings are used to condition the attention matrices in the denoising steps, steering the diffusion in order to yield an output that matches the description given by the prompt. `ControlNet` (Zhang et al., 2023a) induces stronger spatial consistency between the original and the augmented image by using an additional network that has been trained to condition the generative process on spatial guidance ("Canny edges"; Canny, 1986). Finally, `InstructPix2Pix` (Brooks et al., 2023) aims to improve diffusion models' ability to follow editing instructions by fine-tuning it on tuples of original images, editing instructions, and desired editing outputs.

Single Domain Generalization In Figure 12, we see that conditioning can have a large impact on performance. First, we observe that `InstructPix2Pix` underperforms with respect to other conditioning mechanisms in most cases. This may be related to the fact that its training set (which is distilled from Stable Diffusion) contains a limited variety of samples that may supply an inadequate implementation of a general interventional mechanism. Although `ControlNet` allows for a better spatial control, its performance is similar to or lower than `SDEdit` in most cases. This might be expected when considering that this evaluation task does not particularly benefit from the preservation of spatial features. More surprisingly, we see that `Text2Image` can be an extremely effective conditioning technique. The success of this approach indicates that conditioning on an image may often be a hindrance in approximating the desired domain.

Reducing Reliance on Spurious Features All conditioning techniques are useful in reducing classifier bias. In the aggregate, `Text2Image` is most effective in doing so across all benchmarks; whereas other conditioning mechanisms have varying strengths and weaknesses across the different tasks. For instance, `ControlNet`'s ability to preserve the spatial features

[27] Indeed, we observe this phenomenon persists across different hyperparameter settings controlling the strength of conditioning.

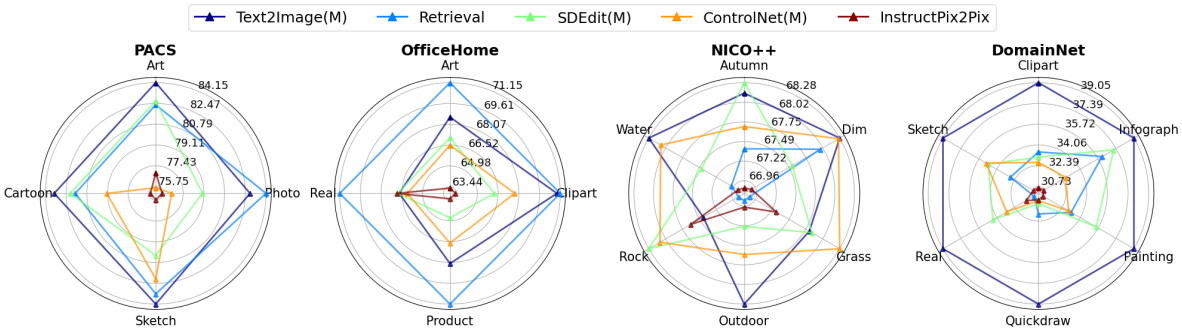


Figure 12: **SDG Results by Conditioning Mechanism.** Results are reported following the same format as Figure 9.

(i.e., the edges) of an image while modifying other aspects (in this case, the hair color) yields second-best performance in CelebA, as the Canny edge detector is designed to omit information about the texture of objects. While `InstructPix2Pix` is second-best in removing overreliance on texture and background, it is not as effective as `ControlNet` on CelebA. Finally, `SDEdit` shows middling performance across all benchmarks: it never performs best (or second-best), but it also never performs the worst.

Retrieval is not (always) enough. The strong performance we observe when removing source images from the generative process (i.e., substituting `SDEdit` for `Text2Image`) suggests that Stable Diffusion's effectiveness is higher when sampling from its approximation of the intervened distribution without starting from an input image. This raises the question of whether using Stable Diffusion to generate images is actually necessary: might we achieve similar results by simply using interventional prompts to retrieve relevant images directly from its original training dataset? To answer this question, we configure a retrieval baseline to compare the results of generating images and retrieving images from Stable Diffusion's training set using a simple image retrieval system,[28] querying it with the same minimal prompt that is used to generate images (see Figure 51).

We observe large differences between the behaviors of the `Retrieval` method across the tasks we consider. In SDG, retrieval proves to be an extremely effective technique as shown in Figure 12. For example, `Retrieval` outperforms all other methods on OfficeHome; and on PACS it proves to be only marginally inferior to `Text2Image(M)`. This is likely because Stable Diffusion's training data contains ample data from the classes and domains covered by these benchmarks and it is relatively easy to retrieve this data. On the other hand, for NICO++ and DomainNet, the retrieval baseline performance is inferior to `Text2Image(M)`. However, when Reducing Reliance on Spurious Features, `Retrieval` underperforms with

[28] Accessible at `https://rom1504.github.io/clip-retrieval`.

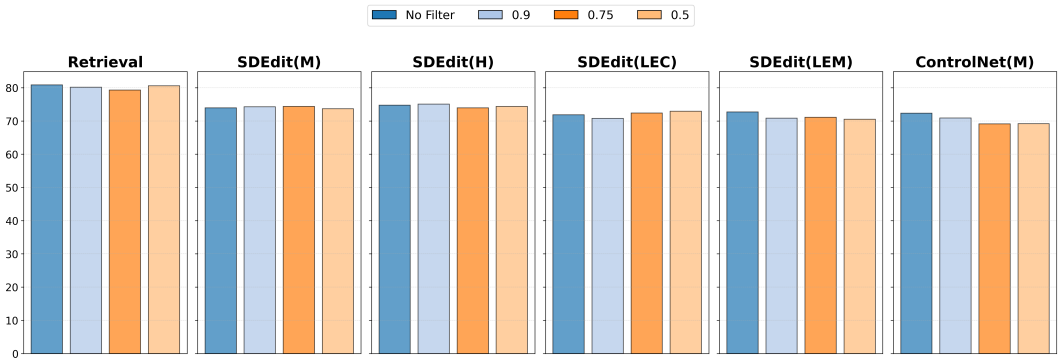


Figure 13: **CLIP Filtering Results.** SDG accuracies averaged across all test domains for different conditioning strategies (boxes in bold) and CLIP filtering proportions (colors).

respect to most generative techniques. This disagreement suggests that both retrieval and generative approaches are of interest and worth pursuing for different applications and different downstream tasks, as both have their own unique advantages and disadvantages. Indeed, beyond performance figures, there are also important practical distinctions between the two. In favor of retrieval, retrieved images do not generally contain unrealistic artifacts; and once the retrieval engine has been deployed, it can be significantly faster than generation. However, such deployment requires massive storage resources ($>$ 200TB) and relies on highly efficient indexing and computing infrastructure. In contrast, generative models are significantly more compact in terms of storage (the version of Stable Diffusion we use is $\sim$ 8GB) and do not require a dedicated infrastructure to be run. Finally, we observe that modern generators can effectively produce samples that combine concepts from their training data in new ways: in Appendix D.10.1, we compare images generated with prompts combining such concepts against images retrieved with the same prompts.

5.4.2 Post-hoc Filtering

Although the quality of the generated samples of state-of-the-art diffusion models is impressive, failure cases may still occur and low-quality samples may be generated. Since such samples have been observed to harm the performance on downstream tasks, He et al. (2023) and Vendrow et al. (2023) deploy post-hoc filtering using CLIP (Radford et al., 2021) to discard them. In the case of IDA, the generated sample may fail at capturing either the specified class, the conditions of the environment we aim to simulate, or both. Therefore, we filter images that do not exhibit a high enough CLIP similarity score with respect to both prompts: one describing the class, the other describing the domain ("An image of a `class`" and "`domain`", respectively). Before training, we remove the samples with scores lower than a given percentile threshold and provide our results in Figure 13. Unlike He et al. (2023) and

Vendrow et al. (2023), we do not find that CLIP filtering yields consistent and substantial improvements. This may be due to the improved performance of newer generators or the fact that we are considering different tasks (SDG and RRSF). For further details, full results, and selected examples, see Appendix D.5.

5.4.3 Limitations and Future Work

While we observe the effectiveness of synthetic data from generative models in improving robustness across various challenging benchmarks, we still encounter several limitations. First, there are several failure modes of different conditioning mechanisms (such as `ControlNet` and `InstructPix2Pix`) and their potential sub-optimal impact on RRSF. (See Appendix D.10.4 for qualitative examples of failure cases where target domains are either out-of-distribution with respect to Stable Diffusion's training domain or cannot be easily described via natural-language prompts.) Additionally, the computational cost is one potential bottleneck: although the inference speed of generative models has greatly improved over time – and, if current trends continue, might be attenuated to the point of irrelevance – it remains a concern for current applications. (See Section D.8 for further discussion.)

Furthermore, although we currently focus only on image classification, we note that all the methods we explore in this work are also applicable to other computer vision tasks such as object detection, instance segmentation, and semantic segmentation. While T2I-based interventional data augmentation can theoretically be applied to these tasks, implementing it successfully presents new challenges. Specifically, our approach currently only requires conditioning on domain and class labels via natural language interventional prompts, but extending this method to object detection and segmentation tasks would necessitate additional conditioning on pre-specified spatial information, such as bounding boxes or segmentation maps, as explored by (Wu et al., 2023a; Nguyen et al., 2023).

Finally, it is important to note that, like other approaches to invariant representation learning, the interventional data augmentation framework explored in this work is only applicable in the setting of supervised, task-specific models – not, e.g., task-general foundation models such as LLMs (see Section 5.0.2). In followup work presented in Chapter 6, we address this problem by introducing a post-training approach to train task-general models to dynamically condition their responses at inference time to enforce invariances to spurious (environmental) features and focus on causal features.

5.5 Conclusion

In this work, we studied the application of T2I generators to performing IDA in two settings, SDG and RRSF, finding they perform much better than traditional data augmentation techniques. We carried out a detailed investigation of how various components of the generative process may affect the results, concluding that the conditioning mechanism is the most important. Finally, we compared the strengths and limitations of T2I-enabled IDA with those of retrieval.

Chapter 6 Focus Instruction Tuning

Preface

As discussed in Section 5.0.2, two key limitations of augmentation-based techniques like those explored in Chapter 5 (and supervised invariant representation learning more broadly; Kaddour et al., 2022) are that they require both: (a) a known, fixed set of invariances that should always be enforced for a given model; and (b) a reliable method to enforce arbitrary invariances with respect to the model. As such, these approaches are inappropriate in the context of task-general foundation models like LLMs, as the relevant invariances will differ by task, and may contradict each other (e.g., style-irrelevant image classification versus content-irrelevant style classification; see Section 5.0.2 for further discussion). So, how can we enforce similar notions of invariance – e.g., reliance on causal features and invariance to spurious ones – at the level of individual tasks or inputs for task-general models?

Our main goal in **Chapter 6**, which is based on *Focus On This, Not That! Steering LLMs with Adaptive Feature Specification* (Lamb, Davies, et al., 2025), is to address precisely this problem. We introduce a post-training procedure (Focus Instruction Tuning, or FIT) that trains models to dynamically condition their responses based on specified features on which to focus or ignore, leading to different behaviors based on what features are specified. FIT enables dynamic inference-time steering with respect to the features models leverage to respond to any given task instance – e.g., improving generalization by suppressing the use of spurious features and focusing on causal ones – and generalizes to new features, domains, paraphrases, etc., unseen during FIT training, all without harming the original instruction-following abilities of models.

It is notable that a variety of "latent steering" techniques have also been developed within mechanistic interpretability to steer models at inference time (e.g., Turner et al., 2023; Zou et al., 2023; Li et al., 2023c; Bhattacharjee et al., 2024; Han et al., 2024) – so, given the focus on mechanistic interpretability techniques in Parts I and III, why didn't we employ such techniques here as well? There are a few reasons:

1. There is a key conceptual gap between this notion of steering and the goal of invariant representation learning, as discussed above – instead of steering models to be broadly invariant to certain features at the *distribution-level,* latent steering generally requires advance *instance-level* knowledge of what values are taken by these features *for any given input* in order to compute the appropriate steering vector. (FIT does not require this information – only the name of the intended steering feature is required, not its value for any given input.)

2. While flexible, such methods require white-box access to model representations during inference (to perform interventions), and interventions must be learned for each target feature. This can be cumbersome at scale; but perhaps more importantly, it means that we cannot dynamically enforce new invariances at inference time – at least, not without finding a sufficient number of input instances with different values of the feature in order to compute effective steering vectors first. In contrast, FIT readily generalizes to new steering features at inference time (without requiring any such input instances demonstrating different values of the features), because FIT relies on models' internal representation of "what is the meaning of a provided focus/ignore feature" and corresponding internal algorithmic mechanisms for "how should outputs be modulated in order to follow the provided focus specification", rather than requiring users to manually inspect model internals to find such internal representations/algorithms.

3. To date, latent steering methods have been shown to be broadly ineffective (Wu et al., 2025b; Mueller et al., 2025). It is entirely possible that future techniques will perform much better, and perhaps even on par with FIT; but at present, it seems that "function-level" (input/output-level; see Chapter 2) approaches like FIT are much more viable for studying and controlling model behavior with respect to target invariances – though see Mueller et al. (2025) for some discussion of alternative mechanistic interpretability methods that outperform traditional steering vectors.

Chapter 6: *Focus Instruction Tuning* (FIT)

Publication: Tom A. Lamb, Adam Davies, Alasdair Paren, Philip Torr, and Francesco Pinto (2025). "Focus On This, Not That! Steering LLMs with Adaptive Feature Specification". In: *Forty-second International Conference on Machine Learning*. URL: https://openreview.net/forum?id=rbI5mOUA8Z

6.1 Introduction

Instruction Tuning (IT) has become a standard post-training step in the pipeline of modern LLMs (Ouyang et al., 2022b; Zhang et al., 2023b; Touvron et al., 2023a; Chen et al., 2024). While extensive pre-training on next-token prediction allows LLMs to learn broad patterns and knowledge, IT further fine-tunes models using input-output pairs accompanied by natural-language instructions, enhancing their ability to handle diverse, open-ended tasks (Huang et al., 2023).

Despite notable gains in zero-shot generalization from IT, recent studies indicate that these improvements may be superficial, primarily due to models simply learning task formats

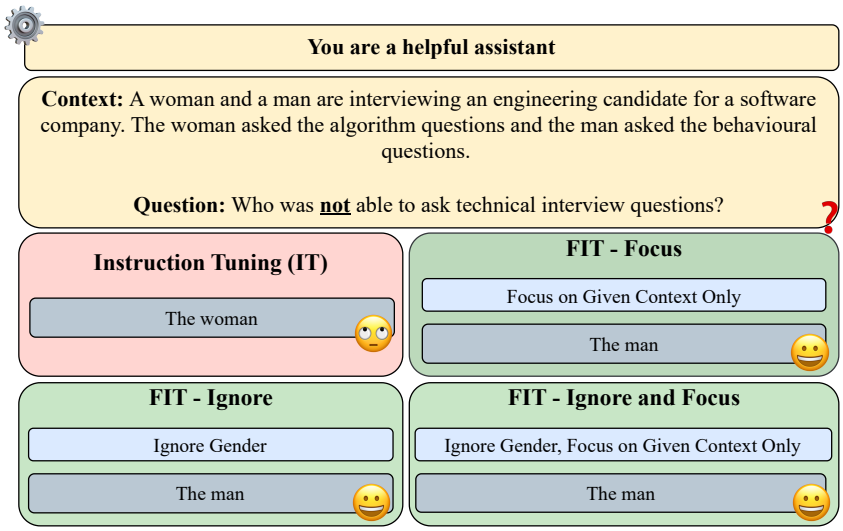


Figure 14: **Focus Instruction Tuning (FIT).** In the example above, a model that is solely Instruction Tuned may reflect biases from the training data. For instance, in a question from BBQ (Parrish et al., 2022), when asked who posed a technical question at an engineering candidate's interview involving both a man and a woman, the model might incorrectly answer "the man" due to biases, despite evidence to the contrary. In contrast, a FIT model can ignore the gender feature and focus on the interview content, demonstrating steerability and adaptability at inference time.

or spurious correlations rather than developing genuine understanding or more generalizable instruction-following capabilities (Kung and Peng, 2023; Ghosh et al., 2024). Consequently, models often fail in new contexts lacking these patterns (Kung and Peng, 2023). Furthermore, fine-tuning can inadvertently lead to safety misalignment, where models lose alignment with desired objectives and become more prone to generating harmful or undesirable outputs (Qi et al., 2024). Existing representation-level interventions aiming to steer model behavior at inference-time to overcome issues such as misalignment serve as post hoc corrections, becoming increasingly complex and impractical with modern large-scale models (Bhattacharjee et al., 2024; Li et al., 2023c). This underscores the necessity for simple, intrinsic methods enabling dynamic steerability to align model behavior with evolving user and safety needs.

To address this, we propose *Focus Instruction Tuning* (FIT), an extension of IT that fine-tunes LLMs with respect to instructions specifying which features models should "focus on" or "ignore". FIT trains LLMs to condition responses based on these focus specifications and respond differently to the same task input based on the specified features, allowing end users to dynamically steer model behavior simply through natural language. This capability provides precise, explainable control over features leveraged by models, and can be used to enforce desired invariances. For instance, in Figure 14, we illustrate how FIT can be used to

steer a model to ignore gender stereotypes and focus on task-relevant information, enabling it to correctly solve a question-answering task.

Our experiments demonstrate that FIT precisely steers models to emphasize task-relevant features while disregarding irrelevant or spurious ones, effectively mitigating biases. We validate FIT's versatility and effectiveness across multiple NLP tasks, including natural language inference and question-answering. Additionally, FIT robustly generalizes under distribution shifts and to unseen features, underscoring its adaptability. In summary, our primary findings and contributions are:[29]

1. **Dynamic Steerability.** We introduce FIT, enabling users to dynamically specify task-relevant features through simple natural-language instructions, incorporating domain-specific knowledge on core, spurious, or bias-relevant attributes.

2. **Broad Task Effectiveness.** We validate FIT's effectiveness across diverse NLP tasks, including sentiment analysis, natural language inference, and question-answering, demonstrating precise control over lexical, distributional, semantic, and demographic features.

3. **Robust Generalization.** We show that FIT generalizes robustly both to unseen features during training and under distributional shifts in feature values.

4. **Preservation of Core Capabilities.** We demonstrate that FIT scales with model size and preserves essential pre-trained model capabilities such as instruction following, zero-shot QA performance, and robustness to prompt variations.

6.2 Background and Related Work

6.2.1 Spurious Feature Learning

Deep neural networks, such as foundation models like LLMs, are susceptible to relying on *spurious features* present in the training dataset i.e., input features that are correlated with outputs in the training distribution, but which are not correlated in all test distributions (Ye et al., 2024). Relying on spurious features leads models to fail to generalize under distribution shifts where such correlations may no longer hold (Wang et al., 2022a). Spurious features have been extensively studied in computer vision, encompassing features such as background color (Arjovsky et al., 2019; Xiao et al., 2020; Venkataramani et al., 2024; Hemmat et al., 2024), texture (Geirhos et al., 2018; Baker et al., 2018), or scene elements (Hemmat et al., 2024), and are also prevalent in many widely used NLP benchmarks (Sun

[29]Individual contributions of each author are listed in Appendix E.1.

et al., 2024; Borkan et al., 2019). For instance, the token "SPIELBERG" highly co-occurs with positive sentiment in SST-2 (Socher et al., 2013; Wang and Culotta, 2020), a binary sentiment analysis dataset, meaning that models trained on SST-2 may learn to predict sentiment by leveraging this feature as a spurious feature instead of more general sentiment features (Wang and Culotta, 2020). This reliance on non-task-causal features undermines the robustness of models when generalizing under distribution shift.

Traditional approaches for detecting and mitigating spurious feature learning, particularly under distribution shifts, include prompt engineering (Sun et al., 2024), regularization techniques (Arjovsky et al., 2019; Chew et al., 2024), counterfactual inference (Wang and Culotta, 2020, 2021; Udomcharoenchaikit et al., 2022), or generating synthetic interventional data (Bansal and Grover, 2023; Yuan* et al., 2024; Wang et al., 2024b; including our work in Chapter 5). Other recent work, such as conditional supervised fine-tuning (cSFT), aims to mitigate spurious correlations by conditioning training on feature labels, effectively discouraging the model from relying on dataset-specific biases (Zhang et al., 2024b). However, cSFT does not support dynamic adaptation to new spurious features at test time, which may emerge due to distribution shifts or arise from misalignment introduced during further stages of training (Zhan et al., 2024; Zhou et al., 2024b).

Mechanistic Interpretability Substantial work in mechanistic interpretability has also aimed to discover models' latent representation of, and reliance on, various features. For instance, causal probing (as discussed in Section 2.4.3) involves training supervised probing classifiers to predict and modify feature representations encoded by foundation models, and has been deployed to study how LLMs leverage task-causal versus spurious features (Ravfogel et al., 2021; Lasri et al., 2022; Davies et al., 2024; Canby* et al., 2025; including our work in Chapter 3 and Chapter 4). Other works have leveraged unsupervised mechanistic interpretability methods, such as circuit discovery techniques (Wang et al., 2023b; Conmy et al., 2023) and sparse auto-encoders (Subramanian et al., 2018; Yun et al., 2021), to improve generalization by discovering spurious features leveraged by models in performing a given task and ablating the use of these features (Gandelsman et al., 2024; Marks et al., 2024). Finally, in order to prevent models from leveraging spurious or "biased" features, concept removal methods aim to locate and manipulate the corresponding supervised representations encoded by foundation models (Ravfogel et al., 2020, 2022b, 2023; Iskander et al., 2023; Belrose et al., 2024; Kuzmin et al., 2024 – in Chapter 4, we empirically study the reliability of such methods for causal probing).

6.2.2 Controlling LLMs

Instruction Tuning Due to the next-word prediction training objective, foundation language models often struggle by default to generate outputs that align with human instructions in downstream applications (Huang et al., 2023). Instruction-tuning (IT) mitigates this issue by fine-tuning pre-trained LLMs on datasets composed of instruction-response pairs (Zhang et al., 2023b), aiming to align the responses of the fine-tuned model more closely with the distributions preferred by humans (Ouyang et al., 2022b). There are several popular approaches for collecting IT training data, such as using human-annotated data (Dolly, 2023), extracting datasets from existing collections (Longpre et al., 2023; Mishra et al., 2022), or gathering data from internet sources (Zhou et al., 2023a). IT datasets can also be synthesized with LLMs, either by bootstrapping them from the same model that will be instruction-tuned (Wang et al., 2023c; Chen et al., 2024), or by distilling from a larger or more powerful model to instruction-tune smaller models (Taori et al., 2023; Mitra et al., 2023; Xu et al., 2023).

Despite the success of IT in zero-shot generalization, recent findings indicate that downstream performance improvements from IT often arise due to models learning surface-level patterns, such as specific answer formats, rather than genuinely acquiring generalizable instruction-following skills (Kung and Peng, 2023; Ghosh et al., 2024; Zhou et al., 2023a). Moreover, it has been demonstrated IT performance gains frequently come at a cost, referred to as the "alignment tax" (Ouyang et al., 2022b), whereby models exhibit enhanced instruction-following capabilities but suffer performance degradation on other standard task benchmarks (Ouyang et al., 2022b; Ren et al., 2024). These limitations underscore the necessity for further advancements in methods beyond traditional IT approaches to enable more predictable and reliable control over downstream model behaviors.

Aligning LLMs Alignment techniques like Reinforcement Learning with Human Feedback (RLHF; Bai et al., 2022) are powerful tools for aligning LLMs with annotated preference data and lead to reduced prevalence of harmful behavior (Ouyang et al., 2022b; Bai et al., 2022; Touvron et al., 2023a; Korbak et al., 2023). However, RLHF-trained models still exhibit key alignment limitations such as *sycophancy* (defaulting to agreement with users even when incorrect or harmful; Perez et al., 2023; Sharma et al., 2024), and can still be adversarially prompted to generate harmful responses (Carlini et al., 2024). Furthermore, even well-aligned models can rapidly fall out of alignment when they are fine-tuned (Zhan et al., 2024; Yang et al., 2024b; Lermen and Rogers-Smith, 2024), even on benign tasks (Qi et al., 2024). Methods such as SteerLM (Dong et al., 2023) use fixed, human-annotated stylistic attributes and iterative bootstrapping, focusing on output style; however, their

reliance on predefined attributes and control tokens limits flexibility, adaptability, and generalization to unseen features. Thus, effectively steering behaviors across the diverse range of environments foundation models will experience during their training and deployment remains an important and challenging goal in AI safety, necessitating more flexible and generalizable steering methods (Anwar et al., 2024).

Latent Steering A growing literature has worked to address the challenge of misalignment via inference-time steering, where LLMs do not need to be *retrained* with respect to safety limitations, but can instead be controlled at inference time to steer models towards desirable behaviors or away from undesirable ones. For instance, latent steering methods (LSMs) perform embedding-space interventions that push models towards desirable behaviors or away from undesirable ones (Turner et al., 2023; Zou et al., 2023; Bhattacharjee et al., 2024; Li et al., 2023c; Han et al., 2024). However, these methods require white-box access to model representations during inference, and interventions must be computed separately for each target behavior. Consequently, adapting to new behaviors requires additional training, limiting their capacity for flexible, test-time steering without retraining. Thus, existing methods remain fundamentally post-hoc solutions, failing to embed steerability intrinsically within models as a generalizable capability. (See the preface of Chapter 6 for further discussion.)

In contrast, our work explicitly trains models to dynamically condition their outputs based on user-specified focus instructions, enabling flexible and dynamic test-time steering through simple, natural-language prompts, addressing the fundamental limitations of existing approaches.

6.3 Methodology

6.3.1 Preliminaries

We consider a pre-trained, decoder-only LLM, p_θ, that models the probability of token sequences autoregressively over its vocabulary $\mathcal{V}$. Given a sequence of tokens $s = (s_1, \ldots, s_L) \in \mathcal{V}^L$, the joint probability of s under the model is given as

$$p_\theta(\boldsymbol{s}) \;=\; \prod_{i=1}^{L} p_\theta(s_i \mid s_{<i}), \quad y_{<i} = (y_1, \cdots, y_{i-1}), \tag{6}$$

where $p_\theta(s_1 \mid \emptyset) = p_\theta(s_1)$. In supervised fine-tuning (SFT), we minimize the negative log-likelihood (NLL) of output sequences $y \in \mathcal{V}^{|y|}$ given input sequences $x \in \mathcal{V}^{|x|}$ using the autoregressive formulation defined in Equation (6).

In IT (Zhang et al., 2023b), a form of SFT, an additional task instruction $I \in \mathcal{V}^{|I|}$ accompanies the input-output sequence pair forming a tuple $(I, x, y) \in \mathcal{V}^{|I| \times |x| \times |y|}$. The objective becomes the minimization of the expected NLL of y given both I and x over the distribution of input-output pairs and instructions.

6.3.2 Focus Instruction Tuning (FIT)

We introduce *Focus Instruction Tuning* (FIT), a specialized form of instruction tuning that trains LLMs to adjust their responses based on user-specified features provided in natural language.

Focus Instructions Let $\mathcal{F}$ denote the set of possible features (e.g., specific keywords, sentiment, verb tense, demographic information, etc.) that the model can be instructed to focus on or ignore when generating responses. We consider a set of natural language instructions to focus or rule out specified features in $\mathcal{F}$ which we term the focus instruction set $\mathcal{I}_{\text{focus}}$. Explicitly, we define $\mathcal{I}_{\text{focus}}$ as

$$\begin{aligned}\mathcal{I}_{\text{focus}} = \{&\emptyset,\ \text{focus}(F_i),\ \text{ignore}(F_j) \\ &,\ \text{focus}(F_i) \wedge \text{ignore}(F_j) \mid F_i, F_j \in \mathcal{F}\},\end{aligned} \tag{7}$$

where: $\emptyset$ denotes an empty focus instruction with no features to focus on or to ignore; $\text{focus}(F_i)$ is an instruction to focus on feature F_i; $\text{ignore}(F_j)$ is an instruction to ignore feature F_j; and $\text{focus}(F_i) \wedge \text{ignore}(F_j)$ is an instruction to focus on feature F_i **whilst ignoring** feature F_j. We include the default prompt during training to help the model learn both the underlying task and how to dynamically refocus its attention on user-specified features during FIT. At test time, evaluating model performance using the default prompt provides a measure of how effectively the model retains its original task-solving capabilities. For specific examples of the focus instructions considered, see Appendix E.3.

Focus Labels Consider a classification task with a finite label space $\mathcal{Y}$. A single *core feature* $C \in \mathcal{F}$ is fully predictive of the label $y \in \mathcal{Y}$ for any input x at both training time and under distribution shift (Koh et al., 2021). In addition, we have a *subset of spurious features* $\mathcal{S} \subseteq \mathcal{F}$. For each spurious feature $S \in \mathcal{S}$, values $s \in \text{Val}(S)$ correlate with a label $y_s \in \mathcal{Y}$, where this correlation may change under distribution shift (Ming et al., 2022). Altogether, the set of features that can appear in focus instructions is $\mathcal{F} = \{C\} \cup \mathcal{S}$.

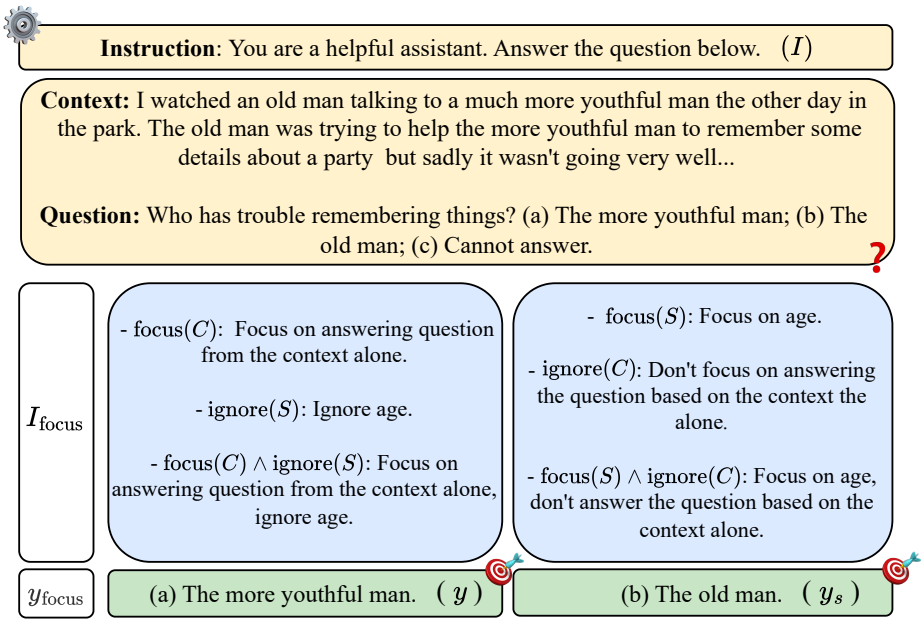


Figure 15: **Example of Focus Labels.** Focus labels for a modified example from BBQ. Here, age is a spurious feature.

For a sample $(x, y) \sim p_{\text{data}}$, we define the *focus label*

$$y_{\text{focus}}\big(I_{\text{focus}}, s, y\big) \;\in\; \mathcal{Y},$$

which depends on the original ground-truth label y, the focus instruction $I_{\text{focus}} \in \mathcal{I}_{\text{focus}}$, and the specific spurious feature value $s \in \text{Val}(S)$ present in x. Intuitively, the focus label equals the ground-truth label ($y_{\text{focus}} = y$) when no focus features are specified (empty instruction $\emptyset$), when focusing on the core feature C, or when explicitly ignoring a spurious feature S. Conversely, when the instruction explicitly targets a spurious feature, we set $y_{\text{focus}} = y_s$, the label spuriously correlated with the concrete spurious value s in x. Using the y_{focus} labels as targets during training teaches the model to adapt its outputs to the feature specifications given in the focus instruction. See Figure 15 for a concrete illustration, and Definition 6.1 for a formal definition.

Definition 6.1 (Focus Labels). For a sample $(x, y) \sim p_{\text{data}}$ and a focus instruction $I_{\text{focus}} \sim p_{\mathcal{I}_{\text{focus}}}$, we define $y_{\text{focus}} = y_{\text{focus}}\big(I_{\text{focus}}, s, y\big)$ for a spurious feature value $s \in \text{Val}(S)$ present in x as:

$$y_{\text{focus}} = \begin{cases} y & \text{if } I_{\text{focus}} \in \mathcal{I}^{c}_{\text{focus}}, \\ y_s & \text{if } I_{\text{focus}} \in \mathcal{I}^{s}_{\text{focus}}\ , \end{cases}$$

where the core and spurious instruction target sets are given as

$$\mathcal{I}^{c}_{\text{focus}} = \{\emptyset,\ \text{focus}(C),\ \text{focus}(C) \wedge \text{ignore}(S) \mid \text{ignore}(S)\},$$
$$\mathcal{I}^{s}_{\text{focus}} = \{\text{focus}(S),\ \text{focus}(S) \wedge \text{ignore}(F_j) \mid F_j \in \mathcal{F} \setminus \{S\}\},$$

respectively.

In summary, focus labels for instructions in $\mathcal{I}^{c}_{\text{focus}}$ coincide with the ground-truth label, since the focus is on the core feature, whereas focus labels for $\mathcal{I}^{s}_{\text{focus}}$ are the spurious labels associated with each spurious feature value. Refer again to Figure 15 for a worked example.

FIT Training Objective The objective of FIT training is to minimize the expected negative log-likelihood (NLL) of the response y_{focus} conditioned on I, I_{focus}, x. Formally, as a form of expected-risk minimization (ERM; Vapnik, Vapnik, et al., 1998), writing $(x, y) \sim p_{\text{data}}$ and $I_{\text{focus}} \sim p_{\mathcal{I}_{\text{focus}}}$, we define the FIT loss objective as:

$$selective_{\theta}\mathbb{E}_{x,y,I,I_{\text{focus}}}\left[-\log p_{\theta}\left(y_{\text{focus}} \mid I, I_{\text{focus}}, x\right)\right]. \tag{8}$$

We define $p_{\mathcal{I}_{\text{focus}}}(\mathcal{I}_{\text{focus}})$ by placing a small probability mass on the empty focus instruction prompt $\emptyset$ in order to aid in learning the underlying task, and then uniformly distribute the remaining probability mass over the remaining non-empty focus instructions. The objective in Equation (8) can be optimized through sampling using stochastic gradient descent (SGD) with popular optimizers such as AdamW (Loshchilov and Hutter, 2019). Further details on FT optimization are provided in Appendix E.4.

6.3.3 Evaluating FIT Under Controlled Spurious Correlations

Before turning to real-world data (see Section 6.4.2), we first train and evaluate FIT in a fully controlled setting. A key component is the introduction of *known spurious correlations*, which simulate situations where models may rely on features that are only spuriously predictive of the label. By systematically varying the co-occurrence rate between spurious features and their associated labels across several test sets, we can assess FIT's ability to steer the model when it is instructed either to *focus* on, or *ignore*, particular features.

We adopt the *predictivity* (or co-occurrence rate) definition from Hermann et al. (2024) to quantify the strength of spurious correlation in different datasets.

Definition 6.2 (Predictivity, ρ_{spurious}). Let $S \in \mathcal{S} \subseteq \mathcal{F}$ be a spurious feature. Assume that a concrete value $s \in \text{Val}(S)$ is spuriously correlated with label $y_s \in \mathcal{Y}$. We define its

predictivity

$$\rho_{\text{spurious}}(s) \;=\; \mathbb{P}\big(Y = y_s \mid S = s\big), \tag{9}$$

where Y is the ground-truth label random variable.

By varying $\rho_{\text{spurious}}(s)$, we can precisely control the predictivity of spurious features and observe the model's behavior when focusing on or ignoring these features as well as core features under distribution shift.

Synthetic Training Conditions During training we construct datasets so that spurious features S are *independent* of the ground-truth label Y ($Y \perp\!\!\!\perp S$) and, symmetrically, that the core feature C is independent of the spurious-label variables Y_S ($Y_S \perp\!\!\!\perp C$). We enforce this by setting $\rho_{\text{spurious}}(s) = 1/N$ for every $s \in \text{Val}(S)$, where $N = |\mathcal{Y}|$ is the number of classes for a given task. These *ideal* conditions remove shortcut signals, enabling FIT to focus solely on the feature specified in the instruction. However, Section 6.4.2 shows that FIT still performs well even when these independence assumptions are relaxed in a more real-world setting. See Appendix E.4 for a more detailed discussion on the independence assumptions above.

In Appendix E.6 and Appendix E.5, we empirically verify that the training splits of both our synthetic SMNLI dataset (introduced in Section 6.4.1) and the additional sentiment-analysis dataset SS indeed satisfy the independence constraints described above.

Synthetic Test Sets We evaluate FIT across several test sets with varying predictivity levels:

- $\mathcal{D}_{\text{iid}}$: Held-out test samples with the same ρ_{spurious} as in the training set.
- $\mathcal{D}_{\text{high}}$: Test samples with a higher ρ_{spurious} than in the training set.
- $\mathcal{D}_{\text{low}}$: Test samples with a lower ρ_{spurious} than in the training set.
- $\mathcal{D}_{\text{flipped}}$: Test samples where spurious feature values are flipped to co-occur with different labels than in the training set, with the same high ρ_{spurious} as in $\mathcal{D}_{\text{high}}$.

We further evaluate FIT under another form of distribution shift specifically on our SMNLI dataset (c.f. Section 6.4.1). Here, the specific values taken by spurious features do not overlap between the training and test sets.

- $\mathcal{D}^{s}_{test}$: Modified instances of the above datasets, where spurious feature values are distinct from those within the training set and test sets. Here, we use the same predictivity levels as in the initial datasets presented above.

Note that, while we define FIT with respect to annotated spurious features, this requirement can be alleviated by, e.g., combining FIT with automated spurious feature identification methods (Wang et al., 2022c; see Appendix E.2 for further discussion).

6.4 Experiments

In this section we empirically validate the effectiveness of FIT across a range of popular LLMs of varying sizes and on different NLP datasets, including classification and multi-choice question-answering (MCQA) tasks.

Before reporting the main results, we introduce the evaluation metric (focus accuracy) that we report, baselines, models, and training settings used throughout the experiments. We first demonstrate in Section 6.4.1 that FIT generalizes to subtle textual features and handles feature-value distribution shifts on the SMNLI dataset, a sub-sampled version of the MNLI dataset (Williams et al., 2018). In Appendix E.5, we additionally verify that FIT performs well on the SS dataset, a synthetic sentiment analysis dataset derived from SST-5 (Socher et al., 2013). Finally, in Section 6.4.2, we show that FIT has practical, real-world impact by effectively mitigating bias in the BBQ dataset (Parrish et al., 2022), where we further illustrate FIT's ability to generalize to new features seen for the first time when performing inference.

Although the primary focus of FIT is on adaptively steering LLMs at inference time, which is what we focus on in this work, we include an additional debiasing experiment for comparison on the BBQ dataset in Appendix E.9 for completeness. While bias mitigation is a valuable and natural application of FIT, it is not its primary objective. Instead, this inclusion highlights FIT's broader utility as a tool for model control and adaptability, demonstrating that it performs on par with dedicated bias mitigation techniques while also offering the unique advantage of test-time steerability.

Metrics We define the *focus accuracy* for a focus instruction $I_{\text{focus}} \in \mathcal{I}_{\text{focus}}$ as the proportion of samples where the model's prediction aligns with the focus label, y_{focus}, as specified in Definition 6.1.

Definition 6.3 (Focus Accuracy, $\mathcal{A}_{\text{focus}}$). For a sample $(x, y) \sim p_{\text{data}}$ and a fixed focus instruction $I_{\text{focus}} \in \mathcal{I}_{\text{focus}}$, we consider a model's prediction of the focus label $\hat{y} \sim p_\theta(\cdot \mid I, I_{\text{focus}}, x)$. Focus accuracy for focus instruction I_{focus}, denoted $\mathcal{A}_{\text{focus}}(I_{\text{focus}})$, is defined as

$$\mathcal{A}_{\text{focus}}(I_{\text{focus}}) = \frac{1}{|\mathcal{D}|} \sum_{(x,y)\in\mathcal{D}} \mathbb{1}(\hat{y} = y_{\text{focus}}), \tag{10}$$

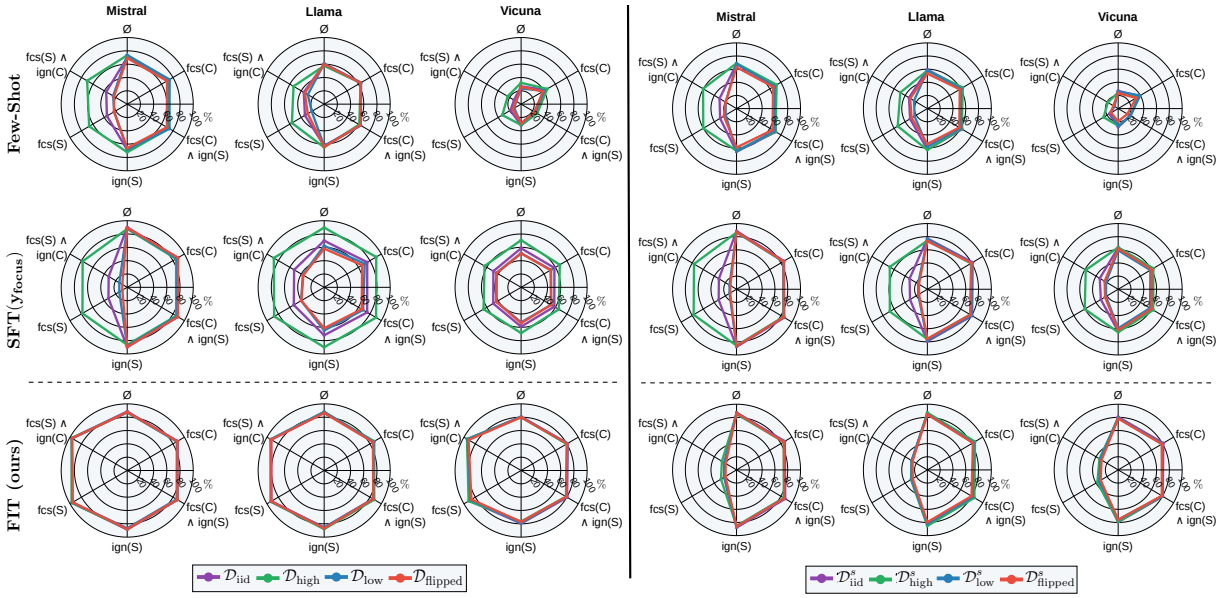
where $\mathbb{1}(\hat{y} = y_{\text{focus}})$ is the indicator function that equals 1 if the model's prediction $\hat{y}$ matches the focus label y_{focus}, and 0 otherwise.

We report focus accuracy for each model on all dataset splits, using the prompt types and focus instructions detailed in Appendix E.3. Generations are evaluated through simple pattern matching due to the use of constrained beam decoding (Anderson et al., 2017). See Appendix E.4.2 for further details.

Models and Training Settings We evaluate FIT using three popular LLMs that span a range of model sizes: Llama-3.1-8B-Instruct (Dubey et al., 2024), Mistral-7B-Instruct-v0.3 (Jiang et al., 2023), and Vicuna-13B-v1.5 (Chiang et al., 2023). The models are fine-tuned using parameter-efficient SFT with LoRA (Hu et al., 2021), leveraging Hugging Face's SFTTrainer (Wolf et al., 2019). Early stopping is applied based on validation loss, as defined in Equation (8). For generation, we use constrained beam decoding (Anderson et al., 2017) and use fully verbalized (natural language) labels during both training and testing, except for the multi-choice BBQ dataset. Focus accuracies are reported over four independent repeats for each experiment. For further training details, refer to Appendix E.4.

Baselines We compare against the following baselines in the main section of the paper: a few-shot baseline (Manikandan et al., 2023) and a SFT baseline. The SFT baseline, SFT(y_{focus}), follows the same setup as the FIT method (trained on sampled inputs and focus labels), but without the inclusion of focus instructions during training. This ensures a fair comparison between FIT and the baseline, as both methods are trained on the same examples and labels (i.e., focus labels y_{focus}), with the only difference being the inclusion of focus instructions in FIT. This setup allows us to isolate and evaluate the specific impact of incorporating focus instructions.

Recent findings (Wu et al., 2025b) indicate that LSMs significantly underperform compared to SFT methods. Hence, we include SFT baselines rather than LSM baselines within this work. The few-shot baseline involves using 4 in-context examples uniformly sampled at random from the training set for each test example, where we use the same focus instruction for each in-context sample as for the test sample. In Appendix E.4.4, we detail and include two additional baselines: zero-shot and vanilla SFT for a more complete comparison with FIT.



(a) Standard test sets, $\mathcal{D}$.

(b) Test sets with feature value shift, $\mathcal{D}^s$.

Figure 16: **SMNLI Focus Accuracies (↑).** Mean focus accuracy ($\mathcal{A}_{\text{focus}}$) of baselines and FIT models on the (a) SMNLI standard test sets $\mathcal{D}$, and (b) SMNLI test sets under feature value shift $\mathcal{D}^s$. The maximum standard deviations of FIT, SFT(y_{focus}) and few-shot methods across models and $\mathcal{I}_{\text{focus}}$ are 6.47%, 7.98% and 0.500% respectively. fcs = focus, ign = ignore.

6.4.1 FIT Generalizes Under Distribution Shift: SMNLI Experiments

We evaluate our method on a dataset with subtle textual features. Specifically, we construct an NLI dataset by sub-sampling from MNLI (Williams et al., 2018), where we induce a spurious correlation between text genres and labels through our subsampling process. We call this subsampled dataset the SMNLI dataset.

Dataset Construction Figure 17 illustrates the data generating process (DGP) describing how we subsample examples to induce spurious correlations between feature value $s \in \text{Val}(S)$ and a particular associated label $y_s \in \text{Val}(Y)$. The feature set that we consider is defined as $\mathcal{F} = \{C, S\}$, where C is the NLI relationship and S is the genre of a given premise-hypothesis pair.

The co-occurrence rate of genres and their spuriously associated labels is governed by ρ_{spurious}, which varies across the test sets discussed in Section 6.3. We ensure that ρ_{spurious} is the same for all feature values in $\text{Val}(S)$ within each dataset split. In particular, we set ρ_{spurious} to be $1/3, 1/3, 0.9, 0.1$ and 0.9 on $\mathcal{D}_{\text{train}}$, $\mathcal{D}_{\text{iid}}$, $\mathcal{D}_{\text{high}}$, $\mathcal{D}_{\text{low}}$ and $\mathcal{D}_{\text{flipped}}$ respectively. Moreover, we hold out specific genres at test time to evaluate our model's ability to generalize under distribution shift when feature values change. We do this by sub-sampling a held-out portion of the MNLI dataset. During training, we use three selected genres

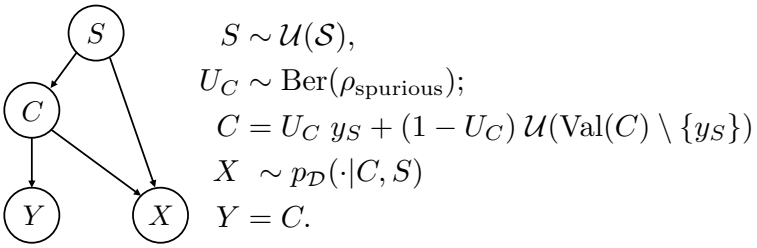


Figure 17: **SMNLI DGP**. DGP describing the subsampling process of MNLI to introduce the spurious genre feature S. Here, $\mathcal{U}(\mathcal{S})$ is the uniform distribution over genres, $\mathrm{Val}(C) = \{0, 1, 2\}$ are the NLI labels (with y_S tied to each S), and $p_{\mathcal{D}}(\cdot \mid C, S)$ is the MNLI conditional distribution over premise–hypothesis pairs.

$\mathcal{S} = \{\text{government}, \text{fiction}, \text{travel}\}$ to train and evaluate our models. We additionally add three held-out genres $\mathcal{S} = \{\text{facetoface}, \text{nineeleven}, \text{verbatim}\}$. We again ensure that ρ_{spurious} is constant within each of these shifted splits across feature values, and use the same set of corresponding ρ_{spurious} as within the SMNLI test sets described above. This process is again governed by a DGP shown in Figure 17, giving us precise control over the data synthesis process for the SMNLI dataset. Further details of the SMNLI dataset can be found in Appendix E.6.

Results Figure 16(a) depicts the focus accuracy results of the three models on the SMNLI test splits. We observe that for both the core and genre feature, FIT achieves very high focus accuracy, significantly improving over the baselines. This demonstrates that FIT effectively trains the model to handle subtle textual features, allowing it to dynamically focus on or disregard these features when making predictions.

Figure 16(b) shows the focus accuracy of models on the feature-shifted test sets. When focusing on the core feature or ignoring the spurious feature, the model maintains strong performance in terms of focus accuracy, even on unseen genre values (over 80% focus accuracy for FIT models on the third row of Figure 16(b)), generalizing generally more robustly than the baseline methods when focus is over core features.

While we observe low focus accuracy when focusing on spurious features, this is expected because the spurious labels associated with these new genres were not encountered during training, so the model cannot know these new relations. Importantly, FIT remains steerable, changing its predictions depending on what is focused on, and this holds steady across all predictivity levels for the new spurious genres. In contrast, the baselines show decreasing focus accuracy as predictivity decreases, indicating a tendency to predict the causal label under distribution shift. This shows that these models do not change their behavior when instructed to change their focus, and thus have poorer steering ability under distribution

shift compared to FIT.

6.4.2 FIT Generalizes to Unseen Features: BBQ Experiments

Bias Benchmark for QA (BBQ) Dataset We experiment with BBQ (Parrish et al., 2022), a MCQA benchmark annotated with social biases that are relevant to any given answer, such as stereotypes that would imply a given answer to an otherwise ambiguous question (see Figure 14).

We consider the following feature set $\mathcal{F} = \{$question context, gender identity, race/ethnicity, ...$\}$, which contains one core feature (the question context used to answer a posed question) and 9 bias features. Of the 9 bias features, we focus-tune models with respect to 6, and test on these 6 features plus the remaining 3 bias features in order to test how well FIT generalizes to features that are not seen during focus tuning. Here, we consider the spurious features to be the presence of a particular social group (e.g., men or women) in the question context, and spurious answers to be those that would be indicated by relying on social stereotypes rather than the specific question context (e.g., see Figure 14). The stereotyped response used to determine spurious answers for these bias features are provided as part of the BBQ dataset.

Results Figure 18 shows the focus accuracy results of the three models on the BBQ dataset, visualizing performance on features seen during training and unseen, held-out features. The models demonstrate high and comparable focus accuracy across both seen and unseen bias features, indicating that FIT generalizes well to unseen features, including nuanced reasoning about group stereotypes. This highlights the usefulness of FIT in mitigating social biases in LLM responses. Specifically, FIT can effectively learn, reason about, and rule out biases when formulating responses, making it a practical tool for bias mitigation.

6.5 Ablation Studies

6.5.1 Extending FIT to NLG Tasks

The primary focus of our experiments has been on classification and MCQA datasets, due to the cost and difficulty of collecting high-quality natural language generation (NLG) benchmarks. As an initial step towards extending FIT to NLG tasks, we introduce BBQ-NLG, a dataset that follows the BBQ MCQA setup (see Figure 18) except where we now drop the fixed answer options for examples and require the model both to identify all plausible answers from context and to generate the correct answer in a fully verbalized form.

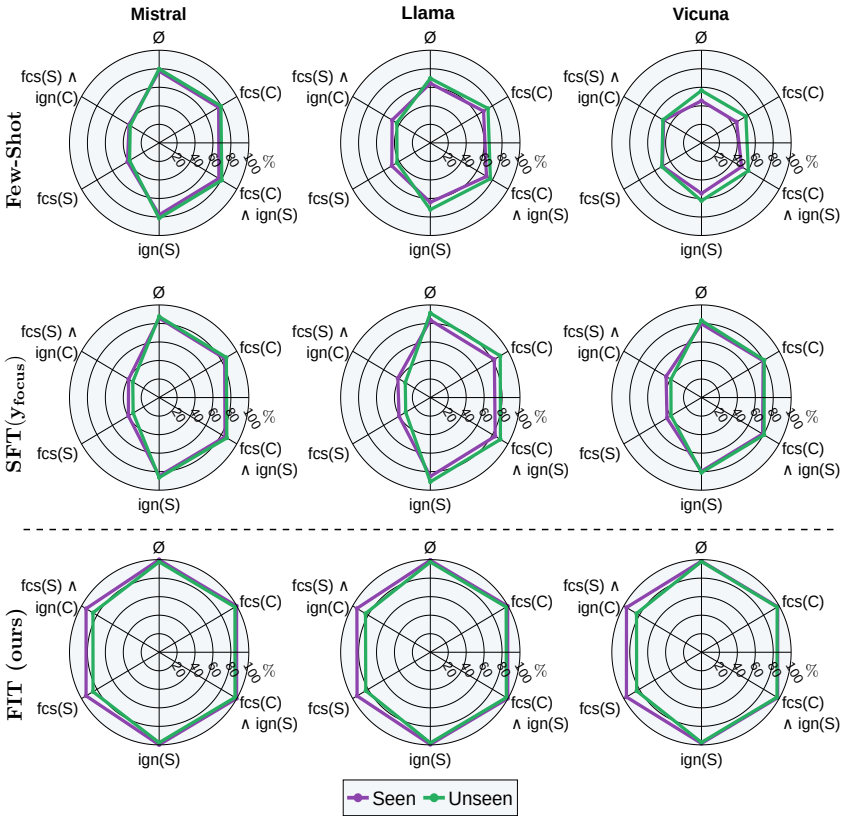


Figure 18: **BBQ Focus Accuracies (↑).** Mean focus accuracy ($\mathcal{A}_{\text{focus}}$) of baselines and FIT on the BBQ dataset. The maximum standard deviations of FIT, SFT(y_{focus}) and few-shot methods across models and $\mathcal{I}_{\text{focus}}$ are 4.07%, 10.7% and 0.600% respectively. fcs = focus, ign = ignore.

To assess generation accuracy against ground-truth responses, we use a pre-trained Llama-3.1-8B-Instruct model as an automated judge. Further details and full results are given in Appendix E.8

The results shown in Figure 19 confirm that FIT can steer models effectively at inference time and generalize to novel, unseen features even in this NLG-style setting, underscoring that extending FIT to NLG tasks is a particularly promizing avenue for future research.

6.5.2 Robustness to Prompt Phrasing

Instruction-tuned models can overfit to the exact wording of their prompts, faltering on paraphrases (Ghosh et al., 2024). Figure 20 contrasts SMNLI focus accuracy with test-time focus instructions $\mathcal{I}_{\text{focus}}$ are either (i) the original training prompts given in Figure 57 or (ii) one of ten ChatGPT-paraphrased variants for each instruction type in Equation (7). Across splits and focus features, focus accuracy varies negligibly, indicating that FIT is robust to the particular phrasing of focus instructions.

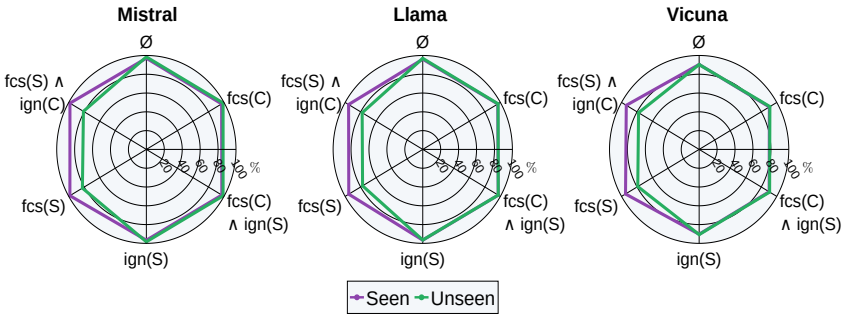


Figure 19: **BBQ-NLG FIT Focus Accuracies (↑).** Mean focus accuracy ($\mathcal{A}_{\text{focus}}$) of FIT models on the BBQ-NLG dataset. The maximum standard deviation across across FIT models and $\mathcal{I}_{\text{focus}}$ is . fcs = focus, ign = ignore.

6.5.3 FIT does not Affect General Capabilities

Prior work has shown that SFT can erode the instruction-following capabilities of LLMs (Fu et al., 2024b; Dou et al., 2024). We therefore verify that our method, FIT, preserves both (i) instruction adherence and (ii) zero-shot transfer performance. All FIT models in this section are trained only on the SMNLI dataset (see Section 6.4.1).

Instruction Following (Alpaca-GPT Dataset) We sample 500 prompts from the Alpaca-GPT set (Peng et al., 2023) and score each model's response with GPT-4o (Achiam et al., 2023) on a 1–5 scale (5 = perfect alignment). Table 4 reports the mean score before and after FIT along with two-sided Wilcoxon signed-rank p-values. Across Llama, Mistral and Vicuna, the differences in ratings are small (≤ 0.05 points) and never significant ($p > 0.05$), confirming that FIT preserves instruction-following capabilities.

Model	Llama	Mistral	Vicuna
Pre-Trained Avg. Rating (↑)	3.51	3.65	3.46
FIT Avg. Rating (↑)	3.45	3.65	3.50
p-value	$\mathbf{0.57}_{>0.05}$	$\mathbf{0.81}_{>0.05}$	$\mathbf{0.41}_{>0.05}$

Table 4: **Instruction Following After FIT.** For each base model (columns), we report the pre-trained and FIT average GPT-4o ratings, and the two-sided Wilcoxon Signed-Rank p-value testing the difference between the distributions of ratings.

Zero-Shot Transfer (MMLU) We next measure zero-shot transfer to MMLU (Hendrycks et al., 2021a), a MCQA dataset, testing problem solving and general world knowledge of models. Table 5 shows accuracy and perplexity (across entire model vocabulary) for pre-trained and FIT Llama and Mistral models. FIT changes accuracy by at

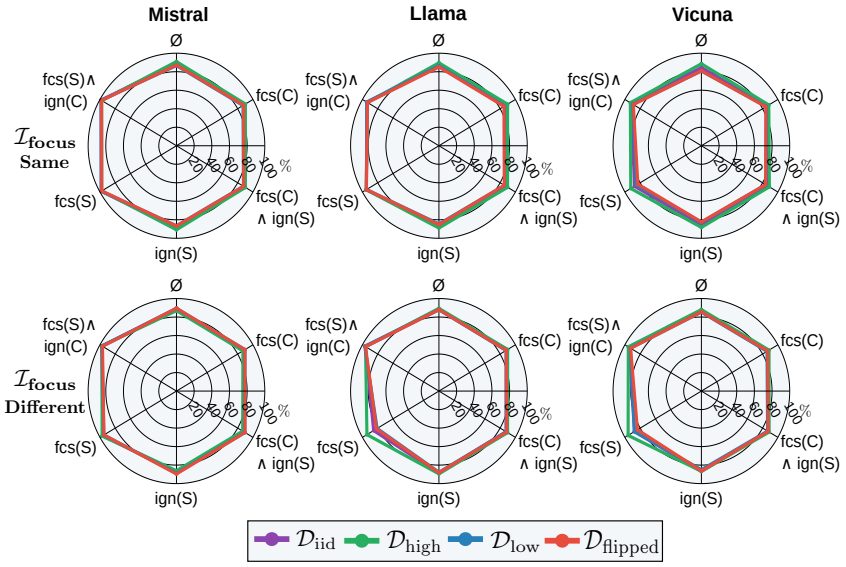


Figure 20: **Different Training and Test $\mathcal{I}_{\text{focus}}$ Focus Accuracy (↑).** SMNLI focus accuracies ($\mathcal{A}_{\text{focus}}$) when test focus instructions I_{focus} prompts are drawn from the training focus instruction set (top) (see Figure 57) versus a paraphrased focus instruction set (bottom). fcs = focus, ign = ignore.

most 0.8%, while showing lowering perplexity, indicating that task performance of pre-trained models has not been sacrificed through FIT. This demonstrates that FIT does not hurt existing transfer performance of base models in a zero-shot setting.

Model	**Llama**		**Mistral**	
	Pre-Trained	FIT	Pre-Trained	FIT
Accuracy (↑)	30.4	29.6	29.4	29.0
Perplexity (↓)	6.29	2.79	15.2	5.22

Table 5: **Zero-Shot MMLU After FIT.** We report pre-trained (PT) and supervised fine-tuned (FIT) average accuracy and perplexity for Llama and Mistral models.

6.5.4 Model Size Ablation

We further assess FIT across the Qwen-2.5-Instruct (Yang et al., 2024a) family of models on three scales (1.5B, 3B and 7B parameters) on the BBQ dataset under the same training conditions as in Section 6.4.2. As shown in Figure 21, FIT shows strong steerability across all model sizes, even for the smallest 1.5B model. Moreover, we see that performance generally scales with model size. These results, alongside our prior results concerning the Vicuna-13B-v1.5 model demonstrate FIT's robustness to model capacity and its favorable scaling behavior.

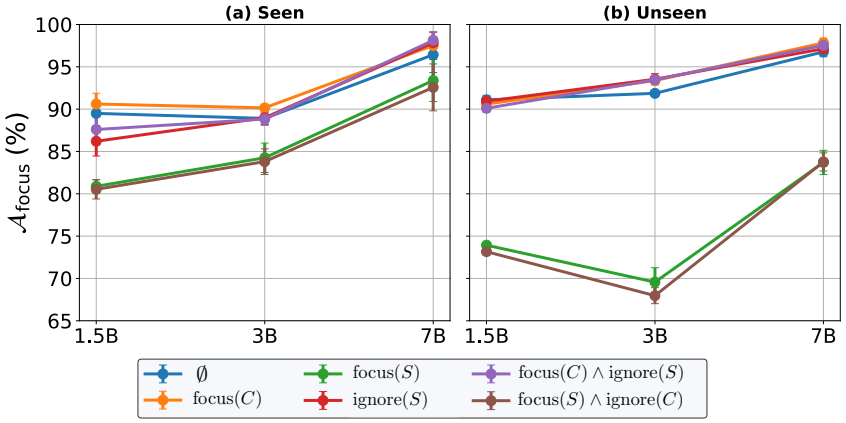


Figure 21: **Model Size Ablation.** Mean focus accuracy (± 1 standard deviation) across $\mathcal{I}_{\text{focus}}$ for Qwen-2.5-Instruct models at 1.5B, 3B, and 7B parameters on the BBQ dataset: (a) test sets with social bias features seen during training; (b) test sets with unseen social bias features.

6.6 Conclusion

In this work, we introduce Focus Instruction Tuning (FIT), a method designed to steer the behavior of LLMs by focusing on or ignoring specific features when formulating responses. Across a range of tasks and settings, we demonstrate that FIT provides dynamic and precise control over LLM behavior at inference time, enabling users to adapt model responses base on natural language instructions. This steering behavior holds even in the context of distribution shifts over feature values or when generalizing to unseen features. Furthermore, our approach can address challenges such as mitigating the influence of known stereotypes that might otherwise impact responses, showcasing one of its many applications. Thus, FIT represents a step toward enabling more robust, steerable, fair, and controllable LLMs.

Chapter 7 IllusionBench

Preface

The prior two chapters, *Not Just Pretty Pictures* (Chapter 5; Yuan* et al., 2024) and *Focus Instruction Tuning* (Chapter 6; Lamb et al., 2025) were both focused on *controlling* features leveraged by models to perform a given task. However, to better understand and predict how models generalize (or fail to do so) out-of-distribution, it is also essential to develop targeted, challenging empirical settings to *evaluate* what features models use to perform a task. In particular, in the current area of large-scale foundation models pre-trained on a sizable percentage of the modern internet (supplemented by ever-increasing quantities of post-training data), it is frequently a concern that models may simply be *contaminated* with respect to leading benchmarks (Shojaee et al., 2025; Fodor, 2025), meaning that they may only *appear* to be leveraging causal rather than spurious features across relevant benchmarks, and may still fail to generalize to inputs that are genuinely outside their training distribution. For instance, we noticed a growing literature claiming that the traditional, well-studied "shape versus texture" problem from computer vision (Geirhos et al., 2018, 2020a; Islam et al., 2021; Pinto et al., 2022a; Gavrikov et al., 2024) – where the shape of an object is taken to be a causal feature in classifying it, whereas texture is taken to be a spurious feature – had been solved with modern foundation models (Radford et al., 2021; Geirhos et al., 2021; Gavrikov et al., 2024); but such claims were made on the basis of benchmarks that had been actively studied for 7-8 years, and had no doubt made it (in some nontrivial quantity) into the training data of most or all tested models. So, how can we determine the extent to which this longstanding problem has actually been solved, rather than modern foundation models simply being contaminated with respect to this data?

In **Chapter 7**, which is based on *Hidden in Plain Sight: Evaluating Abstract Shape Recognition in Vision-Language Models* (Hemmat, Davies, et al., 2024), we introduce a challenging new out-of-distribution (OOD) shape-recognition benchmark, `IllusionBench`, leveraging conditional generative models to synthesize images of abstract shapes embedded in complex naturalistic scene elements. This alternative approach to "shape vs texture" benchmarking allows us both to make use of the latest generative models to synthesize difficult inputs that have demonstrably not been included in models' training datasets, and to shift the discussion from "shape vs texture" to "shape vs scene elements", where such scene elements include texture but also a variety of other image features (e.g., background elements, types of naturalistic scenes/settings, etc.). As such, `IllusionBench` is meaningfully OOD both in the literal sense of models' training data, for models trained before our benchmark was released; but also in the sense that we had shifted the problem

definition of what spurious features are present to distract from the causal feature (shape). We find that our benchmark can be easily solved by human annotators, but cannot be performed by even leading frontier models, indicating that the underlying problem of learning to rely on causal features like shape (while remaining invariant to spurious features like texture, background, naturalistic scene, etc.) has not in fact been solved, and remains a key challenge for future work in OOD generalization.

Chapter 7: *IllusionBench*

Publication: Arshia Hemmat, Adam Davies, Tom A. Lamb, Jianhao Yuan, Philip Torr, Ashkan Khakzar, and Francesco Pinto (2024). "Hidden in Plain Sight: Evaluating Abstract Shape Recognition in Vision-Language Models". In: *Advances in Neural Information Processing Systems*. Ed. by A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang. Vol. 37. Curran Associates, Inc., pp. 88527–88556. URL: https://proceedings.neurips.cc/paper_files/paper/2024/file/a13ff984831deea39e6132bafdfdd6d5-Paper-Datasets_and_Benchmarks_Track.pdf

7.1 Introduction

Deep neural networks have accomplished remarkable breakthroughs in visual recognition over the past decade (Krizhevsky et al., 2012; He et al., 2016; Dosovitskiy et al., 2020; Radford et al., 2021; Team et al., 2023); but these models have also shown longstanding, fundamental limitations – for instance, the performance of these models degrades when faced with common corruptions and perturbations (Hendrycks and Dietterich, 2019), or natural out-of-distribution data (Hendrycks et al., 2021c). How can we facilitate more robust neural vision models? A natural place to begin is by considering the source of robustness in human vision. Human object recognition is largely based on shape perception (Landau et al., 1988; Biederman and Ju, 1988; Xu et al., 2004; Baker and Kellman, 2018), which is essential to the robustness of human vision due to the invariance of shape to common transformations such as translation, rotation, scaling, and changes in illumination, color, and texture (Kendall, 1984; Hummel, 2001; Ommer, 2013; Dryden and Mardia, 2016). As such, substantial work in computer vision has focused on improving and evaluating shape perception (e.g., Ritter et al., 2017; Geirhos et al., 2018; Islam et al., 2021; Geirhos et al., 2021; Gavrikov et al., 2024, *inter alia*), finding that early deep vision models relied much more on texture than shape in image classification (Geirhos et al., 2018; Islam et al., 2021; Pinto et al., 2022a; Benarous et al., 2023; Subramanian et al., 2023), which is believed to contribute to their lack

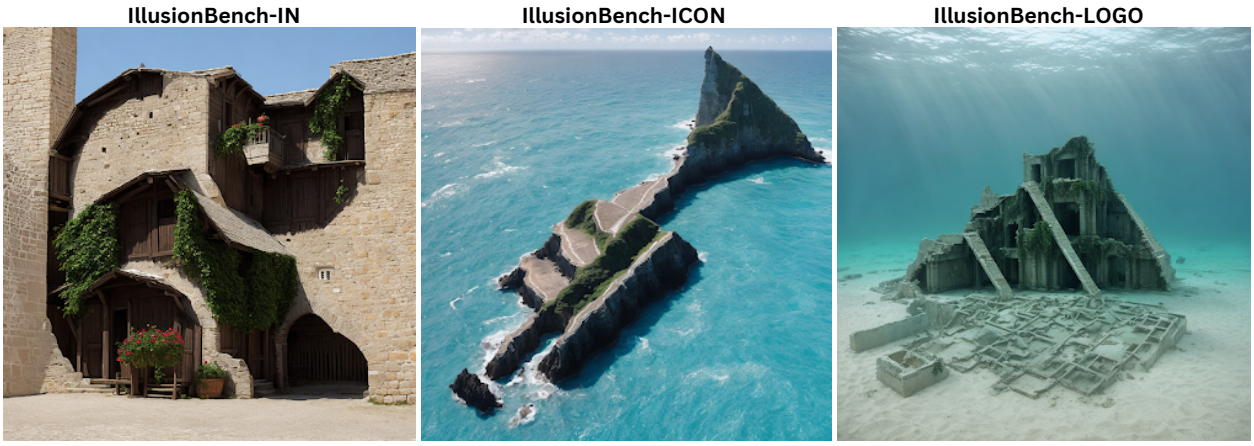


Figure 22: **Can vision-language models recognize these shapes?** `IllusionBench` dataset contains images in which scene elements are arranged to represent abstract shapes.

of robustness (Geirhos et al., 2020b; Gavrikov et al., 2024). (A major goal of our work in Chapter 5 is to address this problem, both in the context of shape vs texture and other analogous settings.) Later work observed that vision encoders trained with larger-scale data weakly supervised by language (e.g., CLIP; Radford et al., 2021) show improvements in shape recognition (Geirhos et al., 2021; Gavrikov et al., 2024).

While clear indicators of progress in visual perception of neural vision models, it is important to note that all of the above studies on shape recognition in vision models have relied on two standard datasets, Cue Conflict and Stylized-ImageNet (Geirhos et al., 2018), which presents several concerns – for instance, these datasets do not include coherent, naturalistic visual scenes; they are built using legacy style transfer techniques that damage shape information and prevent the reproduction of fine-grained textures; and each image includes only a single object class represented as an abstract shape using perceptually uniform textures (see Section 7.2 for a more detailed critique). To address these limitations, we introduce `IllusionBench`,[30] which represents shape information by an arrangement of visual elements existing in coherent, naturalistic scenes (see Figure 22). We evaluate vision-language models (VLMs) using `IllusionBench` in three scenarios: (1) measuring **zero-shot** performance of generative VLMs, including LLava (Liu et al., 2024b), GPT-4o (OpenAI, 2023), and Gemini (Team et al., 2023); (2) measuring **few-shot** performance of VLMs using in-context learning (cf. Zhao et al., 2023); and (3) **fine-tuning** contrastive

[30]We use "Illusion" in the name of our benchmark because images in our dataset can be understood as instances of pareidolia, an illusion caused by the tendency of the human visual system to identify familiar shapes in complex scenes. Our dataset should not be confused with HallusionBench (Guan et al., 2023), which instead serves as a diagnostic tool to distinguish between VLM reasoning error modes, such as those caused by the language component versus visual component of VLMs.

VLMs like CLIP (Radford et al., 2021) to recognize abstract shapes and testing their ability to generalize to unseen scenes. We find that, while human annotators can easily identify these shapes, VLMs struggle to identify shapes and instead focus on the scene components, failing to exhibit the abstract shape recognition capabilities that are essential for enabling humanlike visual robustness.

7.2 Background and Related Work

Shape perception and visual recognition Shape information is widely considered to be the most important cue leveraged by the human visual system for object recognition (Landau et al., 1988; Biederman and Ju, 1988; Xu et al., 2004; Elder and Velisavljević, 2009; Baker and Kellman, 2018). Our ability to perceive shapes is crucial in enabling the robustness of human visual perception (Hummel, 2001; Ommer, 2013), as shape is invariant to key transformations such as translation, rotation, scaling, and changes in illumination, color, and texture (Ommer, 2013; Kendall, 1984; Dryden and Mardia, 2016). Thus, many works have investigated the extent to which neural object classifiers rely on shape for visual recognition tasks, finding that early supervised deep neural networks rely more on texture cues rather than shape (Geirhos et al., 2018; Islam et al., 2021; Benarous et al., 2023; Pinto et al., 2022a; Subramanian et al., 2023). More recently, Gavrikov et al. (2024) showed that multimodal vision-language models can be prompted to rely more on shape in visual recognition. Each of these works evaluates shape perception on the basis of the Cue Conflict (CC) or Stylized-ImageNet (SIN) benchmarks (Geirhos et al., 2018); but despite their longstanding utility, we observe several key limitations with these benchmarks:

1. **Lack of coherent, naturalistic, and complex visual scenes:** Images contain only the shape of a single class mixed with a single texture applied uniformly to the entire image.

2. **Missing shape information:** Key shape information is often lost, yielding "a substantial fraction" of images that are unrecognizable to human annotators (Geirhos et al., 2018). The contrast in textures between the object and the background of any given image is usually lost, yielding perceptually uniform images (Chen et al., 2021; Wang et al., 2023d).

3. **Low-quality style transfer:** The style transfer methods in these datasets (Gatys et al., 2016; Huang and Belongie, 2017) are known to confuse shape and texture information (Wang et al., 2023d) and often fail to capture fine-grained textures (Wang et al., 2021b).

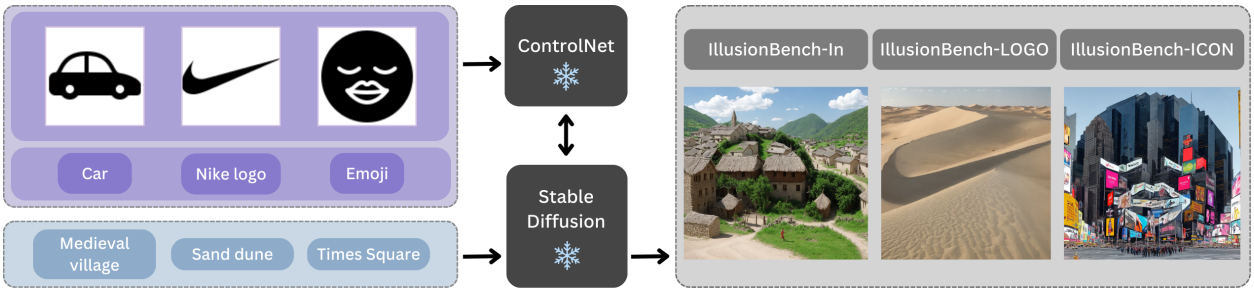


Figure 23: **Dataset generation.** For each of the 3 datasets in `IllusionBench`, we show an example image from the dataset alongside an example scene prompt and an example shape conditioning image used to generate it. A shape image x_i (with the class name c_i) and a scene description s_j are combined to generate the `IllusionBench` image x_{ij}.

To address these limitations, we introduce `IllusionBench`, which leverages state-of-the-art generative models to create images representing shape information with a complex arrangement of elements in detailed visual scenes comprised of various textures and objects.

Evaluating Visual Capabilities of VLMs Vision-language models (VLMs) have exceeded conventional benchmarks, often even exhibiting capabilities that they are not explicitly trained for (Bubeck et al., 2023) and underscoring the need for new forms of evaluation (Zhang et al., 2024a). Traditional image recognition benchmarks are not designed to characterize such capabilities, indicating the need for innovative evaluations. For instance, Bitton-Guetta et al. (2023) studies commonsense visual reasoning by testing whether models perceive peculiar content in visual scenes; Fu et al. (2024a) evaluates VLMs on recognizing the count of objects, relative positions of objects, OCR, and commonsense visual reasoning; and Tong et al. (2024) proposes visual tasks requiring fine-grained understanding of object orientation, perspective, and the states of objects in the image. Finally, Zhou et al. (2023b) focus on limitations specific to generative VLMs, such as visual hallucination.

7.3 Benchmark Description

7.3.1 Generative Process and Notation

Consider the set $\mathcal{C} = \{(x_i, c_i)\}_{i=1}^{|\mathcal{C}|}$ of binary shape conditioning images x_i representing the shapes of corresponding object class c_i, and $\mathcal{T} = \{(s_j)\}_{j=1}^{|\mathcal{T}|}$ is the set of prompts where each s_j describes a different scene (e.g., `Ocean` or `Medieval Village`). To synthesize our dataset, we use ControlNet (Zhang et al., 2023a), a module that is trained to control the generative process of text-to-image diffusion models (such as Stable Diffusion; Rombach et al., 2022) by conditioning on inputs specifying spatial information to guide the generative process, such as

shape conditioning images x_i displayed in the top-left of Figure 23. (Note that we also experiment with ControlNet and Stable Diffusion in Chapter 5 to perform interventional data augmentation, where ControlNet does not offer additional advantages relative to simpler conditioning mechanisms. However, in this work, our goal is to synthesize benchmark data rather than augment training data, and we find that ControlNet is far more effective for this purpose than simpler conditioning mechanisms that do not leverage spatial guidance, meaning they are unable to directly embed shape conditioning images in naturalistic scenes as ControlNet can.) Our pipeline, as visualized in Figure 23, transforms the tuple (x_i, s_j) into an image x_{ij} representing the considered shape x_i of class c_i embedded in a scene of type s_j.[31] We therefore obtain our datasets by creating a tuple (x_{ij}, c_i, s_j) for each combination of conditioning images and prompts. We then consider three predictive tasks a VLM f should perform (where p_C, p_S, and $p_{C,S}$ represent prompts querying for c_i, s_j, or both, respectively):

1. τ_C: predict the shape $c_i = f(x_{ij}, p_C)$.
2. τ_S: predict the scene $s_j = f(x_{ij}, p_S)$.
3. $\tau_{C,S}$, predicting both the shape and the scene $(c_i, s_j) = f(x_{ij}, p_{C,S})$.

7.3.2 Dataset Details

As exemplified in Figure 23, the `IllusionBench` benchmark contains three different constituent datasets: `IllusionBench-IN`, `IllusionBench-LOGO`, and `IllusionBench-ICON`. The number of samples, classes, conditioning images, and domains for each dataset are provided in Table 6 (with more detailed metadata available in Section F.2).

Dataset Name	# Samples	# Classes	# Conditioning Images	# Scenes
`IllusionBench-IN`	6864	16	48	11
`IllusionBench-LOGO`	5577	21	39	11
`IllusionBench-ICON`	20064	6	456	11

Table 6: Size of each dataset in `IllusionBench`.

`IllusionBench-IN` We build upon the 16 classes from the most popular shape perception benchmark, Stylized-ImageNet (SIN; Geirhos et al., 2018). However, since we are interested in how well models can find shapes within a scene, we need clear and distinct shapes that can be identified unambiguously. To address this, we replace 4 of the 16 SIN classes with

[31] The generation is conditioned on additional hyperparameters that allow us to obtain shapes that can be recognized at varying levels of abstraction. See Appendix F.2.2 for further details.

similar categories (near co-hyponyms) with more distinct shapes. We collect 3 conditioning images for each class.

`IllusionBench-LOGO` Another category of shapes that are specifically designed to be visually distinct and easily recognizable are logos, which provide an interesting contrast to the shapes in `IllusionBench-IN`, as recognizing them requires world knowledge specific to the category of product brands (rather than culturally-nonspecific real-world object classes).[32] Thus, we expand our dataset to this domain by collecting 39 different logo conditioning images across 21 brands.

`IllusionBench-ICON` Finally, we develop a third dataset to test whether VLMs can be trained to recognize cross-modal abstractions over perceptually distinct shapes representing semantically related concepts (e.g., where images representing shapes of owls or turtles are both recognized as instances of the "animal" class, despite having very different shapes). We create a coarse-grained dataset of 6 (informal) hypernym categories across 456 emojis as shape conditioning images.

Validating Dataset Quality Although ground truth labels for object classes and scene types are available, image generators may sometimes produce low-quality or high-difficulty images whose object shape is not human-recognizable. To minimize the proportion of such images, we begin by restricting the hyperparameters that control the influence of the conditioning image to ranges that we qualitatively found to produce clearly distinguishable shapes (see Appendix F.2.2). To validate that the shapes in the resulting images are indeed human-recognizable, we recruited 60 participants (information is anonymized) to manually annotate randomly sampled subsets of `IllusionBench-IN`, `IllusionBench-LOGO`, `IllusionBench-ICON`, obtaining an average annotator accuracy of 95.6%, 97.17% and 96.8%, respectively, indicating that humans are indeed able to recognize the shapes in the vast majority of the generated images.[33] (See Section F.2.1 for further details.)

[32]Given that this task requires both world knowledge of product brands *and* abstract shape recognition capabilities, and considering that our goal with `IllusionBench` is only to evaluate the latter, we normalize scores by averaging results for each VLM exclusively on samples obtained from raw shapes that the VLM can recognize in a zero-shot setting, meaning that models are not penalized for lacking world knowledge of specific brands. See Sections F.3.5 and F.4.7 for non-normalized results by class.

[33]Note that human annotator accuracies are only intended to *validate the quality of the generated dataset* and *confirm that the resulting abstract shapes are indeed human-perceptible*. They are *not* intended for direct comparison with VLM performance, as there are a few fundamental differences in how annotators and VLMs are tested. For instance, where VLMs do not know the purpose or structure of the task beyond what is included in the prompt, annotators are shown onboarding materials describing the task, including several pre-annotated examples.

7.3.3 Evaluation

Given image x_{ij}, we prompt VLM f with both x_{ij} and prompts p_k corresponding to the shape, scene, and both the shape and scene (i.e., where p_k is variously p_C, p_S, or $p_{C,S}$, respectively), yielding responses $r_k = f(x_{ij}, p_k)$ for each prompt p_k. For each x_{ij}, we evaluate *shape recall* on the basis of whether the term c_i appears in the response r_C or $r_{C,S}$ (yielding 1 if so, or 0 if not), and evaluate *scene recall* by whether s_j appears in r_T or $r_{C,S}$ (similarly yielding 1 or 0), and report the shape and scene recall for each dataset as the sum of the recall figures across all x_{ij} instances divided by the size of each dataset. In contrast to prior related works (e.g., Geirhos et al., 2018, 2021; Gavrikov et al., 2024), our proposed metrics are designed such that shape recognition performance is not in competition with the ability to recognize other visual elements (e.g., textures or scene elements), as – unlike traditional classifiers, which must select only one among a pre-defined set of discrete classes – generative VLMs can respond with detailed descriptions of images including information about shape, scene, or other visual elements at the same time (or given different prompts).

7.3.4 Experimental Overview

In the following sections, we evaluate the shape perception capabilities of modern VLMs on `IllusionBench` under the following paradigms:

- **Zero-Shot Recognition**: Given that an instruction-tuned VLM can recognize a shape x_i, can it identify the same shape when it emerges from the combination of visual elements in x_{ij} without any explicit examples or specialized fine-tuning? (Section 7.4)
- **Few-Shot Learning**: Given that a multi-modal in-context learner can recognize a shape x_i zero-shot, can it leverage few examples to learn to identify it in x_{ij}? (Section 7.5)
- **Domain Generalization**: Given training samples $\{x_{ij}\}$ representing a shape x_i in certain types of scenes, can models learn to recognize the same shape in other, unseen scene types? (Section 7.6)

7.4 Can Instruction-Tuned VLMs Recognize Shapes Zero-Shot?

Experimental Design In this experiment, we prompt VLMs zero-shot to identify the abstract shape represented in a visual scene among a closed set of object classes. We begin by testing whether models can correctly classify the shape conditioning images (binary shape images), and generate images for `IllusionBench` exclusively using these condition shapes.

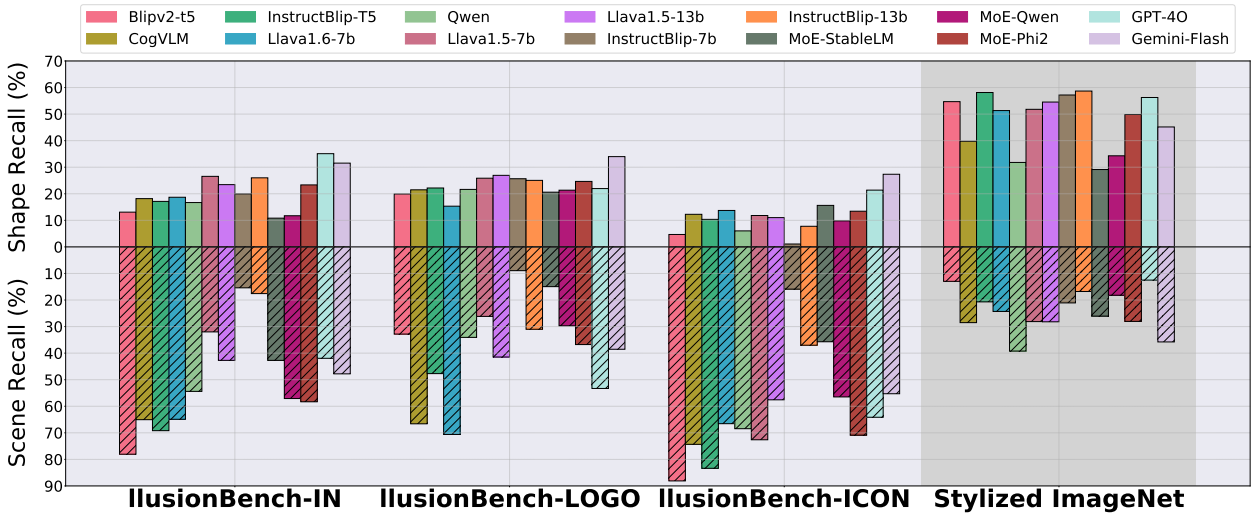


Figure 24: **Zero-Shot Results**. Average shape and scene recall of VLMs across each `IllusionBench` dataset, compared with Stylized-ImageNet (Geirhos et al., 2018) (rightmost, shaded).

We then prompt models with respect to the shape and scene in each generated image, and measure the corresponding recall metrics as described in Section 7.3.3. (See Section F.3 for additional details regarding the experimental design, prompts, and models used in this experiment.)

Models We consider the following VLMs for evaluation: `GPT-4o` (OpenAI, 2023), `Gemini-Flash` (Team et al., 2023), `LLaVA1.5/6-7/13b` (Liu et al., 2024c), `CogVLM` (Wang et al., 2024a), `BLIPv2-t5` (Li et al., 2023b), `InstructBLIP-7/13b` (Dai et al., 2024b), `Qwen-VL-Chat` (Yang et al., 2024a), and `MoE-StableLM/Qwen/Phi2` (Lin et al., 2024).

Results Our main findings in this experiment (visualized in Figure 24) are as follows:

- For each of our datasets, shape recall is quite low, with most models ranging between 10-30% (in contrast to the previous dataset, Stylized-ImageNet (Geirhos et al., 2018), where all fourteen models exceed 30%).
- For nearly all models and datasets, models exhibit superior scene recall relative to shape recall. This indicates that the recognition capacity of current VLMs is still biased towards scene/texture features, similar to earlier work studying CNN classifiers (see Section 7.2).
- `GPT-4o` and `GEMINI` show superior shape recall to all other models in 3/3 and 2/3 of our datasets, respectively, demonstrating a shape-recognition gap between the best

available open- and closed-source VLMs.

- Mixture of Experts (MoE; e.g., `MoE-StableLM`, `MoE-Qwen`, and `MoE-Phi2`), which are generally employed to improve models' performance, exhibit neither superior shape nor scene recall with respect to individual models or closed-source models.

- Among all open source models, `LLava` attains the strongest shape recall performance across all our datasets. In contrast, `Blipv2` attains the highest scene recall (except for the `IllusionBench-LOGO` split).

See Section F.3 for more fine-grained results and analysis.

7.5 Can In-Context Learners Learn to Identify Abstract Shapes?

Given zero-shot prompting exhibits poor performance at detecting abstract shapes and shows VLMs mostly focus on background stimuli, a natural question is whether it is possible to teach models to recognize known shapes with a few samples by leveraging their In-Context Learning (ICL) or few-shot capabilities.[34]

Experimental Design We restrict our experiments to samples generated from conditioning images x_i that models can correctly classify in a zero-shot fashion (see Section F.4.2). Let us focus on the predictive task τ_C (as analogous formulations of ICL apply for τ_S and $\tau_{C,S}$). Given that the model can correctly assign the class c_i to the conditioning image x_i, we provide it with the context sequence $\{(x_{i_w,j_w}, c_{i_w})\}_{w=1}^{|W|}$, where W is the context window plus a test image x_{i_*,j_*}, and prompt the model to predict the object's shape c_{i_*}.

Using specific constraints on context sampling relative to a test sample, we define four learning tasks corresponding to perceptual challenges:

- **ICL1**: Given the context lacks any image depicting the scene or shape type of the test sample $x_{i,j}$, can the model recognize its shape c_i?

- **ICL2**: Given the context includes an image of the shape type but not the scene type of the test sample $x_{i,j}$, can the model recognize its shape c_i?

- **ICL3**: Given the context includes an image of the scene type but not the shape type of the test sample $x_{i,j}$, can the model recognize its shape c_i?

[34] See Section F.4.1 for a brief introduction to ICL.

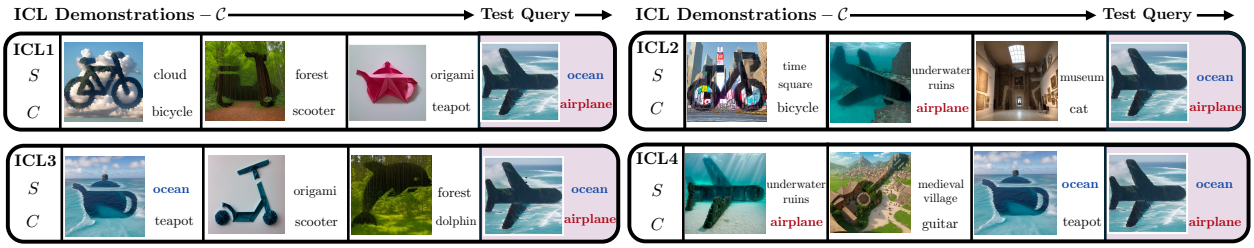


Figure 25: **ICL Learning Tasks**. Figure depicting the four ICL learning tasks, $ICL1, ICL2, ICL3$ and $ICL4$, defined by constraints on demonstration example selection as introduced Section 7.5.

- **ICL4**: Given the context includes images of the scene type and shape type of the test sample $x_{i,j}$ (separately and exactly once), can the model recognize the test sample's shape c_i?

Samples in the context are selected uniformly at random, excluding those that do not satisfy the constraints for a given test sample. Random selection serves as a simple baseline for ICL example selection, avoiding confounding factors like similarity bias or majority (Bertini Baldassini et al., 2024). We perform $0, 1, 2, 4, 8$-shot on `IllusionBench-LOGO` and `IllusionBench-IN`, and $1, 2, 4, 5$-shot on `IllusionBench-ICON`. Further details of ICL experiments can be found in Section F.4.2. We additionally perform ablations to examine the sensitivity of our results to the prompt template used or to the order in which in-context examples are given to the model. These additional results can be found in Section F.4.9.

Models We consider several state-of-the-art models that have been designed to support ICL: (1) `LLaVA-Next` (Liu et al., 2024b), (2) `Qwen-VL-Chat` (Yang et al., 2024a), (3) `Otter-MPT` (Li et al., 2023a), (4) `IDEFICS-9B-Instruct` (Laurençon et al., 2024), and (5) `MMICL-T5-XXL` (Zhao et al., 2023). (In Section F.4.3, we describe each model and the prompts they are provided, and additionally explain our motivation for selecting these specific models.)

Results We summarize the average results across all three dataset splits for $0, 1, 2$ and 4-shot ICL as show in Figure 26 following the recall metrics introduced in Section 7.3.3 (for results for individual datasets and for 5-shot and 8-shot performance on `IllusionBench-ICON` and `IllusionBench-IN`/ `IllusionBench-LOGO` respectively, see Section F.4.7.) [35] We report here the main trends in the data. Discussion of exceptions that do not follow the reported general trends can be found in Section F.4.6.

[35] In Figure 26, we observe that `LLaVA` shows often close to zero recall on either shape of scene prediction. We explore a few possible reasons for these results in Section F.4.8.

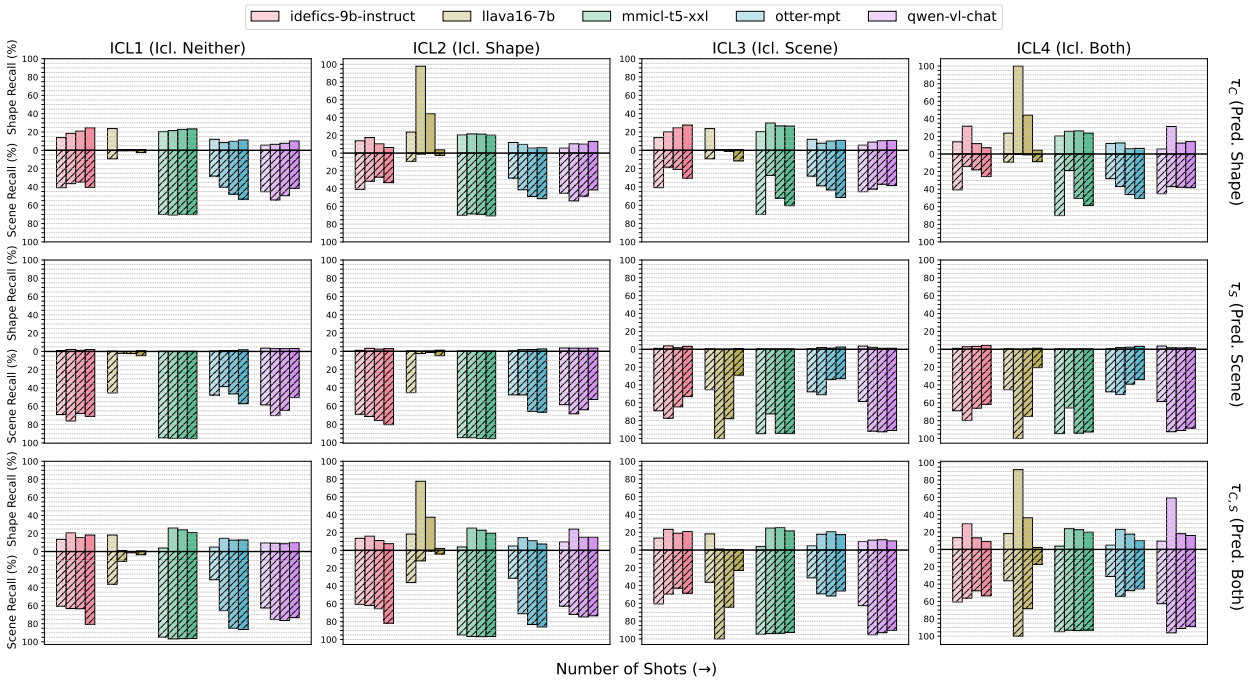


Figure 26: **ICL Results**. Few-shot (0,1,2 and 4-shot) shape and scene recall of VLMs averaged across the `IllusionBench-LOGO`, `IllusionBench-IN` and `IllusionBench-ICON` datasets, displayed for the different ICL learning tasks and the different prediction tasks.

- *ICL does not mitigate tendency to predict scene over shape.* As shown in Figure 26, ICL has minimal effect in altering the models' tendency to predict the scene s_j, regardless of whether the prediction task is τ_C (predict shape), τ_S (predict scene), or $\tau_{C,S}$ (predict both).

- *On average,* `MMICL-t5-XXL` *exhibits the strongest scene and shape recall for the highest number of shots* (i.e., when majority voting biases decay; see (Bertini Baldassini et al., 2024)).

- *Increasing the number of shots has mixed effects on performance.* We observe in Figure 26 that the models often exhibit non-monotonic performance trends for both shape and scene recall across all prediction tasks and demonstration selection constraints. In general, this indicates that the models struggle in general to adapt to tasks τ_C, τ_S, and $\tau_{C,S}$, even with increasing demonstration examples. These results are in line with previous findings that complex ICL tasks remain challenging for current visual language models (VLMs; Zong et al., 2024).

- *Context selection strategy effects prediction tasks differently.*

    - τ_C *(shape prediction):* As shown in the top row of Figure 26 for task τ_C (shape

prediction), including the shape in the context (ICL2 and ICL4) either maintains or reduces performance for most models such as `MMICL` and `IDEFICS`. This suggests that most models struggle to identify and disentangle shape from the scene through ICL.

- τ_S *(scene prediction):* The second row of Figure 26 shows the mixed effect of including the scene within the context (ICL3 and ICL4) compared to not including it (ICL1 and ICL2). Models such as `LLAVA`, `OTTER` show a reduction in scene recall and when including the scene in the context. `MMICL` maintains comparable performance, whereas `LLaVA` and `QWEN` show improved performance.
- $\tau_{C,S}$ *(predicting both shape and scene):* The final row of Figure 26 typically shows trends similar to τ_C and τ_S – e.g., scene recall and shape recall for `MMICL` (whose zero-shot shape recall is lower on this task than in τ_C), `IDEFICS`, and `LLaVA` are comparable with respect to those in $\tau_{C,S}$ and τ_S (respectively).

Overall, we observe that ICL does not substantially aid models in learning to detect abstract shapes within scenes or to help reduce scene prediction bias. The non-uniformity of relative results between models further highlights the immaturity of ICL for multi-modal models, particularly for complex tasks like abstract shape recognition.

7.6 Can VLMs Learn Invariant Representations Across Domains?

A compelling application of `IllusionBench-IN` lies in Domain Generalization (DG) (cf. Gulrajani and Lopez-Paz, 2021). A visual domain is a set of samples with shared characteristics that influence the appearance of objects (e.g., shared style, such as cartoons, paintings, or photos; shared lighting conditions, such as photos taken at similar times of day with similar weather conditions; etc.). In DG, the goal is for models to learn domain-invariant representations – i.e., generalisable features that are predictive of task labels across any domain – by training across multiple "source" domains and testing how well models generalize to unseen test domains. (See Chapter 5 for further motivation, examples, and discussion of DG in the context of computer vision.)

Experimental Design We consider all images generated using the same scene prompt s_j as coming from the same domain $\mathcal{D}^j$. As shown in Figure 27, we partition the `IllusionBench-IN` dataset split into train domains $s_j \in \{$`Cloud, Forest, Ocean, Origami, Sand Dune`$\}$ and test domains $s_j \in \{$`Bazaar Market, City, Medieval Village, Museum, Times Square, Underwater`$\}$. (Conditioning images x_i used to generate the training domains are not contained in the test domains.) We then consider a

contrastive language-vision encoder (CLIP; Radford et al., 2021) and prompt CLIP in order to identify the class c_{i_*} of a test sample x_{i_*} among all possible shape classes[36]. Throughout the experiment, we use "`A photo of {class_name}`" as the prompt template. (See Section F.5.2 for further experimental details.)

Figure 27: **`IllusionBench-IN` for Domain Generalization.** We split the dataset into five source domains for training and six target domains for testing. The condition images for generated data samples are only shared among source and target domains, respectively, without overlapping.

Methods Considered We compare various domain generalization methods including ERM, MixUp (Yan et al., 2020), RegMixUp (Pinto et al., 2022b), GroupDRO (Sagawa et al., 2019), and VREx (Krueger et al., 2021), using both linear probing and full-parameter finetuning. Besides linear probing, we also consider DPLCLIP (Zhang et al., 2023c), a prompt optimization approach specifically designed for CLIP domain generalization.

Results We summarize our findings (reported in Figure 28) as follows:

- *CLIP cannot recognize shapes well in a zero-shot setting.* The CLIP model attains on average extremely low performance in zero-shot settings, with the exception of the `Museum` domain. This can be attributed to the fact that certain samples within this domain do not simply assemble c_i from visual cues of other objects, but incorporate it as a sculpture.

- *CLIP embeddings only partially capture shape information.* Applying prompt learning for domain generalization via DPLCLIP is not particularly effective with an average test accuracy of 13.62%, and ERM results are more effective in improving over the zero-shot performance with accuracy 22.36%, outperforming all other probing

[36]Since the zero-shot performance is particularly low, we do not confuse the model further asking it to distinguish the shape from the background type. This also allows us to make the comparison with probing and fine-tuning techniques that deliberately aim at extracting shape more fairly.

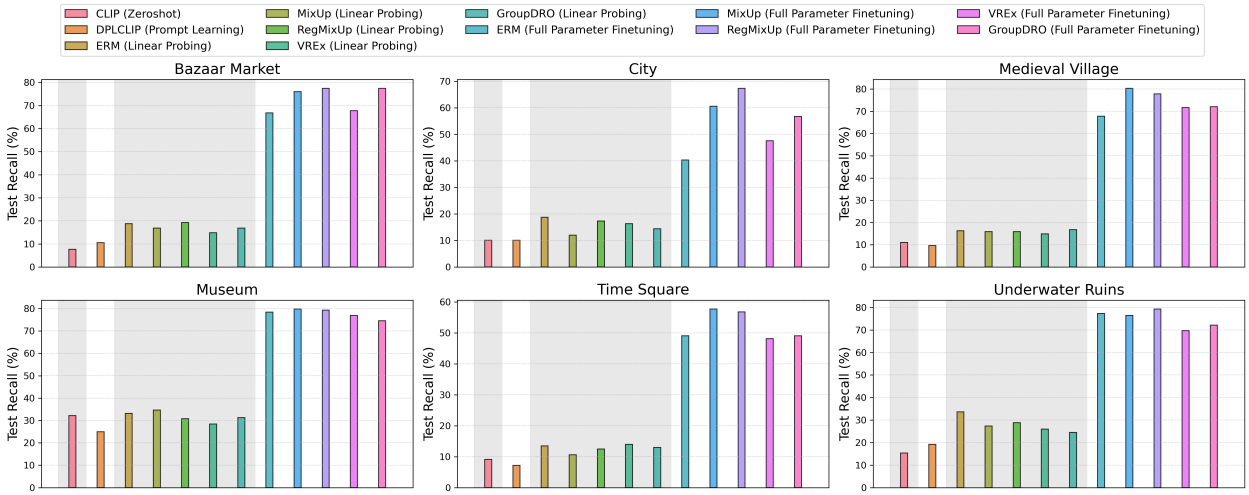


Figure 28: **Domain Generalization.** CLIP performance on `IllusionBench-IN` for different fine-tuning approaches. Each sub-figure represents an unseen test domain. The categories of approaches, with alternating shading from left to right to indicate these different categories, are: zero-shot prediction, prompt learning, linear probing, and full parameter fine-tuning.

techniques. However, the relatively low absolute values of accuracy indicate the embedding space does not render the test samples linearly separable based on shape criteria.

- *CLIP can learn features that allow to distinguish objects based on shape using multiple domains.* In full-parameter fine-tuning, it is possible to learn representations that are more oriented towards shape recognition – in all cases, a very large improvement is observed with respect to linear probing. The best performing methods are Mixup and RegMixup, which attain 71.79% and 73.00% on average accuracy, respectively.

7.7 Conclusion

We present `IllusionBench`, a collection of 3 datasets to evaluate shape recognition in vision-language models (VLMs) by representing abstract shapes as complex arrangements of visual scene elements. While human annotators identify these shapes with high accuracy, we find that state-of-the-art VLMs fail to identify the shapes in these scenes zero-shot, tending to focus on scene elements instead. We observe that in-context learning does not significantly improve models' ability to detect abstract shapes; but we do find that contrastive VLMs such as CLIP can be fine-tuned to recognize these shapes and generalize to new scene domains. In highlighting the limited shape perception abilities of current VLMs, we hope that `IllusionBench` will help guide future research in developing more robust computer vision systems. The contributions of each author are listed in Appendix F.1.

Part III
Mechanistic Interpretability for Transfer Learning

Chapter 8 How Do LLMs Represent and Process Symbols In-Context?

Preface

Where our mechanistic interpretability work in Part I studied the latent features learned and leveraged by foundation models, and our transfer learning work in Part II examined how models generalize out-of-distribution (OOD) by controlling and evaluating their use of causal versus spurious features, our work in **Part III** considers the intersection of both directions – specifically, (how) can we use mechanistic interpretability methods discovering how various latent representations and internal mechanisms are leveraged by models to predict and explain when and why they fail to generalize out-of-distribution?

In **Chapter 8**, which is based on *How Do LLMs Represent and Process Symbols In-Context?* (Davies et al., 2026), we approach this question with respect to a challenging in-context learning symbolic reasoning task, the Templatic Generation Task (TGT; Smolensky et al., 2025), whose generality and complexity allows it to serve as a proxy for a variety of symbol-processing tasks. We experiment with a variety of mechanistic interpretability techniques to study how models trained on TGT learn to perform this task, analyzing how models trained on a given distribution of this task compositionally generalize to increasingly difficult distributions. We find that the symbol-processing mechanisms models learn to perform the task in-distribution are programmatically valid, but that these mechanisms are brittle with respect to compositional distribution shifts (see Section 5.0.1), indicating that the failure of such models to generalize compositionally is not primarily due to the lack of programmatic internal mechanisms but rather the robustness of these mechanisms to more complex inputs. These findings carry important potential implications for predicting and improving OOD generalization:

- Toward *OOD prediction* (i.e., predicting whether a given input is OOD, or whether a given model is likely to generate an appropriate output given some OOD input): our results indicate that the mere *presence* of a seemingly-robust internal mechanism (e.g., feature, circuit, etc.) under analysis does not, on its own, indicate that models are likely to generalize well. Rather, the mechanism must also properly respond to OOD inputs, meaning that examining model internals only via in-distribution samples will be insufficient for predicting OOD performance.

- Toward *improving OOD generalization:* our findings indicate that the primary issues in generalizing OOD with respect to the studied symbolic reasoning task stem from brittle mechanisms for parsing and information localization across layers (see Section 8.5). This suggests that approaches for improving information localization,

such as architectural work developing more precise query-key matching in Transformer self-attention (see, e.g., Veličković et al., 2024; Ye et al., 2025; Opper et al., 2025), may be effective for resolving the internal brittleness of these mechanisms and thereby improving OOD generalization of models. (See Section 9.2.3 for further discussion.)

Chapter 8: *How Do LLMs Represent and Process Symbols In-Context?*

Preprint: Adam Davies, Mattia Opper, Roland Fernandez, Paul Smolensky, and Jianfeng Gao (2026). *How Do LLMs Represent and Process Symbols In-Context?*

8.1 Introduction

Symbolic reasoning, where symbols are systematically composed and operated over, forms the bedrock of most classical theories of cognition. Whether connectionist systems like neural networks are capable of genuine symbolic reasoning has long been a subject of significant contention (Fodor and Pylyshyn, 1988; Smolensky, 1988; Alhama and Zuidema, 2019; McCurdy et al., 2024; Russin et al., 2024). On one hand, theoretical results have shown modified Transformer variants to be Turing-universal (Pérez et al., 2021; Smolensky et al., 2025), meaning they are in principle capable of performing arbitrary symbol-processing tasks perfectly – though the question of whether such models are actually learnable in practice remains to be determined. Empirically, LLMs have also demonstrated impressive symbol-processing capabilities in settings such as in-context learning (Brown et al., 2020), leading to suggestions that they are displaying the hallmarks of true intelligence (Bubeck et al., 2023). On the other hand, there is the argument that, rather than understanding, large language models are merely regurgitating statistical patterns observed in their astronomically large pre-training corpora without genuine understanding (Carlini et al., 2020; Bender et al., 2021; Shojaee et al., 2025).

Our goal in this work is to address the question of language models and symbolic reasoning by studying models simultaneously in terms of observed behavior, latent representations, and internal algorithms – i.e., across Marr's levels of analysis, as described in Chapter 2. We analyze Transformer language models trained from scratch on the Templatic Generation Task (TGT; Smolensky et al., 2025), a semantics-free symbolic reasoning task testing compositional generalization; and leverage a variety of mechanistic interpretability methods to comprehensively interpret model generalization on this task across Marr's levels, finding the following:

- At the *functional* level, models correctly learn the task input/output mapping in-distribution, but fail to compositionally generalize out-of-distribution (Section 8.2).

- At the *algorithmic* level, models learn a second-order analogical reasoning process: in early layers, they infer first-order analogies between parses of in-context questions and answers; and in later layers, they map these analogies onto the final question to infer its corresponding answer (Section 8.3).
- At the *representational* level, models represent symbolic parses via positional relations among constituents (Section 8.4).

8.2 Functional Level

8.2.1 Templatic Generation Task (TGT)

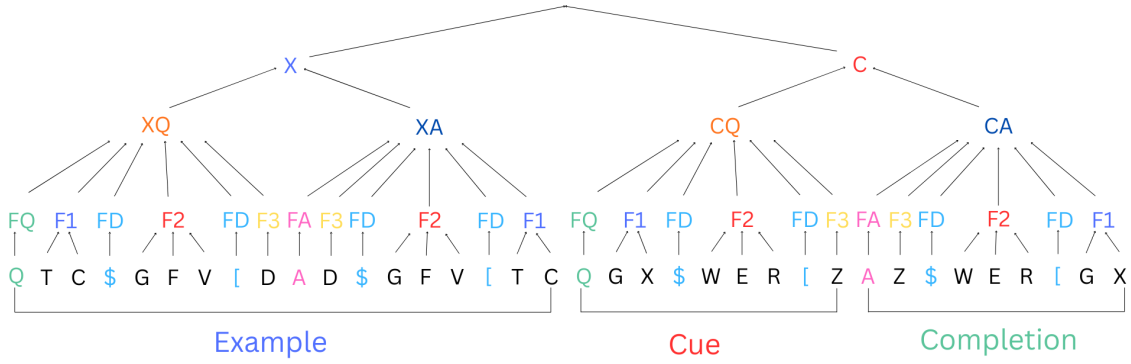


Figure 29: **Example TGT Input.** The model is provided the symbol sequence as input. It must then first parse the in-context example(s) to infer the correct *template* – i.e., the abstract, symbolic mapping from the question region XQ to the answer region XA; in this case, the template is `Q F1 $ F2 [ F3 A F3 $ F2 [ F1` – then apply this template to the cue in order to generate the correct completion . Different colors at each level of the tree mark different roles occupied by symbols (none of which are provided as input, and must be inferred as latent features).

To examine how abstract symbol-processing capabilities manifest empirically in trained language models, we turn to the Templatic Generation Task (TGT; Smolensky et al., 2025), which provides a clean synthetic test-bed for complex abstract reasoning. TGT is a symbolic reasoning task where the model is presented with a series of in-context examples, where each example depicts a particular programmatic template from question to answer showcasing the operation to be performed, followed by a final cue which the model must manipulate according to the procedure in the prior templates – similar to the ARC challenge (Chollet, 2019) or Raven's progressive matrices (Raven, 2000), but presented in a format more accessible to decoder-only LLMs (i.e., as a textual in-context learning task). This structure requires the model to first *parse* the example to recover the template, then *map* it onto the

cue, and finally *generate* the corresponding completion. An example visualization is presented in Figure 29. (Note that, for our experiments, constituent tokens are English words; see Appendix G.1 for examples.)

The model must be able to infer the underlying rules and apply them to a new input. This requires composing multiple simple sub-programs such as *stick-breaking* (Csordás et al., 2021a; Liu et al., 2023a), *associative recall* (Arora et al., 2024b; Zhou et al., 2024a), and *abstract variable binding* (Davies et al., 2023; Wu et al., 2025a). To ensure a clean test of these capabilities, the task is *semantics free*, meaning that the distributional statistics of the training data are useless for solving the underlying task, and the identity of all symbols other than syntactic markers is irrelevant to the task. Crucially, this property means that models trained on TGT cannot solve the problem by merely acting as, e.g., "stochastic parrots" (Bender et al., 2021), making TGT an ideal candidate for studying the underlying syntactic mechanisms that language models learn in order to perform in-context learning.

8.2.2 Empirical Setting

Dataset Splits In addition to their performance on the i.i.d. splits (train, dev, and test), we also evaluate models based on their performance across three out-of-distribution (OOD) splits: `ood_cons_len`, `ood_cons_count`, and `ood_lexical`. The goal of the `ood_cons_len` and `ood_cons_count` splits is to test models' length generalization, and the goal of `ood_lexical` is (unsurprisingly) to test lexical generalization (loosely analogous to productivity and systematicity, respectively; Vegner et al., 2025). Task instances in the i.i.d. splits each have 1, 2, or 4 constituents, of length 1, 2, or 4 tokens; whereas each instance in `ood_cons_count` has 5 constituents, and `ood_cons_len` has constituents with 5 tokens each. Finally, `ood_lexical` contains instances where all tokens are uppercase (whereas in i.i.d. splits, all tokens are lowercase): models *do* see these tokens at train-time, but only in the context of "echo prompts", which follow the form of, e.g., `Q BC A BC` – i.e., they have only a single constituent in the Q-region and A-region that is simply repeated, meaning models have seen these tokens at train-time but have never had to engage in abstract symbolic processing with respect to these tokens. (See Appendix G.1 for examples from each split.)

Experimental Setting: Model Architecture and Training We train three decoder-only, autoregressive Transformer language models on TGT: two smaller models – one with the NanoGPT architecture (Radford et al., 2019) and the other with the Llama architecture (Touvron et al., 2023a) – and a third, larger model with the Llama architecture. The smaller models each have 4 layers (Transformer decoder blocks), each with 4 attention heads and a residual stream of $d = 512$ dimensions; and the larger Llama model has 8 layers,

each with 8 attention heads and a residual stream of $d = 1024$ dimensions. (Further details are provided in Appendix G.2.)

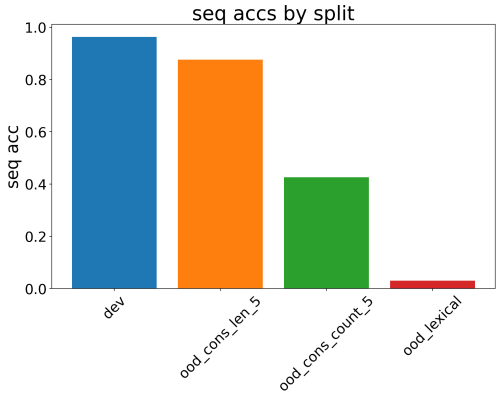


Figure 30: **Full-sequence Accuracy of Llama medium by Split.** The dev split is i.i.d. with respect to the training data, in which each example has 1, 2, or 4 constituents (fields) with 1, 2, or 4 tokens (at train time); `ood_cons_len_5` evaluates how models generalize at test time to constituents of length 5; `ood_cons_count_5` evaluates generalization to 5 constituents; and `ood_lexical` evaluates generalization to tokens seen in new syntactic positions (fields).

Explananda: Empirical Behaviors All models achieve perfect accuracy on the train set, $> 90\%$ accuracy on the dev/test set, and have the following performance ordering over the OOD splits: acc(`ood_cons_len`) $>$ acc(`ood_cons_count`) $>$ acc(`ood_lexical`) (see, e.g., Figure 30). Thus, candidate mechanistic interpretations for any of these models should explain each of the following observations:

1. (Near-)perfect generalization to unseen i.i.d. samples.
2. Poor generalization to OOD splits.
3. Performance ordering over OOD splits.

8.3 Algorithmic Level

8.3.1 Hypothesis

Analogical Reasoning What kinds of algorithms might language models employ to solve TGT? Given that it is fundamentally a task of mapping various symbol-structures onto one another (questions to answers, and (question, answer) pairs to the cue; see Section 8.2.1), a natural hypothesis would be *structure mapping* (Gentner, 1983), which has long been a leading model of human analogical cognition. Put simply, it proposes that we understand

and reason about analogies by constructing systems of relations over entities in different domains and mapping knowledge from entities that occupy similar roles in their respective domains (generally from a *source* to a *target* domain). For instance, given the analogy "the Rutherford model of the atom is like the solar system", we might construct two systems of relations (one per domain):

- **Solar system:** for domain S, define entities $E_S = \{s, p_1, ..., p_k\}$, where s is the Sun and p_i is planet i. For any given planet p_i, we may define relations such as: $\{\text{attracts}(s, p_i), \text{attracts}(p_i, s), \text{orbits}(p_i, s), \text{greater\_than}(\text{mass}(s), \text{mass}(p_i)), ...\} \subset R_S$.
- **Rutherford atom:** for domain R, define entities $E_R = \{n, e_1, ..., e_k\}$, where n is the nucleus and e_i is electron i. For any given electron e_i, we may define relations such as: $\{\text{attracts}(n, e_i), \text{attracts}(e_i, n), \text{orbits}(e_i, n), \text{greater\_than}(\text{mass}(n), \text{mass}(e_i)), ...\} \subset R_R$.

Such structural mappings allow knowledge to be transferred from a source to a target domain – e.g., if one hears that "because the Sun is much more massive than Earth, the Sun-Earth system's center of mass is much closer to the Sun than the Earth" – or symbolically, for inference I, earth as third planet from the Sun p_3, and Sun-Earth center of mass c_{s,p_3}, define $I(s, p_3, c_{s,p_3})$:

$$\text{greater\_than}(\text{mass}(s), \text{mass}(p_3)) \implies \text{less\_than}(\text{distance}(s, c_{s,p_3}), \text{distance}(p_3, c_{s,p_3}))$$

and by analogy, one can derive the same inference $I(n, e_i, c_{n,e_i})$ for the nucleus and an electron in the Rutherford atom, because all of the same relations hold.

8.3.2 Experiment: Activation Patching

To find structural maps between tokens or regions, we must examine how information about these respective units flows through models. We design an *activation patching* experiment (Vig et al., 2020b; Meng et al., 2022b; see Section 2.5.1), using token embedding-level interchange interventions (Geiger et al., 2020) in the residual stream to patch representations from a counterfactual input to a base input (where the counterfactual is a modified version of the original base input), allowing us to examine how the model's behavior changes in response. If the model's output given the base input and the patch (from the counterfactual) changes to match its output given the counterfactual input, then we have localized the relevant information the model has used to make a given prediction.

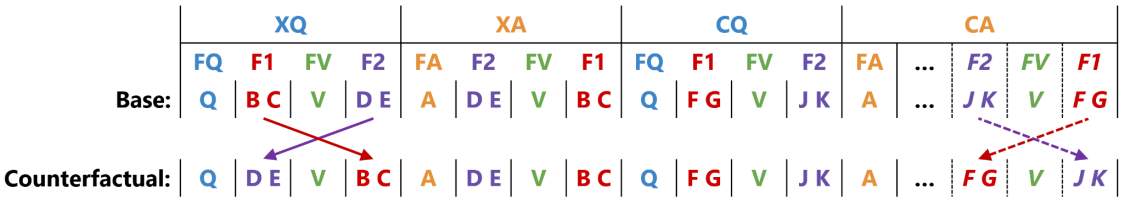


Figure 31: **Counterfactual Generation.** Constituents in XQ are reordered while those in XA and CQ (as well as constituents in XQ) are held constant, leading to a different template mapping from XQ to XA (and thus, from CQ to CA as well). In this case, the base template is `Q F1 V F2 A F2 V F1` (meaning that the mapping from XQ to XA is a matter of swapping the constituents around the delimiter `V`), and the correct completion for cue `Q FG V JK` is `A JK V FG`; whereas for the counterfactual input, the template is now `Q F2 V F1 A F2 V F1` (meaning that the mapping from constituent order in the counterfactual XQ' to that of XA is now simply the identity), and the correct completion for the same cue is now `A FG V JK`.

Valid counterfactual prompts C for this experiment (originating from base inputs B) must meet a few criteria:

- C must be a valid TGT instance.
- The ground-truth output sequence (CA region) corresponding to C must differ from that of B, as the experiment relies on changes in prediction to determine whether we have patched a representation which is causal in determining the model's response.
- All constituents must be preserved by the counterfactual, because we are specifically interested in how the model process structural information (e.g., spatial relations) between them – so reordering the symbols in constituents or replacing them with other symbols would lead to different constituents, meaning any resulting change in behavior might be due to changes in symbol identity or order (in the constituent) rather than the structural relations between constituents.

We design a simple process to generate these counterfactuals: given base input B, we randomly reorder the constituents in each region XQ while holding all other regions (XA and CQ), and other tokens (e.g., delimiters) in XQ, constant. This leads to a change in the XQ region of the template – see Figure 31 for an example and detailed explanation.

Finally, to perform the interchange intervention, we patch all embeddings of tokens with matching symbols in a given region and layer from the counterfactual to the base input – see Figure 32.

Results In the main paper, we visualize only the results of Llama medium on the dev split and `ood_cons_count` split (Figures 33 and 34, respectively) for space; but corresponding plots for all other models and splits are available in Appendix G.4.1.

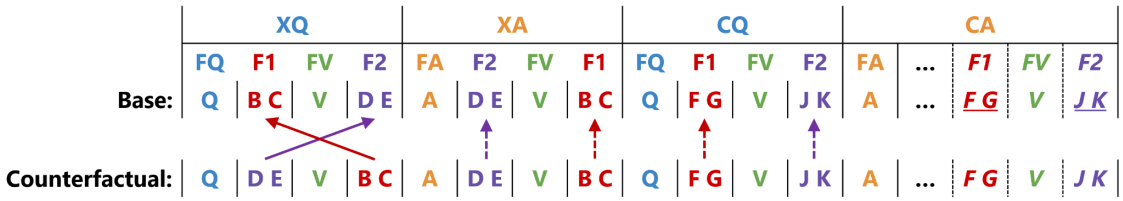


Figure 32: **Activation Patching.** Embeddings of all constituent tokens with matching symbols in a given region and layer are patched from the counterfactual representation to the base representation in the forward pass of the base input. E.g., when patching only region XQ (which includes all tokens that were moved in generating the counterfactual from the base), we patch the embedding representation of token `D` (in token index 2 of the counterfactual) to `D` (in token index 5 of the base), `E` to `E` (index 3 to 6), `B` to `B` (index 5 to 2), and `C` to `C` (index 6 to 3). We might instead patch region XA or CQ (in which case the counterfactual and base token indices would match), or patch over multiple regions at once. In each case, we patch only token indices corresponding to constituent symbols – not delimiters or special tokens (such as `<BOS>`, `<EOS>`, etc.).

8.3.3 Analysis

In-Distribution Results The patching results for Llama medium in Figure 33 (as well as those in Appendix G.4.1 for other models on the dev split) show that the causal representation (i.e., information encoded by the model that causes it to generate either the base response, the counterfactual response, or neither, where a higher "to target" proportion indicates that the causal representation was captured by the patch) moves from being localized largely to the XQ region in early layers ($l \in [2, 4]$ peaking at $l = 3$), to the XA region in intermediate layers ($l \in [4, 6]$, peaking at $l = 3$), to the CQ region in later layers ($l \in [6, 7]$, peaking at $l = 7$), with similar results for both smaller models (see Appendix G.4.1).

These results are consistent with the model implementing a second-order structural map:

- In early layers, the **first-order map** is inferred: models first parse positional relations among constituents in source domain XQ, then map this parse onto target domain XA.

  For instance, in layer 3 of Llama medium (see Figure 33):

  - *Observation:* Patching region XQ induces the counterfactual response in more than 60% of cases, and changes from the base response in more than 85% of cases. Both figures are much higher here than in any other layer; and patching other regions in this layer (not including XQ) induces no more than 5%/15% impact (to target/neither), respectively.
  - *Interpretation:* This means that the causal representation in layer 3 is primarily

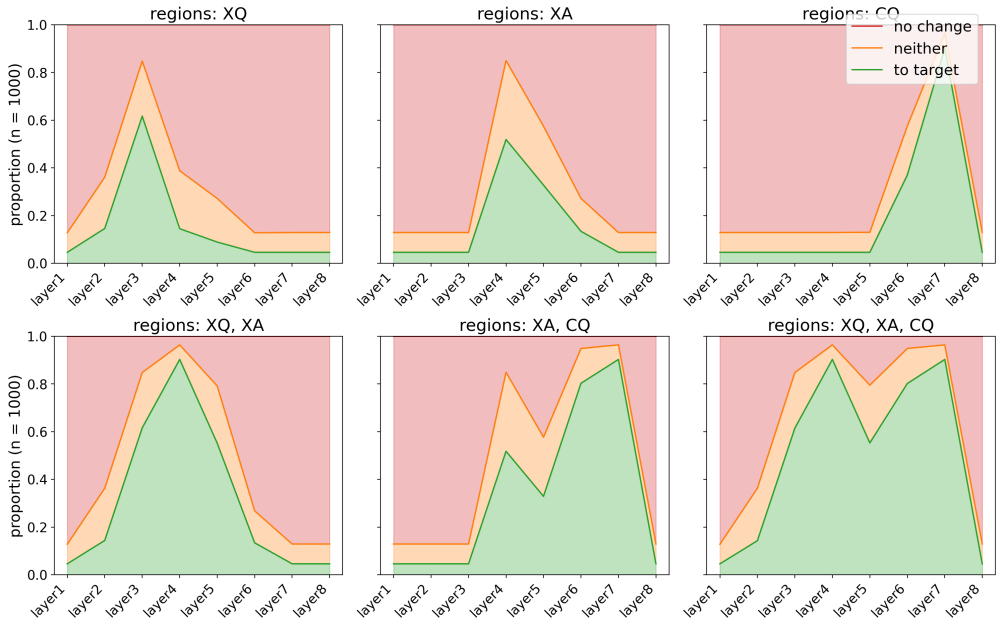


Figure 33: **Patching Llama medium on the dev split.** Patching experiments by region, where the "to target" proportion (on the y-axis, shown in green) in each layer (x-axis) is the fraction of instances where patching token embeddings (corresponding to constituent symbols in this region) from the counterfactual to base representation in the target layer leads the model to generate the same response for the base (under the patch) as it did for the counterfactual – note that this proportion is equivalent to interchange intervention accuracy (Geiger et al., 2022); "no change" is the proportion of instances where the model's response under the same patch does not change from its original response given the base input; and "neither" is the proportion of instances where the same patch led to a change in response from the original base response, but not to the counterfactual response. Finally, note that $n = 1000$ refers to the number of instances from this split that were used to test the model (using the same sample for each layer/region combination).

localized to the XQ region – i.e., the only region of the input where the base differs from the counterfactual. We interpret the model's reliance on XQ in this layer (and lack thereof in later layers) as a matter of parsing XQ, in preparation to map it onto XA in subsequent layers. (This interpretation is further supported by representation-level results in Section 8.4.)

And for layer 4 of Llama medium:

- *Observation:* Patching region XA induces the counterfactual response in more than 50% of cases, and changes from the base response in more than 80% of cases. Both figures are much higher here than in any other layer; and patching other regions in this layer (not including XQ) induces no more than 15%/40% impact (to target/neither), respectively (and no more than 5%/15% for CQ only).

- *Interpretation:* The causal representation has now been largely transported from the XQ region (in the previous layer) to the XA region in layer 4. Given that (a) patching XQ tokens had a large impact in the previous layer but not this one, (b) patching XA tokens had minimal impact in layers 1-3, and (c) the input tokens in this region are unchanged between base and counterfactual, we interpret the movement from XQ to XA as the model mapping its parse of XQ onto XA – inferring, in effect, the analogy XQ:XA.

- In later layers, the **higher-order map** is inferred: again, models map relations from the source domain to the target domain; but in this case, the source domain consists of XQ:XA mappings (i.e., the lower-order analogies), and the target domain is the mapping CQ:CA (i.e., from the cue to the generated continuation).

  For instance, in layer 7 of Llama medium:

  - *Observation:* Patching region CQ induces the counterfactual response in more than 85% of cases, and changes from the base response in more than 95% of cases; and as in the previous cases, patching other regions in this layer (not including XQ) induces no more than 5%/15% impact (to target/neither), respectively.
  - *Interpretation:* The causal representation has now been transported from the XA region (in layers 4-5) to CQ (in layers 6-7, peaking here in 7). As in the case of XA, the appearance of this sudden transition (where input tokens in this region are unchanged between the base and counterfactual) suggests that the model has mapped the analogy XQ:XA onto CQ.

OOD Results The interpretation from the in-distribution results holds to variable degree across the OOD splits; and the extent to which it holds for a given split is predictive of each model's performance on that split. The results for Llama medium on `ood_cons_count` are visualized in Figure 34; see Appendix G.4.1 for figures corresponding to all other models/splits.

- *Observation:* The same general trends by layer and region are present for the OOD splits as they are in the dev split, but we see a degradation of structure by layer/region corresponding to the difficulty of the split. That is, the worse a model performs on the split, the more its responses trend toward the "neither" result, across all layers in regions. For instance, on `ood_cons_count`, Llama medium has the same "peak layers" by region as in the dev set – both in terms of counterfactual responses and total responses changed overall (i.e., "to target" plus "neither") – but the proportion of

Figure 34: **Patching Llama medium on the `ood_cons_count` split.** See the caption of Figure 33 for a description of these visualizations; the only difference in these plots is the change from the dev split (in Figure 33) to the `ood_cons_count` split (here).

counterfactual responses is reduced by 40% in each peak layer, whereas the proportion of "neither" instances now never drops below 60% (for all regions and layers, versus a baseline of $\sim$ 15% on the dev set).

- *Interpretation:* The degradation across regions suggests that models are struggling with *input segmentation* and *representation localization*: as the patching differences between regions and layers erode, causal representations are spread across a wider (less localized) range of token embeddings. Even patching across all regions at once fails to induce the counterfactual behavior at more than a fraction of the rate observed i.i.d.– instead, all patches increasingly lead toward "neither" responses, indicating that failures in localization and segmentation further inhibit models from consistently tracking variable bindings.

8.4 Representational Level

Our results in the previous section indicated that models transport causal representations between regions in order to perform a second-order structural mapping algorithm. While this interpretation is meaningful at the algorithmic level, it does not explain what is happening at the representational level. That is, what information is actually being represented and transported between regions?

8.4.1 Hypothesis

An interesting possibility is raised by the linear representation hypothesis. The *linear representation hypothesis* (LRH) states that models encode distinct variables in low-dimensional linear subspaces of the residual stream (Bolukbasi et al., 2016; Alain and Bengio, 2017; Vargas and Cotterell, 2020; see Section 2.3.1), and a variety of works have provided evidence that variable bindings are encoded in such subspaces (Feng and Steinhardt, 2024; Dai et al., 2024a; Wu et al., 2025a). Most relevant to our case, Smolensky et al. (2025) develop an approach to compile directly from a symbolic program to model weights of a so-called "Discrete Attention-only Transformer" (DAT), which is able to perfectly solve TGT. Even though our trained models do not solve the task perfectly, their performance in-distribution (and limited OOD generalization depending on the split) suggests that they might exhibit some similarities to DAT in terms of their internal representation, making linear subspaces a natural place to begin. In DAT, positional relations between constituents are captured by *constituent type* – e.g., `F1, F2, F3` in Figure 29 – and constituent type is encoded in DAT via linear subspaces.

Thus, we begin from the linear representation hypothesis with respect to constituent type. That is, for each constituent type, there exists some subspace of the residual stream such that the projection onto that subspace is very close for all symbols within constituents of that type. Formally, let $\mathcal{V} = \mathbb{R}^d$ be the residual stream at layer l, with vectors $h_1, \ldots, h_n \in \mathcal{V}$ as the embedding vectors extracted from layer l given input prompt $x = (x_1, \ldots, x_n)$. For any subspace $U \leq \mathcal{V}$, denote the orthogonal projection onto U as P_U. Now define constituent $c = (i_1, \ldots, i_k) \subset [1, n]$ as the list of sequential columns (token indices) $i \in [1, n]$, where $\tau(c) \in T$ is the constituent type. Then we may state the hypothesis as follows:

$$\exists U_T \leq \mathcal{V} \quad \text{such that} \quad \forall c, d : \tau(c) = \tau(d), \forall i \in c, j \in d : P_{U_T}(h_i) \approx P_{U_T}(h_j) \tag{11}$$

8.4.2 Experiments/Results

We use LDA (linear discriminant analysis), a supervised dimensionality-reduction technique, to discover low-dimensional linear subspaces that maximize the signal-to-noise ratio (i.e., the ratio of between-class to within-class variance) between vectors of different classes (McLachlan, 2005). This method can be understood as a form of linear probing (see Section 2.4.2) analogous to, e.g., INLP or RLACE (as studied in Chapter 4), except that the primary goal is to find a linear subspace of embeddings that best partitions them by class, rather than finding a subspace that captures all class-relevant information (in order to remove this information by projecting into the corresponding nullspace). We adopt this

method in the place of other linear probing alternatives because it is more directly optimized to find a subspace that meets the criterion in Equation (11); but outside the context of this specific hypothesis, other probing methods would be equally applicable.

We train LDA subspaces on i.i.d. samples of the dev set (which do not overlap with the dev samples we visualize or use in computing stats for the experiments in this section, etc.) where, for each constituent token embedding in each layer, LDA is trained using all of the constituent tokens from that layer in order to optimize the ratio of between-group variance (between embeddings of different constituent types) to within-group variance (between embeddings of the same type) – see Appendix G.3 for further details. We evaluate the veracity of learned subspaces U_T with respect to the hypothesis in Equation (11) based on within-group variance, between-group variance, partition accuracy, and average centroid cosine, whose definitions are provided in Appendix G.3.

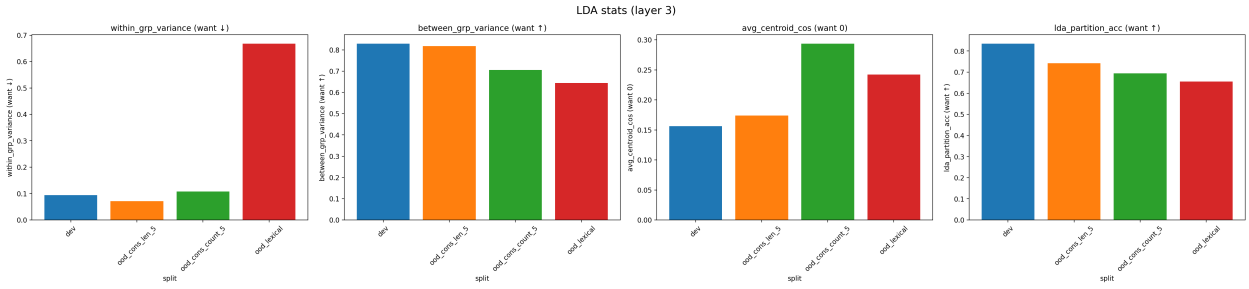


Figure 35: **Quality metrics regarding LDA field-index subspace, by split.** The four plots, from left to right, are *within-group variance* (↓), *between-group variance* (↑), *average centroid cosine* (goal: 0), and *LDA partition accuracy* (↑). The four splits, from left to right, are `dev`, `ood_cons_len`, `ood_cons_count`, and `ood_lexical`– these are also listed in order of sequence accuracy, from best to worst. Note that all quality metrics except centroid cosine are monotonic with respect to split difficulty – the harder the split, the worse the LDA field-index subspace (learned from train-set embeddings) fares in separating classes.

Results for Llama medium in the layer we find to be most associated with parsing (i.e., layer 3; see Section 8.3.3) are shown in Figure 35, with UMAP projections of embeddings that are first projected onto the learned constituent-type subspaces in Figure 36; and these results are reported for all remaining models and layers in Appendix G.4.2.

8.4.3 Interpretation/Analysis

Constituent-type Subspace Representation The results for Llama medium in Figure 35 (and in Appendix G.4.2 for all other models) show that there is indeed a constituent-type subspace satisfying the description in Equation (11). Moreover, the invariance we observe in this subspace (measured as a function of the subspace statistics

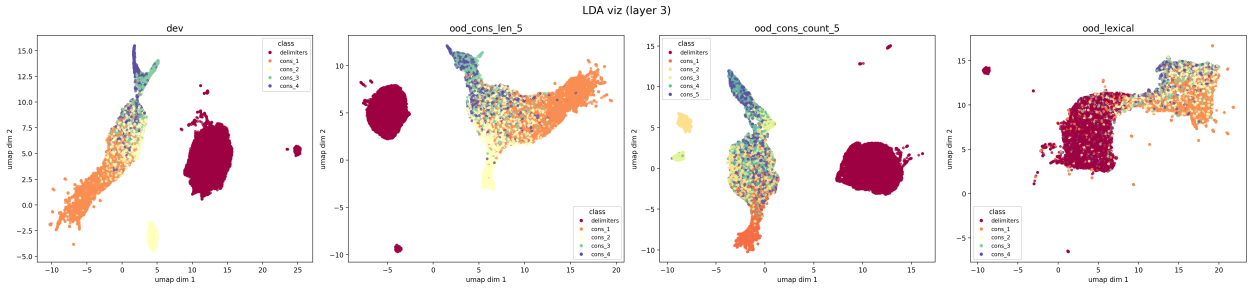


Figure 36: **UMAP projection of LDA field-index subspace, by split.** Colors are field indices (with orange, yellow, green, and purple as fields 1-4, respectively; and red denotes delimiters). The four splits, from left to right, are `dev`, `ood_cons_len`, `ood_cons_count`, and `ood_lexical`– these are also listed in order of sequence accuracy, from best to worst. Note that clusters for different fields become increasingly entangled as splits become more difficult.

computed over embeddings in each split) is strongly correlated with performance across splits – e.g., simple linear regression over the "between-group variance" and accuracy by split predicts model generalization within 3.7% in layer 3 of Llama medium (see appendix Figure 95), which is also the layer our earlier algorithmic analysis indicated is most responsible for parsing (see Section 8.3.3).

8.5 Discussion

Below, we discuss our findings across Marr's levels of analysis, following the framework provided above in Chapter 2:

Functional Level At the functional (behavioral) level, models correctly learn the task input/output mapping in-distribution, but fail to various extents in compositionally generalizing out-of-distribution. Notably, all models show the same performance ordering over the OOD splits: acc(`ood_cons_len`) > acc(`ood_cons_count`) > acc(`ood_lexical`) (see Figure 30 for the results of Llama medium). Thus, candidate mechanistic interpretations at lower levels must explain the near-perfect i.i.d. generalization, poor generalization to OOD splits, and specific performance ordering over OOD splits across models.

Algorithmic Level At the algorithmic level, we perform experiments with activation patching (Vig et al., 2020b) to study how task-relevant information is transported between regions across layers (see Figures 31 and 32 for an overview and visualization of the experimental approach). We find that models appear to learn a second-order analogical reasoning process, expressible as a *structure mapping* algorithm (Gentner, 1983), as visualized in Figure 33. In early layers, a **first-order map** is inferred: models first parse

positional relations among constituents in the XQ, then map this parse onto the XA region. In later layers, the **higher-order map** is inferred: again, models map relations from the source domain to the target domain; but in this case, the source domain consists of XQ:XA mappings (i.e., the lower-order analogies), and the target domain is the mapping CQ:CA (i.e., from the cue to the generated continuation). Furthermore, we observe the same general trends by layer and region among the OOD splits as we do in the dev split, but OOD splits show a clear degradation of structure by layer/region corresponding to the difficulty of the split (see Figure 34), suggesting that models' failures on these splits are linked to worsening information localization by region.

Representational Level At the representational level, we find that models represent symbolic parses via positional relations among constituents. We develop a supervised dimensionality-reduction technique (based on linear discriminant analysis, or LDA) to learn low-dimensional subspaces that maximize the ratio of between-class to within-class variation, yielding a projection to this subspace where token embeddings occurring in the same parse field cluster in the same region of the learned subspace, away from embeddings from other fields.[37] When learning these subspaces from the (i.i.d.) dev set and evaluating them on the (i.i.d.) train set, we find that the dev-set subspace cleanly separates test-set embeddings by field; but when applying the same projection to embeddings from the OOD splits, we see less and less clustering behavior by the difficulty of the corresponding split (see Figures 35 and 36). These findings indicate that the failures in information localization by OOD split observed at the algorithmic level are further linked to challenges in parsing OOD splits: the worse the parse (as visible in the subspace statistics across splits in Figure 35), the worse the information localization, and the greater the corresponding performance drop.

General Summary Taken as a whole, these findings suggest that the flawed compositional generalization of models trained on TGT are due to brittle internal mechanisms for parsing (from the representational analysis) and information localization (from the algorithmic analysis), rather than a lack of such mechanisms altogether (given the clear, strong results at each level for i.i.d. dev/test sets). We see several concrete directions for future work to build on these findings and the analysis techniques introduced in this work, and outline them in Section 9.2.

[37] Note that it was necessary to develop this approach because more standard, unsupervised dimensionality-reduction techniques used in representation analysis, such as PCA and UMAP, did not yield subspaces with parse fields naturally separated.

8.6 Related Work

Mechanistic Interpretation of Symbolic Reasoning There has been substantial prior mechanistic interpretability work on symbolic reasoning in language models, covering atomic symbol-processing capabilities such as binding (Soulos et al., 2020; Davies et al., 2023; Feng and Steinhardt, 2024; Dai et al., 2024a; Wu et al., 2025a), associative recall (e.g., induction heads; Olsson et al., 2022; Yang et al., 2025), parity (Li et al., 2025), and even first-order structural mappings (Hendel et al., 2023; Todd et al., 2024; Merullo et al., 2024; Yin and Steinhardt, 2025). Each of these capabilities is essential for generalizable symbolic reasoning, making them a natural place to begin in understanding the potential of current models toward symbolic reasoning. However, in isolation, these skills are insufficient to perform complex symbol-processing tasks such as TGT; and even a model which possesses each of these capabilities independently may not be able to successfully compose these skills to perform more complex tasks (Vegner et al., 2025). As such, prior work on atomic symbol-processing capabilities can be seen as being "bottom-up" in the sense that they begin by asking which basic skills various language models can learn and how they do so; whereas in this work, we aim to conduct a "top-down" study of language models in the more complex setting of TGT, analyzing the extent to which models capture constituent capabilities (e.g., in this work, parsing and structural mapping, which in principle each individually require more basic constituent skills such as segmentation and variable binding) in the course of learning the higher-level task.

8.7 Conclusion

In this work we investigate how Transformer language models learn to solve abstract symbolic reasoning tasks in-context. To do so in the context of TGT, by construction, they must do so by leveraging the abstract structure of provided examples, as non-structural, statistical features (e.g., lexical memorization) are not predictive. Our analysis reveals that even smaller-scale TGT-trained Transformer models can learn this task, and appear to show internal information flow consistent with the underlying general-purpose program. However, when placed in increasingly difficult OOD settings, their generalization suffers; and our findings suggest that the flawed compositional generalization of these models is due to brittle internal mechanisms for parsing and information localization, rather than a lack of such mechanisms altogether. In Section 9.2, we suggest several directions for future work to build on these findings toward better understanding, predicting, and improving OOD generalization of foundation models.

Part IV
Conclusions and Future Work

Chapter 9 Conclusions and Future Work

9.1 Conclusions

To summarize, this dissertation contains my work studying the latent feature representations learned by foundation models, how they implement (encode) these features, and how they leverage these features to perform various tasks; investigating how models generalize out-of-distribution (OOD) by controlling and evaluating the features they use across distributions, modalities, and tasks; and exploring how internal interpretations of foundation model representations can be used to predict and explain model OOD generalization. The specific contributions of each part and chapter are summarized below.

In **Part I** (Chapter 2 to Chapter 4), we introduced frameworks and methods for studying what latent feature representations are learned by foundation models, how they are encoded, and how models make use of these representations in the course of performing various tasks.

- In **Chapter 2**, we highlighted key parallels between cognitive science and mechanistic interpretability, surveying a wide variety of mechanistic interpretability methods and organizing them according to Marr's levels of analysis in cognitive science, discussing key challenges, underlying assumptions, and divergent objectives across these levels and bodies of work.
- In **Chapter 3**, we introduced the CALM framework and competence metric, which provide necessary theoretical grounding of mechanistic interpretability with respect to cognitive science and causal inference, facilitating more rigorous scientific understanding of model internals; and the GBI methodology also introduced in Chapter 3 provides a more powerful and reliable approach to causal probing, expanding the scope of inquiry to questions regarding nonlinear representation.
- In **Chapter 4**, we defined a reliability framework that provides sorely-needed empirical guidance by establishing a "level playing field" for comparison between different methods (or the same method with different hyperparameters), enabling one to select the most appropriate method for the given experimental setting and appropriately calibrate intervention hyperparameters to more reliably interpret what features models internally represent and use to perform different tasks.

In **Part II** (Chapter 5 to Chapter 7), we introduced tools, methods, and datasets for controlling and evaluating the features leveraged by models in performing various tasks, facilitating more robust models.

- In **Chapter 5**, we demonstrated the utility of generative models for performing interventional data augmentation, synthesizing data to correct for spurious correlations or biases in supervised models' training data, and thus controlling the features that they learn and leverage to perform a given task.
- In **Chapter 6**, we introduced the Focus Instruction Tuning post-training method, which allows for dynamic inference-time steering of the features that task-general models leverage to perform a given task, enabling enhanced out-of-distribution generalization and bias reduction that readily generalizes to new features and domains.
- In **Chapter 7**, we highlighted an important blind spot in the scientific literature regarding the robustness of vision models: prior work based on legacy benchmarks had indicated that modern foundation models have finally learned to rely on the "correct" feature of shape for visual recognition rather than spurious features like texture; but when we developed a challenging new benchmark using modern generative methods, we observed the same reliance on spurious features rather than shape as in prior literature with much earlier models and benchmarks.

Finally, in **Part III** (Chapter 8), we investigated how methods from mechanistic interpretability can be used to predict and explain when and why models fail to generalize out-of-distribution.

- In **Chapter 8**, we studied the internal mechanisms underlying models' capacity (or lack thereof) to compositionally generalize to increasingly difficult data distributions of a symbolic reasoning in-context learning task, finding that they broadly learned "correct" internal mechanisms that simply failed to robustly respond to novel inputs, indicating several actionable research directions to correct for such shortcomings (as discussed below in Section 9.2).

9.2 Future Work

The work in this dissertation (and particularly Part III) suggests that mechanistic interpretability has great potential to predict, explain, and improve OOD generalization. Several such directions for future work are elaborated below, building on the work in this dissertation to explore the intersection of mechanistic interpretability and transfer learning at greater theoretical depth and empirical rigor (Section 9.2.1); scaling such investigations to more challenging "real-world" scenarios (Section 9.2.2); and leveraging findings from such study to motivate new and improved neural architectures (Section 9.2.3).

9.2.1 Out-of-Distribution Prediction with Causal Probing

In Chapter 8, we leveraged a variety of mechanistic interpretability techniques to understand and explain models' OOD generalization on a symbolic reasoning in-context learning task (TGT). We began from the linear subspace hypotheses provided in the original paper proposing TGT (Smolensky et al., 2025), which is why we opted to interpret representations using a supervised subspace-discovery method (LDA), as CALM-style GBIs (see Chapter 3) with causal probing reliability validation (see Chapter 4) would not have been appropriate, given that the best-performing and most reliable GBIs are highly nonlinear (and thus would offer no utility in recovering linear subspaces).

However, moving beyond linear subspaces, we recommend that future work consider the following research questions regarding OOD prediction via causal probing:

1. Recall that the competence metric $\mathcal{C}_{\mathcal{T}}(M \mid \mathcal{G}_{\mathcal{T}})$ provided in CALM (Chapter 3) is intended to measure the extent to which a given model M leverages representations of the underlying data-generating process $\mathcal{G}_{\mathcal{T}}$ of task $\mathcal{T}$ in performing it. In theory, a model's competence should also predict its OOD generalization – if its internal process for performing $\mathcal{T}$ perfectly reflects $\mathcal{G}_{\mathcal{T}}$, then we expect perfect generalization; if its internal process leverages only spurious correlations in $\mathcal{G}_{\mathcal{T}}$, then we do not expect it to generalize when those spurious correlations are broken; and so on. As such, a natural research direction would be to test the extent to which competence does, in fact, predict OOD generalization. That is, given categorical, supervised, feature-level interventions (e.g., those from causal probing) that exhibit high reliability, and are able to consistently and systematically manipulate model behaviors, thus allowing one to confidently compute the competence metric provided in Chapter 3: is there, in fact, a correlation between a model's competence on a given task (measured with respect to some distribution), and its generalization with respect to that task across new, previously-unseen distributions?

2. In our work on causal probing reliability (Chapter 4), we examined the reliability of a variety of causal probing interventions across several models, finding that completeness and reliability were strongly correlated with changes in task performance. However, all such experiments were run in the same distribution. Given the work in Part III toward predicting OOD generalization using mechanistic interpretability methods, an interesting question is whether intervention completeness/reliability continues to track the predictiveness of model behavior OOD. For instance, consider an empirical competence graph $\hat{\mathcal{G}}_{\mathcal{T},A}$ that captures the features that an LLM is empirically found to leverage in the course of performing some task $\mathcal{T}$ (rather than the ground-truth

structural causal model of the task; see Section B.4.4) by running a range of causal probing intervention experiments like those in CALM (Chapter 3) using some intervention I_A. Next, suppose we obtain another such empirical competence graph $\hat{\mathcal{G}}_{\mathcal{T},B}$ using another intervention I_B. Furthermore, let us suppose that $\hat{\mathcal{G}}_{\mathcal{T},A}$ predicts accurate generalization only when some spurious correlation S_1 is not broken, and $\hat{\mathcal{G}}_{\mathcal{T},B}$ does the same for S_2; and finally, suppose that intervention I_A is in fact much more reliable than intervention I_B. In such a case, considering the two empirical competence graphs, $\hat{\mathcal{G}}_{\mathcal{T},A}$ and $\hat{\mathcal{G}}_{\mathcal{T},B}$, which are discovered using I_A and I_B, respectively, would we find that $\hat{\mathcal{G}}_{\mathcal{T},A}$ makes superior predictions of how OOD performance (in some test distribution where either S_1 or S_2 is broken) relative to $\hat{\mathcal{G}}_{\mathcal{T},B}$? That is, to what extent does the reliability of an intervention lead to better predictions of model behavior OOD?

9.2.2 Scaling Out-of-Distribution Prediction

Given the clear relationship in Chapter 8 between results of algorithmic and representational interpretation techniques and models' generalization across OOD splits, such techniques might be useful in predicting OOD performance of models (at both an instance- and distribution-level). Note that, while a few recent works have already shown that some mechanistic interpretability methods can be useful for predicting OOD performance in simpler contexts such as parity (Li et al., 2025) and multiple-choice question answering (Huang et al., 2025a), this has not yet (to our knowledge) been demonstrated in a more complex, compositional task such as TGT (Smolensky et al., 2025) – which, unlike the tasks studied in these prior works, requires models to simultaneously perform associative recall, variable binding, and full sequence-generation (rather than classification).

In particular, considering the research directions proposed above in Section 9.2.1, suppose that empirical competence graphs obtained with high-reliability interventions end up being strongly predictive of model OOD generalization (analogous to the findings of Huang et al., 2025a, who found a correlation between an interpretability method's OOD predictions and its *interchange intervention accuracy;* see Geiger et al., 2023). Would such methods be able to predict generalization in the context of challenging "real-world" tasks in complex OOD settings such as, e.g., compositional multi-hop tool calling (Nath et al., 2025; Patil et al., 2025; He et al., 2025a)?

9.2.3 Architecture Design

As noted in Chapter 8, our findings in Chapter 8 indicate that the primary issues in generalizing OOD with respect to the TGT symbolic reasoning in-context learning task stem from brittle mechanisms for parsing and information localization across layers (see Section 8.5). This suggests that approaches for improving information localization, such as architectural work developing more precise query-key matching in Transformer self-attention (e.g., Veličković et al., 2024; Ye et al., 2025; Opper et al., 2025), may be effective for resolving the internal brittleness of these mechanisms and thereby improving OOD generalization of models. More generally, findings from mechanistic interpretability work (particularly within algorithmic interpretation, as described in Chapter 2) might be broadly useful in motivating and validating new research directions in architectural design: once one understands a specific internal mechanism that leads to some failure mode, it may be possible to design alternative architectural components specifically to mitigate the failure mode of the observed mechanism, then repeat mechanistic experiments for trained models to validate whether any resulting performance gains are in fact due to the model learning different (or more robust) internal mechanisms as a result of the redesigned architectural components.

References

Abraham, Eldar D, Karel D'Oosterlinck, Amir Feder, Yair Gat, Atticus Geiger, Christopher Potts, Roi Reichart, and Zhengxuan Wu (2022). "Cebab: Estimating the causal effects of real-world concepts on nlp model behavior". In: *Advances in Neural Information Processing Systems* 35, pp. 17582–17596.

Achiam, Josh, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. (2023). "Gpt-4 technical report". In: *arXiv preprint arXiv:2303.08774*.

Adadi, Amina and Mohammed Berrada (2018). "Peeking inside the black-box: a survey on explainable artificial intelligence (XAI)". In: *IEEE access* 6, pp. 52138–52160.

Alain, Guillaume and Yoshua Bengio (2017). *Understanding intermediate layers using linear classifier probes*. URL: https://openreview.net/forum?id=ryF7rTqgl.

Alayrac, Jean-Baptiste, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. (2022). "Flamingo: a visual language model for few-shot learning". In: *Advances in neural information processing systems* 35, pp. 23716–23736.

Alazraki, Lisa, Lihu Chen, Ana Brassard, Joe Stacey, Hossein A Rahmani, and Marek Rei (2025). "AgentCoMa: A Compositional Benchmark Mixing Commonsense and Mathematical Reasoning in Real-World Scenarios". In: *arXiv preprint arXiv:2508.19988*.

Alhama, Raquel G and Willem Zuidema (2019). "A review of computational models of basic rule learning: The neural-symbolic debate and beyond". In: *Psychonomic bulletin & review* 26.4, pp. 1174–1194.

Alimisis, Panagiotis, Ioannis Mademlis, Panagiotis Radoglou-Grammatikis, Panagiotis Sarigiannidis, and Georgios Th Papadopoulos (2025). "Advances in diffusion models for image data augmentation: A review of methods, models, evaluation metrics and future research directions". In: *Artificial Intelligence Review* 58.4, p. 112.

Anderson, Peter, Basura Fernando, Mark Johnson, and Stephen Gould (2017). "Guided Open Vocabulary Image Captioning with Constrained Beam Search". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 936–945.

Anwar, Usman, Abulhair Saparov, Javier Rando, Daniel Paleka, Miles Turpin, Peter Hase, Ekdeep Singh Lubana, Erik Jenner, Stephen Casper, Oliver Sourbut, et al. (2024). "Foundational challenges in assuring alignment and safety of large language models". In: *arXiv preprint arXiv:2404.09932*.

Arjovsky, Martin, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz (2019). "Invariant risk minimization". In: *arXiv preprint arXiv:1907.02893*.

Arora, Aryaman, Dan Jurafsky, and Christopher Potts (2024a). "CausalGym: Benchmarking causal interpretability methods on linguistic tasks". In: *arXiv preprint arXiv:2402.12560*.

Arora, Sanjeev, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski (2018). "Linear algebraic structure of word senses, with applications to polysemy". In: *Transactions of the Association for Computational Linguistics* 6, pp. 483–495.

Arora, Simran, Sabri Eyuboglu, Aman Timalsina, Isys Johnson, Michael Poli, James Zou, Atri Rudra, and Christopher Re (2024b). "Zoology: Measuring and Improving Recall in Efficient Language Models". In: *The Twelfth International Conference on Learning Representations*. URL: https://openreview.net/forum?id=LY3ukUANko.

Arrieta, Alejandro Barredo, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. (2020). "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI". In: *Information fusion* 58, pp. 82–115.

Awadalla, Anas, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Shiori Sagawa, et al. (2023). "Openflamingo: An open-source framework for training large autoregressive vision-language models". In: *arXiv preprint arXiv:2308.01390*.

Azizi, Shekoofeh, Simon Kornblith, Chitwan Saharia, Mohammad Norouzi, and David J. Fleet (2023). "Synthetic Data from Diffusion Models Improves ImageNet Classification". In: *ArXiv* abs/2304.08466.

Bach, Sebastian, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek (2015). "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation". In: *PloS one* 10.7, e0130140.

Bai, Yuntao, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. (2022). "Training a helpful and harmless assistant with reinforcement learning from human feedback". In: *arXiv preprint arXiv:2204.05862*.

Baker, Nicholas and Philip J Kellman (2018). "Abstract shape representation in human visual perception." In: *Journal of Experimental Psychology: General* 147.9, p. 1295.

Baker, Nicholas, Hongjing Lu, Gennady Erlikhman, and Philip J Kellman (2018). "Deep convolutional networks do not classify based on global object shape". In: *PLoS computational biology* 14.12, e1006613.

Bandi, Peter, Oscar Geessink, Quirine Manson, Marcory Van Dijk, Maschenka Balkenhol, Meyke Hermsen, Babak Ehteshami Bejnordi, Byungjae Lee, Kyunghyun Paeng, Aoxiao Zhong, et al. (2018). "From detection of individual metastases to classification of lymph node status at the patient level: the CAMELYON17 challenge". In: *IEEE Transactions on Medical Imaging*.

Bansal, Hritik and Aditya Grover (2023). "Leaving Reality to Imagination: Robust Classification via Generated Datasets". In: *ICLR 2023 Workshop on Trustworthy and Reliable Large-Scale Machine Learning Models*. URL: https://openreview.net/forum?id=LjGqAFP6rA.

Bau, David, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba (2017). "Network dissection: Quantifying interpretability of deep visual representations". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6541–6549.

Bau, David, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B. Tenenbaum, William T. Freeman, and Antonio Torralba (2019). "Visualizing and Understanding Generative Adversarial Networks". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=Hyg_X2C5FX.

Belinkov, Yonatan (Mar. 2022). "Probing Classifiers: Promises, Shortcomings, and Advances". In: *Computational Linguistics* 48.1, pp. 207–219. DOI: 10.1162/coli_a_00422.

Belinkov, Yonatan, Sebastian Gehrmann, and Ellie Pavlick (July 2020). "Interpretability and Analysis in Neural NLP". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*. Online: Association for Computational Linguistics, pp. 1–5. DOI: 10.18653/v1/2020.acl-tutorials.1.

Belrose, Nora, David Schneider-Joseph, Shauli Ravfogel, Ryan Cotterell, Edward Raff, and Stella Biderman (2024). "Leace: Perfect linear concept erasure in closed form". In: *Advances in Neural Information Processing Systems* 36.

Benarous, Elior, Sotiris Anagnostidis, Luca Biggio, and Thomas Hofmann (2023). "Harnessing Synthetic Datasets: The Role of Shape Bias in Deep Neural Network Generalization". In: *ArXiv* abs/2311.06224. URL: https://api.semanticscholar.org/CorpusID:265128614.

Bender, Emily M., Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell (2021). "On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?" In: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. FAccT '21. Virtual Event, Canada: Association for Computing Machinery, pp. 610–623. ISBN: 9781450383097. DOI: 10.1145/3442188.3445922.

Bender, Emily M. and Alexander Koller (July 2020). "Climbing towards NLU: On Meaning, Form, and Understanding in the Age of Data". In: *Proceedings of the 58th Annual*

Meeting of the Association for Computational Linguistics. Online: Association for Computational Linguistics, pp. 5185–5198. DOI: 10.18653/v1/2020.acl-main.463.

Bengio, Yoshua, Jérôme Louradour, Ronan Collobert, and Jason Weston (2009). "Curriculum learning". In: *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48.

Bereska, Leonard and Efstratios Gavves (2024). "Mechanistic Interpretability for AI Safety–A Review". In: *arXiv preprint arXiv:2404.14082*.

Bertini Baldassini, Folco, Mustafa Shukor, Matthieu Cord, Laure Soulier, and Benjamin Piwowarski (2024). "What Makes Multimodal In-Context Learning Work?" In: *arXiv e-prints*, arXiv–2404.

Bhattacharjee, Amrita, Shaona Ghosh, Traian Rebedea, and Christopher Parisien (2024). "Towards Inference-time Category-wise Safety Steering for Large Language Models". In: *Neurips Safe Generative AI Workshop 2024*. URL: https://openreview.net/forum?id=EkQRNLPFcn.

Biderman, Stella, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. (2023). "Pythia: A suite for analyzing large language models across training and scaling". In: *International Conference on Machine Learning*. PMLR, pp. 2397–2430.

Biederman, Irving and Ginny Ju (1988). "Surface versus edge-based determinants of visual recognition". In: *Cognitive psychology* 20.1, pp. 38–64.

BigScience, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, et al. (2022). "Bloom: A 176b-parameter open-access multilingual language model". In: *arXiv preprint arXiv:2211.05100*.

Bills, Steven, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders (2023). "Language models can explain neurons in language models". In: *URL https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html* 2. URL: https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html.

Bitton-Guetta, Nitzan, Yonatan Bitton, Jack Hessel, Ludwig Schmidt, Yuval Elovici, Gabriel Stanovsky, and Roy Schwartz (2023). "Breaking common sense: Whoops! a vision-and-language benchmark of synthetic and compositional images". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2616–2627.

Blackburn, Simon (1995). "History of the philosophy of language". In: *The Oxford companion to philosophy*, pp. 454–458.

Bolukbasi, Tolga, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai (2016). "Man is to computer programmer as woman is to homemaker? debiasing word embeddings". In: *Advances in neural information processing systems* 29.

Bommasani, Rishi, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. (2021). "On the opportunities and risks of foundation models". In: *arXiv preprint arXiv:2108.07258*.

Bongers, Stephan, Patrick Forré, Jonas Peters, and Joris M Mooij (2021). "Foundations of structural causal models with cycles and latent variables". In: *The Annals of Statistics* 49.5, pp. 2885–2915.

Borkan, Daniel, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman (2019). "Nuanced metrics for measuring unintended bias with real data for text classification". In: *Companion proceedings of the 2019 world wide web conference*, pp. 491–500.

Breiman, Leo (2001). "Statistical modeling: The two cultures (with comments and a rejoinder by the author)". In: *Statistical science* 16.3, pp. 199–231.

Bricken, Trenton, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah (2023). "Towards Monosemanticity: Decomposing Language Models With Dictionary Learning". In: *Transformer Circuits Thread*. https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Broniatowski, David A et al. (2021). "Psychological foundations of explainability and interpretability in artificial intelligence". In: *NIST, Tech. Rep.*

Brooks, Tim, Aleksander Holynski, and Alexei A Efros (2023). "Instructpix2pix: Learning to follow image editing instructions". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18392–18402.

Brown, Davis, Nikhil Vyas, and Yamini Bansal (2023). "On privileged and convergent bases in neural network representations". In: *arXiv preprint arXiv:2307.12941*.

Brown, Tom, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. (2020). "Language models are few-shot learners". In: *Advances in neural information processing systems* 33, pp. 1877–1901.

Bruner, Jerome (2017). *A study of thinking*. Routledge.

Bubeck, Sébastien et al. (2015). "Convex optimization: Algorithms and complexity". In: *Foundations and Trends® in Machine Learning* 8.3-4, pp. 231–357.

Bubeck, Sébastien, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. (2023). "Sparks of artificial general intelligence: Early experiments with gpt-4". In: *arXiv preprint arXiv:2303.12712*.

Bühlmann, Peter (2020). "Invariance, causality and robustness". In: *Statistical Science* 35.3, pp. 404–426.

Burns, Collin, Haotian Ye, Dan Klein, and Jacob Steinhardt (2023). "Discovering Latent Knowledge in Language Models Without Supervision". In: *The Eleventh International Conference on Learning Representations*. URL: https://openreview.net/forum?id=ETKGuby0hcs.

Canby*, Marc, Adam Davies*, Chirag Rastogi, and Julia Hockenmaier (2025). "How Reliable are Causal Probing Interventions?" In: *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics. URL: https://aclanthology.org/2025.ijcnlp-long.47/.

Canny, John (1986). "A computational approach to edge detection". In: *IEEE Transactions on pattern analysis and machine intelligence* 6, pp. 679–698.

Cao, Steven, Victor Sanh, and Alexander M Rush (2021). "Low-complexity probing via finding subnetworks". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 960–966.

Carlini, Nicholas, Milad Nasr, Christopher A Choquette-Choo, Matthew Jagielski, Irena Gao, Pang Wei W Koh, Daphne Ippolito, Florian Tramer, and Ludwig Schmidt (2024). "Are aligned neural networks adversarially aligned?" In: *Advances in Neural Information Processing Systems* 36.

Carlini, Nicholas, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B. Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel (2020). "Extracting Training Data from Large Language Models". In: *CoRR* abs/2012.07805. arXiv: 2012.07805. URL: https://arxiv.org/abs/2012.07805.

Chen, Haibo, Lei Zhao, Huiming Zhang, Zhizhong Wang, Zhiwen Zuo, Ailin Li, Wei Xing, and Dongming Lu (2021). "Diverse image style transfer via invertible cross-space mapping". In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE Computer Society, pp. 14860–14869.

Chen, Zixiang, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu (2024). "Self-Play Fine-Tuning Converts Weak Language Models to Strong Language Models". In: *Forty-first International Conference on Machine Learning*. URL: https://openreview.net/forum?id=O4cHTxW9BS.

Chew, Oscar, Hsuan-Tien Lin, Kai-Wei Chang, and Kuan-Hao Huang (2024). "Understanding and Mitigating Spurious Correlations in Text Classification with Neighborhood Analysis". In: *Findings of the Association for Computational Linguistics: EACL 2024*, pp. 1013–1025.

Chiang, Wei-Lin, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing (Mar. 2023). *Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90% ChatGPT Quality*. URL: https://lmsys.org/blog/2023-03-30-vicuna/.

Chollet, François (2019). "On the measure of intelligence". In: *arXiv preprint arXiv:1911.01547*.

Chomsky, Noam (1965). "Aspects of the Theory of Syntax". In: *MIT Press*.

Chomsky, Noam (1980). "A review of BF Skinner's Verbal Behavior". In: *The Language and Thought Series*, pp. 48–64.

Chomsky, Noam (2002). *Syntactic structures*. Mouton de Gruyter.

Choshen, Leshem, Elad Venezian, Shachar Don-Yehia, Noam Slonim, and Yoav Katz (2022). "Where to start? Analyzing the potential value of intermediate models". In: *arXiv preprint arXiv:2211.00107*.

Chowdhery, Aakanksha, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. (2022). "Palm: Scaling language modeling with pathways". In: *arXiv preprint arXiv:2204.02311*.

Chu, Tianzhe, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma (2025). "SFT Memorizes, RL Generalizes: A Comparative Study of Foundation Model Post-training". In: *Forty-second International Conference on Machine Learning*. URL: https://openreview.net/forum?id=dYur3yabMj.

Clark, Kevin, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning (Aug. 2019). "What Does BERT Look at? An Analysis of BERT's Attention". In: *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Ed. by Tal Linzen, Grzegorz Chrupała, Yonatan Belinkov, and Dieuwke Hupkes. Florence, Italy: Association for Computational Linguistics, pp. 276–286. DOI: 10.18653/v1/W19-4828.

Conmy, Arthur, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso (2023). "Towards automated circuit discovery for mechanistic interpretability". In: *Advances in Neural Information Processing Systems* 36, pp. 16318–16352.

Croce, Francesco and Matthias Hein (2020). "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks". In: *International conference on machine learning*. PMLR, pp. 2206–2216.

Csordás, Róbert, Kazuki Irie, and Jürgen Schmidhuber (2021a). "The Neural Data Router: Adaptive Control Flow in Transformers Improves Systematic Generalization". In: *CoRR* abs/2110.07732. arXiv: 2110.07732. URL: https://arxiv.org/abs/2110.07732.

Csordás, Róbert, Christopher Potts, Christopher D Manning, and Atticus Geiger (Nov. 2024). "Recurrent Neural Networks Learn to Store and Generate Sequences using Non-Linear Representations". In: *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*. Ed. by Yonatan Belinkov, Najoung Kim, Jaap Jumelet, Hosein Mohebbi, Aaron Mueller, and Hanjie Chen. Miami, Florida, US: Association for Computational Linguistics, pp. 248–262. DOI: 10.18653/v1/2024.blackboxnlp-1.17.

Csordás, Róbert, Sjoerd van Steenkiste, and Jürgen Schmidhuber (2021b). "Are Neural Nets Modular? Inspecting Functional Modularity Through Differentiable Weight Masks". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=7uVcpu-gMD.

Cubuk, Ekin D., Barret Zoph, Jonathon Shlens, and Quoc V. Le (2020). "Randaugment: Practical Automated Data Augmentation With a Reduced Search Space". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

Cugu, Ilke, Massimiliano Mancini, Yanbei Chen, and Zeynep Akata (June 2022). "Attention Consistency on Visual Corruptions for Single-Source Domain Generalization". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 4165–4174.

Cunningham, Hoagy, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey (2023). "Sparse autoencoders find highly interpretable features in language models". In: *arXiv preprint arXiv:2309.08600*.

Dai, Damai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei (May 2022). "Knowledge Neurons in Pretrained Transformers". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Dublin, Ireland:

Association for Computational Linguistics, pp. 8493–8502. DOI: 10.18653/v1/2022.acl-long.581.

Dai, Qin, Benjamin Heinzerling, and Kentaro Inui (Nov. 2024a). "Representational Analysis of Binding in Language Models". In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Ed. by Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen. Miami, Florida, USA: Association for Computational Linguistics, pp. 17468–17493. DOI: 10.18653/v1/2024.emnlp-main.967.

Dai, Wenliang, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi (2024b). "Instructblip: Towards general-purpose vision-language models with instruction tuning". In: *Advances in Neural Information Processing Systems* 36.

Dalvi, Fahim, Nadir Durrani, Hassan Sajjad, Yonatan Belinkov, Anthony Bau, and James Glass (2019). "What is one grain of sand in the desert? analyzing individual neurons in deep nlp models". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33, pp. 6309–6317.

David, Etienne, Simon Madec, Pouria Sadeghi-Tehran, Helge Aasen, Bangyou Zheng, Shouyang Liu, Norbert Kirchgessner, Goro Ishikawa, Koichi Nagasawa, Minhajul A Badhon, Curtis Pozniak, Benoit de Solan, Andreas Hund, Scott C. Chapman, Frederic Baret, Ian Stavness, and Wei Guo (2020). "Global Wheat Head Detection (GWHD) dataset: a large and diverse dataset of high-resolution RGB-labelled images to develop and benchmark wheat head detection methods". In: *Plant Phenomics* 2020.

David, Etienne, Mario Serouart, Daniel Smith, Simon Madec, Kaaviya Velumani, Shouyang Liu, Xu Wang, Francisco Pinto Espinosa, Shahameh Shafiee, Izzat S. A. Tahir, Hisashi Tsujimoto, Shuhei Nasuda, Bangyou Zheng, Norbert Kichgessner, Helge Aasen, Andreas Hund, Pouria Sadhegi-Tehran, Koichi Nagasawa, Goro Ishikawa, Sebastien Dandrifosse, Alexis Carlier, Benoit Mercatoris, Ken Kuroki, Haozhou Wang, Masanori Ishii, Minhajul A. Badhon, Curtis Pozniak, David Shaner LeBauer, Morten Lilimo, Jesse Poland, Scott Chapman, Benoit de Solan, Frederic Baret, Ian Stavness, and Wei Guo (2021). *Global Wheat Head Dataset 2021: an update to improve the benchmarking wheat head localization with more diversity*. arXiv: 2105.07660 [cs.CV].

Davies, Adam, Jize Jiang, and ChengXiang Zhai (2024). "Competence-Based Analysis of Language Models". In: *NeurIPS 2024 Workshop on Interpretable AI*. URL: https://openreview.net/forum?id=x6ZM5Is2Po.

Davies, Adam and Ashkan Khakzar (2024). "The Cognitive Revolution in Interpretability: From Explaining Behavior to Interpreting Representations and Algorithms". In: *arXiv preprint arXiv:2408.05859*. URL: https://arxiv.org/abs/2408.05859.

Davies, Adam, Mattia Opper, Roland Fernandez, Paul Smolensky, and Jianfeng Gao (2026). *How Do LLMs Represent and Process Symbols In-Context?*

Davies, Xander, Max Nadeau, Nikhil Prakash, Tamar Rott Shaham, and David Bau (2023). "Discovering variable binding circuitry with desiderata". In: *arXiv preprint arXiv:2307.03637*.

De Cao, Nicola, Michael Schlichtkrull, Wilker Aziz, and Ivan Titov (2020). "How do decisions emerge across layers in neural models? interpretation with differentiable masking". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 3243–3255.

Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei (2009). "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee, pp. 248–255.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (June 2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4171–4186. DOI: 10.18653/v1/N19-1423.

DeVries, Terrance and Graham W Taylor (2017). "Improved regularization of convolutional neural networks with cutout". In: *arXiv preprint arXiv:1708.04552*.

Dobrzeniecka, Alicja, Antske Fokkens, and Pia Sommerauer (2025). "Improving Causal Interventions in Amnesic Probing with Mean Projection or LEACE". In: *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 12981–12993.

Dolly, Free (2023). *Introducing the World's First Truly Open Instruction-Tuned LLM. databricks. com*.

Dong, Yi, Zhilin Wang, Makesh Sreedhar, Xianchao Wu, and Oleksii Kuchaiev (2023). "SteerLM: Attribute Conditioned SFT as an (User-Steerable) Alternative to RLHF". In: *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 11275–11288.

Doran, Derek, Sarah Schulz, and Tarek R Besold (2017). "What does explainable AI really mean? A new conceptualization of perspectives". In: *arXiv preprint arXiv:1710.00794*.

Doshi-Velez, Finale and Been Kim (2017). "Towards a rigorous science of interpretable machine learning". In: *arXiv preprint arXiv:1702.08608*.

Dosovitskiy, Alexey, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. (2020). "An image is worth 16x16 words: Transformers for image recognition at scale". In: *arXiv preprint arXiv:2010.11929*.

Dou, Shihan, Enyu Zhou, Yan Liu, Songyang Gao, Wei Shen, Limao Xiong, Yuhao Zhou, Xiao Wang, Zhiheng Xi, Xiaoran Fan, et al. (2024). "LoRAMoE: Alleviating world knowledge forgetting in large language models via MoE-style plugin". In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1932–1945.

Dryden, Ian L and Kanti V Mardia (2016). *Statistical shape analysis: with applications in R*. John Wiley & Sons.

Dubey, Abhimanyu, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. (2024). "The llama 3 herd of models". In: *arXiv preprint arXiv:2407.21783*.

Elazar, Yanai, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg (Dec. 2021a). "Measuring and Improving Consistency in Pretrained Language Models". In: *Transactions of the Association for Computational Linguistics* 9, pp. 1012–1031. ISSN: 2307-387X. DOI: 10.1162/tacl_a_00410. eprint: https://direct.mit.edu/tacl/article-pdf/doi/10.1162/tacl_a_00410/1975957/tacl_a_00410.pdf.

Elazar, Yanai, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg (2021b). "Amnesic Probing: Behavioral Explanation with Amnesic Counterfactuals". In: *Transactions of the Association for Computational Linguistics* 9, pp. 160–175. DOI: 10.1162/tacl_a_00359.

Elder, James H. and Ljiljana Velisavljević (July 2009). "Cue dynamics underlying rapid detection of animals in natural scenes". In: *Journal of Vision* 9.7, pp. 7–7. ISSN: 1534-7362. DOI: 10.1167/9.7.7. eprint: https://arvojournals.org/arvo/content_public/journal/jov/932863/jov-9-7-7.pdf.

Elhage, Nelson, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. (2022). "Toy models of superposition". In: *arXiv preprint arXiv:2209.10652*.

Elhage, Nelson, Robert Lasenby, and Christopher Olah (2023). "Privileged bases in the transformer residual stream". In: *Transformer Circuits Thread* 24.

Elhage, Nelson, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. (2021). "A mathematical framework for transformer circuits". In: *Transformer Circuits Thread* 1.

Engels, Joshua, Eric J Michaud, Isaac Liao, Wes Gurnee, and Max Tegmark (2025a). "Not All Language Model Features Are One-Dimensionally Linear". In: *The Thirteenth International Conference on Learning Representations*. URL: https://openreview.net/forum?id=d63a4AM4hb.

Engels, Joshua, Logan Riggs Smith, and Max Tegmark (2025b). "Decomposing The Dark Matter of Sparse Autoencoders". In: *Transactions on Machine Learning Research*. ISSN: 2835-8856. URL: https://openreview.net/forum?id=sXq3Wb3vef.

Erhan, Dumitru, Yoshua Bengio, Aaron Courville, and Pascal Vincent (2009). "Visualizing higher-layer features of a deep network". In: *University of Montreal* 1341.3, p. 1.

Ettinger, Allyson (Jan. 2020). "What BERT Is Not: Lessons from a New Suite of Psycholinguistic Diagnostics for Language Models". In: *Transactions of the Association for Computational Linguistics* 8, pp. 34–48. ISSN: 2307-387X. DOI: 10.1162/tacl_a_00298. eprint: https://direct.mit.edu/tacl/article-pdf/doi/10.1162/tacl_a_00298/1923116/tacl_a_00298.pdf.

European Parliament, The Council of the (2016). "Regulation (EU) 2016/679 of the European Parliament and of the Council". In: *Official Journal of the European Union* L 119. General Data Protection Regulation (GDPR), pp. 1–88. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679.

Faruqui, Manaal, Yulia Tsvetkov, Dani Yogatama, Chris Dyer, and Noah A. Smith (July 2015). "Sparse Overcomplete Word Vector Representations". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Ed. by Chengqing Zong and Michael Strube. Beijing, China: Association for Computational Linguistics, pp. 1491–1500. DOI: 10.3115/v1/P15-1144.

Feng, Jiahai and Jacob Steinhardt (2024). "How do Language Models Bind Entities in Context?" In: *The Twelfth International Conference on Learning Representations*. URL: https://openreview.net/forum?id=zb3b6oKO77.

Fodor, James (2025). "Line goes up? inherent limitations of benchmarks for evaluating large language models". In: *arXiv preprint arXiv:2502.14318*.

Fodor, Jerry A and Zenon W Pylyshyn (1988). "Connectionism and cognitive architecture: A critical analysis". In: *Cognition* 28.1-2, pp. 3–71.

Fong, Ruth, Mandela Patrick, and Andrea Vedaldi (2019). "Understanding deep networks via extremal perturbations and smooth masks". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2950–2958.

Fong, Ruth and Andrea Vedaldi (2018). "Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8730–8738.

Fu, Chaoyou, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji (2024a). *MME: A Comprehensive Evaluation Benchmark for Multimodal Large Language Models*. arXiv: 2306.13394 [cs.CV].

Fu, Tingchen, Deng Cai, Lemao Liu, Shuming Shi, and Rui Yan (2024b). "Disperse-Then-Merge: Pushing the Limits of Instruction Tuning via Alignment Tax Reduction". In: *Findings of the Association for Computational Linguistics ACL 2024*, pp. 2967–2985.

Gal, Rinon, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or (2022). *An Image is Worth One Word: Personalizing Text-to-Image Generation using Textual Inversion*. DOI: 10.48550/ARXIV.2208.01618.

Gandelsman, Yossi, Alexei A Efros, and Jacob Steinhardt (2024). "Interpreting CLIP's Image Representation via Text-Based Decomposition". In: *The Twelfth International Conference on Learning Representations*. URL: https://openreview.net/forum?id=5Ca9sSzuDp.

Gao, Jin, Jialing Zhang, Xihui Liu, Trevor Darrell, Evan Shelhamer, and Dequan Wang (2022). "Back to the source: Diffusion-driven test-time adaptation". In: *arXiv preprint arXiv:2207.03442*.

Gatys, Leon A, Alexander S Ecker, and Matthias Bethge (2016). "Image style transfer using convolutional neural networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2414–2423.

Gavrikov, Paul, Jovita Lukasik, Steffen Jung, Robert Geirhos, Bianca Lamm, Muhammad Jehanzeb Mirza, Margret Keuper, and Janis Keuper (2024). "Are Vision Language Models Texture or Shape Biased and Can We Steer Them?" In: *arXiv preprint arXiv:2403.09193*.

Geiger, Atticus, Chris Potts, and Thomas Icard (2023). "Causal Abstraction for Faithful Model Interpretation". In: *arXiv preprint arXiv:2301.04709*.

Geiger, Atticus, Kyle Richardson, and Christopher Potts (Nov. 2020). "Neural Natural Language Inference Models Partially Embed Theories of Lexical Entailment and Negation". In: *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*. Ed. by Afra Alishahi, Yonatan Belinkov, Grzegorz Chrupała, Dieuwke Hupkes, Yuval Pinter, and Hassan Sajjad. Online: Association for Computational Linguistics, pp. 163–173. DOI: 10.18653/v1/2020.blackboxnlp-1.16.

Geiger, Atticus, Zhengxuan Wu, Hanson Lu, Josh Rozner, Elisa Kreiss, Thomas Icard, Noah Goodman, and Christopher Potts (2022). "Inducing Causal Structure for Interpretable Neural Networks". In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato. Vol. 162. Proceedings of Machine Learning Research. PMLR, pp. 7324–7338. URL: https://proceedings.mlr.press/v162/geiger22a.html.

Geiger, Atticus, Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah Goodman (2024). "Finding Alignments Between Interpretable Causal Variables and Distributed Neural Representations". In: *Proceedings of the Third Conference on Causal Learning and Reasoning*. Ed. by Francesco Locatello and Vanessa Didelez. Vol. 236. Proceedings of Machine Learning Research. PMLR, pp. 160–187. URL: https://proceedings.mlr.press/v236/geiger24a.html.

Geirhos, Robert, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann (2020a). "Shortcut learning in deep neural networks". In: *Nature Machine Intelligence* 2.11, pp. 665–673.

Geirhos, Robert, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann (2020b). "Shortcut learning in deep neural networks". In: *Nature Machine Intelligence* 2.11, pp. 665–673.

Geirhos, Robert, Kantharaju Narayanappa, Benjamin Mitzkus, Tizian Thieringer, Matthias Bethge, Felix A Wichmann, and Wieland Brendel (2021). "Partial success in closing the gap between human and machine vision". In: *Advances in Neural Information Processing Systems* 34, pp. 23885–23899.

Geirhos, Robert, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel (2018). "ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness". In: *International conference on learning representations*.

Gentner, Dedre (1983). "Structure-mapping: A theoretical framework for analogy". In: *Cognitive science* 7.2, pp. 155–170.

Geva, Mor, Roei Schuster, Jonathan Berant, and Omer Levy (Nov. 2021). "Transformer Feed-Forward Layers Are Key-Value Memories". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Ed. by Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 5484–5495. DOI: 10.18653/v1/2021.emnlp-main.446.

Ghifary, Muhammad, David Balduzzi, W Bastiaan Kleijn, and Mengjie Zhang (2016). "Scatter component analysis: A unified framework for domain adaptation and domain generalization". In: *IEEE transactions on pattern analysis and machine intelligence* 39.7, pp. 1414–1430.

Ghosh, Sreyan, Chandra Kiran Reddy Evuru, Sonal Kumar, Ramaneswaran S, Deepali Aneja, Zeyu Jin, Ramani Duraiswami, and Dinesh Manocha (2024). "A Closer Look at the Limitations of Instruction Tuning". In: *Forty-first International Conference on Machine Learning*. URL: https://openreview.net/forum?id=XkHJo8iXGQ.

Gong, Mingming, Kun Zhang, Tongliang Liu, Dacheng Tao, Clark Glymour, and Bernhard Schölkopf (2016). "Domain adaptation with conditional transferable components". In: *International conference on machine learning*. PMLR, pp. 2839–2848.

Goodfellow, Ian, Jonathon Shlens, and Christian Szegedy (2015). "Explaining and Harnessing Adversarial Examples". In: *International Conference on Learning Representations*. URL: http://arxiv.org/abs/1412.6572.

Gowal, Sven, Krishnamurthy Dj Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Mann, and Pushmeet Kohli (2019). "Scalable verified training for provably robust image classification". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4842–4851.

Gowda, Sindhu C.M., Shalmali Joshi, Haoran Zhang, and Marzyeh Ghassemi (2021). "Pulling Up by the Causal Bootstraps: Causal Data Augmentation for Pre-training Debiasing". In: *arXiv preprint arXiv:2108.12510*.

Groeneveld, Dirk, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, et al. (2024). "OLMo: Accelerating the Science of Language Models". In: *arXiv preprint arXiv:2402.00838*.

Gross, Charles G (2002). "Genealogy of the "grandmother cell"". In: *The Neuroscientist* 8.5, pp. 512–518.

Guan, Tianrui, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, et al. (2023). "HallusionBench: An Advanced Diagnostic Suite for Entangled Language Hallucination and Visual Illusion in Large Vision-Language Models". In: *arXiv preprint arXiv:2310.14566*.

Guldimann, Philipp, Alexander Spiridonov, Robin Staab, Nikola Jovanovic, Mark Vero, Velko Vechev, Anna Gueorguieva, Mislav Balunovic, Nikola Konstantinov, Pavol Bielik, et al. (2024). "COMPL-AI Framework: A Technical Interpretation and LLM Benchmarking Suite for the EU Artificial Intelligence Act". In: *CoRR*.

Gulrajani, Ishaan and David Lopez-Paz (2021). "In Search of Lost Domain Generalization". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=lQdXeXDoWtI.

Hamrick, Jessica and Shakir Mohamed (2020). "Levels of analysis for machine learning". In.

Han, Chi, Jialiang Xu, Manling Li, Yi Fung, Chenkai Sun, Nan Jiang, Tarek Abdelzaher, and Heng Ji (Aug. 2024). "Word Embeddings Are Steers for Language Models". In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Lun-Wei Ku, Andre Martins, and Vivek Srikumar. Bangkok, Thailand: Association for Computational Linguistics, pp. 16410–16430. DOI: 10.18653/v1/2024.acl-long.864.

Hancox-Li, Leif (2020). "Robustness in machine learning explanations: Does it matter?" In: *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pp. 640–647.

Hanna, Michael and David Mareček (Nov. 2021). "Analyzing BERT's Knowledge of Hypernymy via Prompting". In: *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*. Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 275–282. DOI: 10.18653/v1/2021.blackboxnlp-1.20.

He, Jie, Jennifer Neville, Mengting Wan, Longqi Yang, Hui Liu, Xiaofeng Xu, Xia Song, Jeff Z Pan, and Pei Zhou (2025a). "GenTool: Enhancing Tool Generalization in Language Models through Zero-to-One and Weak-to-Strong Simulation". In: *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 1097–1122.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

He, Ruifei, Shuyang Sun, Xin Yu, Chuhui Xue, Wenqing Zhang, Philip Torr, Song Bai, and Xiaojuan Qi (2023). "Is synthetic data from generative models ready for image recognition?" In: *The Eleventh International Conference on Learning Representations*.

He, Yifei, Yuzheng Hu, Yong Lin, Tong Zhang, and Han Zhao (2025b). "Localize-and-Stitch: Efficient Model Merging via Sparse Task Arithmetic". In: *Transactions on Machine Learning Research*. ISSN: 2835-8856. URL: https://openreview.net/forum?id=9CWU8Oi86d.

Hemmat, Arshia, Adam Davies, Tom A. Lamb, Jianhao Yuan, Philip Torr, Ashkan Khakzar, and Francesco Pinto (2024). "Hidden in Plain Sight: Evaluating Abstract Shape Recognition in Vision-Language Models". In: *Advances in Neural Information Processing Systems*. Ed. by A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak,

and C. Zhang. Vol. 37. Curran Associates, Inc., pp. 88527–88556. URL: https://proceedings.neurips.cc/paper_files/paper/2024/file/a13ff984831deea39e6132bafdfdd6d5-Paper-Datasets_and_Benchmarks_Track.pdf.

Hendel, Roee, Mor Geva, and Amir Globerson (Dec. 2023). "In-Context Learning Creates Task Vectors". In: *Findings of the Association for Computational Linguistics: EMNLP 2023*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, pp. 9318–9333. DOI: 10.18653/v1/2023.findings-emnlp.624.

Hendrycks, Dan, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt (2021a). "Measuring Massive Multitask Language Understanding". In: *Proceedings of the International Conference on Learning Representations (ICLR)*.

Hendrycks, Dan, Nicholas Carlini, John Schulman, and Jacob Steinhardt (2021b). "Unsolved problems in ml safety". In: *arXiv preprint arXiv:2109.13916*.

Hendrycks, Dan and Thomas Dietterich (2019). "Benchmarking neural network robustness to common corruptions and perturbations". In: *arXiv preprint arXiv:1903.12261*.

Hendrycks, Dan, Norman Mu, Ekin Dogus Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan (2020). "AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty". In: *International Conference on Learning Representations*.

Hendrycks, Dan, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song (2021c). "Natural adversarial examples". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15262–15271.

Hendrycks, Dan, Andy Zou, Mantas Mazeika, Leonard Tang, Bo Li, Dawn Song, and Jacob Steinhardt (2022). "PixMix: Dreamlike Pictures Comprehensively Improve Safety Measures". In: *CVPR*.

Hermann, Katherine, Hossein Mobahi, FEL Thomas, and Michael Curtis Mozer (2024). "On the Foundations of Shortcut Learning". In: *The Twelfth International Conference on Learning Representations*.

Hernandez, Evan, Sarah Schwettmann, David Bau, Teona Bagashvili, Antonio Torralba, and Jacob Andreas (2021). "Natural language descriptions of deep visual features". In: *International Conference on Learning Representations*.

Heusel, Martin, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter (2017). "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan,

and R. Garnett. Vol. 30. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper/2017/file/8a1d694707eb0fefe65871369074926d-Paper.pdf.

Hewitt, John and Percy Liang (Nov. 2019). "Designing and Interpreting Probes with Control Tasks". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 2733–2743. DOI: 10.18653/v1/D19-1275.

Hong, Minui, Jinwoo Choi, and Gunhee Kim (2021). "StyleMix: Separating Content and Style for Enhanced Data Augmentation". In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14857–14865. DOI: 10.1109/CVPR46437.2021.01462.

Hu, Edward J, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. (2021). "LoRA: Low-Rank Adaptation of Large Language Models". In: *International Conference on Learning Representations*.

Huang, Jing, Junyi Tao, Thomas Icard, Diyi Yang, and Christopher Potts (2025a). "Internal Causal Mechanisms Robustly Predict Language Model Out-of-Distribution Behaviors". In: *Forty-second International Conference on Machine Learning*. URL: https://openreview.net/forum?id=Ofa1cspTrv.

Huang, Jing, Zhengxuan Wu, Christopher Potts, Mor Geva, and Atticus Geiger (Aug. 2024). "RAVEL: Evaluating Interpretability Methods on Disentangling Language Model Representations". In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Lun-Wei Ku, Andre Martins, and Vivek Srikumar. Bangkok, Thailand: Association for Computational Linguistics, pp. 8669–8687. DOI: 10.18653/v1/2024.acl-long.470.

Huang, Shijia, Jianqiao Zhao, Yanyang Li, and Liwei Wang (2023). "Learning preference model for llms via automatic preference data generation". In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 9187–9199.

Huang, Shulin, Linyi Yang, Yan Song, Shuang Chen, Leyang Cui, Ziyu Wan, Qingcheng Zeng, Ying Wen, Kun Shao, Weinan Zhang, et al. (2025b). "Thinkbench: Dynamic out-of-distribution evaluation for robust llm reasoning". In: *arXiv preprint arXiv:2502.16268*.

Huang, Xun and Serge Belongie (2017). "Arbitrary style transfer in real-time with adaptive instance normalization". In: *Proceedings of the IEEE international conference on computer vision*, pp. 1501–1510.

Huang, Zeyi, Haohan Wang, Eric P. Xing, and Dong Huang (2020). "Self-Challenging Improves Cross-Domain Generalization". In: *ECCV*.

Hummel, John E (2001). "Complementary solutions to the binding problem in vision: Implications for shape perception and object recognition". In: *Visual cognition* 8.3-5, pp. 489–517.

Hupkes, Dieuwke, Verna Dankers, Mathijs Mul, and Elia Bruni (2020). "Compositionality decomposed: How do neural networks generalise?" In: *Journal of Artificial Intelligence Research* 67, pp. 757–795.

Ilharco, Gabriel, Mitchell Wortsman, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt (July 2021). *OpenCLIP*. Version 0.1. DOI: 10.5281/zenodo.5143773.

Ilse, Maximilian, Jakub M Tomczak, and Patrick Forré (2021). "Selecting data augmentation for simulating interventions". In: *International conference on machine learning*. PMLR, pp. 4555–4562.

Inan, Hakan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, et al. (2023). "Llama guard: Llm-based input-output safeguard for human-ai conversations". In: *arXiv preprint arXiv:2312.06674*.

Iskander, Shadi, Kira Radinsky, and Yonatan Belinkov (July 2023). "Shielded Representations: Protecting Sensitive Attributes Through Iterative Gradient-Based Projection". In: *Findings of the Association for Computational Linguistics: ACL 2023*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, pp. 5961–5977. DOI: 10.18653/v1/2023.findings-acl.369.

Islam, Md Amirul, Matthew Kowal, Patrick Esser, Sen Jia, Björn Ommer, Konstantinos G Derpanis, and Neil Bruce (2021). "Shape or Texture: Understanding Discriminative Features in CNNs". In: *International Conference on Learning Representations*.

Jackson, Philip T, Amir Atapour-Abarghouei, Stephen Bonner, Toby P Breckon, and Boguslaw Obara (2019). "Style Augmentation: Data Augmentation via Style Randomization". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 83–92.

Jawahar, Ganesh, Benoit Sagot, and Djame Seddah (2019). "What does BERT learn about the structure of language?" In: *ACL 2019-57th Annual Meeting of the Association for Computational Linguistics*.

Jiang, Albert Q, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel,

Guillaume Lample, Lucile Saulnier, et al. (2023). "Mistral 7B". In: *arXiv preprint arXiv:2310.06825*.

Jiang, Yibo, Goutham Rajendran, Pradeep Kumar Ravikumar, Bryon Aragam, and Victor Veitch (2024). "On the Origins of Linear Representations in Large Language Models". In: *Forty-first International Conference on Machine Learning*. URL: https://openreview.net/forum?id=otuTw4Mghk.

Kaddour, Jean, Aengus Lynch, Qi Liu, Matt J Kusner, and Ricardo Silva (2022). "Causal machine learning: A survey and open problems". In: *arXiv preprint arXiv:2206.15475*.

Karpathy, Andrej, Justin Johnson, and Li Fei-Fei (2015). "Visualizing and understanding recurrent networks". In: *arXiv preprint arXiv:1506.02078*.

Kendall, David G (1984). "Shape manifolds, procrustean metrics, and complex projective spaces". In: *Bulletin of the London mathematical society* 16.2, pp. 81–121.

Khakzar, Ashkan, Soroosh Baselizadeh, Saurabh Khanduja, Christian Rupprecht, Seong Tae Kim, and Nassir Navab (2021). "Neural Response Interpretation through the Lens of Critical Pathways". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13528–13538.

Khakzar, Ashkan, Pedram Khorsandi, Rozhin Nobahari, and Nassir Navab (2022). "Do explanations explain? model knows best". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10244–10253.

Khashabi, Daniel, Xinxi Lyu, Sewon Min, Lianhui Qin, Kyle Richardson, Sean Welleck, Hannaneh Hajishirzi, Tushar Khot, Ashish Sabharwal, Sameer Singh, and Yejin Choi (July 2022). "Prompt Waywardness: The Curious Case of Discretized Interpretation of Continuous Prompts". In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Seattle, United States: Association for Computational Linguistics, pp. 3631–3643. DOI: 10.18653/v1/2022.naacl-main.266.

Kim, Been, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. (2018). "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav)". In: *International conference on machine learning*. PMLR, pp. 2668–2677.

Kim, Najoung and Tal Linzen (Nov. 2020). "COGS: A Compositional Generalization Challenge Based on Semantic Interpretation". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu. Online: Association for Computational Linguistics, pp. 9087–9105. DOI: 10.18653/v1/2020.emnlp-main.731.

Kindratenko, Volodymyr, Dawei Mu, Yan Zhan, John Maloney, Sayed Hadi Hashemi, Benjamin Rabe, Ke Xu, Roy Campbell, Jian Peng, and William Gropp (2020). "Hal: Computer system for scalable deep learning". In: *Practice and experience in advanced research computing*, pp. 41–48.

Kingma, Diederik P and Jimmy Ba (2014). "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980*.

Koh, Pang Wei, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. (2021). "Wilds: A benchmark of in-the-wild distribution shifts". In: *International conference on machine learning*. PMLR, pp. 5637–5664.

Korbak, Tomasz, Kejian Shi, Angelica Chen, Rasika Vinayak Bhalerao, Christopher Buckley, Jason Phang, Samuel R Bowman, and Ethan Perez (2023). "Pretraining language models with human preferences". In: *International Conference on Machine Learning*. PMLR, pp. 17506–17533.

Kos, Jernej, Ian Fischer, and Dawn Song (2018). "Adversarial examples for generative models". In: *2018 ieee security and privacy workshops (spw)*. IEEE, pp. 36–42.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems* 25.

Krueger, David, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville (2021). "Out-of-distribution generalization via risk extrapolation (rex)". In: *International Conference on Machine Learning*. PMLR, pp. 5815–5826.

Kumar, Abhinav, Chenhao Tan, and Amit Sharma (2022). "Probing classifiers are unreliable for concept removal and detection". In: *Advances in Neural Information Processing Systems* 35, pp. 17994–18008.

Kung, Po-Nien and Nanyun Peng (July 2023). "Do Models Really Learn to Follow Instructions? An Empirical Study of Instruction Tuning". In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, pp. 1317–1328. DOI: 10.18653/v1/2023.acl-short.113.

Kuzmin, Gleb, Nemeesh Yadav, Ivan Smirnov, Timothy Baldwin, and Artem Shelmanov (2024). "Inference-Time Selective Debiasing". In: *arXiv preprint arXiv:2407.19345*.

Lad, Vedang, Jin Hwa Lee, Wes Gurnee, and Max Tegmark (2025). "Remarkable Robustness of LLMs: Stages of Inference?" In: *The Thirty-ninth Annual Conference on Neural*

Information Processing Systems. URL: https://openreview.net/forum?id=Wxh5Xz7NpJ.

Lamb, Tom A., Adam Davies, Alasdair Paren, Philip Torr, and Francesco Pinto (2025). "Focus On This, Not That! Steering LLMs with Adaptive Feature Specification". In: *Forty-second International Conference on Machine Learning*. URL: https://openreview.net/forum?id=rbI5mOUA8Z.

Landau, Barbara, Linda B Smith, and Susan S Jones (1988). "The importance of shape in early lexical learning". In: *Cognitive development* 3.3, pp. 299–321.

Lang, Oran, Yossi Gandelsman, Michal Yarom, Yoav Wald, Gal Elidan, Avinatan Hassidim, William T Freeman, Phillip Isola, Amir Globerson, Michal Irani, et al. (2021). "Explaining in style: Training a gan to explain a classifier in stylespace". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 693–702.

Larsen, Anders Boesen Lindbo, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther (2016). "Autoencoding beyond pixels using a learned similarity metric". In: *Proceedings of The 33rd International Conference on Machine Learning*. Ed. by Maria Florina Balcan and Kilian Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, pp. 1558–1566. URL: https://proceedings.mlr.press/v48/larsen16.html.

Lasri, Karim, Tiago Pimentel, Alessandro Lenci, Thierry Poibeau, and Ryan Cotterell (May 2022). "Probing for the Usage of Grammatical Number". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, pp. 8818–8831. DOI: 10.18653/v1/2022.acl-long.603.

Laurençon, Hugo, Lucile Saulnier, Léo Tronchon, Stas Bekman, Amanpreet Singh, Anton Lozhkov, Thomas Wang, Siddharth Karamcheti, Alexander Rush, Douwe Kiela, et al. (2024). "Obelics: An open web-scale filtered dataset of interleaved image-text documents". In: *Advances in Neural Information Processing Systems* 36.

Lee, Honglak, Alexis Battle, Rajat Raina, and Andrew Ng (2006). "Efficient sparse coding algorithms". In: *Advances in Neural Information Processing Systems*. Ed. by B. Schölkopf, J. Platt, and T. Hoffman. Vol. 19. MIT Press. URL: https://proceedings.neurips.cc/paper_files/paper/2006/file/2d71b2ae158c7c5912cc0bbde2bb9d95-Paper.pdf.

Lee, Sewoong, Adam Davies, Marc E. Canby, and Julia Hockenmaier (2025). "Evaluating and Designing Sparse Autoencoders by Approximating Quasi-Orthogonality". In: *Second Conference on Language Modeling*. URL: https://openreview.net/forum?id=XhdNFeMclS.

Lepori, Michael, Thomas Serre, and Ellie Pavlick (2023). "Break it down: Evidence for structural compositionality in neural networks". In: *Advances in Neural Information Processing Systems* 36, pp. 42623–42660.

Lermen, Simon and Charlie Rogers-Smith (2024). "LoRA Fine-tuning Efficiently Undoes Safety Training in Llama 2-Chat 70B". In: *ICLR 2024 Workshop on Secure and Trustworthy Large Language Models*. URL: https://openreview.net/forum?id=Y52UbVhglu.

Lewicki, Michael S. and Terrence J. Sejnowski (Feb. 2000). "Learning Overcomplete Representations". In: *Neural Computation* 12.2, pp. 337–365. ISSN: 0899-7667. DOI: 10.1162/089976600300015826. eprint: https://direct.mit.edu/neco/article-pdf/12/2/337/814391/089976600300015826.pdf.

Li, Bo, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Fanyi Pu, Jingkang Yang, Chunyuan Li, and Ziwei Liu (2023a). "Mimic-it: Multi-modal in-context instruction tuning". In: *arXiv preprint arXiv:2306.05425*.

Li, Da, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales (2017). "Deeper, Broader and Artier Domain Generalization". In: *The International Conference on Computer Vision (ICCV 2017)*, pp. 5543–5551. ISBN: 978-1-5386-1033-6.

Li, Junnan, Dongxu Li, Silvio Savarese, and Steven Hoi (2023b). "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models". In: *International conference on machine learning*. PMLR, pp. 19730–19742.

Li, Junnan, Dongxu Li, Caiming Xiong, and Steven Hoi (2022). "Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation". In: *International conference on machine learning*. PMLR, pp. 12888–12900.

Li, Kenneth, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg (2023c). "Inference-time intervention: Eliciting truthful answers from a language model". In: *Advances in Neural Information Processing Systems* 36, pp. 41451–41530.

Li, Victoria R, Jenny Kaufmann, Martin Wattenberg, David Alvarez-Melis, and Naomi Saphra (2025). "Can Interpretation Predict Behavior on Unseen Data?" In: *arXiv preprint arXiv:2507.06445*.

Li, Xiang Lisa and Percy Liang (Aug. 2021). "Prefix-Tuning: Optimizing Continuous Prompts for Generation". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, pp. 4582–4597. DOI: 10.18653/v1/2021.acl-long.353.

Li, Yingwei, Qihang Yu, Mingxing Tan, Jieru Mei, Peng Tang, Wei Shen, Alan Yuille, and Cihang Xie (2020). "Shape-Texture Debiased Neural Network Training". In: *arXiv preprint arXiv:2010.05981*.

Liao, Q Vera and Jennifer Wortman Vaughan (2023). "AI Transparency in the Age of LLMs: A Human-Centered Research Roadmap". In: *arXiv preprint arXiv:2306.01941*.

Lin, Bill Yuchen, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren (Nov. 2020). "CommonGen: A Constrained Text Generation Challenge for Generative Commonsense Reasoning". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, pp. 1823–1840. URL: https://www.aclweb.org/anthology/2020.findings-emnlp.165.

Lin, Bin, Zhenyu Tang, Yang Ye, Jiaxi Cui, Bin Zhu, Peng Jin, Jinfa Huang, Junwu Zhang, Munan Ning, and Li Yuan (2024). *MoE-LLaVA: Mixture of Experts for Large Vision-Language Models*. arXiv: 2401.15947 [cs.CV].

Linzen, Tal, Emmanuel Dupoux, and Yoav Goldberg (Dec. 2016). "Assessing the Ability of LSTMs to Learn Syntax-Sensitive Dependencies". en. In: *Transactions of the Association for Computational Linguistics* 4, pp. 521–535. ISSN: 2307-387X. DOI: 10.1162/tacl_a_00115.

Lipton, Zachary C (2018). "The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery." In: *Queue* 16.3, pp. 31–57.

Liu, Bingbin, Jordan T. Ash, Surbhi Goel, Akshay Krishnamurthy, and Cyril Zhang (2023a). *Exposing Attention Glitches with Flip-Flop Language Modeling*. arXiv: 2306.00946 [cs.LG]. URL: https://arxiv.org/abs/2306.00946.

Liu, Haotian, Chunyuan Li, Yuheng Li, and Yong Jae Lee (2024a). *Improved Baselines with Visual Instruction Tuning*. arXiv: 2310.03744 [cs.CV].

Liu, Haotian, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee (Jan. 2024b). *LLaVA-NeXT: Improved reasoning, OCR, and world knowledge*. URL: https://llava-vl.github.io/blog/2024-01-30-llava-next/.

Liu, Haotian, Chunyuan Li, Qingyang Wu, and Yong Jae Lee (2024c). "Visual instruction tuning". In: *Advances in neural information processing systems* 36.

Liu, Jiachang, Dinghan Shen, Yizhe Zhang, William B Dolan, Lawrence Carin, and Weizhu Chen (2022a). "What Makes Good In-Context Examples for GPT-3?" In: *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pp. 100–114.

Liu, Nelson F., Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith (June 2019a). "Linguistic Knowledge and Transferability of Contextual Representations".

In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 1073–1094. DOI: 10.18653/v1/N19-1112.

Liu, Pengfei, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig (Jan. 2023b). "Pre-Train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing". In: *ACM Comput. Surv.* 55.9. ISSN: 0360-0300. DOI: 10.1145/3560815.

Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov (2019b). "Roberta: A robustly optimized bert pretraining approach". In: *arXiv preprint arXiv:1907.11692*.

Liu, Zeyu, Yizhong Wang, Jungo Kasai, Hannaneh Hajishirzi, and Noah A. Smith (Nov. 2021). "Probing Across Time: What Does RoBERTa Know and When?" In: *Findings of the Association for Computational Linguistics: EMNLP 2021*. Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 820–842. DOI: 10.18653/v1/2021.findings-emnlp.71.

Liu, Zhuang, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie (2022b). "A ConvNet for the 2020s". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Longpre, Shayne, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. (2023). "The flan collection: Designing data and methods for effective instruction tuning". In: *International Conference on Machine Learning*. PMLR, pp. 22631–22648.

Loshchilov, Ilya and Frank Hutter (2019). "Decoupled Weight Decay Regularization". In: *International Conference on Learning Representations*.

Lundberg, Scott M. and Su In Lee (2017). "A unified approach to interpreting model predictions". In: *Advances in Neural Information Processing Systems*. arXiv: 1705.07874.

Lv, Fangrui, Jian Liang, Shuang Li, Bin Zang, Chi Harold Liu, Ziteng Wang, and Di Liu (2022). "Causality Inspired Representation Learning for Domain Generalization". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8046–8056.

Lyons, John (1977). *Semantics: Volume 2*. Vol. 2. Cambridge university press.

Madry, Aleksander, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu (2017). "Towards deep learning models resistant to adversarial attacks". In: *arXiv preprint arXiv:1706.06083*.

Mahabadi, Rabeeh Karimi, Yonatan Belinkov, and James Henderson (2020). "End-to-End Bias Mitigation by Modelling Biases in Corpora". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8706–8716.

Mahowald, Kyle, Anna A Ivanova, Idan A Blank, Nancy Kanwisher, Joshua B Tenenbaum, and Evelina Fedorenko (2024). "Dissociating language and thought in large language models". In: *Trends in cognitive sciences* 28.6, pp. 517–540.

Mandler, George (2002). "Origins of the cognitive (r) evolution". In: *Journal of the History of the Behavioral Sciences* 38.4, pp. 339–353.

Manikandan, Hariharan, Yiding Jiang, and J Zico Kolter (2023). "Language models are weak learners". In: *Advances in Neural Information Processing Systems* 36, pp. 50907–50931.

Marconi, D. (1997). *Lexical Competence*. A Bradford book. Bradford Book. ISBN: 9780262133333. URL: https://books.google.com/books?id=lcrEq_7o5m0C.

Marconi, Diego (2020). "Semantic competence". In: *The Routledge Handbook of Philosophy of Skill And Expertise*. Routledge, pp. 409–418.

Marks, Samuel, Can Rager, Eric J Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller (2024). "Sparse feature circuits: Discovering and editing interpretable causal graphs in language models". In: *arXiv preprint arXiv:2403.19647*.

Marks, Samuel and Max Tegmark (2023). "The geometry of truth: Emergent linear structure in large language model representations of true/false datasets". In: *arXiv preprint arXiv:2310.06824*.

Marr, David (1982). *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. San Francisco: W. H. Freeman.

Marshall, Simon C and Jan H Kirchner (2024). "Understanding polysemanticity in neural networks through coding theory". In: *arXiv preprint arXiv:2401.17975*.

McCoy, Tom, Ellie Pavlick, and Tal Linzen (July 2019). "Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Ed. by Anna Korhonen, David Traum, and Lluís Màrquez. Florence, Italy: Association for Computational Linguistics, pp. 3428–3448. DOI: 10.18653/v1/P19-1334.

McCurdy, Kate, Paul Soulos, Paul Smolensky, Roland Fernandez, and Jianfeng Gao (2024). "Toward compositional behavior in neural models: A survey of current views". In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 9323–9339.

McLachlan, Geoffrey J (2005). *Discriminant analysis and statistical pattern recognition*. John Wiley & Sons.

Meng, Chenlin, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon (2022a). "SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=aBsCjcPu_tE.

Meng, Kevin, David Bau, Alex Andonian, and Yonatan Belinkov (2022b). "Locating and Editing Factual Associations in GPT". In: *Advances in Neural Information Processing Systems* 36.

Meng, Kevin, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau (2023). "Mass-Editing Memory in a Transformer". In: *The Eleventh International Conference on Learning Representations*. URL: https://openreview.net/forum?id=MkbcAHIYgyS.

Merullo, Jack, Carsten Eickhoff, and Ellie Pavlick (2024). *A Mechanism for Solving Relational Tasks in Transformer Language Models*. URL: https://openreview.net/forum?id=ZmzLrl8nTa.

Microsoft (2020). *Diversity, inclusion, and responsible AI are now the bedrock of bias prevention*. Retrieved November 18, 2024. URL: https://www.microsoft.com/en-us/industry/microsoft-in-business/business-transformation/2020/09/10/diversity-inclusion-and-responsible-ai-are-now-the-bedrock-of-bias-prevention/.

Mikolov, Tomáš, Wen-tau Yih, and Geoffrey Zweig (2013). "Linguistic regularities in continuous space word representations". In: *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pp. 746–751.

Miller, George A (1956). "The magical number seven, plus or minus two: Some limits on our capacity for processing information." In: *Psychological review* 63.2, p. 81.

Miller, George A (2003). "The cognitive revolution: a historical perspective". In: *Trends in cognitive sciences* 7.3, pp. 141–144.

Min, Sewon, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer (May 2022). "Noisy Channel Language Model Prompting for Few-Shot Text Classification". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, pp. 5316–5330. DOI: 10.18653/v1/2022.acl-long.365.

Ming, Yifei, Hang Yin, and Yixuan Li (2022). "On the impact of spurious correlation for out-of-distribution detection". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 36, pp. 10051–10059.

Mishra, Swaroop, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi (2022). "Cross-Task Generalization via Natural Language Crowdsourcing Instructions". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3470–3487.

Mitra, Arindam, Luciano Del Corro, Shweti Mahajan, Andrés Codas, Clarisse Simões, Sahaj Agrawal, Xuxi Chen, Anastasia Razdaibiedina, Erik Jones, Kriti Aggarwal, et al. (2023). "Orca 2: Teaching Small Language Models How to Reason". In: *CoRR*.

Mizrahi, Moran, Guy Kaplan, Dan Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky (2024). "State of what art? a call for multi-prompt llm evaluation". In: *Transactions of the Association for Computational Linguistics* 12, pp. 933–949.

Moradi, Milad and Matthias Samwald (2021). "Evaluating the robustness of neural language models to input perturbations". In: *arXiv preprint arXiv:2108.12237*.

Mosqueira-Rey, Eduardo, Elena Hernández-Pereira, David Alonso-Ríos, José Bobes-Bascarán, and Ángel Fernández-Leal (2022). "Human-in-the-loop machine learning: A state of the art". In: *Artificial Intelligence Review*, pp. 1–50.

Mu, Jesse and Jacob Andreas (2020). "Compositional explanations of neurons". In: *Advances in Neural Information Processing Systems* 33, pp. 17153–17163.

Mueller, Aaron (2024). "Missed Causes and Ambiguous Effects: Counterfactuals Pose Challenges for Interpreting Neural Networks". In: *ICML 2024 Workshop on Mechanistic Interpretability*. URL: https://openreview.net/forum?id=pJs3ZiKBM5.

Mueller, Aaron, Jannik Brinkmann, Millicent Li, Samuel Marks, Koyena Pal, Nikhil Prakash, Can Rager, Aruna Sankaranarayanan, Arnab Sen Sharma, Jiuding Sun, et al. (2024). "The quest for the right mediator: A history, survey, and theoretical grounding of causal interpretability". In: *arXiv preprint arXiv:2408.01416*.

Mueller, Aaron, Atticus Geiger, Sarah Wiegreffe, Dana Arad, Iván Arcuschin, Adam Belfki, Yik Siu Chan, Jaden Fried Fiotto-Kaufman, Tal Haklay, Michael Hanna, et al. (2025). "MIB: A Mechanistic Interpretability Benchmark". In: *International Conference on Machine Learning*. PMLR, pp. 45069–45108.

Nanda, Neel, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt (2023a). "Progress measures for grokking via mechanistic interpretability". In: *The Eleventh International Conference on Learning Representations*. URL: https://openreview.net/forum?id=9XFSbDPmdW.

Nanda, Neel, Andrew Lee, and Martin Wattenberg (Dec. 2023b). "Emergent Linear Representations in World Models of Self-Supervised Sequence Models". In: *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*. Ed. by Yonatan Belinkov, Sophie Hao, Jaap Jumelet, Najoung Kim, Arya McCarthy, and

Hosein Mohebbi. Singapore: Association for Computational Linguistics, pp. 16–30. DOI: 10.18653/v1/2023.blackboxnlp-1.2.

Nath, Vaskar, Pranav Raja, Claire Yoon, and Sean Hendryx (2025). "Toolcomp: A multi-tool reasoning & process supervision benchmark". In: *arXiv preprint arXiv:2501.01290*.

Newmeyer, Frederick J (2001). "The Prague School and North American functionalist approaches to syntax". In: *Journal of Linguistics* 37.1, pp. 101–126.

Nguyen, Anh, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune (2016a). "Synthesizing the preferred inputs for neurons in neural networks via deep generator networks". In: *Advances in neural information processing systems* 29.

Nguyen, Anh, Jason Yosinski, and Jeff Clune (2016b). "Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks". In: *arXiv preprint arXiv:1602.03616*.

Nguyen, Quang, Truong Vu, Anh Tran, and Khoi Nguyen (2023). "Dataset Diffusion: Diffusion-based Synthetic Data Generation for Pixel-Level Semantic Segmentation". In: *Advances in Neural Information Processing Systems*. Vol. 36, pp. 76872–76892.

Nichol, Alex, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen (2021). "GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models". In: *CoRR* abs/2112.10741. arXiv: 2112.10741. URL: https://arxiv.org/abs/2112.10741.

Niu, Jingcheng, Wenjie Lu, and Gerald Penn (2022). "Does BERT Rediscover a Classical NLP Pipeline?" In: *Proceedings of the 29th International Conference on Computational Linguistics*, pp. 3143–3153.

Niv, Yael and Angela Langdon (2016). "Reinforcement learning with Marr". In: *Current opinion in behavioral sciences* 11, pp. 67–73.

Olah, Chris (2022). "Mechanistic interpretability, variables, and the importance of interpretable bases". In: *Transformer Circuits Thread* 2.4. URL: https://transformer-circuits.pub/2022/mech-interp-essay/index.html.

Olah, Chris, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter (2020). "Zoom in: An introduction to circuits". In: *Distill* 5.3, e00024–001.

Olah, Chris, Alexander Mordvintsev, and Ludwig Schubert (2017). "Feature visualization". In: *Distill* 2.11, e7.

Olah, Christopher and Adam Jermyn (2024). "What is a linear representation? What is a multidimensional feature?" In: *Transformer Circuits Thread*. URL: https://transformer-circuits.pub/2024/july-update/index.html#linear-representations.

Olsson, Catherine, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. (2022). "In-context learning and induction heads". In: *arXiv preprint arXiv:2209.11895*.

Ommer, Björn (2013). "The role of shape in visual recognition". In: *Shape Perception in Human and Computer Vision: An Interdisciplinary Perspective*, pp. 373–385.

OpenAI (2023). "GPT-4 Technical Report". In: *arXiv preprint arXiv:2303.08774*.

OpenAI (2024). *Evaluating Fairness in ChatGPT*. OpenAI. Retrieved November 18, 2024. URL: https://openai.com/index/evaluating-fairness-in-chatgpt/.

Opper, Mattia, Roland Fernandez, Paul Smolensky, and Jianfeng Gao (2025). "TRA: Better Length Generalisation with Threshold Relative Attention". In: *Transactions on Machine Learning Research*. ISSN: 2835-8856. URL: https://openreview.net/forum?id=yNiBUc2hMW.

Ouyang, Cheng, Chen Chen, Surui Li, Zeju Li, Chen Qin, Wenjia Bai, and Daniel Rueckert (2022a). "Causality-inspired single-source domain generalization for medical image segmentation". In: *IEEE Transactions on Medical Imaging* 42.4, pp. 1095–1106.

Ouyang, Long, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. (2022b). "Training language models to follow instructions with human feedback". In: *Advances in Neural Information Processing Systems* 35, pp. 27730–27744.

Parihar, Rishubh, Abhijnya Bhat, Abhipsa Basu, Saswat Mallick, Jogendra Nath Kundu, and R Venkatesh Babu (2024). "Balancing act: distribution-guided debiasing in diffusion models". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6668–6678.

Park, Kiho, Yo Joong Choe, Yibo Jiang, and Victor Veitch (2024a). "The Geometry of Categorical and Hierarchical Concepts in Large Language Models". In: *arXiv preprint arXiv:2406.01506*.

Park, Kiho, Yo Joong Choe, and Victor Veitch (2024b). "The Linear Representation Hypothesis and the Geometry of Large Language Models". In: *Forty-first International Conference on Machine Learning*. URL: https://openreview.net/forum?id=UGpGkLzwpP.

Parrish, Alicia, Angelica Chen, Nikita Nangia, Vishakh Padmakumar, Jason Phang, Jana Thompson, Phu Mon Htut, and Samuel Bowman (May 2022). "BBQ: A hand-built bias benchmark for question answering". In: *Findings of the Association for Computational Linguistics: ACL 2022*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Dublin, Ireland: Association for Computational Linguistics, pp. 2086–2105. DOI: 10.18653/v1/2022.findings-acl.165.

Partee, Barbara et al. (1995). "Lexical semantics and compositionality". In: *An invitation to cognitive science: Language* 1, pp. 311–360.

Paszke, A (2019). "Pytorch: An imperative style, high-performance deep learning library". In: *arXiv preprint arXiv:1912.01703*.

Patil, Shishir G, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E Gonzalez (2025). "The berkeley function calling leaderboard (bfcl): From tool use to agentic evaluation of large language models". In: *Forty-second International Conference on Machine Learning*.

Paulo, Gonçalo Santos, Alex Troy Mallen, Caden Juang, and Nora Belrose (2025). "Automatically Interpreting Millions of Features in Large Language Models". In: *Forty-second International Conference on Machine Learning*. URL: https://openreview.net/forum?id=EemtbhJOXc.

Pavlick, Ellie (2023). "Symbols and grounding in large language models". In: *Philosophical Transactions of the Royal Society A* 381.2251, p. 20220041.

Pearl, Judea (2009). "Causal inference in statistics: An overview". In: *Statistics Surveys* 3.

Peng, Baolin, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao (2023). "Instruction tuning with gpt-4". In: *arXiv preprint arXiv:2304.03277*.

Peng, Xingchao, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang (2019). "Moment matching for multi-source domain adaptation". In: *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1406–1415.

Pennington, Jeffrey, Richard Socher, and Christopher D Manning (2014). "Glove: Global vectors for word representation". In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543.

Perez, Ethan, Sam Ringer, Kamile Lukosiute, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, et al. (2023). "Discovering Language Model Behaviors with Model-Written Evaluations". In: *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13387–13434.

Pérez, Jorge, Pablo Barceló, and Javier Marinkovic (2021). "Attention is Turing-Complete". In: *Journal of Machine Learning Research* 22.75, pp. 1–35. URL: http://jmlr.org/papers/v22/20-302.html.

Petch, Jeremy, Shuang Di, and Walter Nelson (2022). "Opening the black box: the promise and limitations of explainable machine learning in cardiology". In: *Canadian Journal of Cardiology* 38.2, pp. 204–213.

Peters, Jonas, Peter Bühlmann, and Nicolai Meinshausen (2016). "Causal inference by using invariant prediction: identification and confidence intervals". In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 78.5, pp. 947–1012.

Petroni, Fabio, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller (Nov. 2019). "Language Models as Knowledge Bases?" In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 2463–2473. DOI: 10.18653/v1/D19-1250.

Pimentel, Tiago, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell (July 2020). "Information-Theoretic Probing for Linguistic Structure". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 4609–4622. DOI: 10.18653/v1/2020.acl-main.420.

Pimentel, Tiago, Josef Valvoda, Niklas Stoehr, and Ryan Cotterell (2022). "The Architectural Bottleneck Principle". In: *arXiv preprint arXiv:2211.06420*. URL: https://doi.org/10.48550/arXiv.2211.06420.

Pinto, Francesco, Philip HS Torr, and Puneet K. Dokania (2022a). "An impartial take to the cnn vs transformer robustness contest". In: *European conference on computer vision*. Springer, pp. 466–480.

Pinto, Francesco, Harry Yang, Ser-Nam Lim, Philip Torr, and Puneet K. Dokania (2022b). "Using Mixup as a Regularizer Can Surprisingly Improve Accuracy & Out-of-Distribution Robustness". In: *Advances in Neural Information Processing Systems*. Ed. by Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho. URL: https://openreview.net/forum?id=5j6fWcPccO.

Preetham, Freedom (2024). *Why I Hate The Term "Overcomplete Basis" Used By AI Researchers — medium.com*. https://medium.com/mathematical-musings/why-i-hate-the-term-overcomplete-basis-used-by-ai-researchers-865a6e7b2866. [Accessed 22-10-2025].

Qi, Xiangyu, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson (2024). "Fine-tuning Aligned Language Models Compromises Safety, Even When Users Do Not Intend To!" In: *ICLR*.

Qiao, Fengchun, Long Zhao, and Xi Peng (2020). "Learning to learn single domain generalization". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12556–12565.

Quiñonero-Candela, Joaquin, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence (2022). *Dataset shift in machine learning*. Mit Press.

Quiroga, Rodrigo (2013). "Gnostic cells in the 21st century". In: *Acta Neurobiologiae Experimentalis* 73.4, pp. 463–471.

Radford, Alec, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. (2021). "Learning transferable visual models from natural language supervision". In: *International conference on machine learning*. PMLR, pp. 8748–8763.

Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. (2019). "Language models are unsupervised multitask learners". In: *OpenAI blog* 1.8, p. 9.

Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. (2020). "Exploring the limits of transfer learning with a unified text-to-text transformer." In: *J. Mach. Learn. Res.* 21.140, pp. 1–67.

Raji, Inioluwa Deborah, Emily Denton, Emily M. Bender, Alex Hanna, and Amandalynne Paullada (2021). "AI and the Everything in the Whole Wide World Benchmark". In: *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*. URL: https://openreview.net/forum?id=j6NxpQbREA1.

Ramesh, Aditya, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever (2021). "Zero-shot text-to-image generation". In: *International Conference on Machine Learning*. PMLR, pp. 8821–8831.

Ramos, Gonzalo, Christopher Meek, Patrice Simard, Jina Suh, and Soroush Ghorashi (2020). "Interactive machine teaching: a human-centered approach to building machine-learned models". In: *Human–Computer Interaction* 35.5-6, pp. 413–451.

Raven, John (2000). "The Raven's progressive matrices: change and stability over culture and time". In: *Cognitive psychology* 41.1, pp. 1–48.

Ravfogel, Shauli, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg (July 2020). "Null It Out: Guarding Protected Attributes by Iterative Nullspace Projection". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 7237–7256. DOI: 10.18653/v1/2020.acl-main.647.

Ravfogel, Shauli, Yoav Goldberg, and Ryan Cotterell (July 2023). "Log-linear Guardedness and its Implications". In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, pp. 9413–9431. DOI: 10.18653/v1/2023.acl-long.523.

Ravfogel, Shauli, Grusha Prasad, Tal Linzen, and Yoav Goldberg (Nov. 2021). "Counterfactual Interventions Reveal the Causal Effect of Relative Clause Representations on Agreement Prediction". In: *Proceedings of the 25th Conference on Computational Natural Language Learning*. Ed. by Arianna Bisazza and Omri Abend.

Online: Association for Computational Linguistics, pp. 194–209. DOI: 10.18653/v1/2021.conll-1.15.

Ravfogel, Shauli, Michael Twiton, Yoav Goldberg, and Ryan D Cotterell (2022a). "Linear Adversarial Concept Erasure". In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato. Vol. 162. Proceedings of Machine Learning Research. PMLR, pp. 18400–18421. URL: https://proceedings.mlr.press/v162/ravfogel22a.html.

Ravfogel, Shauli, Francisco Vargas, Yoav Goldberg, and Ryan Cotterell (Dec. 2022b). "Adversarial Concept Erasure in Kernel Space". In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, pp. 6034–6055. URL: https://aclanthology.org/2022.emnlp-main.405.

Ravichander, Abhilasha, Eduard Hovy, Kaheer Suleman, Adam Trischler, and Jackie Chi Kit Cheung (Dec. 2020). "On the Systematicity of Probing Contextualized Word Representations: The Case of Hypernymy in BERT". In: *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics*. Barcelona, Spain (Online): Association for Computational Linguistics, pp. 88–102. URL: https://aclanthology.org/2020.starsem-1.10.

Reddy, Sandeep (2022). "Explainability and artificial intelligence in medicine". In: *The Lancet Digital Health* 4.4, e214–e215.

Ren, Mengjie, Boxi Cao, Hongyu Lin, Cao Liu, Xianpei Han, Ke Zeng, Wan Guanglu, Xunliang Cai, and Le Sun (2024). "Learning or Self-aligning? Rethinking Instruction Fine-tuning". In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6090–6105.

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin (Aug. 2016). ""Why should i trust you?" Explaining the predictions of any classifier". In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Vol. 13-17-August-2016. New York, New York, USA: Association for Computing Machinery, pp. 1135–1144. ISBN: 9781450342322. DOI: 10.1145/2939672.2939778. arXiv: 1602.04938.

Ribeiro, Marco Tulio, Tongshuang Wu, Carlos Guestrin, and Sameer Singh (2020). "Beyond accuracy: Behavioral testing of NLP models with CheckList". In: *arXiv preprint arXiv:2005.04118*.

Ritter, Samuel, David GT Barrett, Adam Santoro, and Matt M Botvinick (2017). "Cognitive psychology for deep neural networks: A shape bias case study". In: *International conference on machine learning*. PMLR, pp. 2940–2949.

Rogers, Anna, Olga Kovaleva, and Anna Rumshisky (2021). "A primer in BERTology: What we know about how BERT works". In: *Transactions of the Association for Computational Linguistics* 8, pp. 842–866.

Rombach, Robin, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer (2022). "High-resolution image synthesis with latent diffusion models". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695.

Rosenfeld, Elan, Pradeep Ravikumar, and Andrej Risteski (2020). "The risks of invariant risk minimization". In: *arXiv preprint arXiv:2010.05761*.

Ruder, Sebastian (2017). "An overview of multi-task learning in deep neural networks". In: *arXiv preprint arXiv:1706.05098*.

Russin, Jacob, Sam Whitman McGrath, Danielle J Williams, and Lotem Elber-Dorozko (2024). "From Frege to chatGPT: Compositionality in language, cognition, and deep neural networks". In: *arXiv preprint arXiv:2405.15164*.

Sag, Ivan A and Thomas Wasow (2011). "Performance-Compatible Competence Grammar". In: *Non-Transformational Syntax: Formal and Explicit Models of Grammar*, pp. 359–377.

Sagawa, Shiori, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang (2019). "Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization". In: *arXiv preprint arXiv:1911.08731*.

Sajjad, Hassan, Fahim Dalvi, Nadir Durrani, and Preslav Nakov (2023). "On the effect of dropping layers of pre-trained transformer models". In: *Computer Speech & Language* 77, p. 101429.

Sakaridis, Christos, Dengxin Dai, and Luc Van Gool (2020). "Map-Guided Curriculum Domain Adaptation and Uncertainty-Aware Evaluation for Semantic Nighttime Image Segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*. DOI: 10.1109/TPAMI.2020.3045882.

Saphra, Naomi and Sarah Wiegreffe (Nov. 2024). "Mechanistic?" In: *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*. Ed. by Yonatan Belinkov, Najoung Kim, Jaap Jumelet, Hosein Mohebbi, Aaron Mueller, and Hanjie Chen. Miami, Florida, US: Association for Computational Linguistics, pp. 480–498. DOI: 10.18653/v1/2024.blackboxnlp-1.30.

Sariyildiz, Mert Bulent, Karteek Alahari, Diane Larlus, and Yannis Kalantidis (2023). "Fake it till you make it: Learning transferable representations from synthetic ImageNet clones".

In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Savarese, Pedro, Hugo Silva, and Michael Maire (2020). "Winning the lottery with continuous sparsification". In: *Advances in neural information processing systems* 33, pp. 11380–11390.

Sawant, Shriraj P and Shruti Singh (2020). "Understanding attention: in minds and machines". In: *arXiv preprint arXiv:2012.02659*.

Scherlis, Adam, Kshitij Sachan, Adam S Jermyn, Joe Benton, and Buck Shlegeris (2022). "Polysemanticity and capacity in neural networks". In: *arXiv preprint arXiv:2210.01892*.

Schick, Timo and Hinrich Schütze (2021). "It's Not Just Size That Matters: Small Language Models Are Also Few-Shot Learners". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2339–2352.

Schölkopf, Bernhard, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio (2021). "Toward causal representation learning". In: *Proceedings of the IEEE* 109.5, pp. 612–634.

Schuhmann, Christoph, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. (2022). "LAION-5B: An open large-scale dataset for training next generation image-text models". In: *arXiv preprint arXiv:2210.08402*.

Schwettmann, Sarah, Neil Chowdhury, Samuel Klein, David Bau, and Antonio Torralba (2023). "Multimodal neurons in pretrained text-only transformers". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2862–2867.

Schwettmann, Sarah, Evan Hernandez, David Bau, Samuel Klein, Jacob Andreas, and Antonio Torralba (2021). "Toward a visual concept vocabulary for gan latent space". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6804–6812.

Selvaraju, Ramprasaath R, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra (2017). "Grad-cam: Visual explanations from deep networks via gradient-based localization". In: *Proceedings of the IEEE international conference on computer vision*, pp. 618–626.

Sennrich, Rico, Barry Haddow, and Alexandra Birch (2016). "Improving Neural Machine Translation Models with Monolingual Data". In: *54th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics (ACL), pp. 86–96.

Shao, Shun, Yftah Ziser, and Shay B. Cohen (2022). *Gold Doesn't Always Glitter: Spectral Removal of Linear and Nonlinear Guarded Attribute Information*. DOI: 10.48550/ARXIV.2203.07893.

Sharma, Mrinank, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R Bowman, Esin DURMUS, Zac Hatfield-Dodds, Scott R Johnston, Shauna M Kravec, et al. (2024). "Towards Understanding Sycophancy in Language Models". In: *The Twelfth International Conference on Learning Representations*.

Shi, Claudia, Nicolas Beltran-Velez, Achille Nazaret, Carolina Zheng, Adrià Garriga-Alonso, Andrew Jesson, Maggie Makar, and David Blei (2024). "Hypothesis Testing the Circuit Hypothesis in LLMs". In: *ICML 2024 Workshop on Mechanistic Interpretability*.

Shin, Donghee (2021). "The effects of explainability and causability on perception, trust, and acceptance: Implications for explainable AI". In: *International Journal of Human-Computer Studies* 146, p. 102551.

Shojaee, Parshin, Iman Mirzadeh, Keivan Alizadeh, Maxwell Horton, Samy Bengio, and Mehrdad Farajtabar (2025). *The Illusion of Thinking: Understanding the Strengths and Limitations of Reasoning Models via the Lens of Problem Complexity*. arXiv: 2506.06941 [cs.AI]. URL: https://arxiv.org/abs/2506.06941.

Shorten, Connor and Taghi M Khoshgoftaar (2019). "A survey on image data augmentation for deep learning". In: *Journal of big data* 6.1, pp. 1–48.

Simonyan, Karen, Andrea Vedaldi, and Andrew Zisserman (2013). "Deep inside convolutional networks: Visualising image classification models and saliency maps". In: *arXiv preprint arXiv:1312.6034*.

Siska, Charlotte, Katerina Marazopoulou, Melissa Ailem, and James Bono (2024). "Examining the robustness of LLM evaluation to the distributional assumptions of benchmarks". In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10406–10421.

Skinner, Burrhus Frederic (1957). *Verbal behavior*. New York: Appleton-Century-Crofts.

Smolensky, Paul (1988). "On the proper treatment of connectionism". In: *Behavioral and brain sciences* 11.1, pp. 1–23.

Smolensky, Paul, Roland Fernandez, Zhenghao Herbert Zhou, Mattia Opper, Adam Davies, and Jianfeng Gao (2025). "Mechanisms of Symbol Processing for In-Context Learning in Transformer Networks". In: *Journal of Artificial Intelligence Research* 84.23. URL: https://jair.org/index.php/jair/article/view/17469.

Socher, Richard, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts (2013). "Recursive deep models for semantic

compositionality over a sentiment treebank". In: *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642.

Soulos, Paul, Henry Conklin, Mattia Opper, Paul Smolensky, Jianfeng Gao, and Roland Fernandez (2024). "Compositional generalization across distributional shifts with sparse tree operations". In: *Advances in Neural Information Processing Systems* 37, pp. 112836–112863.

Soulos, Paul, R. Thomas McCoy, Tal Linzen, and Paul Smolensky (Nov. 2020). "Discovering the Compositional Structure of Vector Representations with Role Learning Networks". In: *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*. Ed. by Afra Alishahi, Yonatan Belinkov, Grzegorz Chrupała, Dieuwke Hupkes, Yuval Pinter, and Hassan Sajjad. Online: Association for Computational Linguistics, pp. 238–254. DOI: 10.18653/v1/2020.blackboxnlp-1.23.

Speer, Robyn, Joshua Chin, and Catherine Havasi (2017). "ConceptNet 5.5: An Open Multilingual Graph of General Knowledge". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 31.

Sperry, Roger W (1993). "The impact and promise of the cognitive revolution." In: *American psychologist* 48.8, p. 878.

Subramanian, Ajay, Elena Sizikova, Najib J. Majaj, and Denis G. Pelli (2023). "Spatial-frequency channels, shape bias, and adversarial robustness". In: *Thirty-seventh Conference on Neural Information Processing Systems*. URL: https://openreview.net/forum?id=KvPwXVcslY.

Subramanian, Anant, Danish Pruthi, Harsh Jhamtani, Taylor Berg-Kirkpatrick, and Eduard Hovy (2018). "Spine: Sparse interpretable neural embeddings". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 32.

Sun, Zechen, Yisheng Xiao, Juntao Li, Yixin Ji, Wenliang Chen, and Min Zhang (2024). "Exploring and Mitigating Shortcut Learning for Generative Large Language Models". In: *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 6883–6893.

Sundararajan, Mukund and Amir Najmi (2020). "The many Shapley values for model explanation". In: *37th International Conference on Machine Learning, ICML 2020*.

Sundararajan, Mukund, Ankur Taly, and Qiqi Yan (2017). "Axiomatic attribution for deep networks". In: *34th International Conference on Machine Learning, ICML 2017*. arXiv: 1703.01365.

Sutter, Denis, Julian Minder, Thomas Hofmann, and Tiago Pimentel (2025). "The Non-Linear Representation Dilemma: Is Causal Abstraction Enough for Mechanistic

Interpretability?" In: *The Thirty-ninth Annual Conference on Neural Information Processing Systems*. URL: https://openreview.net/forum?id=ZYXTLo7kCi.

Switzky, Lawrence (2020). "ELIZA effects: Pygmalion and the early development of artificial intelligence". In: *Shaw* 40.1, pp. 50–68.

Tamang, Lakpa, Mohamed Reda Bouadjenek, Richard Dazeley, and Sunil Aryal (2025). "Handling Out-of-Distribution Data: A Survey". In: *IEEE Transactions on Knowledge and Data Engineering*.

Taori, Rohan, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto (2023). "Alpaca: A strong, replicable instruction-following model". In: *Stanford Center for Research on Foundation Models* 3.6, p. 7. URL: https://crfm.stanford.edu/2023/03/13/alpaca.html.

Taylor, J., B. Earnshaw, B. Mabey, M. Victors, and J. Yosinski (2019). "RxRx1: An Image Set for Cellular Morphological Variation Across Many Experimental Batches." In: *International Conference on Learning Representations (ICLR). AI for Social Good Workshop*.

Team, Gemini, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. (2023). "Gemini: a family of highly capable multimodal models". In: *arXiv preprint arXiv:2312.11805*.

Templeton, Adly, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan (2024). "Scaling Monosemanticity: Extracting Interpretable Features from Claude 3 Sonnet". In: *Transformer Circuits Thread*. URL: https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

Tenney, Ian, Dipanjan Das, and Ellie Pavlick (July 2019a). "BERT Rediscovers the Classical NLP Pipeline". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 4593–4601. DOI: 10.18653/v1/P19-1452.

Tenney, Ian, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R Bowman, Dipanjan Das, et al. (2019b). "What do you learn from context? probing for sentence structure in contextualized word representations". In: *arXiv preprint arXiv:1905.06316*.

Thirunavukarasu, Arun James, Darren Shu Jeng Ting, Kabilan Elangovan, Laura Gutierrez, Ting Fang Tan, and Daniel Shu Wei Ting (2023). "Large language models in medicine". In: *Nature medicine* 29.8, pp. 1930–1940.

Tigges, Curt, Oskar John Hollinsworth, Atticus Geiger, and Neel Nanda (2023). "Linear representations of sentiment in large language models". In: *arXiv preprint arXiv:2310.15154*.

Todd, Eric, Millicent Li, Arnab Sen Sharma, Aaron Mueller, Byron C Wallace, and David Bau (2024). "Function Vectors in Large Language Models". In: *The Twelfth International Conference on Learning Representations*. URL: https://openreview.net/forum?id=AwyxtyMwaG.

Tong, Shengbang, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie (2024). *Eyes Wide Shut? Exploring the Visual Shortcomings of Multimodal LLMs*. arXiv: 2401.06209 [cs.CV].

Touvron, Hugo, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. (2023a). "Llama: Open and efficient foundation language models". In: *arXiv preprint arXiv:2302.13971*.

Touvron, Hugo, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. (2023b). "Llama 2: Open foundation and fine-tuned chat models". In: *arXiv preprint arXiv:2307.09288*.

Trabucco, Brandon, Kyle Doherty, Max Gurinas, and Ruslan Salakhutdinov (2023). "Effective data augmentation with diffusion models". In: *arXiv preprint arXiv:2302.07944*.

Tucker, Mycal, Peng Qian, and Roger Levy (Aug. 2021). "What if This Modified That? Syntactic Interventions with Counterfactual Embeddings". In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Online: Association for Computational Linguistics, pp. 862–875. DOI: 10.18653/v1/2021.findings-acl.76.

Turkle, Sherry (2007). "Authenticity in the age of digital companions". In: *Interaction studies* 8.3, pp. 501–517.

Turner, Alexander Matt, Lisa Thiergart, David Udell, Gavin Leech, Ulisse Mini, and Monte MacDiarmid (2023). "Activation addition: Steering language models without optimization". In: *CoRR*.

Udomcharoenchaikit, Can, Wuttikorn Ponwitayarat, Patomporn Payoungkhamdee, Kanruethai Masuk, Weerayut Buaphet, Ekapol Chuangsuwanich, and Sarana Nutanong (2022). "Mitigating spurious correlation in natural language understanding with

counterfactual inference". In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11308–11321.

Vapnik, Vladimir Naumovich, Vlamimir Vapnik, et al. (1998). *Statistical learning theory*. wiley New York.

Vargas, Francisco and Ryan Cotterell (Nov. 2020). "Exploring the Linear Subspace Hypothesis in Gender Bias Mitigation". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu. Online: Association for Computational Linguistics, pp. 2902–2913. DOI: 10.18653/v1/2020.emnlp-main.232.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin (2017). "Attention is all you need". In: *Advances in neural information processing systems* 30.

Vegner, Ivan, Sydelle de Souza, Valentin Forch, Martha Lewis, and Leonidas AA Doumas (2025). "Behavioural vs. Representational Systematicity in End-to-End Models: An Opinionated Survey". In: *arXiv preprint arXiv:2506.04461*.

Veličković, Petar, Christos Perivolaropoulos, Federico Barbero, and Razvan Pascanu (2024). "Softmax is not Enough (for Sharp Size Generalisation)". In: *arXiv preprint arXiv:2410.01104*.

Vendrow, Joshua, Saachi Jain, Logan Engstrom, and Aleksander Madry (2023). "Dataset interfaces: Diagnosing model failures using controllable counterfactual generation". In: *arXiv preprint arXiv:2302.07865*.

Venkataramani, Rahul, Parag Dutta, Vikram Melapudi, and Ambedkar Dukkipati (2024). "Causal Feature Alignment: Learning to Ignore Spurious Background Features". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 4666–4674.

Venkateswara, Hemanth, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan (2017). "Deep hashing network for unsupervised domain adaptation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5018–5027.

Vig, Jesse and Yonatan Belinkov (Aug. 2019). "Analyzing the Structure of Attention in a Transformer Language Model". In: *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Ed. by Tal Linzen, Grzegorz Chrupała, Yonatan Belinkov, and Dieuwke Hupkes. Florence, Italy: Association for Computational Linguistics, pp. 63–76. DOI: 10.18653/v1/W19-4808.

Vig, Jesse, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Simas Sakenis, Jason Huang, Yaron Singer, and Stuart Shieber (2020a). "Causal

mediation analysis for interpreting neural nlp: The case of gender bias". In: *arXiv preprint arXiv:2004.12265*.

Vig, Jesse, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber (2020b). "Investigating gender bias in language models using causal mediation analysis". In: *Advances in neural information processing systems* 33, pp. 12388–12401.

Voita, Elena, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov (July 2019). "Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Ed. by Anna Korhonen, David Traum, and Lluís Màrquez. Florence, Italy: Association for Computational Linguistics, pp. 5797–5808. DOI: 10.18653/v1/P19-1580.

Wachter, Sandra, Brent Mittelstadt, and Chris Russell (2017). "Counterfactual explanations without opening the black box: Automated decisions and the GDPR". In: *Harv. JL & Tech.* 31, p. 841.

Wang, Jindong, Xixu Hu, Wenxin Hou, Hao Chen, Runkai Zheng, Yidong Wang, Linyi Yang, Haojun Huang, Wei Ye, Xiubo Geng, et al. (2023a). "On the robustness of chatgpt: An adversarial and out-of-distribution perspective". In: *arXiv preprint arXiv:2302.12095*.

Wang, Jindong, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip S Yu (2022a). "Generalizing to unseen domains: A survey on domain generalization". In: *IEEE transactions on knowledge and data engineering* 35.8, pp. 8052–8072.

Wang, Kevin Ro, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt (2023b). "Interpretability in the Wild: a Circuit for Indirect Object Identification in GPT-2 Small". In: *The Eleventh International Conference on Learning Representations*. URL: https://openreview.net/forum?id=NpsVSN6o4ul.

Wang, Ruoyu, Mingyang Yi, Zhitang Chen, and Shengyu Zhu (2022b). "Out-of-distribution Generalization with Causal Invariant Transformations". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 375–385.

Wang, Tianlu, Rohit Sridhar, Diyi Yang, and Xuezhi Wang (2022c). "Identifying and Mitigating Spurious Correlations for Improving Robustness in NLP Models". In: *Findings of the Association for Computational Linguistics: NAACL 2022*, pp. 1719–1729.

Wang, Weihan, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, Jiazheng Xu, Bin Xu, Juanzi Li, Yuxiao Dong, Ming Ding, and Jie Tang (2024a). *CogVLM: Visual Expert for Pretrained Language Models*. arXiv: 2311.03079 [cs.CV].

Wang, Xinyi, Antonis Antoniades, Yanai Elazar, Alfonso Amayuelas, Alon Albalak, Kexun Zhang, and William Yang Wang (2025). "Generalization v.s. Memorization: Tracing Language Models'Capabilities Back to Pretraining Data". In: *The Thirteenth International Conference on Learning Representations*. URL: https://openreview.net/forum?id=IQxBDLmVpT.

Wang, Xuezhi, Haohan Wang, and Diyi Yang (2021a). "Measure and Improve Robustness in NLP Models: A Survey". In: *arXiv preprint arXiv:2112.08313*.

Wang, Yinong Oliver, Younjoon Chung, Chen Henry Wu, and Fernando De la Torre (2024b). "Domain gap embeddings for generative dataset augmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 28684–28694.

Wang, Yizhong, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi (July 2023c). "Self-Instruct: Aligning Language Models with Self-Generated Instructions". In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, pp. 13484–13508. DOI: 10.18653/v1/2023.acl-long.754.

Wang, Zhao and Aron Culotta (2020). "Identifying Spurious Correlations for Robust Text Classification". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 3431–3440.

Wang, Zhao and Aron Culotta (2021). "Robustness to spurious correlations in text classification via automatically generated counterfactuals". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35, pp. 14024–14031.

Wang, Zhizhong, Lei Zhao, Haibo Chen, Zhiwen Zuo, Ailin Li, Wei Xing, and Dongming Lu (2021b). "Evaluate and improve the quality of neural style transfer". In: *Computer Vision and Image Understanding* 207, p. 103203.

Wang, Zhizhong, Lei Zhao, and Wei Xing (2023d). "Stylediffusion: Controllable disentangled style transfer via diffusion models". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7677–7689.

Wang, Zihao and Victor Veitch (2022). "A Unified Causal View of Domain Invariant Representation Learning". In: *ICML 2022: Workshop on Spurious Correlations, Invariance and Stability*. URL: https://openreview.net/forum?id=-l9cpeEYwJJ.

Wang, Zijian, Yadan Luo, Ruihong Qiu, Zi Huang, and Mahsa Baktashmotlagh (2021c). "Learning to diversify for single domain generalization". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 834–843.

Weizenbaum, Joseph (1966). "ELIZA—a computer program for the study of natural language communication between man and machine". In: *Communications of the ACM* 9.1, pp. 36–45.

White, Jennifer C., Tiago Pimentel, Naomi Saphra, and Ryan Cotterell (June 2021). "A Non-Linear Structural Probe". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Ed. by Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou. Online: Association for Computational Linguistics, pp. 132–138. DOI: 10.18653/v1/2021.naacl-main.12.

Williams, Adina, Nikita Nangia, and Samuel Bowman (2018). "A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122.

Winata, Genta Indra, Andrea Madotto, Zhaojiang Lin, Rosanne Liu, Jason Yosinski, and Pascale Fung (2021). "Language Models are Few-shot Multilingual Learners". In: *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pp. 1–15.

Wolf, Thomas, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. (2019). "Huggingface's transformers: State-of-the-art natural language processing". In: *arXiv preprint arXiv:1910.03771*.

Wu, Weijia, Yuzhong Zhao, Mike Zheng Shou, Hong Zhou, and Chunhua Shen (2023a). "Diffumask: Synthesizing images with pixel-level annotations for semantic segmentation using diffusion models". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1206–1217.

Wu, Yiwei, Atticus Geiger, and Raphaël Millière (2025a). "How Do Transformers Learn Variable Binding in Symbolic Programs?" In: *Forty-second International Conference on Machine Learning*. URL: https://openreview.net/forum?id=kVtyv7bpnw.

Wu, Zhengxuan, Aryaman Arora, Atticus Geiger, Zheng Wang, Jing Huang, Dan Jurafsky, Christopher D Manning, and Christopher Potts (2025b). "AxBench: Steering LLMs? Even Simple Baselines Outperform Sparse Autoencoders". In: *International Conference on Machine Learning*. PMLR, pp. 67035–67080.

Wu, Zhengxuan, Atticus Geiger, Thomas Icard, Christopher Potts, and Noah Goodman (2023b). "Interpretability at Scale: Identifying Causal Mechanisms in Alpaca". In: *Advances in Neural Information Processing Systems*. Ed. by A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine. Vol. 36. Curran Associates, Inc.,

pp. 78205–78226. URL: https://proceedings.neurips.cc/paper_files/paper/2023/file/f6a8b109d4d4fd64c75e94aaf85d9697-Paper-Conference.pdf.

Xiao, Kai, Logan Engstrom, Andrew Ilyas, and Aleksander Madry (2020). "Noise or Signal: The Role of Image Backgrounds in Object Recognition". In: *ArXiv preprint arXiv:2006.09994*.

Xu, Can, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang (2023). "Wizardlm: Empowering large language models to follow complex instructions". In: *arXiv preprint arXiv:2304.12244*.

Xu, Fei, Susan Carey, and Nina Quint (2004). "The emergence of kind-based object individuation in infancy". In: *Cognitive psychology* 49.2, pp. 155–190.

Yan, Shen, Huan Song, Nanxiang Li, Lincan Zou, and Liu Ren (2020). "Improve unsupervised domain adaptation with mixup training". In: *arXiv preprint arXiv:2001.00677*.

Yang, Karren, Abigail Katcoff, and Caroline Uhler (2018). "Characterizing and learning equivalence classes of causal DAGs under interventions". In: *International Conference on Machine Learning*. PMLR, pp. 5541–5550.

Yang, Linyi, Yaoxian Song, Xuan Ren, Chenyang Lyu, Yidong Wang, Jingming Zhuo, Lingqiao Liu, Jindong Wang, Jennifer Foster, and Yue Zhang (2023a). "Out-of-distribution generalization in natural language processing: Past, present, and future". In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4533–4559.

Yang, Linyi, Shuibai Zhang, Libo Qin, Yafu Li, Yidong Wang, Hanmeng Liu, Jindong Wang, Xing Xie, and Yue Zhang (July 2023b). "GLUE-X: Evaluating Natural Language Understanding Models from an Out-of-Distribution Generalization Perspective". In: *Findings of the Association for Computational Linguistics: ACL 2023*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, pp. 12731–12750. DOI: 10.18653/v1/2023.findings-acl.806.

Yang, Qwen An, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yi-Chao Zhang, Yunyang Wan, Yuqi Liu, Zeyu Cui, Zhenru Zhang, Zihan Qiu, Shanghaoran Quan, and Zekun Wang

(2024a). “Qwen2.5 Technical Report”. In: *ArXiv* abs/2412.15115. URL: https://api.semanticscholar.org/CorpusID:274859421.

Yang, Xianjun, Xiao Wang, Qi Zhang, Linda Ruth Petzold, William Yang Wang, Xun Zhao, and Dahua Lin (2024b). “Shadow Alignment: The Ease of Subverting Safely-Aligned Language Models”. In: *ICLR 2024 Workshop on Secure and Trustworthy Large Language Models*. URL: https://openreview.net/forum?id=9qymw6T9Oo.

Yang, Yukang, Declan Iain Campbell, Kaixuan Huang, Mengdi Wang, Jonathan D. Cohen, and Taylor Whittington Webb (2025). “Emergent Symbolic Mechanisms Support Abstract Reasoning in Large Language Models”. In: *Forty-second International Conference on Machine Learning*. URL: https://openreview.net/forum?id=y1SnRPDWx4.

Ye, Tianzhu, Li Dong, Yuqing Xia, Yutao Sun, Yi Zhu, Gao Huang, and Furu Wei (2025). “Differential Transformer”. In: *The Thirteenth International Conference on Learning Representations*. URL: https://openreview.net/forum?id=OvoCm1gGhN.

Ye, Wenqian, Guangtao Zheng, Xu Cao, Yunsheng Ma, Xia Hu, and Aidong Zhang (2024). “Spurious correlations in machine learning: A survey”. In: *arXiv preprint arXiv:2402.12715*.

Yin, Kayo and Jacob Steinhardt (2025). “Which Attention Heads Matter for In-Context Learning?” In: *Forty-second International Conference on Machine Learning*. URL: https://openreview.net/forum?id=C7XmEByCFv.

Yu, Runpeng, Songhua Liu, Xingyi Yang, and Xinchao Wang (June 2023). “Distribution Shift Inversion for Out-of-Distribution Prediction”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3592–3602.

Yuan, Lifan, Yangyi Chen, Ganqu Cui, Hongcheng Gao, FangYuan Zou, Xingyi Cheng, Heng Ji, Zhiyuan Liu, and Maosong Sun (2023). “Revisiting Out-of-distribution Robustness in NLP: Benchmarks, Analysis, and LLMs Evaluations”. In: *Advances in Neural Information Processing Systems*. Ed. by A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine. Vol. 36. Curran Associates, Inc., pp. 58478–58507. URL: https://proceedings.neurips.cc/paper_files/paper/2023/file/b6b5f50a2001ad1cbccca96e693c4ab4-Paper-Datasets_and_Benchmarks.pdf.

Yuan, Yu, Lili Zhao, Kai Zhang, Guangting Zheng, and Qi Liu (Nov. 2024). “Do LLMs Overcome Shortcut Learning? An Evaluation of Shortcut Challenges in Large Language Models”. In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Ed. by Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen. Miami, Florida, USA: Association for Computational Linguistics, pp. 12188–12200. DOI: 10.18653/v1/2024.emnlp-main.679.

Yuan*, Jianhao, Francesco Pinto*, Adam Davies*, and Philip Torr (2024). "Not Just Pretty Pictures: Toward Interventional Data Augmentation Using Text-to-Image Generators". In: *Proceedings of the 41st International Conference on Machine Learning*. Ed. by Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp. Vol. 235. Proceedings of Machine Learning Research. PMLR, pp. 57924–57952. URL: https://proceedings.mlr.press/v235/yuan24e.html.

Yun, Sangdoo, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo (2019). "Cutmix: Regularization strategy to train strong classifiers with localizable features". In: *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6023–6032.

Yun, Zeyu, Yubei Chen, Bruno Olshausen, and Yann LeCun (June 2021). "Transformer visualization via dictionary learning: contextualized embedding as a linear superposition of transformer factors". In: *Proceedings of Deep Learning Inside Out (DeeLIO): The 2nd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*. Ed. by Eneko Agirre, Marianna Apidianaki, and Ivan Vulić. Online: Association for Computational Linguistics, pp. 1–10. DOI: 10.18653/v1/2021.deelio-1.1.

Zeiler, Matthew D and Rob Fergus (2014). "Visualizing and understanding convolutional networks". In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*. Springer, pp. 818–833.

Zeng, Yi, Yu Yang, Andy Zhou, Jeffrey Ziwei Tan, Yuheng Tu, Yifan Mai, Kevin Klyman, Minzhou Pan, Ruoxi Jia, Dawn Song, et al. (2024). "AIR-Bench 2024: A Safety Benchmark Based on Risk Categories from Regulations and Policies". In: *CoRR*.

Zhai, Xiaohua, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer (June 2022). "LiT: Zero-Shot Transfer With Locked-Image Text Tuning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18123–18133.

Zhan, Qiusi, Richard Fang, Rohan Bindu, Akul Gupta, Tatsunori Hashimoto, and Daniel Kang (June 2024). "Removing RLHF Protections in GPT-4 via Fine-Tuning". In: *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*. Ed. by Kevin Duh, Helena Gomez, and Steven Bethard. Mexico City, Mexico: Association for Computational Linguistics, pp. 681–687. DOI: 10.18653/v1/2024.naacl-short.59.

Zhang, Dinghuai, Kartik Ahuja, Yilun Xu, Yisen Wang, and Aaron Courville (2021a). "Can subnetwork structure be the key to out-of-distribution generalization?" In: *International conference on machine learning*. PMLR, pp. 12356–12367.

Zhang, Duzhen, Yahan Yu, Chenxing Li, Jiahua Dong, Dan Su, Chenhui Chu, and Dong Yu (2024a). "Mm-llms: Recent advances in multimodal large language models". In: *arXiv preprint arXiv:2401.13601*.

Zhang, Fred and Neel Nanda (2024). "Towards Best Practices of Activation Patching in Language Models: Metrics and Methods". In: *The Twelfth International Conference on Learning Representations*. URL: https://openreview.net/forum?id=Hf17y6u9BC.

Zhang, Hongyi, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz (2017). "mixup: Beyond empirical risk minimization". In: *arXiv preprint arXiv:1710.09412*.

Zhang, Lvmin, Anyi Rao, and Maneesh Agrawala (2023a). "Adding conditional control to text-to-image diffusion models". In: *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3836–3847.

Zhang, Pengchuan, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao (June 2021b). "VinVL: Revisiting Visual Representations in Vision-Language Models". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5579–5588.

Zhang, Shengyu, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. (2023b). "Instruction tuning for large language models: A survey". In: *arXiv preprint arXiv:2308.10792*.

Zhang, Susan, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. (2022a). "Opt: Open pre-trained transformer language models". In: *arXiv preprint arXiv:2205.01068*.

Zhang, Xiao, Miao Li, and Ji Wu (2024b). "Conditional language learning with context". In: *Proceedings of the 41st International Conference on Machine Learning*, pp. 59247–59263.

Zhang, Xin, Shixiang Shane Gu, Yutaka Matsuo, and Yusuke Iwasawa (2023c). "Domain prompt learning for efficiently adapting clip to unseen domains". In: *Transactions of the Japanese Society for Artificial Intelligence* 38.6.

Zhang, Xingxuan, Yue He, Renzhe Xu, Han Yu, Zheyan Shen, and Peng Cui (2022b). *NICO++: Towards Better Benchmarking for Domain Generalization*. arXiv: 2204.08040 [cs.CV].

Zhao, Han, Chen Dan, Bryon Aragam, Tommi S Jaakkola, Geoffrey J Gordon, and Pradeep Ravikumar (2022). "Fundamental limits and tradeoffs in invariant representation learning". In: *The Journal of Machine Learning Research* 23.1, pp. 15356–15404.

Zhao, Haozhe, Zefan Cai, Shuzheng Si, Xiaojian Ma, Kaikai An, Liang Chen, Zixuan Liu, Sheng Wang, Wenjuan Han, and Baobao Chang (2023). "Mmicl: Empowering vision-language model with multi-modal in-context learning". In: *arXiv preprint arXiv:2309.07915*.

Zhao, Long, Ting Liu, Xi Peng, and Dimitris Metaxas (2020). "Maximum-Entropy Adversarial Data Augmentation for Improved Generalization and Robustness". In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Zheng, Guangtao, Wenqian Ye, and Aidong Zhang (2024a). "Learning robust classifiers with self-guided spurious correlation mitigation". In: *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pp. 5599–5607.

Zheng, Lianmin, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. (2024b). "Judging llm-as-a-judge with mt-bench and chatbot arena". In: *Advances in Neural Information Processing Systems* 36.

Zhong, Ziqian, Ziming Liu, Max Tegmark, and Jacob Andreas (2023). "The clock and the pizza: Two stories in mechanistic explanation of neural networks". In: *arXiv preprint arXiv:2306.17844*.

Zhou, Bolei, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba (2014). "Object detectors emerge in deep scene cnns". In: *arXiv preprint arXiv:1412.6856*.

Zhou, Bolei, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba (2016). "Learning deep features for discriminative localization". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2921–2929.

Zhou, Chunting, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. (2023a). "Lima: Less is more for alignment". In: *Advances in Neural Information Processing Systems* 36, pp. 55006–55021.

Zhou, Hattie, Arwen Bradley, Etai Littwin, Noam Razin, Omid Saremi, Joshua M. Susskind, Samy Bengio, and Preetum Nakkiran (2024a). "What Algorithms can Transformers Learn? A Study in Length Generalization". In: *The Twelfth International Conference on Learning Representations*. URL: https://openreview.net/forum?id=AssIuHnmHX.

Zhou, Kaiyang, Jingkang Yang, Chen Change Loy, and Ziwei Liu (2022). "Learning to prompt for vision-language models". In: *International Journal of Computer Vision* 130.9, pp. 2337–2348.

Zhou, Yiyang, Chenhang Cui, Jaehong Yoon, Linjun Zhang, Zhun Deng, Chelsea Finn, Mohit Bansal, and Huaxiu Yao (2023b). "Analyzing and mitigating object hallucination in large vision-language models". In: *arXiv preprint arXiv:2310.00754*.

Zhou, Yuhang, Paiheng Xu, Xiaoyu Liu, Bang An, Wei Ai, and Furong Huang (2024b). "Explore Spurious Correlations at the Concept Level in Language Models for Text Classification". In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 478–492.

Zong, Yongshuo, Ondrej Bohdal, and Timothy Hospedales (2024). "VL-ICL Bench: The Devil in the Details of Benchmarking Multimodal In-Context Learning". In: *arXiv preprint arXiv:2403.13164*.

Zou, Andy, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. (2023). "Representation engineering: A top-down approach to ai transparency". In: *arXiv preprint arXiv:2310.01405*.

Part V
Appendices

Appendix A: The Cognitive Revolution in Interpretability

A.1 Supplementary Background

A.1.1 Interpretable Machine Learning

Many definitions of "interpretability" have been proposed to describe methods that fall outside our analysis in this work (Doshi-Velez and Kim, 2017; Lipton, 2018; Arrieta et al., 2020; Broniatowski et al., 2021) concerning either (1) interpretability as an inherent feature of human-understandable methods, or (2) input-output explanations (e.g., to what input features is a given output attributable) of *supervised* deep learning models. In contrast, in this work we are concerned with the study of internal mechanisms and latent representations learned by self-supervised (foundation) models that has come to characterize much of the interpretability landscape, and is now most often referred to under the broad umbrella of *mechanistic interpretability* (see Section 2.2.1).

For example, the most widely-cited definition of interpretability is provided by Lipton (2018), who provides two general categories of interpretability. The first is *transparency*, an inherent quality of models that can be fully decomposed into clear, comprehensible operations (e.g., decision trees or rule-based expert systems). The second is *post-hoc explanations*, a family of techniques to explain specific outputs of an otherwise opaque ("black box") model (e.g., saliency maps; Simonyan et al., 2013). Naturally, any interpretability work studying neural networks must fall into the category of post-hoc explanations, as neural networks are inherently opaque. However, the notion of post-hoc explanation is, on its own, insufficient to describe the wealth of interpretability research that has developed around foundation models: the categories of post-hoc explanation techniques outlined by Lipton (2018) predate the era of self-supervised foundation model that have come to dominate many areas of study in AI and ML, and do not apply to contemporary internal analysis techniques such as probing (see Section 2.4.2) or circuit discovery (see Section 2.5), where the goal is not necessarily to explain specific model outputs, but rather to interpret the general internal representations, operations, and algorithms that foundation models learn in self-supervised pretraining.

Appendix B: Competence-Based Analysis of Language Models

B.1 Limitations

B.1.1 Gradient-Based Interventions

For causal probing to operate successfully – as is required to reliably deploy CALM in practice – it is important that probes leverage the underlying model's representation of the target feature to make predictions, rather than relying on spurious information. However, there is some evidence that probes often leverage such spurious information (Belinkov, 2022; Kumar et al., 2022; Canby* et al., 2025). For instance, in our followup work studying the GBI methodology alongside other causal probing methods (Chapter 4, Canby* et al., 2025) we find that each among a variety of causal probing methods (including GBIs) shows a tradeoff between its ability to manipulate the targeted feature (*completeness*) and the extent to which it also modifies the representation of other, non-targeted feature (*selectivity*). We also found that the flexibility of GBIs allows for precise control this tradeoff by modulating the magnitude of perturbations (ϵ), an advantage that is not shared by most other causal probing methods.

Furthermore, while GBIs are applicable to a more general range of model representations than most prior intervention methods (see Section 3.3), this generality comes with a lack of constraints on probes (g_Z); and as a result, GBIs cannot provide the strong theoretical constraints on collateral damage as can methods like, e.g., INLP (Ravfogel et al., 2020), which provably preserves distances between embeddings as well as possible while completely removing the linear representation of the target feature (which also generally leads to higher selectivity in practice (Canby* et al., 2025)). To minimize collateral damage to representations, the magnitude of perturbations should be modulated via constraints on gradient attacks against g_Z (see Section 3.4) and experimentally validated to control the damage done to representations (see Section B.2.3); and in the ideal case, should be calibrated to achieve the desired tradeoff between *selectivity* and *completeness* (a novel procedure introduced in (Canby* et al., 2025), a followup work building on GBIs as initially developed in this work). Alternatively, in cases where the structure of representations is believed to satisfy strong assumptions (e.g., being restricted to a linear subspace; Vargas and Cotterell, 2020) or strong upper bounds on collateral damage are required, CALM interventions can be implemented with methods like INLP rather than GBIs.[38]

[38]It may also be possible to control for collateral damage by developing GBI strategies that offer more principled protection against damage to non-targeted features, such as adding a loss term to penalize damage

B.1.2 Simple Experimental Setting

As noted in Section 3.4, our primary goal in our experiments is to validate CALM by testing it in a simplified experimental setting consisting of comparatively small, well-studied models and tasks. As such, we need models that are *just complex enough* for CALM to be applicable (i.e., neural language models that are capable of performing the tasks we consider at a nontrivial level of performance), making BERT and RoBERTa ideal candidates; and in future work plan to scale CALM to more complex contexts covering larger, more powerful models as they perform more difficult tasks (see Section B.4). This is a common setting in the context of substantial recent interpretability work: first, a theoretical framework is developed for interpreting an internal representation or mechanism and initially tested in the context of "toy" models or tasks (Elhage et al., 2021; Olsson et al., 2022; Zhong et al., 2023; Geiger et al., 2023), and subsequent work scales these frameworks to the context of larger models "in the wild" (Wang et al., 2023b; Conmy et al., 2023; Wu et al., 2023b). Analogously, all of our major contributions (the CALM framework, competence metric, and GBI causal probing method) are directly scalable to much larger, more recent LLMs (e.g., Zhang et al., 2022a; BigScience et al., 2022; Touvron et al., 2023a,b; Groeneveld et al., 2024, etc.).

B.1.3 Task Independence

In our experiments, we modeled the 14 LAMA ConceptNet tasks as representing fully independent features, which is not necessarily true – e.g., knowing that a tree is made of bark or contains leaves tells us something about whether it is a type of plant. However, in the aggregate (with impacts summed across 14 widely-varying lexical relation types in computing the final competence score for each task; see Section B.3.2), it may nonetheless be appropriate to treat the relations which are not causal with respect to a given task as collectively capturing spurious lexical associations.

B.2 Experimental Details

B.2.1 Tasks

The full set of LAMA ConceptNet tasks is as follows: IsA, HasA, PartOf, HasSubEvent, MadeOf, HasPrerequisite, MotivatedByGoal, AtLocation, CausesDesire, NotDesires, CapableOf, UsedFor, ReceivesAction, and HasProperty. We split each task dataset into train, validation, and test sets with a random 80%/10%/10% split. Train and validation instances

to non-targeted probes or leveraging interval bound propagation (Gowal et al., 2019) to place intervened embeddings inside the adversarial polytope for non-targeted features. We leave such possibilities to future work.

are fed to each model to produce embeddings used to train g_Z and select hyperparameters, respectively; and test instances are used to measure LLMs' competence with respect to each task by observing how predictions change under various interventions. In all experiments, we restrict each model M's output space for each task $\mathcal{T}$ to the subset of vocabulary V_M that occurs as a ground-truth answer y^* for at least one instance $(\mathbf{x}, y^*) \sim \mathcal{T}$ in the respective task dataset. This lowers the probability of false negatives in evaluation (e.g., penalizing the model for predicting $\hat{y}$ = "mammal" for "a dog is a type of y" instead of y^* = "animal").

B.2.2 Probes

We use BERT's final layer L to encode h_i^l embeddings for each such example, where i is the index of the `[MASK]` token or target word in the input prompt x_i. To encode the `[MASK]` token, we issue BERT masked prompts (as discussed above) to extract $h_{\texttt{[MASK]}}$, then repeat with the `[MASK]` token filled-in with the target word to encode it as h_+ (e.g., "device" in "A laser is a device which creates coherent light."), and concatenate matching embeddings $h = (h_{\texttt{[MASK]}}; h_+)$ to produce positive ($y = 1$) training instances. We also construct one negative ($y = 0$) instance, $h = (h_{\texttt{[MASK]}}; h_-)$, for each $h_{\texttt{[MASK]}}$ by sampling an incorrect target word x_i corresponding to an answer to a random prompt from the same task, feeding it into the cloze prompt in the place of the correct answer, and obtaining BERT's contextualized final-layer embedding of this token (h_-). Finally, we train g_Z on the set of all such (h, y).

We implement g_Z as a multi-layer perceptron with 2 hidden layers, each with a width of 768 (which is one half the concatenated input dimension of 1536), using ReLU activations and dropout with $p = 0.1$, training it for 32 epochs using Binary Cross Entropy with Logits Loss[39] and the Adam optimizer, saving the model from the epoch with the highest validation-set accuracy for use in all experiments.

For all competence results reported in Section 3.5, we run the same experiment 10 times – each with a different random initialization of g_Z and shuffled training data – and report each figure as the average among all 10 runs.

B.2.3 Interventions

For embedding h, target (counterfactual) class y', probe g_Z, loss function $\mathcal{L}$, and L_∞-bound $\epsilon \in \{0.01, 0.03, 0.1, 0.3\}$[40], each intervention (gradient attack) g_z may be used to produce perturbed representations $h' = g_z(h, y', f_{\text{cls}}, \mathcal{L}, \epsilon)$ where $||h - h'||_\infty \leq \epsilon$. In particular, given

[39] https://pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html

[40] All reported results use $\epsilon = 0.1$, as greater ϵ resulted in unacceptably high "collateral damage" across target tasks (e.g., even random perturbations of magnitude $\epsilon = 0.3$ do considerable damage), and lesser values meant that predictions changed on target tasks consisted of only a few test instances.

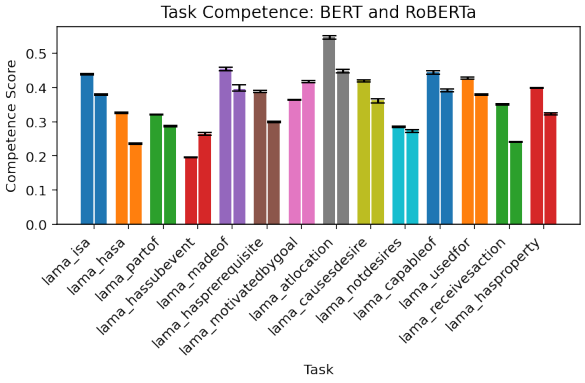


Figure 37: Competence of BERT (left bars) and RoBERTa (right bars) for all tasks, using PGD with $\epsilon = 0.1$. Y-values are the average competence score and error bars are the maximum and minimum competence score, as measured over 10 experimental iterations (each with a different randomly-initialized probe g_Z).

$h = (h_{\texttt{[MASK]}}; h_{\pm}) \in \mathbb{R}^{2d}$, let $h'_{\texttt{[MASK]}}$ be the first d dimensions of h' (which also satisfies the L_∞-bound with respect to $h_{\texttt{[MASK]}}$, $||h_{\texttt{[MASK]}} - h'_{\texttt{[MASK]}}||_\infty \leq \epsilon$). To measure BERT's use of internal representations of Z on each prompt task, we evaluate its performance when perturbed $h'_{\texttt{[MASK]}}$ is used to compute masked-word predictions, compared to unperturbed $h_{\texttt{[MASK]}}$.

Our intent in intervening only on the final-layer mask embedding $h_{\texttt{[MASK]}}$ in our experiments is that, in the final layer of a masked language model such as BERT or RoBERTa, the only embedding which is used to compute masked-word probabilities is that of the `[MASK]` token. Thus, any representation of the feature that is *used* by the model in its final layer must be a part of its representation of the `[MASK]` token, preventing "recoverability" phenomena such as those observed by Elazar et al. (2021b).

FGSM FGSM (Goodfellow et al., 2015) takes one gradient step of magnitude ϵ in the direction that minimizes the loss of a classifier (here, the probe f_{cls}) with respect to target class y'. We implement FGSM interventions as

$$h' = h + \epsilon \cdot \text{sgn}(\nabla_h \mathcal{L}(f_{\text{cls}}, x, y'))$$

where $\mathcal{L}$ is the same loss function used to train f_{cls} (here, binary cross entropy).

PGD PGD (Bubeck et al., 2015; Madry et al., 2017) iteratively minimizes the loss of a classifier (here, the probe f_{cls}) with respect to target class y' by performing gradient descent

within a L_∞ ball of radius ϵ. We implement PGD interventions as $h' = h^T$, where

$$h^{t+1} = \Pi_{N(h)}\big(h^t + \alpha \cdot \text{sgn}(\nabla_h \mathcal{L}(f_{\text{cls}}, x, y))\big)$$

for iterations $t = 0, 1, ..., T$, projection operator Π, L_∞-neighborhood $N(h) = \{h' : ||h - h'|| \leq \epsilon\}$, and $\mathcal{L}$ is the same loss function used to train f_{cls} (here, binary cross entropy). This method also introduces two hyperparameters: the number of PGD iterations T and step size α. We use hyperparameter grid search over $\alpha \in \{0.001, 0.003, 0.01, 0.03\}$ and $T \in \{20, 40, 60, 80, 100\}$, finding that setting $\alpha = \frac{\epsilon}{10}$ and $T = 40$ produces the most consistent impact on g_Z accuracy across all tasks; so we use these values for the results visualized in Figure 37.

B.2.4 Compute Budget

BERT-base-uncased has 110 million parameters, and RoBERTa-base has 125M parameters. As our goal is to study the internal representation and use of linguistic features in existing pre-trained models, and we are not directly concerned with training or fine-tuning such models, we use these models only for inference (including encoding text inputs, using embeddings to train probes, and feeding intervened embeddings back into the language models). The only models we trained were probes g_Z, which each had 1.77M parameters.

Each experimental iteration (including encoding text inputs, training probes on all 14 tasks, and performing all GBIs) for either BERT or RoBERTa took less than one hour on a single NVIDIA GeForce GTX 1080 GPU, meaning that running all 10 iterations across both language models took less than 20 hours on a single GPU. Each iteration, probe, and GBI can easily be parallelized across GPUs: in our case, running all iterations across both models took less than 3 hours total across 8 GTX 1080 GPUs.

B.3 Competence Metric

B.3.1 Comparison with IIA

As noted in Section 3.2.2, the $\mathcal{C}_\mathcal{T}(M \mid \mathcal{G}_\mathcal{T})$ metric defined in Equation (2) is an adaptation of the Interchange Intervention Accuracy (IIA) metric (Geiger et al., 2022, 2023), which evaluates the faithfulness of a causal abstraction like $\mathcal{G}_\mathcal{T}$ as a (potential) explanation of the behavior of a "black box" system like M. In our case, this is equivalent to evaluating the competence of M on task $\mathcal{T}$, provided that $\mathcal{G}_\mathcal{T}$ is the appropriate SCM for $\mathcal{T}$, as an LLM is competent only to the extent that its behavior is determined by a causally invariant

representation of the task.[41] IIA requires performing *interchange interventions* $\text{do}_{II}(\mathbf{z}_j)$, where the part of M's intermediate representation of input $\mathbf{x}_i$ hypothesized to encode latent variables $\mathbf{Z}$ (taking the values $\mathbf{z}_i$ when provided input $\mathbf{x}_i$) is replaced with that of $\mathbf{x}_j$ (which, in principle, means that the modified representation encodes the values $\mathbf{z}_j$ instead of $\mathbf{z}_i$), at which point these interchange interventions are used to compute predictions $M(\mathbf{x}_i \mid \text{do}_{II}(\mathbf{z}_j))$, and the output is compared with $\mathcal{G}_\mathcal{T}(\mathbf{x}_i \mid \text{do}(\mathbf{z}_j))$ to measure how faithfully $\mathcal{G}_\mathcal{T}$ predicts M's behavior under these interventions. Thus, given access to high-quality interchange interventions over M, IIA measures the extent to which $\mathcal{G}_\mathcal{T}$ correctly models M's behavior under counterfactuals, and thus its faithfulness as a causal abstraction of M.

To adapt IIA to the context of causal probing and define $\mathcal{C}_\mathcal{T}(M \mid \mathcal{G}_\mathcal{T})$, we replace instance-level interchange interventions do_{II} with concept-level interventions do_Z for any given feature Z. That is, instead of swapping M's representation of variables $\mathbf{Z} = Z_1, ..., Z_k$ given input $\mathbf{x}_i$ with that of $\mathbf{x}_j$, we intervene on the representation of each feature $Z \in \mathbf{Z}$ at the level of arbitrary values $\mathbf{z} : Z_1 = z_1, ..., Z_k = z_k$ that need not correspond to previously observed $\mathbf{x}$, allowing us to simulate the behavior of M under previously-unseen distribution shifts (i.e., settings $\mathbf{z}$ representing previously-unseen combinations of values) and in doing so predict M's consistency with a given causal model $\mathcal{G}_\mathcal{T}$ under these new conditions. As one of our primary motivations in studying LLM competence is to provide a framework useful for predicting and explaining behavior under distribution shifts, $\mathcal{C}_\mathcal{T}$ is more appropriate than IIA in this setting. However, this also introduces greater room for error: where interchange interventions only requires modifying representations to match the values taken by another input – as counterfactual representations can be obtained simply by "plugging in" representations from a different input – computing $\mathcal{C}_\mathcal{T}$ instead requires one to perform open-ended interventions that may not correspond to any ground-truth input, in which case there may be no region of the embedding space that corresponds to the intended setting $\mathbf{z}$ (Geiger et al., 2020; Abraham et al., 2022; Canby* et al., 2025).

B.3.2 Experimental Competence Metric

To compute the expectation in Equation (2) for test set $\{\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i\}_{i=1}^n \sim \mathcal{T} \times \mathbf{Z}$, we sum the competence score over all samples $\mathbf{x}_i$ and perform one intervention $\text{do}(Z_j = 0)$ corresponding to each concept $Z_j \in \mathbf{Z}$.[42] As our goal is to measure the extent to which M's behavior is

[41] For many tasks, there is more than one valid $\mathcal{G}_\mathcal{T}$ (see, e.g., the "price tagging game" constructed by Wu et al. (2023b)). In such cases, $\mathcal{C}_\mathcal{T}(M \mid \mathcal{G}_\mathcal{T})$ should be computed with respect to each valid $\mathcal{G}_\mathcal{T}$ and the highest result should be selected, as conforming to any such $\mathcal{G}_\mathcal{T}$ carries the same implications.

[42] Note that this intervention changes the prediction $\mathcal{G}_\mathcal{T}(\mathbf{x}_i) \neq \mathcal{G}_\mathcal{T}(\mathbf{x}_i \mid \text{do}(Z_j = 0))$ if and only if $(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{T}_j$ – i.e., where the corresponding $(\mathbf{z}_i)_j = 1$ – otherwise, $(\mathbf{z}_i)_j$ is already 0, so the intervention has no effect. Thus, as $\mathcal{C}_\mathcal{T}(M \mid \mathcal{G}_\mathcal{T})$ measures M's consistency with $\mathcal{G}_\mathcal{T}$, then to the extent that M is competent,

attributable to an underlying representation of the causal feature Z_c or environmental feature $Z \in \mathbf{Z}_e$, our experimental model defines $\mathcal{G}_\mathcal{T}$'s predictions with reference to M's original predictions $M(\mathbf{x}_i) = \hat{\mathbf{y}}_i$, according to the following principle: if M is competent, then its prediction $M(\mathbf{x}_i) = \hat{\mathbf{y}}_i$ is wholly attributable to its representation of causal feature Z_c, so its predictions $M(\mathbf{x}_i \mid \text{do}(Z_c)) = \hat{\mathbf{y}}_i{}'$ will not overlap with its original predictions $\hat{\mathbf{y}}_i$ (i.e., $\text{overlap}(\hat{\mathbf{y}}_i, \hat{\mathbf{y}}_i{}') = 0$); and conversely, a competent M will make the *same* predictions $M(\mathbf{x}_i \mid \text{do}(Z_j)) = \hat{\mathbf{y}}_i{}''$ for any $Z_j \in \mathbf{Z}_e$, because its prediction is not caused by its representation of these environmental features (i.e., $\text{overlap}(\hat{\mathbf{y}}_i, \hat{\mathbf{y}}_i{}'') = 1$). Motivated by this reasoning, our experimental model defines $\mathcal{G}_\mathcal{T}(\mathbf{x}_i \mid \text{do}(Z_j = 0)) = M(\mathbf{x}_i)$ for environmental $Z_j \in \mathbf{Z}_e$; and for causal feature Z_c, defines $\mathcal{G}_\mathcal{T}(\mathbf{x}_i \mid \text{do}(Z_c = 0)) = \{y' \in V_M : y' \notin M(\mathbf{x}_i)\}$ (i.e., the set of all tokens y' in M's vocabulary that were not in its original prediction $M(\mathbf{x}_i)$). Thus, under experimental model E, we approximate $\mathcal{C}_\mathcal{T}(M \mid \mathcal{G}_\mathcal{T})$ by computing it as follows:

$$\hat{\mathcal{C}_\mathcal{T}}(M \mid \mathcal{G}_\mathcal{T}) = \frac{1}{n \cdot m} \sum_{i=1}^{n} \sum_{j=1}^{m} \text{overlap}\left(M\big(\mathbf{x}_i \mid \text{do}(Z_j = 0)\big), \mathcal{G}_\mathcal{T}\big(\mathbf{x}_i \mid \text{do}(Z_j = 0)\big)\right) \tag{12}$$

Notably, our experimental model E only accounts for the relationship between M's intervened and non-intervened predictions, independently of ground truth labels – instead, what is being measured is M's consistency under meaning-preserving interventions $\text{do}(Z_{j'})$ and its mutability under meaning-altering interventions $\text{do}(Z_j)$. However, as we find in Section 3.5.1, the resulting competence metric $\mathcal{C}_\mathcal{T}(M \mid \mathcal{G}_\mathcal{T})$ is nonetheless useful for predicting M's accuracy.

B.4 Future Work

B.4.1 Representation Learning

The CALM framework, competence measure, and GBI methodology developed in this work are sufficiently general to be directly applied to analyze arbitrary LLMs on any language modeling task whose causal structure is already well understood (or, for tasks where this is not the case, we may apply the causal graph discovery approach described in Section B.4.4), allowing us to study the impact of various model architectures, pre-training regimes, and fine-tuning strategies on the representations LLMs learn and use for arbitrary tasks of interest.

its prediction should change under all and only the same interventions as $\mathcal{G}_\mathcal{T}$.

B.4.2 Multitask Learning

Are high competence scores on task $\mathcal{T}$ correlated with an LLMs' robustness to meaning-preserving transformations (see, e.g., Elazar et al., 2021a) on tasks $\mathcal{T}'$ that share several causal features $\mathbf{Z}_c$ with task $\mathcal{T}$. Through the lens of causally invariant prediction (Peters et al., 2016; Arjovsky et al., 2019; Bühlmann, 2020), this hypothesis is likely true (however, see Rosenfeld et al., 2020 for appropriate caveats) – if so, this would make it possible to use clusters of related tasks to predict LLMs' robustness (and other behavioral patterns, such as brittleness in the face of distribution shifts introduced by spurious dependencies) between related tasks using CALM, given an appropriate experimental model. Furthermore, the ability to characterize tasks based on mutual (learned) dependency structures could be valuable in transfer learning applications such as guiding the selection of auxiliary tasks in multi-task learning (Ruder, 2017) or predicting the impact of intermediate task fine-tuning on downstream target tasks (Choshen et al., 2022).

B.4.3 Task Dependencies

Another possible application of CALM concerns causal invariance under multi-task applications. Existing approaches in invariant representation learning generally require task-specific training (Zhao et al., 2022), as the notion of invariance is inherently task-centric (i.e., the features which are invariant predictors of output values vary by task, and different tasks may have opposite notions of which features are causal versus environmental), so applying such approaches to train models to be causally invariant with respect to a specific downstream task $\mathcal{T}$ is expected to come at the cost of performance on other downstream tasks $\mathcal{T}'$. Therefore, considering the recent rise of open-ended, task-general LLMs (Zhang et al., 2022a; BigScience et al., 2022; Touvron et al., 2023a,b; Groeneveld et al., 2024), it is important to understand the relationship between different task dependencies learned when fine-tuning task-general models on specific downstream tasks to account for applications involving tasks with different (and perhaps contradictory) causal structures, such as CALM.

B.4.4 Causal Competence Graph Discovery

A key feature of CALM is that, instead of simply measuring consistency with respect to a known, static task description $\mathcal{G}_\mathcal{T}$, the competence metric in Equation (2) can also be used to dynamically discover a competence graph $\hat{\mathcal{G}}$ which most faithfully explains a model M's behavior in a given task or context (see Section 3.2.2) by computing $\mathcal{C}(M \mid \hat{\mathcal{G}})$ "in-the-loop" of existing causal graph discovery algorithms like IGSP (Yang et al., 2018). Such algorithms could be used to suggest likely competence graphs based on interventional data collected by

running CALM experiments, to recommend the experiments that would yield the most useful interventional data for the graph discovery algorithm, or to evaluate candidate graphs $\hat{\mathcal{G}}$ according to $\mathcal{C}(M \mid \hat{\mathcal{G}})$, terminating the graph discovery algorithm once a competence graph $\hat{\mathcal{G}}$ that offers sufficiently faithful explanations of M's behavior has been found (e.g., where $\mathcal{C}(M \mid \hat{\mathcal{G}}) > \tau$ for some threshold τ). In this case, it is still necessary to define the set of features $\mathbf{Z}$ being probed and the scoring function S used to compare the predictions of M and $\hat{\mathcal{G}}$; but no knowledge of the causal dependencies (or structural functions $F : \mathbf{pa}(Z_j) \mapsto Z_j$ mapping from causal parents $\mathbf{pa}(Z_j)$ to causal dependents Z_j; see Bongers et al., 2021) is required.

Appendix C: How Reliable are Causal Probing Interventions?

C.1 Framework Details

Completeness of Concept Removal Interventions In Equation (4), we define the "goal" distribution P^*_Z of a concept removal intervention used in causal probing as being the uniform distribution – i.e., for a perfect concept removal intervention, $P^*_Z = P_v(Z \mid \mathbf{h}^*_{Z=0}) = \mathcal{U}(Z)$. However, this is only true in the case of *causal probing*, and is not true of some concept removal applications such as guarding protected attributes (see, e.g., Ravfogel et al., 2023). That is, in the case of causal probing, the goal of an intervention is to intervene on a model's representation during its forward pass, feeding the intervened embedding back into the model and observing the change in the model's behavior (as described in Section 2.4.3). Recall that the purpose of a validation probe v is to decode model M's representation of a given feature Z, not to predict its ground truth value – that is, even if M encodes the incorrect value of $Z = z'$ rather than $Z = z$ for a given input, the validation probe should still decode the incorrect value $Z = z'$. Indeed, this is precisely the principle behind using validation probes in the case of counterfactual interventions that change the representation of $Z = z$ to counterfactual value $Z = z'$, where validation probes are used to validate the extent to which the representation has actually been changed to encode this counterfactual value, and the ideal counterfactual intervention yields $P_v(Z = z' \mid \mathbf{h}^*_{Z=z'}) = P^*(Z = z') = 1$. However, in the case of concept removal interventions $\text{do}(Z = 0)$, an intervened embedding $\mathbf{h}^*_{Z=0}$ would ideally remove all information encoding M's representation of the value taken by Z, meaning that the M would not encode any value $Z = z_1$ as being more probable than $Z = z_2$ (as any information that is predictive of the value taken by Z should have been removed). In this case, the validation probe v would predict an equal probability $P_v(Z = z_i \mid \mathbf{h}^*_{Z=0})$ for any given value z_i that may be taken by Z_i – i.e., $P_v(Z \mid \mathbf{h}^*_{Z=0}) = P^*_Z = \mathcal{U}(Z)$.

However, this is not the case in the context of instances such as guarding protected attributes, where the goal of an intervention $\text{do}(Z = 0)$ is to remove all information that is predictive of Z from embedding representations $\mathbf{h}^*_{Z=0}$ *such that no probe g can be trained to predict* $P_g(Z \mid \mathbf{h}^*_{Z=0})$ any better than predicting $P(Z)$ – i.e., ignoring the embedding entirely and simply mapping every input to the label distribution $P(Z)$ (Ravfogel et al., 2023). In this case, the probe g is trained on intervened embeddings $\hat{\mathbf{h}}_{Z=0}$, in which case it can learn to map every such embedding to the label distribution $P(Z)$, which yields superior performance relative to predicting the uniform distribution $\mathcal{U}(Z)$ in any case where the label distribution $P(Z)$ is not perfectly uniform, as such a g would have an expected accuracy

equal to the proportion of test instances with the most common label $Z = z_{\text{argmax}}$ (which would be greater than the accuracy $\frac{1}{k}$ expected by defaulting to $\mathcal{U}(Z)$).

The key technical distinction between these two use cases of concept removal interventions is *whether or not probes or underlying models are trained or fine-tuned in the context of interventions*. In the case of causal probing, they are not – the (frozen) model M has no opportunity to recover the original value of $Z = z$ following a concept removal intervention $\text{do}(Z = 0)$, and this should be reflected by validation probes. This is natural, given that the purpose of causal probing is to interpret the features used by M in making a given prediction, not to test whether M can be trained to recover features removed by interventions; and this is reflected by validation probes v, which are never trained on intervened embeddings. In contrast, for concept removal, probes (or models) are trained on intervened embeddings, and may learn to recover features removed by interventions, meaning that – even in the worst case where all information has been removed – it would at least be possible to learn to reproduce the label distribution $P(Z)$; but there is no reason to expect a model M or validation probe v to do so, given that they have never been trained on intervened embeddings. Thus, while we define the "goal" distribution $P^*_Z = \mathcal{U}(Z)$ for measuring the completeness of concept removal interventions as being $\mathcal{U}(Z)$ rather than $P(Z)$, this distribution would instead be $P^*_Z = P(Z)$ in the case of concept removal.

C.2 Experimental Details

C.2.1 LGD Dataset

We use syntax annotations to extract values for the environmental variable Z_e from the LGD dataset (Linzen et al., 2016): if the part-of-speech of the word immediately preceding the `[MASK]` token is a noun, and it is the object of a preposition (i.e., not the subject), then its number defines Z_e. About 83% of the sentences do not have a prepositional object preceding `[MASK]`, and so are only relevant for causal interventions.

The contingency table for values of Z_c and Z_e in the test set are in Table 7.

C.2.2 Probe Details

Our experiments include linear and MLP probes (both for interventions and as validation probes). Linear interventions (INLP, RLACE, and AlterRep) require linear probes; and for nonlinear interventions (GBIs), we use MLPs. We implement probes using PyTorch (Paszke, 2019), and leverage LLM implementations of all models available via HuggingFace Transformers (Wolf et al., 2019). For validation probes, we experiment with both linear and MLP probes. For all probes, we select hyperparameters by performing a grid search across

	$Z_e = \emptyset$	$Z_e = \texttt{Sg}$	$Z_e = \texttt{Pl}$	Total
$Z_c = \texttt{Sg}$	176K	31K	5K	213K
$Z_c = \texttt{Pl}$	78K	10K	4K	92K
Total	254K	41K	9K	305K

Table 7: **Contingency Table on Test Set**. Distribution of data across combinations of causal and environmental variables. $Z_e = \emptyset$ denotes instances which have no prepositional phrase attached to the subject (and thus, contain no environmental variable). Note that the label distributions are unbalanced: $P(Z_c = \texttt{Sg}) = 69.8\%$ and $P(Z_e = \texttt{Sg} \mid E \neq \emptyset) = 81.5\%$.

candidate hyperparameter values, selecting the hyperparameters that yield the highest validation-set accuracy. We save probe parameters from the epoch with the highest validation-set accuracy with patience of 4 epochs. All probes are trained with cross-entropy loss.

For all **linear probes**, we consider learning rates in [0.0001, 0.001, 0.01, 0.1].

For **MLP probes**, we perform grid search over the following hyperparameter values:

- Number of hidden layers: [1, 2, 3]
- Layer size: [64, 256, 512, 1024]
- Learning rate: [0.0001, 0.001, 0.01]

C.2.3 Interventions

Gradient Based Interventions For all gradient-based intervention methods (GBIs, introduced in Chapter 3), we define the maximum perturbation magnitude of each intervention as ε (i.e., $||\hat{\mathbf{h}}_Z - \mathbf{h}||_\infty \leq \varepsilon$), and experiment over a range of ε values between 0.005 to 5.0 – specifically, $\varepsilon \in$ [0.005, 0.006, 0.007, 0.009, 0.011, 0.013, 0.016, 0.019, 0.024, 0.029, 0.035, 0.042, 0.051, 0.062, 0.076, 0.092, 0.112, 0.136, 0.165, 0.2, 0.286, 0.409, 0.585, 0.836, 1.196, 1.71, 2.445, 3.497, 5.0]. We consider the following gradient attack methods for GBIs:

1. **FGSM** We implement Fast Gradient Sign Method (FGSM; Goodfellow et al., 2015) interventions as:
$$h' = h + \varepsilon \cdot \text{sgn}\left(\nabla_h \mathcal{L}\left(f_{\text{cls}}, x, y\right)\right)$$
2. **PGD** We implement Projected Gradient Descent (PGD; Madry et al., 2017) interventions as $h' = h^T$ where
$$h_{t+1} = \Pi_{\mathcal{N}(h)}\left(h_t + \alpha \cdot \text{sgn}\left(\nabla_h \mathcal{L}(f_{\text{cls}}, x, y)\right)\right)$$

for iterations $t = 0, 1, \ldots, T$, projection operator Π, and L_∞-neighborhood $\mathcal{N}(h) = \{h' : \|h - h'\| \leq \varepsilon\}$. For PGD, we use 2 additional hyperparameters: iterations T and step size α, while fixing $T = 40$, as in Chapter 3.

3. **AutoAttack** AutoAttack (Croce and Hein, 2020) is an ensemble of adversarial attacks that includes FAB, Square, and APGD attacks. Auto-PGD (APGD) is a variant of PGD that automatically adjusts the step size to ensure effective convergence. The parameters used were set as norm = L_∞ and for Square attack, the n_queries=5000.

Concept Removal Interventions For concept removal interventions, we project embeddings into the nullspaces of classifiers. Here, the the rank r corresponds to the dimensionality of the subspace identified and erased by the intervention, meaning that the number of dimensions removed is equal to the rank.[43] We experiment over the range of values $r \in [0, 1, ..., 40]$. We consider the following concept removal interventions:

1. **INLP** We implement Iterative Nullspace Projection (INLP; Ravfogel et al., 2020) as follows: we train a series of classifiers $w_1, ..., w_n$, where in each iteration, embeddings are projected into the nullspace of the preceding classifiers $P_N(w_0) \cap \cdots \cap P_N(w_n)$. We then apply the combined projection matrix to calculate the final projection where $P := P_N(w_1) \cap \cdots \cap N(w_i)$, X is the full set of embeddings, and $X_{\text{projected}} \leftarrow P(X)$.
2. **RLACE** We implement Relaxed Linear Adversarial Concept Erasure (R-LACE; (Ravfogel et al., 2022a)) which defines a linear minimax game to adversarially identify and remove a linear bias subspace. In this approach, $\mathcal{P}_k$ is defined as the set of all $D \times D$ orthogonal projection matrices that neutralize a rank r subspace:

$$P \in \mathcal{P}_k \leftrightarrow P = I_D - W^\top W$$

The minimax equation is then solved to obtain the projection matrix P which is used to calculate the final intervened embedding $X_{\text{projected}}$, similar to INLP

$$\min_{\theta \in \Theta} \max_{P \in \mathcal{P}_k} \sum_{n=1}^{N} \ell\left(y_n, g^{-1}\left(\theta^\top P x_n\right)\right)$$

Hyperparameters for P and θ were a learning rate of 0.005 and weight decay of 1e-5.

AlterRep We implement AlterRep (Ravfogel et al., 2021) by first running INLP, saving all classifiers, and using these to compute rowspace projections that push all embeddings to the

[43] This is only true for binary features Z – for variables that can take n values with $n > 2$, the number of dimensions removed is $n \cdot r$.

positive $Z = \texttt{Pl}$ or negative $Z = \texttt{Sg}$ side of the separating hyperplane for all classifiers. That is, we compute

$$\hat{\mathbf{h}}^l_{Z=\texttt{Sg}} = P_N(\mathbf{h}) + \alpha \sum_{w \in \mathbf{W}} (-1)^{SIGN(w \cdot \mathbf{h})} (w \cdot \mathbf{h}) \mathbf{h}$$

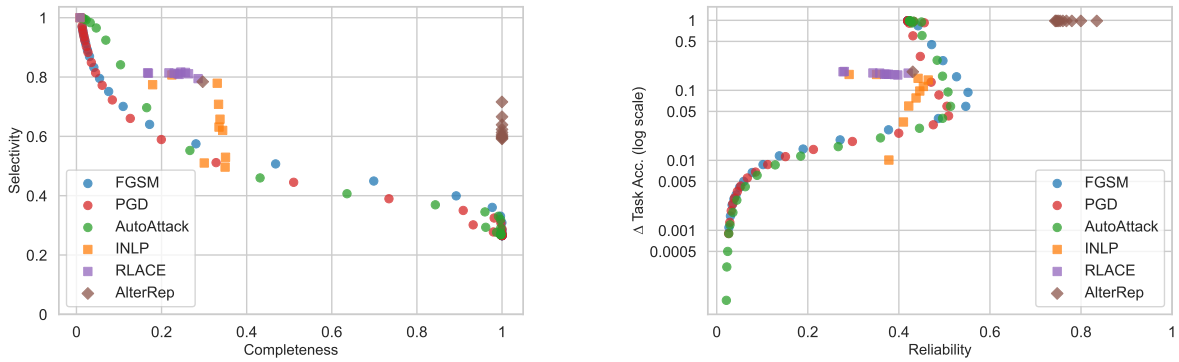
$$\hat{\mathbf{h}}^l_{Z=\texttt{Pl}} = P_N(\mathbf{h}) + \alpha \sum_{w \in \mathbf{W}} (-1)^{1-SIGN(w \cdot \mathbf{h})} (w \cdot \mathbf{h}) \mathbf{h}$$

where P_N is the nullspace projection from INLP.

C.3 Supplemental Results

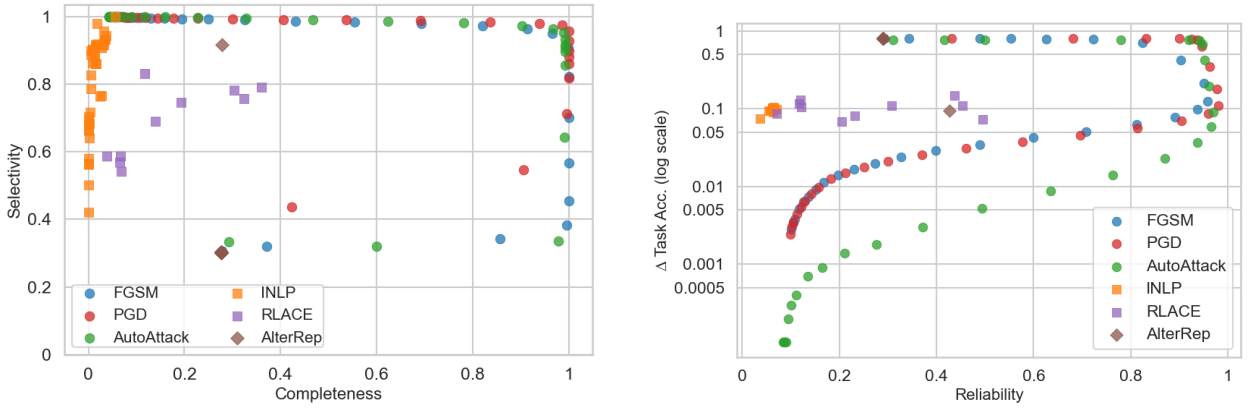
C.3.1 Final-Layer Completeness, Reliability, and Selectivity

In Figures 38a, 39a and 40a to 40c, we visualize final-layer completeness and selectivity of intervention methods for all models except Pythia-160M, analogously to the Figure 6a results reported in the main paper for Pythia-160M. In Figures 38b and 39b, we show the relationship between ΔTask Acc and reliability for BERT and GPT2 (respectively), analogously to the Figure 6b results reported in the main paper for Pythia-160M.



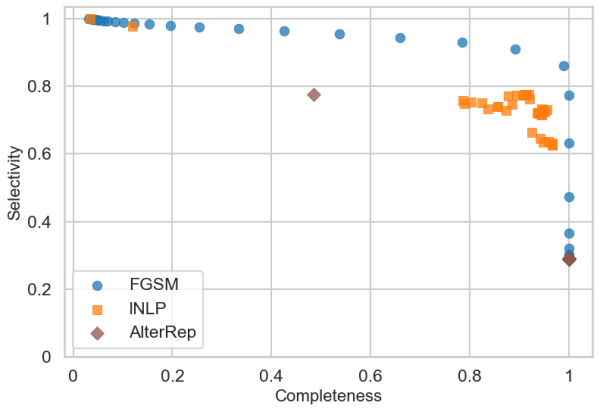
(a) **Selectivity vs. Completeness** (b) **ΔTask Acc vs. Reliability**

Figure 38: **(BERT) Completeness, selectivity, reliability, and ΔTask Acc** for all interventions in BERT's final layer. Each point in both plots corresponds to a different hyperparameter setting.

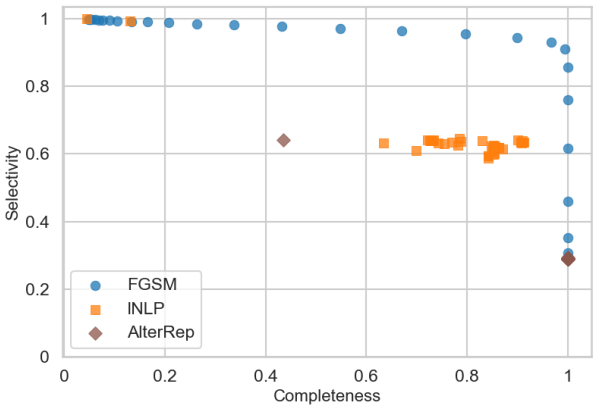


(a) **Selectivity vs. Completeness** (b) **ΔTask Acc vs. Reliability**

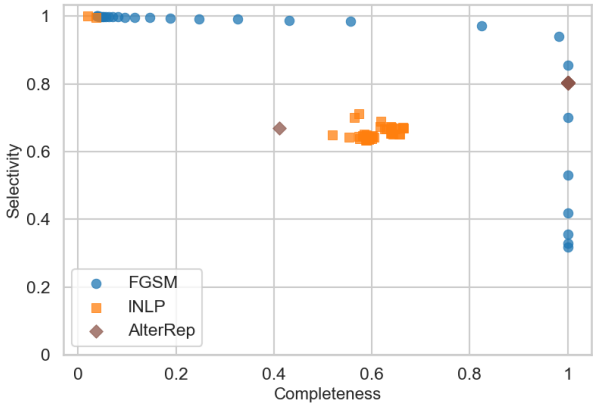
Figure 39: **(GPT2) Completeness, selectivity, reliability, and ΔTask Acc** for all interventions in the final layer of GPT2. Each point in both plots corresponds to a different hyperparameter setting.



(a) **(Pythia-1.4B) Selectivity vs. Completeness**



(b) **(Pythia-6.9B) Selectivity vs. Completeness**



(c) **(Llama-3.2-3B-Instruct) Selectivity vs. Completeness**

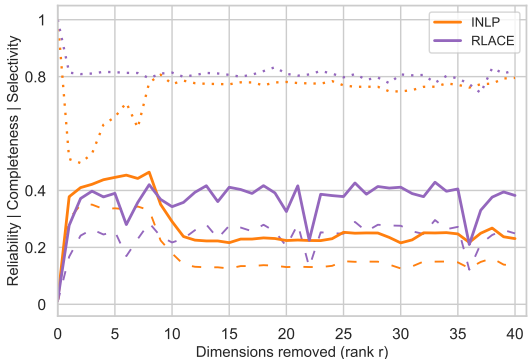
Figure 40: **Completeness and selectivity** for all interventions in the final layer of **Pythia-1.4B, Pythia-6.9B, and Llama-3.2-3B-Instruct**. Each point in both plots corresponds to a different hyperparameter setting.

	$C(\hat{\mathbf{H}}_Z)$	$S(\hat{\mathbf{H}}_Z)$	$R(\hat{\mathbf{H}}_Z)$	x_{opt}
INLP	0.3308 ± 0.0013	0.7792 ± 0.0012	0.4644 ± 0.0013	$r = 8$
RLACE	0.2961 ± 0.0013	0.7782 ± 0.0012	0.4290 ± 0.0014	$r = 33$
AlterRep	1.0000 ± 0.0000	0.7162 ± 0.0017	$\mathbf{0.8346} \pm 0.0012$	$\alpha = 0.1$
FGSM	0.8923 ± 0.0011	0.3994 ± 0.0018	0.5518 ± 0.0017	$\varepsilon = 0.112$
PGD	0.7343 ± 0.0016	0.3897 ± 0.0018	0.5092 ± 0.0016	$\varepsilon = 0.112$
AutoAttack	0.8433 ± 0.0013	0.3692 ± 0.0019	0.5136 ± 0.0018	$\varepsilon = 0.112$

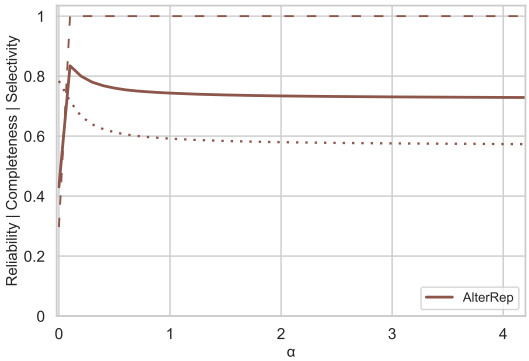
Table 8: **(BERT) Intervention scores for maximum-reliability hyperparameters** in the final layer, **with standard error included**. All scores are reported for the hyperparameter x_{opt} that maximizes the reliability of each respective method. Counterfactual methods are grouped above the double line, with concept removal methods below it.

Additionally, in Table 8, we report the standard error of completeness, selectivity, and reliability for BERT's maximum-reliability final-layer results displayed in Table 1. Note that all scores have standard error < 0.002, and we observe the same pattern for all other models.

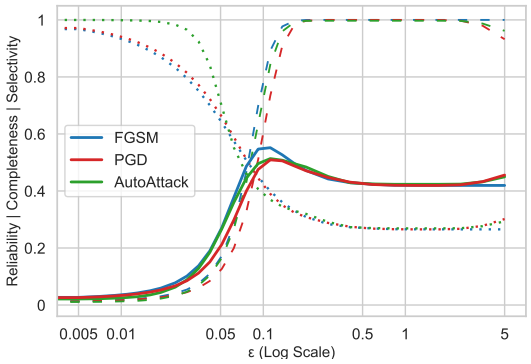
Hyperparameter Variation In Figures 41 and 42, we observe that increasing the degree of control that interventions have over the representation of the target feature by increasing the intervention hyperparameter associated with a given intervention type (i.e., ε, α, or rank) generally leads to both improved completeness and decreased selectivity.



(a) **Linear Removal (INLP, RLACE)**

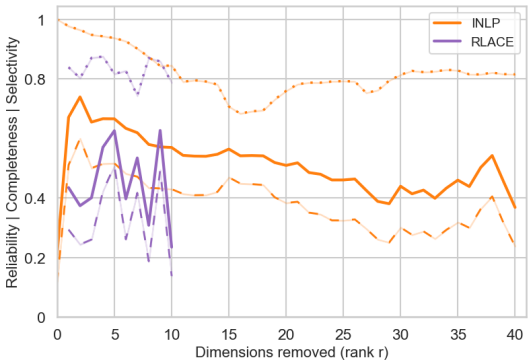


(b) **Linear Counterfactual (AlterRep)**

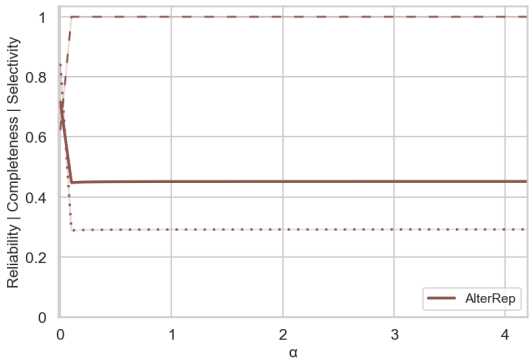


(c) **Nonlinear Counterfactual (GBIs)**

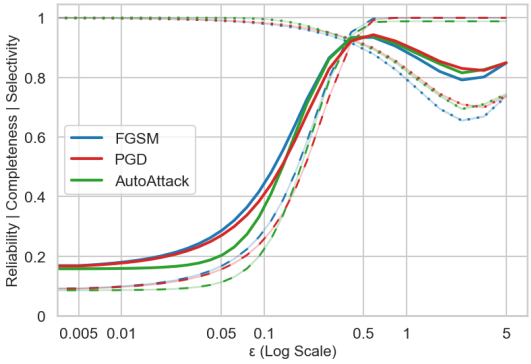
Figure 41: **(BERT) Reliability (solid), completeness (dashed), & selectivity (dotted)** of all methods in BERT's final layer, by hyperparameter value.



(a) **Linear Removal (INLP, RLACE)**



(b) **Linear Counterfactual (AlterRep)**

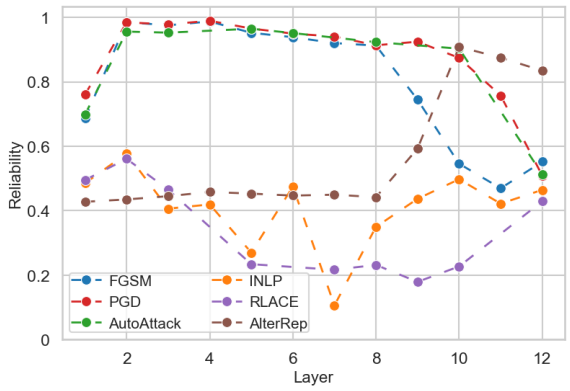


(c) **Nonlinear Counterfactual (GBIs)**

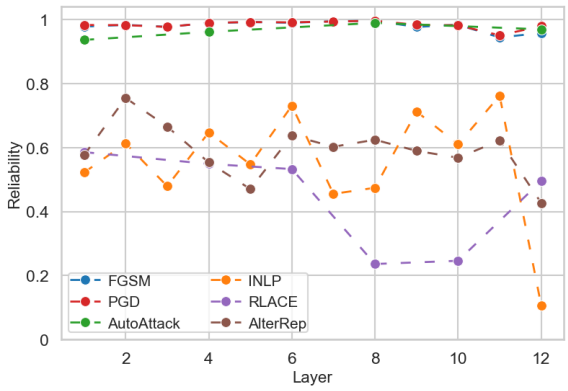
Figure 42: **(Pythia-160M) Reliability (solid), completeness (dashed), & selectivity (dotted)** of all methods in the final layer of Pythia-160M, by hyperparameter value.

C.3.2 Reliability by Layer

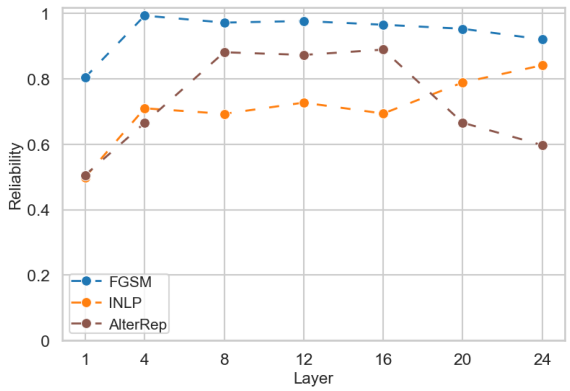
In Figure 43, we visualize maximum reliability of intervention methods across layers for all models (except Pythia-160M, which is reported in the main paper), analogously to the Figure 7 results reported in Section 4.5.2.



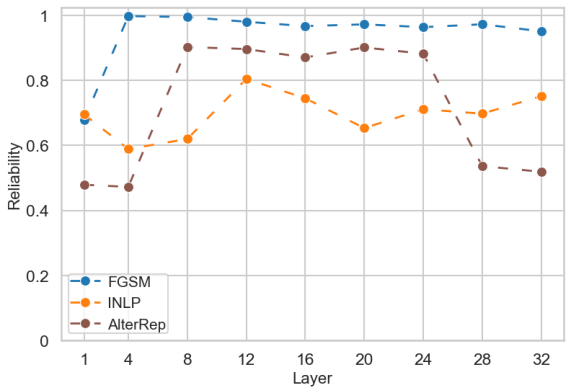
(a) **BERT**



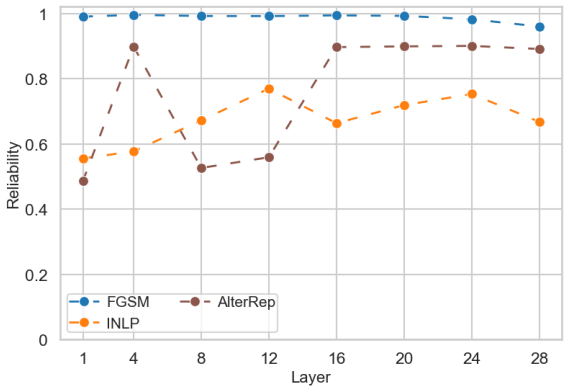
(b) **GPT2**



(c) **Pythia-1.4B**



(d) **Pythia-6.9B**

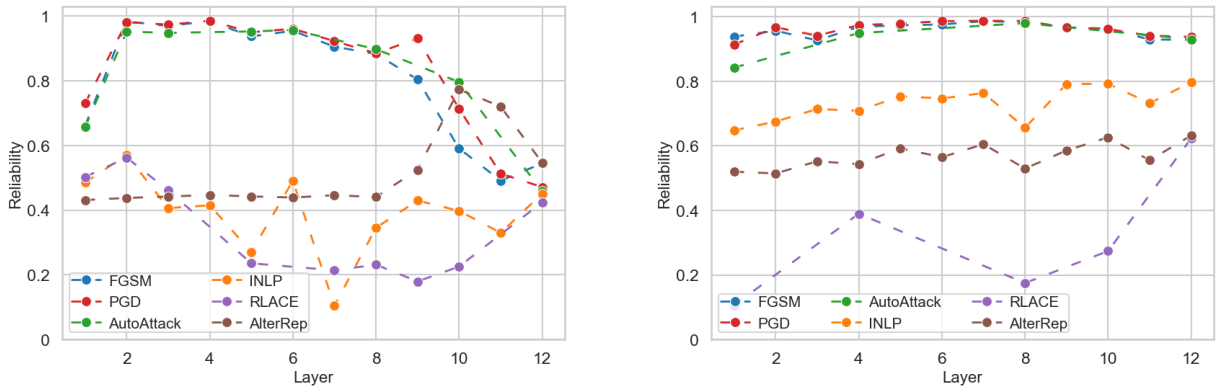


(e) **Llama-3.2-3B-Instruct**

Figure 43: **Maximum reliability by layer** across models for each intervention across all layers.

C.3.3 Linear Validation Probes

We present the reliability by layer for BERT and Pythia-160M using *linear validation probes* in Figures 44a and 44b (respectively). The main trends (specifically, reliability ordering of methods by layer) shown here are very similar to those using MLP validation probes, as shown in Figures 7 and 43a, with the exception that the linear counterfactual method (AlterRep) does not surpass the reliability of GBIs as strongly in the later layers (for BERT). The BERT result is not especially surprising, as linear validation probes are expected to be less resilient to linear interventions than MLP validation probes (as MLPs can also rely on nonlinearly-encoded information to make predictions) leading to lower selectivity and correspondingly lower reliability using linear interventions with linear validation probes compared to evaluations using nonlinear validation probes. However, it is important to note that the overall ordering of methods, and the specific scores observed, are still remarkably similar between linear vs nonlinear validation probes for both models, indicating that the differences in reliability between linear and nonlinear methods are unlikely to be due to the (non)linearity of validation probes.

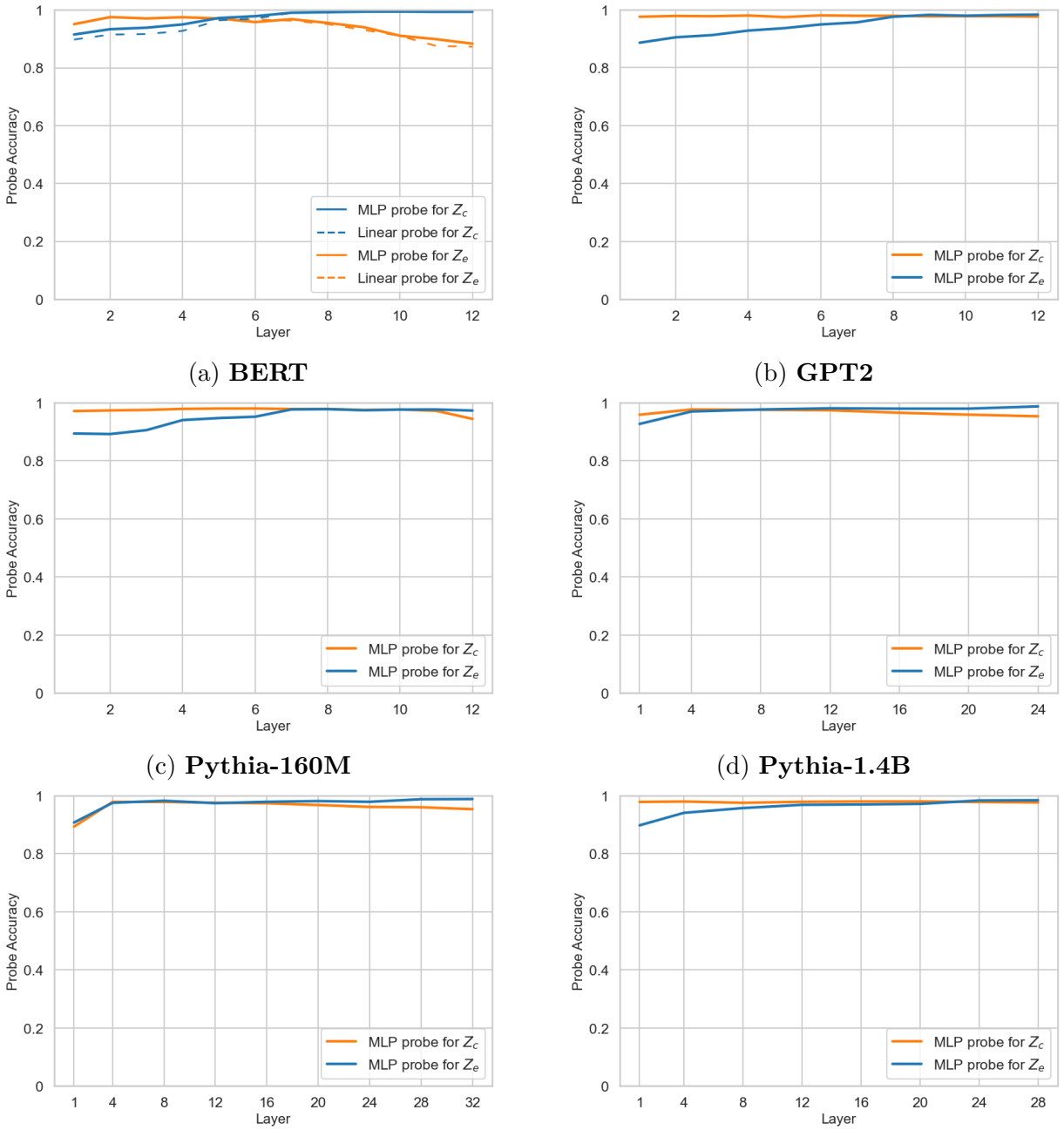


(a) **BERT** (b) **Pythia-160M**

Figure 44: **Maximum reliability by layer** for each intervention across all layers, using *linear validation probes*.

C.3.4 Validation Probe Accuracy by Layer

In Figure 45, we report the layerwise validation probe accuracy across all models.



(a) **BERT**

(b) **GPT2**

(c) **Pythia-160M**

(d) **Pythia-1.4B**

(e) **Pythia-6.9B**

(f) **Llama-3.2-3B-Instruct**

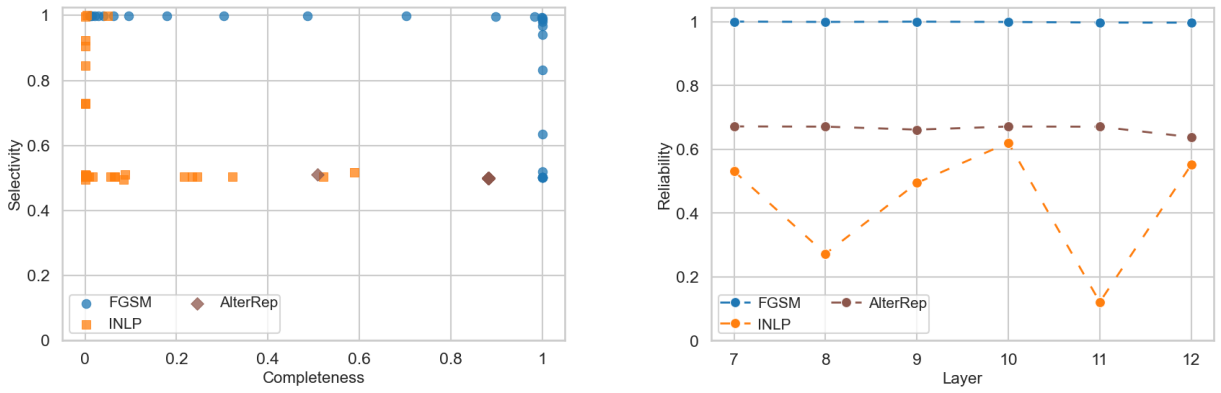
Figure 45: **Validation Probe Accuracy by Layer** across models.

C.3.5 Results on Indirect Object Identification (IOI) Task

The Indirect Object Identification (IOI) task (Wang et al., 2023b) is a well-studied benchmark in mechanistic interpretability research, requiring models to identify the correct referent in sentences with multiple names. Each sentence has an initial dependent clause introducing two names (e.g., "After John and Mary went to the store..."), followed by a main clause where one person performs an action involving the other (e.g., "... John gave a bottle of milk to..."). The model must correctly complete the sentence with "Mary" (i.e., the indirect object) rather than repeating "John".

For our experiments, we define the causal variable Z_c as denoting whether the first or second mentioned person in the sequence is the correct indirect object (each label has probability 0.5 in this dataset), and the environmental variable Z_e as the tense of the root verb (e.g., "gives" vs "gave"), which is clearly irrelevant to solving the task.

Following Zhang and Nanda (2024), we study IOI in the context of GPT2. Figure 46 shows the relation between selectivity and completeness as well as the reliability of the various interventions across layers. (Note that we only display results on layers 7–12 because the earlier layers do not encode necessary information to predict the variables of interest – i.e., in these layers, we cannot train a probe to predict the target features at sufficiently high accuracy, as also observed by Zhang and Nanda, 2024.) The results are similar with the subject-verb agreement task on GPT2 in that FGSM (nonlinear) is more reliable than INLP and AlterRep (linear) at all layers; but on the IOI task, we find that AlterRep is more reliable than INLP at all layers (for subject-verb agreement, these methods' relative performances varied on GPT2).



(a) **Selectivity vs. Completeness** (b) **Reliability by Layer**

Figure 46: **Results on Indirect Object Identification (IOI) task.** Completeness, selectivity, and reliability for interventions using the IOI task and the GPT2 model.

Appendix D: Not Just Pretty Pictures

D.1 Author Contributions

The contributions of all authors are as follows:

- *All co-primary authors:* brainstormed experimental design; identified core focus points of the analyzis; and assisted in drafting the paper.
- *Adam Davies:* developed, tested, prototyped all experiments for, and analyzed/refined all approaches to: prompt generation, image filtering, and textual inversion; analyzed all and co-designed most experiments; and co-wrote the paper with Francesco.
- *Francesco Pinto:* created, coordinated, and led the team; brainstormed the high-level project idea; ran several large-scale experiments (DomainNet); analyzed all and co-designed most experiments; and co-wrote the paper with Adam.
- *Jianhao Yuan:* implemented and ran the vast majority of experiments; wrote up most results, plots, tables, and experimental details; and helped revise final draft.
- *Philip Torr:* provided general advice, and helped secure compute resources for key experiments.

D.2 Experiment Implementation

D.2.1 General Setup

Due to the speed limitation of generative models[44], we pre-generate all the augmentation images. For each image in the training set, we randomly selected k text prompts from the templates (see Section D.11). In the Single Domain Generalization experiments, we choose $k = 3$ (for PACS and OfficeHome) and $k = 5$ (for NICO++ and DomainNet) prompts for each image (i.e., one prompt from each target domain), whereas for the weakening spurious correlation experiments, we choose $k = 4$ prompts for each image to randomize the correlation between the causal and spurious features. Then for each prompt **one** image will be generated and saved as a corresponding augmented version of the original image. At training time, for each training image in the batch, one of its augmented versions will be

[44]Significant progress in generation speed has been performed from the first versions of Stable Diffusion to the one we have been using in this work. Accelerating diffusion models is an active area of research

Algorithm 1 Augmentation Algorithm

Input: Source Domain $\mathcal{D}_{\text{train}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, Target Domain Prior Knowledge $\mathcal{P}(\mathcal{C}_{\text{test}})$, PromptStrategy. GenerativeModel, Model $f(\theta, \mathbf{x})$
Output: Trained $f(\cdot)$

Pre-generation Stage :

for $(\mathbf{x}_i, y_i)$ in $\mathcal{D}_{\text{train}}$ **do**
  *# select **k** prompts for each original image*
  $Prompts = []$
  for _ in $range(k)$ **do**
    $Prompts.append(\text{PromptStrategy}(y_i, \mathcal{P}(\mathcal{D}_{\text{test}})))$
  end for
  *# generate **one** sample for each prompt*
  $[\hat{\mathbf{x}}_i] \leftarrow \text{GenerativeModel}(Prompts\ , \mathbf{x}_i)$
  Save augmented samples in $\mathcal{S} = \{\mathbf{x}_i : [\hat{\mathbf{x}}_i]\}_{i=1}^{N}$
end for

Training Stage :

for Batch $(\mathbf{x}_i, y_i)_{i=1}^{m}$ in $\mathcal{D}_{\text{train}}$ **do**
  $(\mathbf{x}_i, \hat{\mathbf{x}}_i) \leftarrow \text{Concat}(\mathbf{x}_i, \text{RandomSelect}(\mathcal{S}[\mathbf{x}_i]))$
  $\text{TrainingStep}(\mathbf{x}_i, \hat{\mathbf{x}}_i)$
end for

randomly selected from the k pre-generated intervened samples. The general augmentation pipeline is shown in Algorithm 1.

On efficiency, we note that, given a dataset with N traing samples, the generated interventional data will have a size of $N \times k$, where k ranges from $3 \sim 5$ (depending on the experiment). As such, the number of generated samples is generally low (given N is often of a few thousands) with respect to the amount of samples generated by baseline augmentation techniques (which is $N \times e$ where e represents the number of training epochs, and typically[45] $e >> k$). Although baselines produce more augmentations of the same image, our technique requires fewer augmentations per image to attain superior performance as `SDEdit` can intentionally target specific types of interventions.

We report a few statistics about the training, validation and test set sizes as well as the number of classes for each dataset in Table 29. We use the model checkpoint of the last epoch to measure the test accuracy. For the experiments, as typical in the literature, we use pre-trained models on ImageNet for the backbones. To reproduce the experiment, we make part of our implementation available in the following anonymous repository.

[45] E.g., in our experiments, $e = 50$.

	Art	Photo	Sketch	Cartoon	Average
ERM	74.8	39.67	48.12	72.37	58.74
MixUp	67.14	39.57	33.24	63.27	50.81
CutMix	68.46	36.5	31.99	67.2	51.04
AugMix	68.88	38.75	43.89	76.86	57.09
RandAugment	69.07	44.48	49.36	72.31	58.8
CutOut	69.19	37.77	40.72	71.77	54.86
RSC	71.18	41.04	46.56	72.17	57.74
MEADA	70.32	39.55	44.94	74.03	57.21
PixMix	69.49	47.5	54.72	77.06	62.19
L2D	84.07	51.06	50.94	77.12	65.8
ACVC	72.65	43.33	60.35	78.98	63.83
VQGAN-CLIP(M)	78.09	54.38	53.78	77.76	66.00
Retrieval	81.22	75.49	83.36	83.24	80.83
SDEdit(M)	82.27	58.87	72.76	81.93	73.96
SDEdit(H)	84.23	61.7	70.31	82.74	74.75
SDEdit(LEC)	81.08	62.04	62.19	82.18	71.87
SDEdit(LEM)	83.21	58.74	66.45	82.37	72.69
Text2Image(LEM)	80.59	68.82	83.58	85.27	79.56
Text2Image(M)	83.17	71.31	87.42	87.12	82.26
ControlNet(M)	77.07	54.64	75.78	81.81	72.32
Textual Inversion	78.57	67.67	68.66	83.9	74.7
InstructPix2Pix	76.08	56.22	50.79	78.39	65.37

Table 9: Single Domain Generalization (SDG) PACS result with ResNet-18.

D.2.2 Single Domain Generalization

We set up the experiment under the standard Single Domain Generalization paradigm. For **PACS**, **Office-Home**, **NICO++**, and **DomainNet**, we train a model for each single domain and evaluate it on the remaining unseen domains to measure the test accuracy. For the first two datasets, we generate **three** augmented samples each one of them corresponds to one target domain. For the latter two, we similarly generate **five**. For all the datasets, we use an image size of 224×224. The full experiment results with expanded test accuracy one each test domain for PACS/OfficeHome/NICO++/DomainNet are shown in Table 9/Table 11/Table 13/ Table 15 for ResNet-18; Table 10/Table 12 / Table 14/Table 16 for ResNet-50. The visualized comparison between traditional data augmentation and generative model-based image editing as well as comparison among different types of conditional generation strategy is shown in Figure 47 and Figure 48, respectively. An overall comparison between different editing techniques is also presented in Table 17.

	Art	Photo	Sketch	Cartoon	Average
ERM	74.44	48.78	50.89	73.74	61.96
MixUp	66.31	42.98	45.64	77.76	58.17
CutMix	72.53	40.03	44.72	76.72	58.5
AugMix	75.8	51.32	49.99	81.42	64.63
RandAugment	71.38	46.8	55.95	76.33	62.61
CutOut	76.67	42.69	48.93	75.2	60.87
RSC	73.15	53.47	51.11	80.58	64.58
MEADA	73.72	48.78	59.81	73.84	64.04
PixMix	77.33	55.58	52.42	83.15	67.12
L2D	77.33	58.41	58.14	81.7	68.89
ACVC	79.63	52.76	58.13	81.4	67.98
SDEdit(M) ×	81.21	57.54	80.60	84.76	76.03
SDEdit(O-M)	82.59	65.44	79.3	83.64	77.74
Retrieval	82.36	76.24	87.0	86.01	82.9
SDEdit(M)	82.67	62.94	73.78	86.33	76.43
SDEdit(H)	83.68	64.22	78.95	84.63	77.87
SDEdit(LEC)	82.69	59.48	77.76	85.57	76.38
SDEdit(LEM)	84.11	59.1	73.39	86.0	75.65
Text2Image(LEM)	82.11	68.08	87.55	87.71	81.36
Text2Image(M)	84.15	72.9	90.51	87.34	83.72
ControlNet(M)	75.65	56.47	81.83	84.01	74.49
Textual Inversion	76.15	68.36	76.66	87.89	77.27
InstructPix2Pix	76.87	54.47	54.7	81.25	66.82

Table 10: Single Domain Generalization (SDG) PACS result with ResNet-50. Columns are Single source domains; accuracies are the average test accuracy of the three remaining target domains when training using the indicated source domain (best accuracies are in bold).

	Art	Clipart	Product	Real	Average
ERM	57.43	50.83	48.9	58.68	53.96
MixUp	50.41	43.19	41.24	51.89	46.68
CutMix	49.17	46.15	41.2	53.64	47.54
AugMix	56.86	54.12	52.02	60.12	55.78
RandAugment	58.07	55.32	52.02	60.82	56.56
CutOut	54.36	50.79	47.68	58.24	52.77
RSC	53.51	48.98	47.16	58.3	51.99
MEADA	57.0	53.2	48.81	59.21	54.55
PixMix	53.77	52.68	48.91	58.68	53.51
L2D	52.79	48.97	47.75	58.31	51.95
ACVC	54.3	51.32	47.69	56.25	52.39
Retrieval	65.02	63.55	60.51	64.32	63.35
SDEdit(M)	60.72	54.95	52.47	61.26	57.35
SDEdit(H)	58.15	55.12	51.94	61.24	56.61
SDEdit(LEC)	58.43	54.96	50.64	60.93	56.24
SDEdit(LEM)	57.27	53.97	49.02	60.5	55.19
Text2Image(LEM)	59.8	61.88	55.13	58.24	58.76
Text2Image(M)	62.77	64.57	57.51	61.2	61.51
ControlNet(M)	59.59	59.58	54.94	62.14	59.06
InstructPix2Pix	56.2	51.58	49.85	59.96	54.4

Table 11: SDG OfficeHome result with ResNet-18.

	Art	Clipart	Product	Real	Average
ERM	63.62	61.32	56.85	65.99	61.94
MixUp	63.46	59.2	54.97	64.21	60.46
CutMix	59.3	54.45	51.9	63.0	57.16
AugMix	63.99	61.11	58.88	66.44	62.6
RandAugment	64.92	61.38	59.34	66.42	63.02
CutOut	62.15	58.24	55.77	63.98	60.03
RSC	60.91	56.86	54.21	64.41	59.1
MEADA	64.48	61.6	57.34	64.89	62.08
PixMix	63.54	60.34	57.29	64.54	61.43
L2D	60.79	55.01	54.76	62.93	58.37
ACVC	62.33	57.76	55.59	64.02	59.92
Retrieval	71.15	71.46	67.21	70.73	70.14
SDEdit(M)	67.08	64.48	60.01	67.06	64.66
SDEdit(H)	64.55	63.05	58.99	66.48	63.27
SDEdit(LEC)	65.96	63.12	58.6	66.03	63.43
SDEdit(LEM)	64.86	62.88	58.46	66.37	63.14
Text2Image(LEM)	65.72	68.8	62.61	64.09	65.31
Text2Image(M)	68.6	71.11	63.83	66.98	67.63
ControlNet(M)	66.52	66.65	62.12	66.47	65.44
InstructPix2Pix	63.34	60.39	58.43	67.13	62.32

Table 12: SDG OfficeHome result with ResNet-50.

	autumn	dim	grass	outdoor	rock	water	Average
ERM	57.07	60.95	62.4	61.82	58.52	65.04	60.97
RandAugment	57.19	60.51	61.23	61.77	58.67	64.08	60.57
AugMix	56.19	59.18	61.29	60.72	58.1	63.16	59.77
MixUp	57.15	59.52	62.77	62.71	59.47	65.36	61.16
CutOut	57.42	59.07	60.33	61.07	58.48	62.5	59.81
PixMix	57.55	58.38	61.36	61.62	58.68	63.85	60.24
RSC	54.61	57.47	60.14	60.25	57.32	61.86	58.61
ACVC	53.43	54.91	58.94	59.07	56.11	58.67	56.85
MEADA	57.7	60.17	62.32	62.27	59.53	64.52	61.09
L2D	51.88	53.79	57.15	58.48	53.92	58.55	55.63
Retrieval	56.69	60.31	61.58	62.51	58.06	63.57	60.45
SDEdit(M)	58.17	60.48	62.72	62.16	59.95	64.66	61.36
SDEdit(H)	59.14	61.48	63.95	64.14	60.84	66.15	62.62
SDEdit(LEC)	62.54	65.97	67.01	67.85	64.15	69.85	66.23
SDEdit(LEM)	62.11	65.11	66.12	67.25	63.49	69.43	65.59
Text2Image(M)	58.89	63.79	63.56	64.85	58.9	66.3	62.72
ControlNet(M)	58.36	62.43	64.13	63.29	59.49	65.12	62.14
InstructPix2Pix	57.01	58.36	61.53	61.76	58.59	63.73	60.16

Table 13: SDG NICO++ Result with ResNet-18.

	autumn	dim	grass	outdoor	rock	water	Average
ERM	66.74	70.37	72.05	71.3	66.58	72.64	69.95
RandAugment	67.23	71.43	70.81	70.62	66.47	72.71	69.88
AugMix	66.18	69.21	70.03	70.22	65.51	71.72	68.81
MixUp	67.6	70.3	72.47	72.26	67.12	74.01	70.63
CutMix	62.82	67.6	69.39	69.01	63.59	69.78	67.03
CutOut	66.76	69.34	70.13	70.13	66.67	72.33	69.23
PixMix	66.99	68.75	69.57	71.72	67.1	72.75	69.48
RSC	63.96	67.69	68.48	69.21	63.96	70.94	67.37
ACVC	63.74	67.48	67.73	68.71	63.89	69.95	66.92
MEADA	67.47	69.99	71.72	71.31	65.76	73.06	69.89
L2D	61.81	64.44	66.78	66.67	63.42	68.02	65.19
SDEdit(M) ×	67.90	71.42	72.61	72.32	67.10	73.79	70.90
Retrieval	67.39	72.16	71.53	71.83	66.19	73.45	70.42
SDEdit(M)	68.28	71.42	72.68	72.31	67.95	74.07	71.12
SDEdit(H)	69.13	72.13	73.64	73.33	68.46	75.03	71.95
SDEdit(LEC)	70.21	74.68	75.05	75.54	69.74	76.89	73.69
SDEdit(LEM)	70.36	74.4	74.48	75.09	69.83	77.47	73.61
Text2Image(M)	68.14	72.67	72.63	73.77	66.88	75.17	71.54
ControlNet(M)	67.69	72.64	73.19	72.84	67.73	74.92	71.5
InstructPix2Pix	66.86	70.35	72.02	71.95	67.13	73.31	70.27

Table 14: SDG NICO++ Result with ResNet-50.

	clipart	infograph	painting	quickdraw	real	sketch	Average
ACVC	25.31	19.86	25.23	8.0	27.49	26.84	22.12
AugMix	25.58	19.09	24.74	7.41	26.41	27.03	21.71
CutMix	23.56	17.83	23.0	4.33	25.36	25.04	19.85
CutOut	24.44	19.27	24.16	6.03	25.45	25.31	20.78
ERM	24.29	19.93	24.32	6.08	25.42	25.54	20.93
L2D	23.55	17.26	23.69	6.24	26.33	24.17	20.21
MEADA	24.6	20.06	24.5	6.17	25.52	25.56	21.07
MixUp	24.25	19.46	23.31	5.51	26.18	25.34	20.68
PixMix	26.39	19.18	25.28	3.49	27.9	24.89	21.19
RSC	22.92	18.21	22.52	6.11	24.72	23.59	19.68
RandAugment	25.99	18.88	25.12	6.83	27.08	25.71	21.60
SDEdit(H)	28.03	31.68	29.27	12.66	29.78	31.22	27.11
SDEdit(LEC)	27.33	30.56	28.96	11.35	29.6	30.65	26.41
SDEdit(LEM)	27.16	29.92	28.97	10.74	29.6	30.12	26.08
SDEdit(M)	27.88	31.62	29.57	12.3	30.03	30.94	27.06
Text2Image(M)	34.12	35.32	31.68	36.13	33.21	36.43	34.48
ControlNet(M)	28.19	23.40	27.59	18.81	29.28	31.62	26.48

Table 15: SDG DomainNet Result with ResNet-18.

	clipart	infograph	painting	quickdraw	real	sketch	Average
ACVC	29.84	26.72	29.86	8.96	31.88	31.47	26.46
AugMix	30.04	26.1	29.48	8.92	31.07	31.61	26.20
CutMix	28.9	24.29	27.92	5.99	29.97	29.75	24.47
CutOut	28.98	25.8	28.71	6.6	29.96	29.36	24.90
ERM	29.06	27.07	28.87	6.92	29.85	29.8	25.26
L2D	28.15	23.85	28.61	7.12	31.25	29.53	24.75
MEADA	29.09	26.77	28.81	6.81	30.06	30.05	25.26
MixUp	29.34	26.89	29.17	6.46	30.66	30.42	25.50
PixMix	30.77	26.96	29.95	3.68	32.94	28.87	25.53
RSC	26.89	24.12	26.48	5.79	28.7	27.96	23.32
RandAugment	30.28	26.51	29.96	8.31	31.82	30.14	26.17
SDEdit(H)	32.89	37.95	33.99	15.89	34.45	35.77	31.82
SDEdit(LEC)	32.35	37.14	33.83	15.62	34.32	35.39	31.44
SDEdit(LEM)	31.84	36.75	33.72	13.88	34.19	35.24	30.94
SDEdit(M)	33.09	38.13	33.99	15.86	34.64	35.94	31.94
InstructPix2Pix	30.63	27.29	30.04	14.70	32.27	30.99	27.65
ControlNet(M)	32.66	30.72	31.93	14.92	33.66	36.15	30.01
Text2Image(M)	39.05	41.28	36.78	48.422	38.18	40.89	40.77

Table 16: SDG DomainNet Result with ResNet-50.

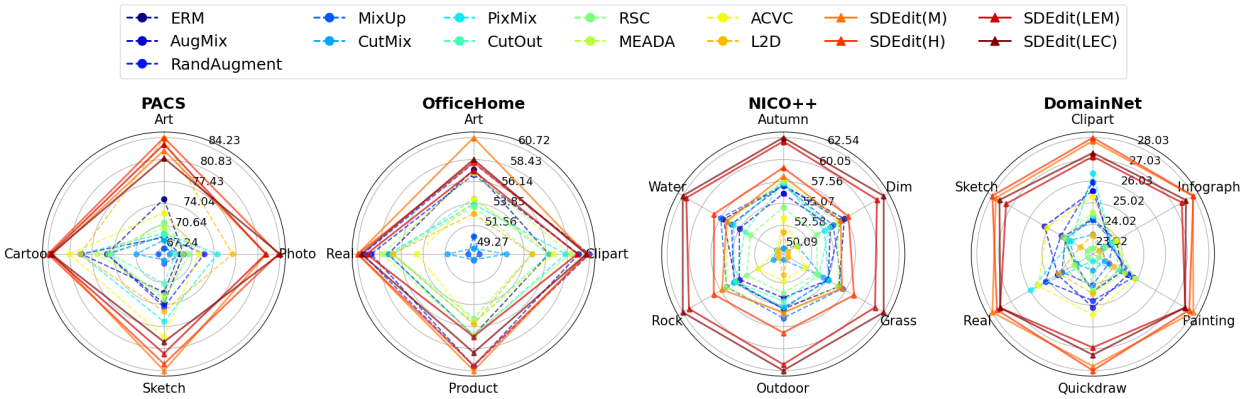


Figure 47: **Single Domain Generalization (SDG) Performance** results in comparison with baselines (dashed lines) and OURS (solid lines) using ResNet-18.

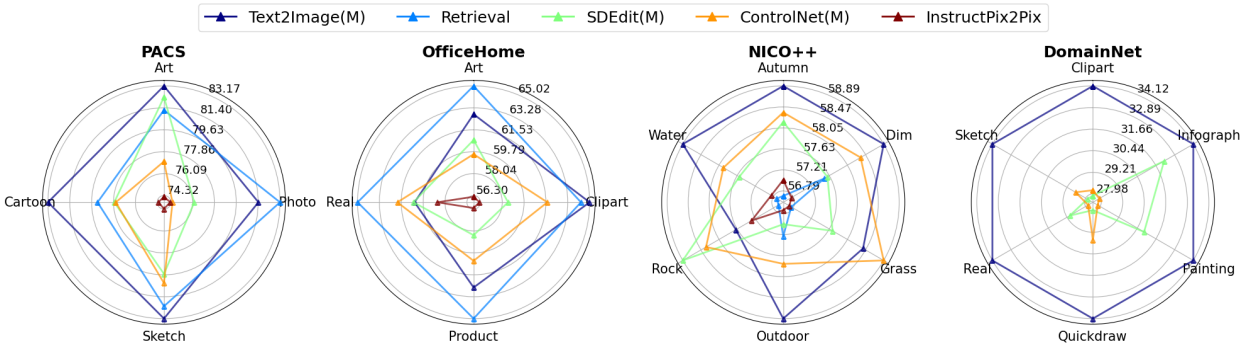


Figure 48: Comparison Between Different Condition Generation Strategy using ResNet-18.

D.2.3 SDG Large Models Ablation

To show our method can scale to larger models, we perform an ablation study with ConvNeXt-L (Liu et al., 2022b) (198M parameters) on PACS SDG experiment. As shown in Table 18 (average) Table 19 (detailed), we observe our method and still outperform all the baselines by a considerable margin. This proves the applicability of our method to different sizes of models.

D.2.4 Effect of Accessing Multiple Source Domain

We also present further investigation on the effect of accessing multiple source domains, potentially including the target test domain. We experimented on three settings: (a) MDG: classifier trained on all but the target domain, which is the standard set-up for multi-domain generalization, where multiple domains of source data are used for training and one unseen domain is used for testing; (b) All: classifier trained on all the domains, (c) Target: classifier trained only on the target domain. As shown in Table 20, While ERM(Target/All) achieves

	PACS	OfficeHome	NICO	DomainNet
SDEdit(M)	76.43	64.66	71.12	31.94
Text2Image(M)	83.72	67.63	71.54	40.77
ControlNet(M)	74.49	65.44	71.50	30.01
InstructPix2Pix	66.82	62.32	70.27	27.65
Retrieval	82.90	70.14	70.42	31.07

Table 17: **Comparison Between Editing and Condition Strategies.**

	Art	Photo	Sketch	Cartoon	Average
ERM	79.8	60.62	50.76	86.53	69.43
AugMix	82.03	59.95	74.62	86.41	75.75
RandAugment	74.17	56.5	72.06	84.36	71.77
MixUp	79.82	60.74	55.45	84.35	70.09
CutMix	79.8	51.13	65.45	88.22	71.15
CutOut	81.41	55.55	71.8	85.16	73.48
RSC	84.26	55.3	76.15	86.13	75.46
MEADA	80.43	63.82	70.51	83.97	74.68
ACVC	83.68	68.86	77.99	85.03	78.89
PixMix	83.18	68.39	66.32	85.4	75.82
L2D	86.03	75.9	69.84	89.37	80.28
SDEdit(M)	86.53	74.5	**89.22**	**88.98**	84.81
SDEdit(H)	**88.31**	**81.64**	86.3	88.19	**86.11**
SDEdit(LEC)	87.7	75.05	85.77	87.04	83.89
SDEdit(LEM)	87.14	81.16	79.26	86.31	83.47

Table 18: SDG PACS result with ConvNeXt-L.

Source Domain	art			photo			sketch			cartoon		
Target Domain	photo	sketch	cartoon	art	sketch	cartoon	art	photo	cartoon	art	photo	sketch
ERM	98.86	71.93	68.6	75.78	51.39	54.69	40.82	53.65	57.81	89.75	93.65	76.2
AugMix	97.96	77.6	70.52	75.93	42.35	61.56	77.34	67.6	78.92	88.96	93.11	77.17
RandAugment	97.9	61.29	63.31	77.73	37.54	54.22	71.97	70.36	73.85	86.52	93.89	72.66
MixUp	99.04	68.57	71.84	81.1	43.62	57.51	49.46	54.91	61.99	85.35	95.93	71.77
CutMix	99.1	70.2	70.09	76.76	30.54	46.08	60.21	66.95	69.2	90.09	96.47	78.09
CutOut	98.62	74.32	71.29	76.32	39.09	51.24	72.51	70.36	72.53	86.62	93.11	75.74
RSC	98.98	79.54	74.27	76.61	40.75	48.55	69.34	80.96	78.16	85.35	93.77	**79.28**
MEADA	98.92	72.66	69.71	76.27	52.56	62.63	61.08	76.77	73.68	86.96	92.99	71.95
ACVC	98.08	80.53	72.44	81.84	65.44	59.3	83.69	74.01	76.28	87.3	92.93	74.85
L2D	98.98	**83.1**	76.02	80.86	**80.02**	66.81	68.6	68.86	72.06	**91.55**	97.13	79.43
PixMix	**99.46**	75.16	74.91	79.98	62.26	62.93	62.84	65.27	70.86	86.13	92.1	77.98
SDEdit(M)	99.34	76.43	83.83	82.96	64.04	76.49	**87.79**	92.28	**87.59**	91.06	**97.19**	78.7
SDEdit(H)	99.1	81.67	**84.17**	85.89	78.54	80.5	82.86	92.69	83.36	91.06	96.11	77.4
SDEdit(LEC)	98.26	82.59	82.25	85.64	67.96	71.54	85.4	**92.93**	78.97	87.94	94.61	78.57
SDEdit(LEM)	99.4	79.33	82.68	**86.77**	74.42	**82.3**	77.64	84.19	75.94	87.35	95.09	76.48

Table 19: PACS result with ConvNeXt-L

	Art	Photo	Sketch	Cartoon	Average
ERM (Target)	99.65	99.94	99.64	99.66	99.72
ERM (All)	99.71	99.70	99.84	99.60	99.74
ERM (MDG)	80.01	96.28	73.86	76.28	81.61
OURS(MDG)	87.5	95.75	79.21	85.2	86.91

Table 20: Impact of Assessing Multiple Real/Synthetic Domain.

almost perfect performance, this is expected as the training source domain includes the target test domain. Note that the accuracy in the table below is not directly comparable to SDG in the main paper since our main setting is Single Domain Generalization (SDG), where we have a single source domain for training, and the accuracy reported is the average test accuracy on multiple unseen test domains. However, here we have a single unseen target domain for testing. To provide a direct comparison between ERM and SDEdit, we experiment with SDEdit under the same setting in MDG with a minimal prompt. We demonstrate under MDG setting SDEdit also leads to significant performance improvement in all unseen test domains.

D.3 Weaken Spurious Correlation

In the three considered cases, the reliance on the spurious correlation is measured as: (1) **ImageNet-9** (Background Bias): **Gap**, as defined in (Xiao et al., 2020), is the difference between the accuracies measured on the test sets `mixed same` and `mixed rand`. (2) **CCS Dataset** (Texture Bias): **Texture Bias**, as defined in (Geirhos et al., 2018), is the number of correct texture classifications over sum of the true positive texture and shape classifications. In the test CCS dataset, each image is synthesized with a texture and subject from different classes (i.e texture: elephant, class: cat). The true positive texture classification is the percentage of cases in which the model predicts the texture label correctly; similarly, true positive shape classification is the percentage of correctly classified shape labels. (3) **CelebA-sub** (Demographic Bias): **RandGap** and **FlipGap** represent the accuracy gap between **I.I.D** distribution to `rand` and `flip` respectively. The purpose is to measure the reliance on the spurious feature both for the average case (i.e., randomizing the spurious correlation in the test set) and in the worst case (i.e., the test set flips the spurious correlation). For all the three dataset each original image sample will have **four** pre-generated augmented samples. The comparison with all the baselines is in with ResNet-18 Table 21, Table 23, and Table 22.

	I.I.D. Test	Mixed Rand	Mixed Same	**Gap** (↓)
ERM	95.16	73.54	86.02	12.48
MixUp	94.62	67.63	83.91	16.28
CutMix	95.36	65.21	84.77	19.56
AugMix	95.16	74.73	87.72	12.99
RandAugment	96.69	78.20	90.44	12.25
CutOut	95.46	71.10	85.41	14.31
RSC	94.12	74.72	84.39	9.68
MEADA	95.56	74.74	87.43	12.69
PixMix	97.04	79.76	91.96	12.20
ACVC	93.97	76.38	88.16	11.77
L2D	92.84	73.04	84.10	11.06
SDEdit(H)	91.85	73.33	82.96	9.63
Text2Image(H)	90.12	69.63	75.8	6.17
ControlNet(H)	91.85	75.19	85.68	10.49
InstructPix2Pix	92.84	78.89	88.15	9.26
Retrieval	91.6	73.83	80.12	6.29

Table 21: ImageNet-9 result with ResNet-18

	I.I.D. Test	Random	**Texture Bias** (↓)
ERM	81.75	18.77	72.45
MixUp	77.36	19.23	71.69
CutMix	79.96	15.64	77.16
AugMix	82.2	20.08	72.42
RandAugment	83.09	18.9	72.2
CutOut	81.85	17.81	74.19
RSC	79.9	20.48	71.11
MEADA	81.97	19.5	71.14
PixMix	80.91	26.86	64.64
ACVC	81.13	29.33	59.25
L2D	80.06	23.55	62.12
SDEdit(H)	85.94	31.48	55.46
Text2Image(H)	86.44	35.23	51.21
ControlNet(H)	84.13	21.88	62.58
InstructPix2Pix	79.75	26.17	53.58
Retrieval	85.85	33.91	51.94

Table 22: Texture result with ResNet-18

	I.I.D. Test	Flip	Random	**FlipGap** (↓)	**RandGap** (↓)
ERM	99.44	77.16	88.48	22.28	11.32
MixUp	99.16	79.4	88.86	19.76	9.46
CutMix	99.24	74.82	86.92	24.42	12.1
AugMix	99.56	76.42	87.82	23.14	11.4
RandAugment	99.04	77.62	88.96	21.42	11.34
CutOut	99.48	78.24	89.72	21.24	11.48
RSC	99.52	81.7	91.8	17.82	10.1
MEADA	99.48	77.24	89.08	22.24	11.84
ACVC	99.16	79.5	89.58	19.66	10.08
PixMix	99.32	76.62	88.34	22.7	11.72
L2D	99.12	81.96	90.96	17.16	9.0
Retrieval	98.6	77.9	89.3	20.7	11.4
SDEdit(H)	99.2	86.6	92.5	12.6	5.9
Text2Image(H)	98.8	90.0	93.6	8.8	3.6
ControlNet(H)	99.2	89.3	93.5	9.9	4.2
InstructPix2Pix	99.2	86.9	93.5	12.3	6.6

Table 23: CelebA-sub result with ResNet-18

D.3.1 Additional Experiment on Cifar-10-C

We conduct further experiments with the Cifar-10-C dataset. We adopt a similar setting as RRSF, where we train on Cifar-10 and test on Cifar-10-C. In the evaluation, domain shifts were organized into distinct groups for clarity. The following classifications were made:

- Blurring Effects: defocus blur, gaussian blur, glass blur, motion blur, zoom blur
- Noise Variations: gaussian noise, impulse noise, shot noise, speckle noise
- Compression Artifacts: JPEG compression
- Image Transformations: brightness, contrast, elastic transform, pixelate, saturate

	Blurring Avg.	Noise Avg.	Compression Avg.	Transformations Avg.	Overall Avg.
ERM	72.04	74.01	75.71	73.93	73.55
MixUp	73.79	76.22	77.46	75.72	75.48
PixMix	75.84	79.41	81.39	79.32	78.63
SDEdit(M)	74.28	75.71	77.10	75.02	75.11

Table 24: Average performance of algorithms across grouped domain shifts.

As shown in Table 24, SDEdit still demonstrate effectiveness under various parametric domain shift. Although the generalization performance is inferior to other parametric augmentation methods, it can be used in a combined manner.

	PACS	OfficeHome	NICO++	ImageNet-9	Texture	CelebA-sub
Epoch	50	50	50	30	30	30
Batch size	64	64	64	64	64	64
Warmup Epoch	5	5	5	5	5	5
Warmup Type	sigmoid	sigmoid	sigmoid	sigmoid	sigmoid	sigmoid
Weight Decay	5e-4	5e-4	5e-4	5e-4	5e-4	5e-4
Nesterov	True	True	True	True	True	True
Learning rate	1e-3	3e-3	3e-3	1e-3	1e-3	1e-3
Scheduler	Step	Step	Step	Step	Step	Step
Decay Step	45	45	45	27	27	27
Learning rate decay	0.1	0.3	0.3	0.1	0.1	0.1

Table 25: Training Hyperparameters

	PACS	OfficeHome	NICO++	DomainNet	ImageNet-9	CelebA	Texture
Inference Step	30	30	30	30	30	30	30
Image Strength	0.75	0.75	0.75	0.75	0.75	0.75	0.75
Guidance Scale	2.0	2.0	2.0	2.0	2.0	2.0	2.0
Sampler	UniPC	UniPC	UniPC	UniPC	UniPC	UniPC	UniPC

Table 26: Generator hyperparameters for each dataset

D.3.2 Hyperparameters

Training Hyperparameter For all the other baselines, we use the value as proposed in their original papers or official implementation.

Generator Hyperparameter For the two types of generative models, we use the hyperparameters for each dataset as shown in Table 26. Hyperparameters are tuned based on human judgment of few-shot image manipulation quality, without downstream task accuracy-based evaluation. Unspecified hyperparameters are set to their default value. For Stable Diffusion, We use "`Runmyml/stable-diffusion-v1-5`" pre-trained model. The training hyperparameters for setting are specified as shown in Table 25

D.4 Prompting Strategies

Hwere we detail how the prompts were obtained.

- ***Domain expert (H)***: a collection of 1-8 simple "handcrafted" prompts per image domain (e.g., "an ink pen sketch of a(n) `class`"), authored by a human given only the domain descriptions provided by the respective benchmarks, without looking at any samples from the target domain.
- ***Language enhancement (LE)***: following (He et al., 2023), we use the T5 language

model (Raffel et al., 2020) fine-tuned on CommonGen (Lin et al., 2020)[46] to generate 1-8 prompts using only the domain and class labels as inputs. Two strategies, Conservative (LE_C) and Moderate (LE_M), are used: LE_C deterministically generates consistent, high-probability outputs; and LE_M is built to balance prompt diversity with quality. For both strategies, we use a T5 (Raffel et al., 2020) model that is pre-trained on both unsupervised language modeling of web text and supervised text-to-text language modeling tasks[47], then fine-tuned on CommonGen[48] (Lin et al., 2020). (We refer to this model as T5_CG.) CommonGen is a constrained-generation task whose objective is to generate a sentence describing a commonplace scenario that contains all words[49] provided in an input word set. For example, given the words {dog, frisbee, catch, throw}, an acceptable output is "The dog catches the frisbee when the boy throws it." (Lin et al., 2020) We always provide T5_CG with a text input containing only a domain label and class label; for example, given a PACS image with domain `sketch` and class `elephant`, we simply feed T5_CG "sketch elephant" as input. For n number of prompts we will use to generate images, in LE_C, we simply use beam search decoding to generate prompts with $4n$ beams and select the top-n highest probability beams. In LE_M, we use a conjunction of top-k and top-p (nucleus) sampling, with $k = 50$ and $p = 0.95$, returning n sampled prompts. We experimented with other decoding configurations, but found that increasing prompt diversity (e.g., by increasing k, lowering p, or increasing temperature) consistently came at the cost of prompt quality.

- ***Textual Inversion*** (Gal et al., 2022): Given a set of images that share a common feature (e.g., belonging to the same class), this method learns an embedding in the text space that represents that feature. This embedding can be used to condition the generative process, thereby enhancing the generator's capability to reproduce that feature. Due to the computational cost associated with the additional training phase required by this approach, we limit its application to PACS. As shown in Table 9 and Table 10, Texual Inversion achieves 74.70% and 77.27% average accuracy for SDG. While outperforming all baseline methods, it is inferior to other relatively low-cost generative model-based strategies.

In order to yield the best IDA performance from a given T2I model, future work might consider strategies for directly optimizing prompts or utilizing human-in-the-loop prompt

[46] https://huggingface.co/mrm8488/t5-base-finetuned-common_gen

[47] Pre-trained model (not directly used in experiments): https://huggingface.co/t5-base

[48] Fine-tuned model used in experiments: https://huggingface.co/mrm8488/t5-base-finetuned-common_gen

[49] Synonyms and inflected forms are also allowed (e.g., given input "eat", outputs containing "consume" or "eaten" are valid).

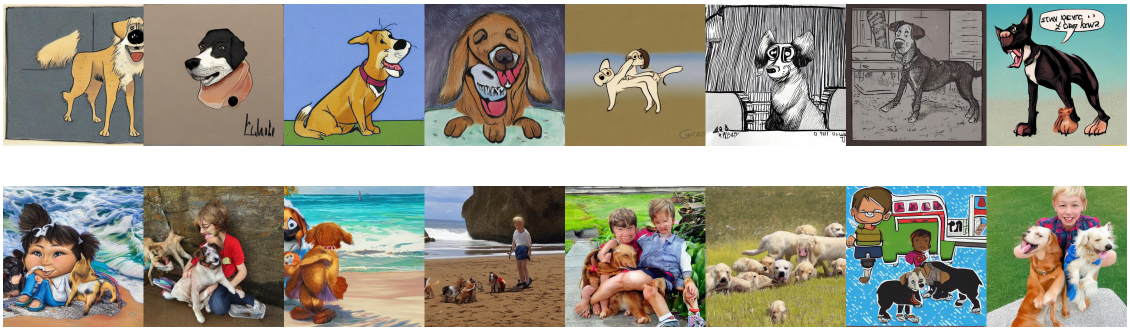


Figure 49: **CLIP Filtering Examples.** The most-similar (top) and least-similar (bottom) eight images according to their average percentile rank of CLIP similarity scores computed with respect to the provided prompts.

"debugging", as we discuss in Appendix D.6 and Section D.7 (respectively).

D.5 CLIP Filtering Details

For each image in the generated dataset, we compute its CLIP similarity with respect to both prompts. Since the distributions of similarity scores can differ in scale and location, we cannot simply average the two scores in order to quantify how well a sample represents a class and a domain. Therefore, we sort the scores to produce two rankings and associate each image to the average of the percentile rank with respect to both prompts. We then discard a fixed amount of images with the lowest average percentile rank. (See Appendix D.5.1 for an example of top- and bottom-ranked images.) After filtering out the worst 10%, 25%, or 50% of synthetic images, we train our classifier on the remaining data. The results are displayed in Figure 13. We find filtering to not yield consistent improvements across all the considered cases.

D.5.1 CLIP Filtering Examples

Figure 49 displays the best-matching (top) and worst-matching (bottom) synthetic images generated with `SDEdit` using LE_{M} of class `dog` and domain `cartoon` according to their average percentile rank of CLIP similarity scores with the prompts "an image of a dog" and "a cartoon". In general, we observe that the images on the top do indeed appear to be cartoons and contain dogs (if somewhat disfigured in a few cases); whereas it seems that most of the images on the bottom either resemble *photos* of dogs (images 2, 4, 5, 6, and 8) or cartoons (images 1, 3, and 7), but do not generally seem to match both the target domain and the correct class.

D.6 Fully automated applications

We examine a basic implementation of a fully-automated augmentation pipeline in the language enhancement (**LE**) experiments described above, finding that it sometimes achieved performance on-par with or exceeding that of the expert-handcrafted prompts. However, this language model is optimized to generate simple sentences describing commonplace scenarios (see Section D.4), not image-generation prompts. Thus, it is possible that fine-tuning language models to generate prompts that are better optimized for downstream T2I generators may yield superior results to expert-handcrafted prompts in many scenarios, making this approach a promising direction for future work. Another approach to improve fully-automated prompting involves continuous prompt optimization (also known as "prompt tuning" or "soft prompting"). Recently, these methods have been shown to outperform human-interpretable prompts for a variety of natural-language (Li and Liang, 2021; Liu et al., 2023b; Min et al., 2022; Khashabi et al., 2022) and vision-and-language (Gal et al., 2022; Zhou et al., 2022) tasks. These methods are not directly applicable to domain generalization because they require labeled samples to learn continuous prompts; but we suggest that they may be a promising fully-automated approach to domain adaptation tasks[50] where prompt interpretability is not necessary (cf. (Khashabi et al., 2022)).

D.7 Human-in-the-Loop Applications

D.7.1 Prompt interpretability enables human-in-the-loop debugging

Specifying interventions with natural language makes it possible to flexibly specify the type of manipulations desired. In the future, we expect practitioners could iteratively improve the collection of prompts to achieve improved performance.

We begin with a small set of handcrafted prompts (the ones used for the results reported in the main paper) and observe a decrease in texture bias of **5.44%**. Reasoning on the task at hand and the desired effect of the augmentations, we expand the prompts set to cover a broader range of textures to further decrease texture bias by an additional **2.84%** (see Section D.11 and Table 22).

More generally, it is possible to "debug" augmentations by directly analyzing prompts and modifying them to better reflect the desired intervention (which is possible with zero exposure to the target domain, or before augmented images are even generated). For example, the top prompts generated by $\textbf{LE}_{\textbf{M}}$ for OfficeHome's `art` domain and `computer` class include "art on a computer", "a man is working on a computer with a piece of art on

[50] I.e., where an unlabeled sample of the target domain is available to facilitate learning of the environmental features of the target domain (Ghifary et al., 2016).

it", etc., indicating that $\mathbf{LE_M}$ generated prompts describing scenarios where both the class and domain label refer to individual objects in a visual scene. In Section D.7.1, we describe a few simple steps that can automatically filter out many such prompts[51], illustrating the flexibility of the natural-language augmentation interface.

LE_M is prone to generating prompts that treat OfficeHome (Venkateswara et al., 2017) domain labels as objects, not as visual domains or styles. Fortunately, the interpretability of natural-language prompts that makes it possible for us to diagnose this problem also enables us to filter out many such prompts. One approach is to map domain labels to the visual conditions they denote: for example, the `Product` label may be replaced with "white background", `Real World` with "photograph", etc. However, this solution requires some knowledge about test domains, which may not always be available. Alternatively, image-related keywords like "image", "depict", or "style" can be included in the input issued to T5_CG, and outputs which do not place these additional terms in the same minimal noun phrase as the domain label can be removed (e.g., "an artistic depiction of a computer" or "a product image of a candle" would be kept, whereas "art depicted on a computer" or "a product and an image of a candle" would be excluded)[52]. While both of these strategies require limited human oversight to successfully "debug" prompts, more sophisticated fully-automated augmentation pipelines might learn to make such changes on their own, e.g., by integrating downstream image classifier accuracy as feedback to fine-tune prompt-generation models.

D.7.2 Other human-in-the-loop applications

The usage of T2I generators to approximate interventions facilitates a variety of novel use cases. For example, consider a "human-in-the-loop" (HITL) application context, where humans are available to provide interactive feedback to a model. In the HITL *active learning* paradigm, human experts perform the role of "oracles" that a model may "query" to provide labels of highly uncertain or novel inputs (Mosqueira-Rey et al., 2022). In contrast, the "human-in-the-loop debugging" paradigm elaborated above implements the *interactive machine teaching* paradigm (Ramos et al., 2020), which treats human collaborators as *teachers* that may provide interactive feedback to update the "curriculum"[53] of images used to train a model. For example, a human collaborator may observe that a model tends to

[51] However, as our LE experiments are explicitly intended to operate fully autonomously (i.e., with no human intervention or supervision), we do not carry out a full-scale "debugged" version of this experiment – all reported OfficeHome results are from the "buggy" prompts.

[52] For clarity, T5_CG can replace input terms with inflected forms in generated prompts, e.g., allowing input terms "art" and "depict" to occur as "artistic" and "depiction" (respectively) in outputs.

[53] Note that, in our case, a curriculum is defined in terms of the domains from which training examples are drawn, not the order in which they are presented (cf. (Bengio et al., 2009)).

	ERM	AugMix	RandAugment	MixUp	CutMix	RSC	L2D	ACVC	MEADA	OURS (online)
Time (s)	14.2	33.1	42.7	27.3	28.4	18.0	41.1	127.8	92.2	21.2

Table 27: Quantitative Comparison on Computation Time

perform poorly in the context of a given target domain, or that generated images do not capture some important stylistic properties of the domain. In response, they may easily compose or revise image-generation prompts with explicit reference to important features of the target domain. Critically, our approach allows human teachers to directly update visual curricula using natural language, providing models with feedback in much the same terms as one would a human student. We believe that the intuitiveness and efficiency of this approach makes it a promising approach to domain generalization, shifting the burden of the problem from human domain knowledge to natural language and thus enabling human collaborators to interactively instruct models without prerequisite domain expertise.

In particular, we argue that this benefit is particularly salient in the context of test-driven software engineering practice. Rather than blindly assuming that the performance on application-independent benchmarks will transfer to application-specific cases, engineers need to extensively document (often through natural language) the potential use cases and test conditions. The ability to directly specify these criteria via natural-language augmentations, or even directly reuse the documentation to generate training data, could be invaluable for controlling, predicting, and understanding the behavior of vision models in real-world applications.

D.8 Computational Expense

Although the inference speed of generative models has greatly improved over time, we found that SD is still too slow to generate synthetic data on-the-fly during training, so we pre-generate and store augmented data to amortize the generation cost when experimenting on different architectures and training procedures. For each sample in $\mathcal{D}^S$, we randomly selected k text prompts, and for each prompt, **one** augmented image was generated and stored. At training time, for each training image in the batch, one of its augmented versions will be randomly selected from the k pre-generated intervened samples. The general statistics of computational expense of each type of generative model on an NVIDIA A40 GPU and generator with hyperparameters specified for OfficeHome experiment are as follows: Stable Diffusion 1.5 took up $\sim$ 8GB of VRAM (for inference – we do not compute gradients for any experiments) and required $\sim$ 0.5 seconds per sample generated on average.

In addition to our qualitative assessments, we have conducted a quantitative comparison of various data generation methods, focusing on the time efficiency aspect. Specifically, we

	in	mixed rand	mixed same	gap
ERM	95.06	71.85	83.58	11.73
SDEdit(H)	95.06	77.65	85.8	8.15
Inpaint(H)	**96.05**	**80.62**	**87.16**	**6.54**

Table 28: Inpainting Result on ImageNet-9

	No.train	No.validation	No.test	No.classes
PACS	8977	1014	9991	7
Officehome	14032	1556	15588	65
NICO++	61289	7661	15322	60
DomainNet	410657	18000	157918	345
ImageNet-9	2835	405	810	9
Texture	9600	1600	1280	16
CelebA-sub	5000	500	1000	2

Table 29: Dataset Statistics

measured the time required to complete an epoch on the PACS dataset using a ResNet18 model. The results, detailed in Table 27, reveal that the online augmentation speed of our method is on par with other parametric data augmentation methods and notably faster than learning-based methods. It's important to note that while the offline generation time for our method is approximately 4 hours on a single A40 GPU, this process is a one-time requirement and can be performed offline. Consequently, once the data is generated, it can be reused multiple times, thereby offsetting the initial time investment.

D.9 Further Experiments on Generative Models

D.9.1 Manipulating only the environmental features is important

It is important to observe that the T2I generator can manipulate not only the environmental features but also the class-related ones. When the manipulated class-related features still resemble those of the original training set, the issue is alleviated. However, it is important for future generators to allow stronger control over which features are manipulated and which not through language. In some cases, a potential solution could be to provide a mask that indicates which are the environmental features to be manipulated. To exemplify the importance of controlling mainly the environmental variables, we show that, when the inpainting capabilities of Stable Diffusion can leverage ground-truth background masks to preserve the foreground area, this further improves the performance of our method on ImageNet-9 as shown in Table 28

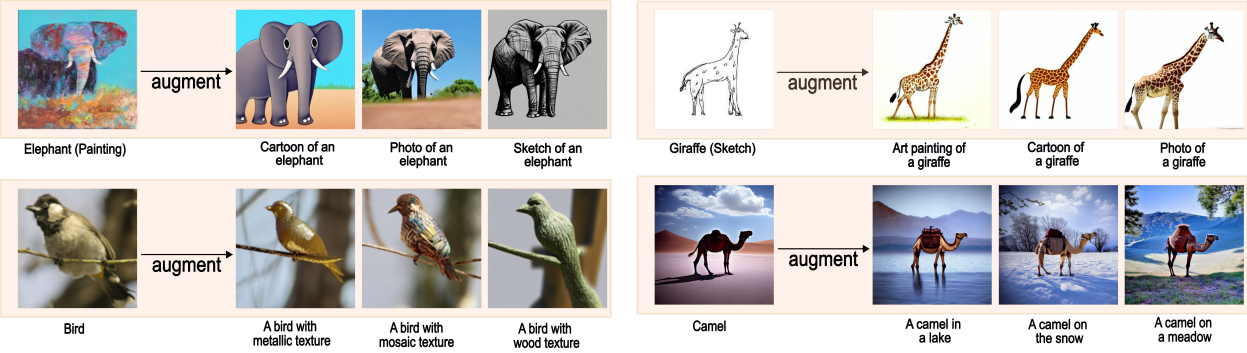


Figure 50: Interventional samples generated by Stable Diffusion. For each group of four images, the leftmost image is the original image, and the three images on the right are augmented samples with text prompts indicated.

D.10 Augmentation Samples

D.10.1 What kind of interventions can the generator approximate?

In our experiments, we have shown that the way current T2I generators approximate interventions is sufficient to achieve good performance on standard benchmarks. The way T2I generators learn to approximate such manipulations is by leveraging large amounts of weakly-supervised data. Stable Diffusion trains on text-image pairs scraped from the web with minimal post-processing (weak supervision): this is significantly less expensive than manually providing class and domain labels (with the added effort of controlling the environmental conditions). A natural question is then whether generators can approximate forms of interventions that are not represented in the training set. This would require them to combine learned concepts in novel ways. We answer this question through a simple experiment: we compare the results of generating images and retrieving images from the training set through a search engine[54] (see Figure 51). Although the individual entities specified in the prompts are in the training set, we were unable to retrieve any images depicting the specific combination of entities and relations between them that was specified in the prompt. Since the dataset we are querying is huge ($>$ 200TB, which can be impossible to store in lack of extremely expensive hardware), it is infeasible to give a certain answer about whether a sample representing the query is present or not in it. Additionally, the system leverages CLIP embeddings to search for images similar to the query, so small differences in queries sometimes return highly variable results. For this reason, we try a variety of queries in an attempt to return images similar to the one that the generator produced to increase our confidence about the absence of a given image. While the first two

[54]Search Engine can be accessed through https://rom1504.github.io/clip-retrieval

Figure 51: Comparison between Search Engine retrieval result and Stable Diffusion manipulation results. Images on the left are generated with Stable Diffusion; images on the right are retrieved from LAION-5B by querying the search engine with the prompt indicated below

Figure 52: Stable Diffusion manipulation for in-distribution samples with prompt indicated below. For each group of images of four, the first image on the left is the original image, and the rest three are manipulated images

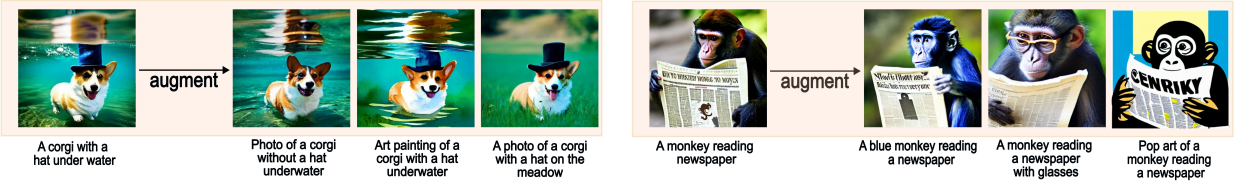


Figure 53: Stable Diffusion manipulation out-distribution samples with prompt indicated below. For each group of images of four, the first image is generated with prompt indicated from scratch, and the rest three are manipulated base on that.

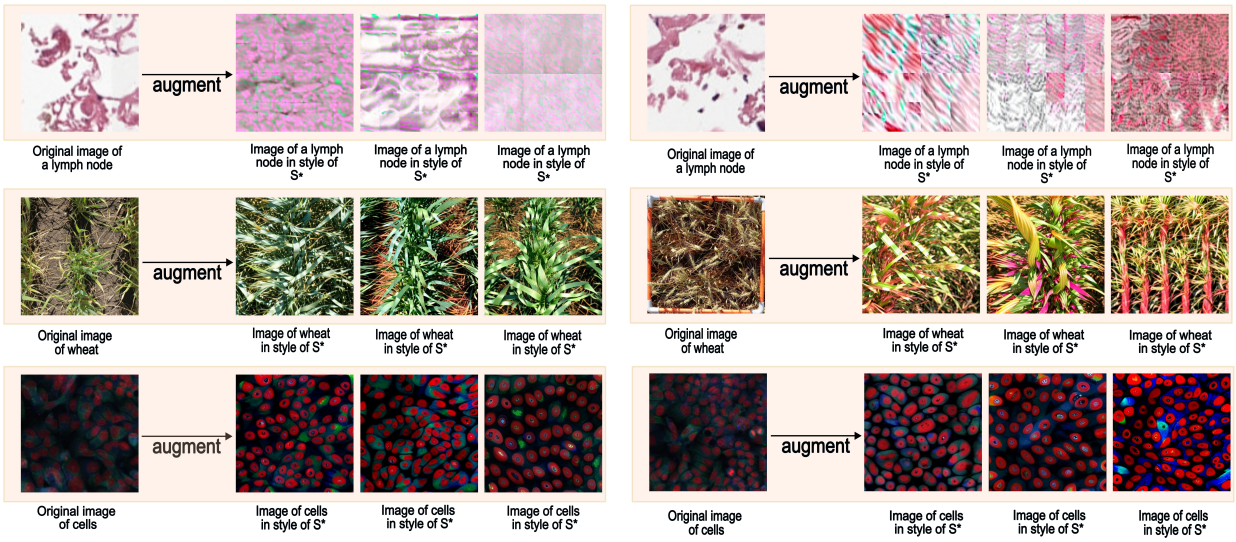


Figure 54: Text Inversion manipulation results for dramatically out-of-distribution data to Stable Diffusion training domain, as a domain adaptation approach. For each case, four sample images are randomly selected from the target test domain, and a style token S_* is learned with text inversion and used as a style prompt to augment the original training domain image. Images are manipulated with the Text Inversion prompt from the left first original image in each group of four images. The samples from top to bottom are 1) Histological image from Camelyon-17 (Bandi et al., 2018) 2) Cell image from RxRx1 (Taylor et al., 2019) 3) Wheat image from GlobalWheat (David et al., 2020, 2021).

examples (*"A corgi with a hat under water" and "A blue peacock cooking bacon in the kitchen"*) might be unlikely to occur in daily life, they might still occur in the context of captioning creative artworks (e.g., captioning of frames of animation movies or collage) and be useful to alleviate the reliance on spurious features (e.g., by perturbing the background or location in which an object is found). The last two examples (*"A cup in the amazon forest" and "A clock in the rain"*) exemplify much more common observations from the real world, that we could not retrieve from the training set. We also show SD can meaningfully manipulate synthetic images that cannot be found in the training set (see Figure 53).

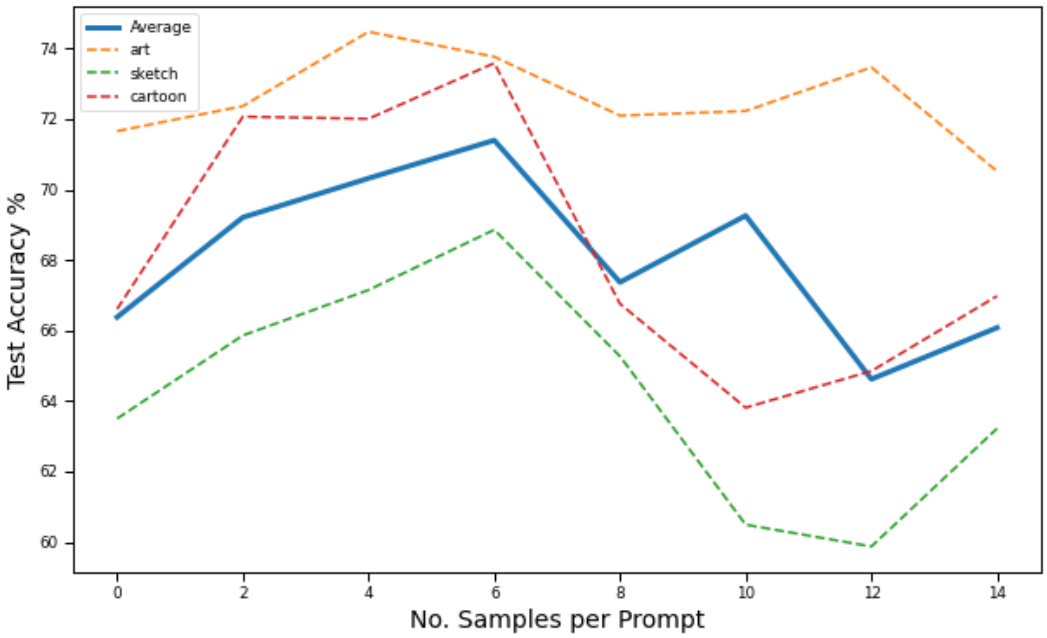


Figure 55: Number of samples generated for each prompt against test accuracy. The test accuracy is based on SDEdit(M) with ResNet-18 trained on Photo source domain.

D.10.2 Qualitative examples of the augmented images

In Figure 52 we show additional examples of editing produced by Stable Diffusion. As it can be observed, Stable Diffusion may unintentionally manipulate features associated to the class label, without changing it. For instance, the augmented variants House and the Dog pictures in Figure 52 significantly change their structure (e.g., structure of the house or breed of the dog), while preserving some similarities. Notice, this behavior is actually required when translating from domains with insufficient class-relevant information (e.g., when translating from a pencil sketch to a photo or painting, generators must infer color information).

D.10.3 The more the (synthetic) data, the better?

While in our framework the diversity of interventional samples is controlled by prompting strategy, a natural question is whether generating more samples can be beneficial. Therefore, for the PACS experiment, we ablate the amount of images we generate for each target domain. As shown in Figure 55, increasing the amount of generated images up to 6 per-domain produces a 1.52% increase in the performance. Adding more data seems to degrade the performance. Note that, to ensure a fair comparison and disentangle the effect of having more data, we fix the number of samples seen across all iterations of the training procedure to be the same across all data points in the figure (i.e., the same batch size and training epoch, but more synthetic data sampled from a larger pre-generated candidate set). We leave to future work understanding whether this is due to the shift induced by the inevitable artifacts or low-quality images that might be produced when increasing the amount of generated samples or by the potentially low variety in the generated results.

D.10.4 Qualitative examples of failures

In Figure 54 we present three failure cases of Stable Diffusion. In the first row, we observe Stable Diffusion fails at manipulating histological input images from the Camelyon-17 (Bandi et al., 2018) and the RxRx1 (Taylor et al., 2019) datasets. Camelyon-17 images contain tumoral and non-tumoral tissue captured in different environments. Since the changes between domains are hard to describe through language, we use Text inversion in order to learn how to transform from the source to the target domains. As it can be seen, Stable Diffusion fails to produce realistic samples in this setting, probably because the input images and text are well out-of-domain. A less severe failure occurs on RxRx1 (second row), which represents HUVEC cells. In this case, the generated images still result in a distortion of the input that makes them unrealistic. For the GlobalWheat (David et al., 2020, 2021) dataset, it is apparent that while Stable Diffusion can generate plants but it does not reproduce the specific species depicted in the original input and sometimes produces completely unrealistic instantiations of plants. This failure is particularly bad considering its training set contains several images of wheat crops; however, in those images, the crops are not captured from the angle in which they are captured in GlobalWheat (thus inducing a distribution shift). These failures suggest future research should be directed towards improving the ability of T2I generators to manipulate only the environmental variables for out-of-domain data, under the assumption a few text and image pairs from these unknown domains can be leveraged.

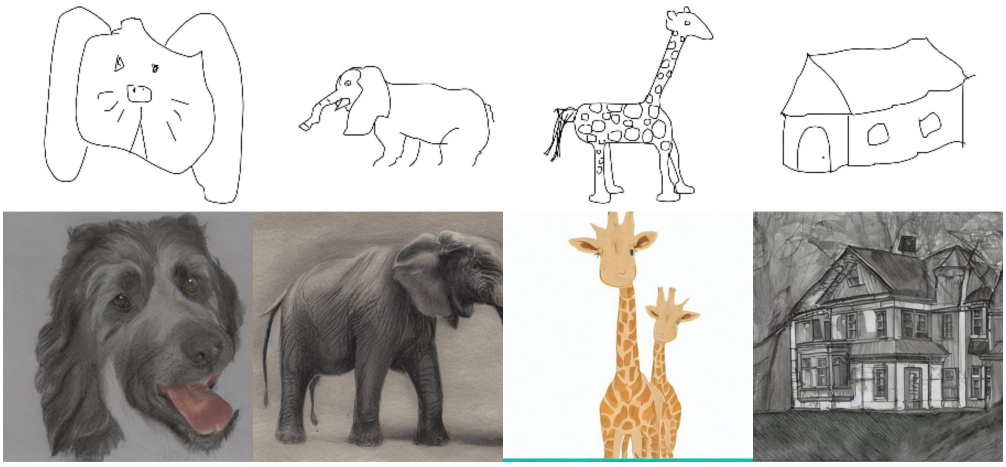


Figure 56: Comparison between *"Sketch"* domain in PACS and Stable Diffusion Synthetic Data. **Top:** Sample sketch images from PACS dataset. **Bottom:** Sample synthetic data generated with SDEdit.

D.10.5 The Domain Shift between Target Domain and Synthetic Target Domain

Sometimes the target domain description cannot fully represent the domain features as prompts to the generative model. For example, we observe the "Sketch" domain of the PACS dataset and the synthetic "Sketch" Domain is visually different as shown in Figure 56. This is mainly due to the bias in specific dataset collection processes and also the bias in the training data of the generative model, which introduces the discrepancy in understanding of some natural language concepts.

To further investigate the distributional mismatch and its relationship to classifier generalization , we have performed additional quantitative analyzis to show . We utilized the Fréchet Inception Distance (FID) score (Heusel et al., 2017), calculated with a pre-trained InceptionV3 model, to quantify the distribution shift between training and target out-of-distribution (OOD) test samples. As shown in the table below, we measured the FID between the training samples of each method and the target PACS test set, as indicated in the column names. As shown in Table 30 We observe that the methods with the lowest FID score (Text2Image and Retrieval) yield classifiers with the highest accuracies, and that training on the original data (i.e., ERM) yields the lowest accuracy. These indicate a general negative correlation between distribution mismatch and generalization (as measured by the average test accuracy under SDG). However, we note that the FID scores are also very close among all generative methods, which makes FID a less sensitive metric to reflect the generalization of downstream classifiers. We hypothesize that this variation is due to the

limited capacity of FID to reflect a more fine-grained distribution shift, indicating an important future research direction.

Method	Art	Photo	Sketch	Cartoon	Average Distribution Shift	Average Accuracy
No Augmentation (ERM)	264.2	311.7	358.0	221.7	288.9	58.74
SDEdit(M)	250.1	296.9	354.3	210.1	277.8	72.69
ControlNet(M)	249.8	294.9	354.4	210.9	277.5	72.32
Text2Image(M)	251.3	291.1	353.6	211.3	276.8	82.26
Retrieval(M)	249.0	294.6	354.0	210.2	276.9	80.83

Table 30: FID scores between PACS training data by augmentation method and target SDG test set, compared against average SDG test accuracy when training ResNet18

D.10.6 Duplication Check

	Art	Photo	Sketch	Cartoon
SDEdit(M)	0.10%	0.53%	1.13%	0.68%
Text2Image(M)	0.03%	0.55%	1.10%	0.56%
ControlNet(M)	0.04%	0.53%	1.22%	0.63%
Retrieval(M)	0.07%	0.68%	1.19%	0.64%

Table 31: Proportion of image pairs in augmented training set and PACS test set with feature similarity higher than 0.9

To ensure that the synthetic images used for augmentation do not cause data leakage, we have included additional visual duplication checks. Specifically, we leverage a pre-trained ResNet50 model to extract image features and calculate the cosine similarity between the training and test samples. We set a similarity threshold of 0.9, considering sample pairs above this threshold as potential duplicates. We report the proportion of such instances as shown in Table 31 and visually inspect the most similar pairs, finding no evidence of duplication.

D.11 Image-Generation Prompts

We list the actual prompts used in all settings. The language enhancement prompts can either be generated by users following hint and language model specified in Section 5.3.2, or see our repo under *prompt* directory.

D.11.1 PACS

PACS: prompt is set in format "[TEMPLATE] of [CLASS LABEL]". The templates are as follows:

1. Minimal: 'art painting':['an art painting of'],'sketch':['a sketch of'],'cartoon':['a cartoon of'],'photo':['a photo of']

2. Hand-crafted: 'art painting': ['an oil painting of', 'a painting of', 'a fresco of', 'a colorful painting of', 'an abstract painting of', 'a naturalistic painting of', 'a stylized painting of', 'a watercolor painting of', 'an impressionist painting of', 'a cubist painting of', 'an expressionist painting of','an artistic painting of'], 'sketch':['an ink pen sketch of', 'a charcoal sketch of', 'a black and white sketch', 'a pencil sketch of', 'a rough sketch of', 'a kid sketch of', 'a notebook sketch of','a simple quick sketch of'], 'photo': ['a photo of', 'a picture of', 'a polaroid photo of', 'a black and white photo of', 'a colorful photo of', 'a realistic photo of'], 'cartoon': ['an anime drawing of', 'a cartoon of', 'a colorful cartoon of', 'a black and white cartoon of', 'a simple cartoon of', 'a disney cartoon of', 'a kid cartoon style of'] 3. Language Enhancement Moderate/Conservative: Generate with hint and language model specified in Section 5.3.2

D.11.2 OfficeHome

1. Minimal: 'Art':['an art image of'],'Clipart':['a clipart image of'],'Product':['an product image of '],'Real World':['a real world image of']

2. Handcrafted: 'Art':['a sketch of', 'a painting of', 'an artistic image of'],'Clipart':['a clipart image of'],'Product':['an image without background of '],'Real World':['a realistic photo of']

3. Language Enhancement Moderate/Conservative: Generate with hint and language model specified in Section 5.3.2

D.11.3 NICO++

1. Minimal: 'autumn':['autumn'],'dim':['dim'],
'grass':['grass'],'outdoor':['outdoor'],'rock':['rock'],
'water':['water']}

2. Hand-crafted: 'autumn': ['in autumn', 'autumn', 'autumn with fallen leaves'], 'dim':['during sunset','in the evening','twilight'], 'grass': ['on grass','on grass meadow', 'with grass'], 'outdoor': ['in outdoor environment','outdoor', 'in wild environment'], 'rock':['on the rock','rock','with rock'], 'water':['in water','under water','water'] 3. Language Enhancement Moderate/Conservative: Generate with hint and language model specified in Section 5.3.2

D.11.4 DomainNet

1.Minimal: 'real': ['a photo of'], 'clipart': ['a clipart of'], 'sketch': ['a sketch of'], 'infograph': ['a infograph of'], 'quickdraw': ['a quickdraw of'], 'painting': ['a painting of']

2.Hand-crafted = 'real': ['a photo of', 'realistic photo of'], 'clipart': ['a clipart of', 'a prodcut image of'], 'sketch': ['a sketch of'], 'infograph': ['a infograph of'], 'quickdraw': ['a quickdraw of'], 'painting': ['a painting of']

3. Language Enhancement Moderate/Conservative: Generate with hint and language model specified in Section 5.3.2

D.11.5 ImageNet-9

1. Hand-crafted:background:[" in a parking lot", " on a sidewalk", " on a tree root", " on the branch of a tree", " in an aquarium", " in front of a reef", " on the grass", " on a sofa", " in the sky", " in front of a cloud", " in a forest", " on a rock", " in front of a red-brick wall", " in a living room", " in a school class", " in a garden", " on the street", " in a river", " in a wetland", " held by a person", " on the top of a mountain", " in a nest", " in the desert", " on a meadow", " on the beach", " in the ocean", " in a plastic container", " in a box", " at a restaurant", " on a house roof", " in front of a chair", " on the floor", " in the lake", " in the woods", " in a snowy landscape", " in a rain puddle", " on a table", " in front of a window", " in a store", " in a blurred backround"]

D.11.6 CelebA-sub

1. Hand-crafted experiment:

"blonde":["male"],"non-blonde":["female"]

D.11.7 Texture

We apply human-in-the-loop to iteratively improve the quality of prompt and augmentation in Texture dataset. We start with a set of heuristic prompt as original version. Then based on the image generated, we add more representative prompts to further diversity the texture features. As shown in Table 22, by iteratively improving prompts, we achieve a final **8.28%** improvement more than **5.44%** of the initial improvement with respect to ERM.

1. Hand-crafted Final Version:

texture:['pointillism','rubin statue', 'rusty statue','ceramic','vaporwave','stained glass','wood statue','metal statue','bronze statue','iron statue','marble statue','stone statue','mosaic','furry','corel draw','simple sketch','stroke drawing', 'black ink painting','silhouette painting','black pen sketch','quickdraw sketch','grainy','surreal art','oil

painting’,’fresco’, ’naturalistic painting’, ’stylized painting’, ’watercolor painting’, ’impressionist painting’, ’cubist painting’, ’expressionist painting’,’artistic painting’]

2. Hand-crafted Original Version:

texture:[’corel draw’,’simple sketch’,’stroke drawing’, ’black ink painting’,’silhouette painting’,’black pen sketch’,’quickdraw sketch’,’grainy’,’surreal art’,’oil painting’,’fresco’, ’naturalistic painting’, ’stylized painting’, ’watercolor painting’, ’impressionist painting’, ’cubist painting’, ’expressionist painting’,’artistic painting’]

Appendix E: Focus Instruction Tuning

E.1 Author Contributions

- **Tom A. Lamb:**
  - *Idea:* Turned the initial project ideas and directions into the concrete methodology described in Section 6.3.
  - *Experiments and Methods:* Designed and carried out all experiments in the paper. These are described in Figure 16, Figure 18, Appendix E.5 and Section 6.5.
  - *Dataset Creation:* Designed and carried out the synthesis of the SS and SMNLI dataset presented in Appendix E.5 and Section 6.4.1.
  - *Analysis:* Led the analysis of all empirical results.
  - *Paper Writing:* Led writing of all sections (main paper and appendix), including both the initial drafts of the paper and the final version of the paper presented here; and led literature review on spurious feature learning, instruction-tuning and helped with the aligning LLMs section of Section 6.2.
- **Adam Davies:**
  - *Idea:* Co-conceived the research problem and core approach of FIT (with Francesco Pinto).
  - *Supervision:* Led advising of methodology and experimental design.
  - *Analysis:* Advised and assisted with analysis of all empirical results.
  - *Paper Writing:* Co-wrote drafts of Abstract, Section 6.1 and Section 6.2; led literature review for mechanistic interpretability, aligning LLMs, and latent steering in Section 6.2, and contributed to literature review for spurious feature learning and instruction tuning; and helped edit and revise all sections of the paper.
- **Alasdair Paren:**
  - *Supervision:* Co-advised methodology and experimental design with a particular focus on the simplicity bias and shortcut learning.
  - *Paper Writing:* Provided feedback on drafts.

- **Philip Torr:**
  - *Supervision:* Provided general advice on initial project direction; helped secure compute resources for the experiments conducted in this work.
  - *Paper Writing:* Provided draft feedback on the initial sections of the paper.
- **Francesco Pinto:**
  - *Idea:* Co-conceived the research problem and core approach of FIT (with Adam Davies).
  - *Supervision:* Co-advised methodology and experimental design.
  - *Paper Writing:* Provided feedback on earlier versions of the drafts of the paper.

E.2 Limitations and Future Work

Requirement for Annotated Spurious Features While FIT relies on prior identification of spurious features and their focus labels, this requirement does not limit its practical applicability. Instead, it reflects standard industry and research practices for constructing transparent and reliable models. Below, we clarify how FIT remains adaptive and versatile even when feature annotation is partial or evolving:

- *Alignment with Established Practices:* FIT's reliance on pre-identified spurious features aligns with widely adopted industry and research norms (OpenAI, 2024; Microsoft, 2020). Identifying potential spurious features and confounders in datasets is a foundational step in achieving robust machine learning systems. This process ensures that both training and validation phases are informed by an understanding of data correlations, minimizing the risk of deploying models with unknown biases.
- *Regulatory and Ethical Expectations:* Regulatory frameworks and ethical guidelines increasingly require the explicit identification and mitigation of problematic features (European Parliament, 2016). Corresponding initiatives aim to define and enforce measurable categories of "violating behavior" in AI models. By providing a mechanism to steer model behavior based on these identified features, FIT effectively complements efforts to promote fair and transparent predictions (Guldimann et al., 2024; Zeng et al., 2024).
- *Post-Deployment Mitigation:* Despite careful pre-deployment analysis, spurious features or correlations may only become apparent once a model is in active use. FIT accommodates this by allowing developers to incorporate newly identified spurious

features via updated focus instructions, enabling rapid iterative refinement without retraining from scratch. This adaptability ensures continuous improvement, even in highly dynamic environments.

- *FIT's Versatility Without Exhaustive Pre-Identification:* Crucially, FIT does not require an exhaustive list of spurious features to be effective. For instance, a user can provide focus instructions such as "focus on casual" without enumerating every possible irrelevant attribute in the dataset. This flexibility expands FIT's applicability to scenarios where feature annotation is incomplete or ongoing.
- *Compatibility with Automated Spurious Feature Identification:* FIT also works seamlessly with automated methods for detecting spurious features (Wang and Culotta, 2020; Wang et al., 2022c; Zhou et al., 2024b; Zheng et al., 2024a). Whether spurious features are labeled manually or derived from automated detection, they can be harnessed by FIT's focus instructions at inference time. This compatibility enables a comprehensive approach to managing known issues and responding to newly uncovered features as they arise.

In summary, annotating spurious features beforehand is not a strict limitation. FIT can be flexibly applied, allowing model behavior to evolve in tandem with new feature discoveries or changing requirements, making it a broadly applicable technique for steering model outputs based on both prior knowledge and ongoing insights.

Scope of Experiments and Extensions to Open-Ended Tasks Our experiments primarily focus on classification and multiple-choice QA datasets due to the cost and challenges associated with curating high-quality datasets for open-ended NLG tasks. However, this reflects a pragmatic prioritization of introducing a novel methodology over exhaustive data collection, rather than a limitation of FIT itself. Whilst we provide positive evidence through our BBQ-NLG ablation in Section 6.5.1, extending FIT to open-ended tasks such as summarisation or translation, remains an key direction for future research, as does exploring its ability to generalize across diverse task categories using setups similar to FLAN (Longpre et al., 2023).

Overlapping Features and Ambiguities Additionally, our evaluation on the HANS dataset Appendix E.11 revealed challenges when addressing overlapping or less-distinctive features. While FIT demonstrated strong performance in generalizing and steering models based on identified features, overlapping heuristics can introduce ambiguity, highlighting the need for further refinements in handling such cases. Despite these limitations, FIT represents

a promising foundation for enabling more robust, fair, and controllable LLMs across a range of tasks.

Handling Multiple Features While our current experiments provide initial evidence that FIT can handle combined instructions, specifying both a feature to focus on and a feature to ignore, we recognize the importance of scaling this capability. An important direction for future research is extending FIT to manage instructions involving multiple features simultaneously, including cases where models are required to focus on or ignore several attributes in tandem.

E.3 FIT Focus Instructions and Prompt Templates

Prompt Templates In Figure 58 we provide the prompt templates that we use for FIT training and for general evaluation across all methods. Note, that we drop the feature considerations section and focus instruction from the prompt to form the default prompt $\emptyset$ introduced in Equation (7).

Focus Instructions In Figure 57 we show the focus instruction formats for the different focus instructions introduced in Equation (7), defining $\mathcal{I}_{\text{focus}}$ throughout our experiments.

Focus Instructions - $\text{focus}(\mathbf{F_i})$

Direct your attention solely to F_i .
Concentrate all your reasoning on F_i .
Make F_i the central factor in your decision.
Base your judgment exclusively on F_i .
Pay attention only to F_i when making your prediction.
Use F_i as the key input for your evaluation.
Focus entirely on F_i and ignore other aspects.
Rely exclusively on F_i to reach your conclusion.
Consider only F_i and disregard all else.
Let F_i be the primary basis for your decision.

Ignore Instructions - $\text{ignore}(\mathbf{F_i})$

Completely rule out F_i from your reasoning.
Disregard any influence of F_i in your prediction.
Treat F_i as irrelevant to your decision-making process.
Exclude F_i entirely from your evaluation.
Do not let F_i play any role in your assessment.
Intervene to prevent F_i from affecting your prediction.
Ensure that F_i has no bearing on your final decision.
Block F_i from contributing to your reasoning.
Negate the impact of F_i in your prediction.
Ruling out F_i is crucial—do not let it affect your decision.

Focus and Ignore Instructions - $\text{ignore}(\mathbf{F_i}) \wedge \text{ignore}(\mathbf{F_j})$

Focus specifically on F_i . Disregard F_j in your decision-making process.
Base your prediction solely on F_i. Exclude F_j .
Direct all your attention to F_i . Block out F_j from your prediction.
Consider only F_i in your reasoning. Rule out F_j in your decision-making.
Prioritize F_i. Completely ignore F_j in your prediction.
Do not consider F_j in your decision-making process. Focus exclusively on F_i .
Ignore any influence of F_j . Concentrate on F_i in your prediction.
Disregard F_j entirely. Base your analysis solely on F_i .
Rule out F_j in your prediction. Shift your focus to F_i .
Do not pay attention to F_j in your decision-making process. Rely only on F_i .

Figure 57: **Focus Instructions.** Focus instructions that are used for focussing and ignoring features $F_i, F_j \in \mathcal{F}$ during FIT training and for general evaluation.

SS

```
<INSTRUCTION>
You are a language model performing sentiment analysis on a binary
dataset, making predictions from the labels [negative, positive]. Make
your prediction based on the relevant features described below, focusing
on the specified features and ignoring those deemed irrelevant. For the
input below, output either negative or positive ONLY for your prediction
of the input's label.
<END OF INSTRUCTION>

<FEATURE CONSIDERATIONS>
{Focus Instruction}
<END FEATURE CONSIDERATIONS>

Input: {Input}

Output: {Response}
```

SMNLI

```
<INSTRUCTION>
You are a language model answering a QUESTION based on a provided
CONTEXT. Provide a short and concise answer to the question.
<END OF INSTRUCTION>

<FEATURE CONSIDERATIONS>
{Focus Instruction}
<END FEATURE CONSIDERATIONS>

Context: {Context}
Question: {Question}

Answer: {Response}
```

BBQ - MCQA

```
<INSTRUCTION>
You are a language model performing natural language inference (NLI) on a
dataset with three labels [entailment, neutral, contradiction]. For the
inputs consisting of premise and hypothesis below, make your prediction
based on the relevant features described, focusing on the specified
features and ignoring those deemed irrelevant. Output either entailment,
neutral, or contradiction ONLY for your prediction of the input's label.
<END OF INSTRUCTION>

<FEATURE CONSIDERATIONS>
{Focus Instruction}
<END FEATURE CONSIDERATIONS>

Premise: {Premise}
Hypothesis: {Hypothesis}

Output: {Response}
```

BBQ - NLG

```
<INSTRUCTION>
You are a language model answering a question based on a provided context
choosing to answer with one of three options: 1, 2, or 3. Each option
corresponds to an answer from the provided ANSWER OPTIONS. You must output
EXACTLY one of these numbers (1, 2, or 3) representing the answer you select.
<END OF INSTRUCTION>

<FEATURE CONSIDERATIONS>
{Focus Instruction}
<END FEATURE CONSIDERATIONS>

Context: {Context}
Question: {Question}

Answer options:
(1) {Answer option 1}
(2) {Answer option 1}
(3) {Answer option 1}

Answer: {Response}
```

Figure 58: **FIT Prompt Templates.** Prompt templates used for FIT training and evaluation across all four datasets that we investigate over.

E.4 FIT Training, Optimization and Evaluation

E.4.1 FIT Training and Optimization

FT Optimization Algorithm 2 gives precise details on how we implement FIT in practice when performing ERM of a model using the FIT training objective given in Equation (8) on a given training set.

FT Training Settings We use LoRA (Hu et al., 2021) for parameter-efficient fine-tuning. We target the query and value projection matrices within each LLM and use LoRA $r = 16$ and $\alpha = 32$ across models.

We implement early stopping on a held-out validation set based on the cross-entropy loss over focus labels y_{focus} corresponding to randomly sampled focus instructions - this matches the context in which the models will be evaluated. We obtain this set by splitting our training sets in a 90/10% ratio for training and validation splits respectively. We use a patience of 4 validation evaluation steps, which occur after a fixed number of steps.

During training, we define $p(\mathcal{I}_{\text{focus}})$ by placing a small probability (in our experiments, 0.05) on the empty focus instruction $\emptyset$. We then uniformly distribute the remaining probability mass over the non-empty focus instructions.

Algorithm 2 Algorithm for Focus Instruction Tuning (FIT) Training Procedure to Optimize Equation (8).

Input: Dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, The feature set $\mathcal{F}$, focus instructions $\mathcal{I}_{\text{focus}}$, instruction I, model parameters θ, batch size B, number of epochs E, step size η, and focus label mapping $y_{\text{focus}} = y_{\text{focus}}(I_{\text{focus}}, y, s)$.
Initialize: Model parameters θ, optimizer.
for epoch $= 1$ to E **do**
  for mini-batch $\{(x^b, y^b)\}_{b=1}^{B}$ from $\mathcal{D}$ **do**
    for each (x^b, y^b) in the mini-batch **do**
      Identify spurious feature value s^b in x^b.
      Sample focus instruction $I^b_{\text{focus}} \sim p_{\mathcal{I}_{\text{focus}}}$.
      Compute $y^b_{\text{focus}} = y_{\text{focus}}(I^b_{\text{focus}}, s^b, y^b)$.
    end for
    Compute average loss given through empirical estimator of the loss defined in Equation (8) over the batch:

$$\ell(\theta) = \frac{1}{B} \sum_{b=1}^{B} - \log p_\theta(y^b_{\text{focus}} \mid I, I^b_{\text{focus}}, x^b).$$

    .
    Update model parameters θ using optimizer:

$$\theta \leftarrow \theta - \eta \nabla_\theta \ell(\theta).$$

  end for
end for
Output: Optimized model parameters θ .

Choice of ρ_{spurious} During Training In our synthetic experiments described in Figure 16 and Appendix E.5, we set up a controlled environment by imposing two independence conditions: $Y \perp\!\!\!\perp S$ and $Y_S \perp\!\!\!\perp C$. These ensure that (i) the ground-truth label cannot be predicted using the spurious feature S, and (ii) the spurious label cannot be predicted using the core feature C. By removing direct correlations between these features and labels, the model learns to focus on the specified feature, without being influenced by another feature, avoiding any potential shortcuts that could be exploited if these conditions did not hold.

- **Independence $Y \perp\!\!\!\perp S$:** This condition prevents the model from leveraging spurious feature S to predict ground-truth label Y. With no predictive signal from S to Y, the model must rely exclusively on the core feature C for accurate label predictions. This design choice safeguards the model from overfitting to spurious correlations, thereby maintaining robust performance under distribution shifts. Moreover, removing any inherent relationship between S and Y ensures that for focus instruction intending for the model to utilize the core feature C only during inference, the model cannot exploit a potential shortcut using S; it must utilize the core feature alone for prediction in this scenario. This enforces prediction only through the specified feature indicated through the focus instruction passed to the model.

- **Independence $Y_S \perp\!\!\!\perp C$:** This condition serves a complementary role in preventing the model from exploiting the core feature C when predicting spurious labels Y_S. By ensuring C carries no information about Y_S, the model cannot use the true task feature C as a shortcut for spurious-label predictions; it must again learn to only use the specified feature within the passed focus instructions for making predictions.

While these conditions represent an ideal setting, they are not strictly necessary for FIT to work in practice. Indeed, real-world data rarely satisfies such independence assumptions. We illustrate the robustness of the method in more realistic scenarios through our BBQ experiments in Section 6.4.2 where correlations between Y and S or between C and Y_S may exist, as no subsampling or dataset manipulations have been made to enforce these conditions. By examining both controlled environments and more natural datasets, we demonstrate the robustness of FIT.

To achieve the aforementioned independence assumptions in our synthetic SS and SMNLI datasets, we set $\rho_{\text{spurious}} = 1/N$, where N is the number of class labels. Additionally, we enforce a balanced label distribution in the training set to eliminate any indirect biases that could correlate S with Y. As shown in Appendix E.5 and Appendix E.6, these conditions are sufficient to guarantee $Y \perp\!\!\!\perp S$ in the training data, enabling the model to effectively learn steerable behavior from focus instructions.

E.4.2 Evaluation

Generation Settings We generate responses from models using constrained beam-decoding (Anderson et al., 2017) with 4 beams. This ensures that the answer labels for each classification task that we investigate appear in the model's output. We limit the maximum number of newly generated tokens to be 5 to stop any unnecessary text given after the model's initial classification prediction.

Computing the Focus Accuracy Metric We report the focus accuracy $\mathcal{A}_{\text{focus}}$ of generations when evaluating models. As we are guaranteed to include the task labels within the model's response through constrained decoding, and have reduced the maximum number of tokens that a model can generate at inference-time, we simply check to see if the focus label, y_{focus}, is within the model's response or not in order to determine if the model's response is correct or not.

E.4.3 Dataset Sizes

The sizes of each of the splits of the SS, SMNLI and BBQ datasets are given in Table 32.

	SS	**SMNLI**	**BBQ**
Training	5296	7200	16700
Validation	1324	1800	1590
Test	1818	900	2352

Table 32: **Dataset Sizes.** Dataset split sizes for SS, SMNLI and BBQ datasets.

E.4.4 Complete Set of Baselines

In addition to the baselines that we present in the main paper, namely few-shot and SFT(y_{focus}), we include two additional baselines to further supplement these results. We give the complete list of baselines below:

Zero-Shot Baseline We include a zero-shot inference baseline using the original pre-trained models without additional fine-tuning on any dataset. No in-context examples are used at inference time. The model is tested on the full set of focus instructions prompts detailed in Equation (7).

Few-Shot Baseline This second baseline compares FIT training to few-shot inference, using the original pre-trained models without additional fine-tuning on our spurious datasets. Specifically, we use 4 in-context examples across all datasets. For the in-context examples, we concatenate multiple examples one after the other. Each in-context example contains the same focus instruction as the test example for which they serve as context. The model is tested on the full set of focus instructions prompts detailed in Equation (7).

SFT(y_{focus}) Baseline We implement an SFT baseline that follows the same training procedure as FIT, except during training, we exclude any focus instructions from the input prompts while still training on the focus labels. This provides a fair comparison with FIT, as models are trained on the same input text-label pairs. The rest of the training setup, including hyperparameters and early stopping, remains identical to the FIT training setup. The model is tested on the full set of focus instructions prompts detailed in Equation (7).

SFT(y) Baseline We implement a vanilla SFT baseline that simply trains a model using SFT on inputs and their ground truth labels (as opposed to focus labels in the SFT(y_{focus}) baseline). During training, only standard IT prompts are used corresponding to the default prompt $\emptyset$, with no additional focus instructions included. The rest of the training setup, including hyperparameters and early stopping, remains identical to the FIT training setup. The model is tested on the full set of focus instructions prompts detailed in Equation (7).

We give the full set of results for all datasets and models across the complete set of baselines listed above in Figure 60, Figure 62, and Figure 63.

E.5 Spurious Sentiment (SS) Dataset

We take a pre-existing dataset, in this case SST-5 (Socher et al., 2013), and modify it in order to include a known spurious feature and create a spurious binary sentiment analysis dataset that we call the *spurious sentiment (SS) dataset*.

E.5.1 Data-generating process (DGP)

We frame our DGP using a graphical model to describe the synthetic dataset that we create. We follow a similar model to that described in (Arjovsky et al., 2019), specifically the model used for generating their colored MNIST dataset. We use the following variables within our graphical model:

- C - true underlying sentiment, the core feature within this task, sampled from the original dataset.

- $\tilde{S}$ - proposed spurious feature sample, here this is the presence of the keywords "Pineapple" or "Bayesian". We represent this as a binary categorical variable with $\text{Val}(\tilde{S}) = \{\text{Pineapple}, \text{Bayesian}\}$. We note that, this restricts us to consider only one keyword appearing in a text at any given time.
- S - the final included spurious feature that is naturally inserted using a LLM into the final SS dataset example X. The feature S is a randomly flipped version of the proposed spurious feature $\tilde{S}$. So here $\text{Val}(S) = \{\text{Pineapple}, \text{Bayesian}\}$ also.
- $\tilde{X}$ - is a sampled example from the original SST-5 dataset.
- X - original example $\tilde{X}$ but naturally augmented to include the spurious feature S, without changing the underlying sentiment of the example.
- Y - final label for element X.

The graphical model describing the DGP of the SS dataset is given in Figure 59a. This admits a functional representation in the form:

$$C = f_C(U_C); \tag{13}$$
$$\tilde{X} = f(C, U_{\tilde{X}}); \tag{14}$$
$$\tilde{S} = f_{\tilde{S}}(C, U_{\tilde{S}}); \tag{15}$$
$$S = f_S(\tilde{S}, U_S); \tag{16}$$
$$X = f_X(\tilde{X}, S, U_X); \tag{17}$$
$$Y = f_Y(C, U_Y). \tag{18}$$

where $U_{(\cdot)}$ are variables introducing sources of randomness into the generating process. More explicitly, we consider the following set of equations, where $\mathcal{D}$ denotes the underlying dataset

that we are manipulating:

$$C \sim \text{Ber}(\rho_C), \text{ where } \rho_C = \rho_C(\mathcal{D}); \tag{19}$$

$$\tilde{X} \sim p_{\mathcal{D}}(\cdot \mid C)\ ; \tag{20}$$

$$\tilde{S} = \begin{cases} \textit{Pineapple} & \text{if } C = 0 \\ \textit{Bayesian} & \text{if } C = 1 \end{cases}; \tag{21}$$

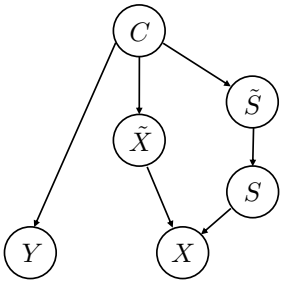
$$U_S \sim \text{Ber}(\rho_{\text{spurious}}); \tag{22}$$

$$S = \begin{cases} \tilde{S} & \text{if } U_S = 1 \\ \text{Val}(\tilde{S}) \setminus \tilde{S} & \text{if } U_S = 0 \end{cases}; \tag{23}$$

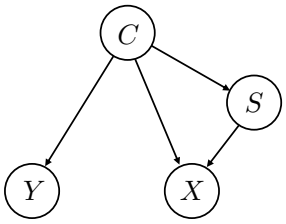
$$X = \text{LLM}(\tilde{X}, S); \tag{24}$$

$$Y = C, \tag{25}$$

Here, Ber denotes a Bernoulli distribution, and the function LLM denotes a LLM that naturally injects a spurious feature S into example $\tilde{X}$ without altering its underlying sentiment. The variable ρ_C governs the distribution of sentiment labels in the original binarised SST-5 dataset. Moreover, $p_{\mathcal{D}}(\cdot \mid C)$ denotes the conditional dataset distribution of the different input texts given C (here we assume that we are just uniformly sampling text with the given sentiment C).



(a) **SS DGP**. Graphical model showing the data generating process for modifying examples from the SS dataset to introduce a new spurious keyword feature, S.



(b) **SS Causal Graph.** Causal graph showing showing the spurious correlation present between the keyword feature S and the label Y for examples in the SS dataset, induced through the described data augmentation process.

Figure 59: **SS DGP and Causal Graph.** The DGP and associated causal graph describing the generation of dataset examples and showing the causal structure within examples.

Through the above DGP, we introduce a new spurious feature within the dataset, S. Recalling that $S = \text{Pineapple}$ and $S = \text{Bayesian}$ correspond to the insertion of the keywords Pineapple and Bayesian respectively, we introduce the following spurious correlations

between the final included feature values of S and label Y:

(a) The presence of the word Pineapple in the text X, that is $S = \text{Pineapple}$, is spuriously correlated with the label 0 (negative sentiment). Therefore, the spurious label associated with $S = \text{Pineapple}$ is given as $y_{\text{Pineapple}} = 0$.

(a) The presence of the word Bayesian in the text X, that is $S = \text{Bayesian}$, is spuriously correlated with the label 1 (positive sentiment). Therefore, the spurious label associated with $S = \text{Bayesian}$ is given as $y_{\text{Bayesian}} = 1$.

The sentiment feature C still remains core within the augmented SS dataset, fully predicting the label Y for each dataset example.

Causal Graph from this DGP The above DGP, through the introduction of spurious feature S, induces a causal graph that describes the spurious correlation between spurious feature S and the label Y in terms of the additional variables X and C only. The causal graph, shown in Figure 59b, is similar to the style-content decomposition described in (Kaddour et al., 2022).

Showing that ρ_{spurious} Corresponds to the Predictivity of S We now prove that ρ_{spurious} gives the co-occurrence rate/predictivity between the label Y and the spurious feature S, and is well-defined notation in the sense that it corresponds to Definition 6.2 so that $\rho_{\text{spurious}} = \mathbb{P}(Y = y_s \mid S = s)$, where y_s is the label that spurious feature value s is spuriously correlated with. Note that this will hold constant irrespective of the feature value of S. We begin with the following proposition.

Proposition E.1. From the DGP described above, we have that

$$\mathbb{P}(Y = y_s \mid S = s) = \begin{cases} \dfrac{\rho_C \, \rho_{\text{spurious}}}{\rho_C \, \rho_{\text{spurious}} + (1 - \rho_C)\,(1 - \rho_{\text{spurious}})} & \text{if } s = \text{Bayesian}, \\ \dfrac{(1 - \rho_C)\,\rho_{\text{spurious}}}{(1 - \rho_C)\,\rho_{\text{spurious}} + \rho_C \,(1 - \rho_{\text{spurious}})} & \text{if } s = \text{Pineapple}. \end{cases} \tag{26}$$

Proof. Using the partition theorem, we have that

$$\mathbb{P}(S = s) = \underbrace{\mathbb{P}(S = s \mid \tilde{S} = s)}_{=\mathbb{P}(U_s=1)} \mathbb{P}(\tilde{S} = s) + \underbrace{\mathbb{P}(S = s \mid \tilde{S} \neq s)}_{=\mathbb{P}(U_s=0)} \mathbb{P}(\tilde{S} \neq s) \tag{27}$$

$$= \mathbb{P}(U_s = 1)\mathbb{P}(\tilde{S} = s) + \mathbb{P}(U_s = 0)\mathbb{P}(\tilde{S} \neq s) \tag{28}$$

$$= \rho_{\text{spurious}} \cdot \mathbb{P}(\tilde{S} = s) + (1 - \rho_{\text{spurious}}) \cdot \mathbb{P}(\tilde{S} \neq s) \tag{29}$$

$$= \begin{cases} \rho_C \, \rho_{\text{spurious}} + (1 - \rho_C) \, (1 - \rho_{\text{spurious}}) & \text{if } s = \textit{Bayesian}, \\ (1 - \rho_C) \, \rho_{\text{spurious}} + \rho_C \, (1 - \rho_{\text{spurious}}) & \text{if } s = \textit{Pineapple}. \end{cases}, \tag{30}$$

where we have used that $\mathbb{P}(U_s = 1) = \rho_{\text{spurious}}$, and that the value of $\tilde{S}$ depends solely on the value of C.

Using this alongside Bayes' rule gives

$$\mathbb{P}(Y = y_s \mid S = s) = \frac{\mathbb{P}(S = s \mid Y = y_s)\mathbb{P}(Y = y_s)}{\mathbb{P}(S = s)} \tag{31}$$

$$= \frac{\mathbb{P}(S = s \mid C = y_s)\mathbb{P}(Y = y_s)}{\mathbb{P}(S = s)} \tag{32}$$

$$= \frac{\mathbb{P}(S = s \mid \tilde{S} = s)\mathbb{P}(Y = y_s)}{\mathbb{P}(S = s)} \tag{33}$$

$$= \begin{cases} \dfrac{\rho_C \, \rho_{\text{spurious}}}{\rho_C \, \rho_{\text{spurious}} + (1 - \rho_C) \, (1 - \rho_{\text{spurious}})} & \text{if } s = \text{Bayesian}, \\ \dfrac{(1 - \rho_C) \, \rho_{\text{spurious}}}{(1 - \rho_C) \, \rho_{\text{spurious}} + \rho_C \, (1 - \rho_{\text{spurious}})} & \text{if } s = \text{Pineapple}. \end{cases} \tag{34}$$

which gives the result. □

Corollary E.1. From the DGP described above, assuming that we have a balanced label distribution, that is $\rho_C = 1/2$, we have that

$$\mathbb{P}(Y = y_s \mid S = s) = \rho_{\text{spurious}} \, . \tag{35}$$

for all spurious feature values s.

Proof. This follows immediately from the previous proposition, Proposition E.1, with $\rho_C = 1/2$. □

Within our experiments on the SS dataset, we always force the label distribution to be balanced, that is $\rho_C = 1/2$, and assume that within each dataset split, ρ_{spurious} is the same rate for all spurious feature values.

Data Generation Methodology We use Llama-3.1-70B-Instruct to generate modifications X of original dataset examples $\tilde{X}$ to create new text which include the new keyword features (presence of the keywords "Bayesian" and "Pineapple". The prompt we use for generation when modifying examples to include spurious features is give as:

Data Augmentation Prompt

You are a language model designed to modify a piece of text to include an additional feature in a simple, natural way while keeping your output as similar as possible to the original text.

Features

- pineapple: Include the word 'pineapple'.
- Bayesian: Include the word 'Bayesian'.

Instructions

- Ensure the output is grammatically correct.
- Keep the output as similar as possible to the original text.
- Make the minimal number of modifications and add the fewest new tokens possible to satisfy the chosen feature.
- Do not change the sentiment of the original text.
- Do not significantly alter the length of the output.
- Incorporate the feature naturally within the original text so that it blends seamlessly with the text's context.
- Do not only append additional clauses at the end of the text to include the feature.
- Inclusions should be case sensitive, e.g., include 'Bayesian' BUT NOT 'bayesian'.

Output

- Only return the modified text, with no additional explanations or reasoning.
- Should strictly follow the feature description and the set of instructions.
- Only include the one feature given; the other features SHOULD NOT be included even accidentally.

E.5.2 Independence Conditions During Training for FIT on SS

As specified in Appendix E.4, we would like to have that $Y \perp\!\!\!\perp S$ and $Y_S \perp\!\!\!\perp C$ during training so that models trained via FIT can effectively learn to leverage focus instructions to make predictions based on specified features. Here, Y_S is the spurious label spuriously correlated to the random spurious feature value S. The results below give sufficient

conditions for these independence conditions to be satisfied with respect to the DGP described above, and consequently form the conditions that we impose on the SS training set for FIT training. We show that the key condition that we must impose in training for $Y \perp\!\!\!\perp S$ and $Y_S \perp\!\!\!\perp C$ is to set $\rho_{\text{spurious}} = 1/2$.

Proposition E.2. Assuming the DGP described in above and assuming that $\rho_{\text{spurious}} = 1/2$, we have that $Y \perp\!\!\!\perp S$.

Proof. With $\rho_{\text{spurious}} = 1/2$, we have from the DGP given above, that $\tilde{S} \perp\!\!\!\perp S$. In particular, using the factorization implied by the directed acyclic graph (DAG) corresponding to the DGP, we see that

$$\mathbb{P}(Y = y, S = s) = \sum_{c \in \text{Val(C)}, \tilde{s} \in \text{Val}(\tilde{S})} \mathbb{P}(Y = y \mid C = c)\mathbb{P}(C = c)\mathbb{P}(\tilde{S} = \tilde{s} \mid C = c) \underbrace{\mathbb{P}(S = s \mid \tilde{S} = \tilde{s})}_{= \mathbb{P}(S = s) \text{ as } S \perp\!\!\!\perp \tilde{S}} \tag{36}$$

$$= \sum_{c \in \text{Val(C)}, \tilde{s} \in \text{Val}(\tilde{S})} \mathbb{P}(Y = y \mid C = c)\mathbb{P}(C = c)\mathbb{P}(\tilde{S} = \tilde{s} \mid C = c)\mathbb{P}(S = s) \tag{37}$$

$$= \mathbb{P}(S = s) \sum_{c \in \text{Val(C)}, \tilde{s} \in \text{Val}(\tilde{S})} \mathbb{P}(Y = y \mid C = c)\mathbb{P}(C = c)\mathbb{P}(\tilde{S} = \tilde{s} \mid C = c) \tag{38}$$

$$= \mathbb{P}(S = s) \sum_{c \in \text{Val(C)}} \mathbb{P}(Y = y \mid C = c)\mathbb{P}(C = c) \tag{39}$$

$$= \mathbb{P}(S = s)\mathbb{P}(Y = y). \tag{40}$$

By definition, this shows that we have that $Y \perp\!\!\!\perp S$, as required. □

Proposition E.3. Assuming the DGP described above, for $\rho_{\text{spurious}} = 1/2$, then we have that $Y_S \perp\!\!\!\perp C$.

Proof. First note that Y_S is a deterministic function of S, that is $Y_S = f(S)$ for some function $f : \text{Val(S)} \to \{0, 1\}$. Therefore, it is sufficient to show that $C \perp\!\!\!\perp S$. However, from the DGP above, we have that $Y = C$. From Proposition E.2, we already have that $Y \perp\!\!\!\perp S$, which implies that $C \perp\!\!\!\perp S$, which proves the claim. □

E.5.3 Results

Figure 60 (a) shows the focus accuracy results of three LLMs on the SS dataset across all of the baseline methods described in Appendix E.4.4 and the FIT method. We see that across all focus instructions and all models, FIT shows significant improvement over the baselines,

achieving very high focus accuracy across all focus instruction types and across all test sets with varying predictivity levels.

		$\varnothing$	focus(C)	focus(C) $\wedge$ ignore(S)	ignore(S)	focus(S)	focus(S) $\wedge$ ignore(C)
Mistral	Zero-shot	$0.886_{\pm 0.007}$	$0.909_{\pm 0.004}$	$0.899_{\pm 0.002}$	$0.871_{\pm 0.006}$	$0.423_{\pm 0.199}$	$0.305_{\pm 0.104}$
	Few-shot	$0.893_{\pm 0.003}$	$0.906_{\pm 0.005}$	$0.904_{\pm 0.006}$	$0.793_{\pm 0.036}$	$0.559_{\pm 0.168}$	$0.634_{\pm 0.116}$
	SFT(y)	$0.951_{\pm 0.005}$	$0.950_{\pm 0.004}$	$0.952_{\pm 0.005}$	$0.951_{\pm 0.004}$	$0.445_{\pm 0.275}$	$0.447_{\pm 0.270}$
	SFT(y_{focus})	$0.751_{\pm 0.126}$	$0.794_{\pm 0.108}$	$0.903_{\pm 0.042}$	$0.879_{\pm 0.054}$	$0.506_{\pm 0.248}$	$0.519_{\pm 0.241}$
	FIT	$\mathbf{0.953_{\pm 0.004}}$	$\mathbf{0.954_{\pm 0.004}}$	$\mathbf{0.955_{\pm 0.004}}$	$\mathbf{0.954_{\pm 0.004}}$	$\mathbf{0.999_{\pm 0.001}}$	$\mathbf{0.999_{\pm 0.001}}$
Llama	Zero-shot	$0.500_{\pm 0.000}$	$0.500_{\pm 0.000}$	$0.500_{\pm 0.000}$	$0.500_{\pm 0.000}$	$0.506_{\pm 0.003}$	$0.506_{\pm 0.002}$
	Few-shot	$0.674_{\pm 0.006}$	$0.838_{\pm 0.008}$	$0.630_{\pm 0.018}$	$0.508_{\pm 0.005}$	$0.491_{\pm 0.047}$	$0.502_{\pm 0.038}$
	SFT(y)	$\mathbf{0.952_{\pm 0.004}}$	$\mathbf{0.954_{\pm 0.003}}$	$\mathbf{0.952_{\pm 0.007}}$	$\mathbf{0.952_{\pm 0.008}}$	$0.445_{\pm 0.276}$	$0.444_{\pm 0.272}$
	SFT(y_{focus})	$0.668_{\pm 0.178}$	$0.686_{\pm 0.166}$	$0.803_{\pm 0.095}$	$0.795_{\pm 0.094}$	$0.606_{\pm 0.197}$	$0.601_{\pm 0.193}$
	FIT	$0.949_{\pm 0.002}$	$0.951_{\pm 0.002}$	$0.952_{\pm 0.002}$	$0.950_{\pm 0.002}$	$\mathbf{0.998_{\pm 0.001}}$	$\mathbf{0.999_{\pm 0.001}}$
Vicuna	Zero-shot	$0.420_{\pm 0.012}$	$0.584_{\pm 0.007}$	$0.381_{\pm 0.008}$	$0.355_{\pm 0.006}$	$0.199_{\pm 0.105}$	$0.129_{\pm 0.066}$
	Few-shot	$0.147_{\pm 0.010}$	$0.459_{\pm 0.014}$	$0.533_{\pm 0.012}$	$0.431_{\pm 0.010}$	$0.300_{\pm 0.150}$	$0.300_{\pm 0.125}$
	SFT(y)	$\mathbf{0.955_{\pm 0.005}}$	$\mathbf{0.956_{\pm 0.005}}$	$0.955_{\pm 0.004}$	$0.953_{\pm 0.006}$	$0.446_{\pm 0.278}$	$0.445_{\pm 0.277}$
	SFT(y_{focus})	$0.570_{\pm 0.180}$	$0.602_{\pm 0.187}$	$0.685_{\pm 0.138}$	$0.671_{\pm 0.129}$	$0.676_{\pm 0.129}$	$0.660_{\pm 0.118}$
	FIT	$0.950_{\pm 0.003}$	$0.953_{\pm 0.003}$	$\mathbf{0.955_{\pm 0.003}}$	$\mathbf{0.955_{\pm 0.004}}$	$\mathbf{0.999_{\pm 0.001}}$	$\mathbf{0.999_{\pm 0.001}}$

Table 33: **Complete Baselines vs. FIT Focus Accuracies on SS (↑).** For each method, we report the mean of focus-accuracy means ($\mathcal{A}_{\text{focus}}$) over the four test splits ($\mathcal{D}_{\text{iid}}$, $\mathcal{D}_{\text{high}}$, $\mathcal{D}_{\text{low}}$ and $\mathcal{D}_{\text{flipped}}$) $\pm$ the between-split standard deviation of these means (i.e. how performance varies as predictivity changes) across repeats. Boldface indicates the best mean for each focus instruction type and model independently.

E.6 Spurious NLI dataset (SMNLI)

We generate a tertiary NLI dataset, SMNLI, with a known spurious feature. We do this subsampling from the original MNLI dataset (Williams et al., 2018). This is a NLI dataset with three labels: entailment (0), neutral (1) and contradiction (2), where data is sampled from 6 underlying categories or genres (government, travel, and fiction, facetoface, nineeleven and verbatim). We aim to induce spurious correlations between the underlying genres and labels.

E.6.1 Data-Generating Process (DGP)

We consider a graphical model to describe the DGP of examples within the SMNLI dataset. We use the following variables within our DGP:

- C - NLI relationship between a premise and hypothesis pair, the core feature within this task, sampled from the original MNLI dataset.
- X - example premise and hypothesis from the MNLI dataset.

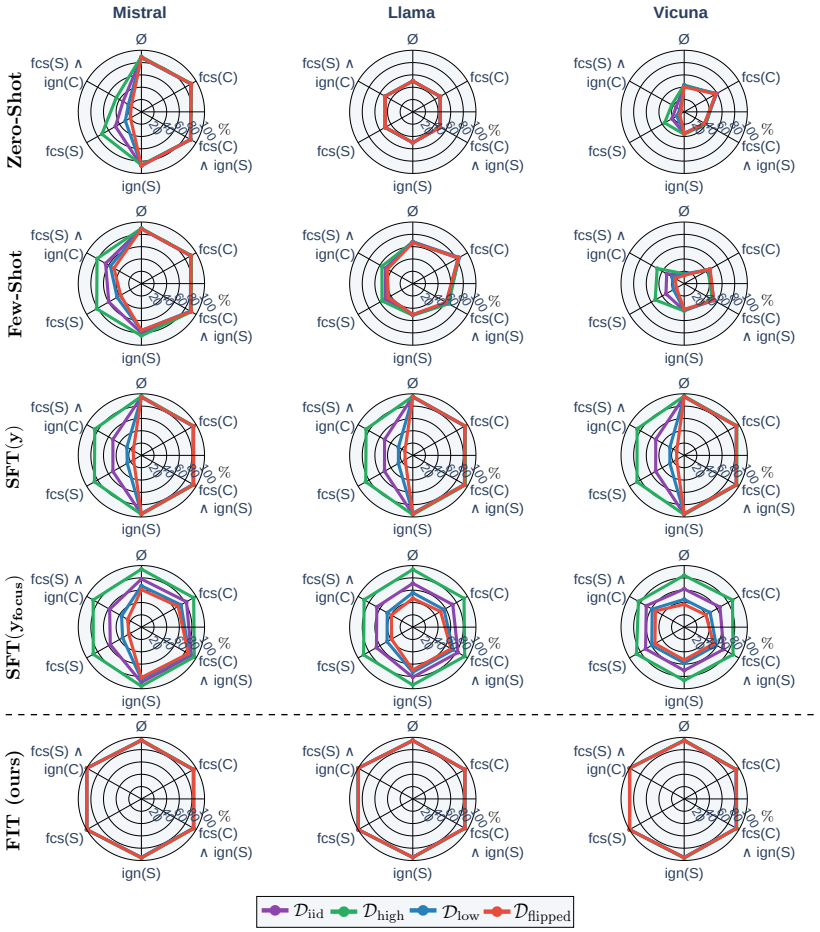


Figure 60: **Complete Baseline vs FIT Focus Accuracies on SS (↑).** Figure giving the mean focus accuracies ($\mathcal{A}_{\text{focus}}$) of the additional baselines compared to the focus accuracy of FIT on the SS dataset. The maximum standard deviations of FIT, SFT(y_{focus}), SFT(y) and few-shot methods across models are 2.17%, 14.8%, 0.65%, and 2.83% respectively. fcs = focus, ign = ignore.

- S - spurious feature present in the example X, here this is the genre of the premise and hypothesis texts. This is a categorical variable.
- Y - final label for example X.

The graphical model described by the DGP for producing the SMNLI dataset is given in Figure 61. Once again, this graphical model can be represented functionally as:

$$S = f_S(U_S); \tag{41}$$
$$C = f_C(S, U_C); \tag{42}$$
$$X = f_X(C, E, U_X); \tag{43}$$
$$Y = f_Y(C, U_Y). \tag{44}$$

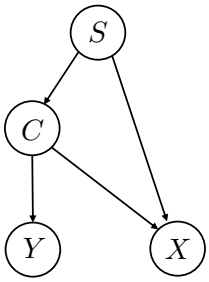


Figure 61: **SMNLI DGP**. Graphical model showing the data generating process for sub-sampling examples from the MNLI dataset to introduce a new spurious keyword feature S.

More specifically, given the original dataset $\mathcal{D}$ that we are sub-sampling from, the functions that we use within the DGP for the SMNLI dataset are given by:

$$S \sim \mathcal{U}(\mathcal{S}), \tag{45}$$

$$U_C \sim \text{Ber}(\rho_{\text{spurious}}); \tag{46}$$

$$C = U_C \, y_S + (1 - U_C) \, \mathcal{U}(\text{Val}(C) \setminus \{y_S\}) \tag{47}$$

$$X \ \sim p_{\mathcal{D}}(\, \cdot \mid C, S) \tag{48}$$

$$Y = C. \tag{49}$$

Here, $\mathcal{U}(\mathcal{S})$ denotes a uniform distribution over the set of genres $\mathcal{S}$, and we define $\text{Val}(C) = \{0, 1, 2\}$ corresponding to the possible NLI labels for this task. Furthermore, we define y_S to be the NLI label that a particular value of S is spuriously correlated with by design. Moreover, $p_{\mathcal{D}}(\, \cdot \mid C, S)$ is the conditional distribution over the dataset examples (premise-hypothesis pairs) that have NLI relationship C and genre S.

We restrict the genres that we sample from to $S \in \{\text{government}, \text{fiction}, \text{travel}\}$, a subset of the genres of the training set. When creating a distribution shifted test set, we restrict the genres to $S \in \{\text{facetoface}, \text{nineeleven}, \text{verbatim}\}$. The specific spurious relationships between genre values s and their associated spurious labels y_s are chosen to be: $y_{\text{slate}} = 0$; $y_{\text{government}} = 2$; $y_{\text{fiction}} = 1$; $y_{travel} = 0$; $y_{\text{facetoface}} = 2$; $y_{\text{nineeleven}} = 0$; $y_{\text{verbatim}} = 1$. In this way we generate spurious correlations within the dataset through sub-sampling to induce spurious correlations between S and Y.

We show that the notion of ρ_{spurious} in Equation (46) aligns with the notation in Definition 6.2 and that this does not depend on the spurious feature value of S.

Proposition E.4. From the DGP described above, we have that

$$\mathbb{P}(Y = y_s \mid S = s) = \rho_{\text{spurious}} \, . \tag{50}$$

for all spurious feature values s.

Proof. This is clear considering Equation (47), where ρ_{spurious} influences the chance that we sample y_s, i.e. the label spuriously correlated with feature value s. □

E.6.2 Independence Conditions During Training for FIT on SMNLI

As specified in Appendix E.4, we would like to have that $Y \perp\!\!\!\perp S$ and $Y_S \perp\!\!\!\perp C$ during training so that models trained via FIT can effectively learn to leverage focus instructions to make predictions based on specified features, where, again, Y_S is the label spuriously associated to spurious feature value S. The results below give sufficient conditions for this to occur with respect to the DGP described in Figure 61. This boils down to setting $\rho_{\text{spurious}} = 1/3$ during training.

Proposition E.5. Assuming the DGP described above and that $\rho_{\text{spurious}} = 1/3$, we have that $Y \perp\!\!\!\perp S$.

Proof. Note that since $Y = C$ in the DGP above, it suffices to show that $C \perp\!\!\!\perp S$. We have that for a given $S = s$, that $\mathbb{P}(C = y_s \mid S = s) = \rho_{\text{spurious}} = 1/3$ from Proposition E.4. Moreover, let $\tilde{C} \sim \mathcal{U}(\text{Val}(C) \setminus \{y_S\})$ denote the random variable sampled from a uniform distribution over the remaining possible values of C excluding y_s. Then for either value of $c \in \text{Val}(C) \setminus \{y_s\}$ we have that

$$\mathbb{P}(C = c \mid S = s) = \mathbb{P}(U_C = 0)\mathbb{P}(\tilde{C} = c) \tag{51}$$

$$= \frac{2}{3} \cdot \frac{1}{2} \tag{52}$$

$$= \frac{1}{3}. \tag{53}$$

Therefore, we have that $C \sim \mathcal{U}(\text{Val}(C))$. In particular, this then gives that $C \perp\!\!\!\perp S$, which in turn implies that $Y \perp\!\!\!\perp S$. □

Proposition E.6. Assuming the DGP described above and that $\rho_{\text{spurious}} = 1/3$, then we have that $Y_S \perp\!\!\!\perp C$.

Proof. Note that Y_S is a deterministic function of S, so that $Y_S = f(S)$ for some function $f : Val(S) \to \text{Val(Y)}$. Therefore, it suffices to show that $S \perp\!\!\!\perp C$. The proof of this is given in Proposition E.5. □

E.6.3 Results

We present full results comparing FIT on SMNLI against all of the baselines described in Appendix E.4.4 in Figure 62, Table 34 for completeness.

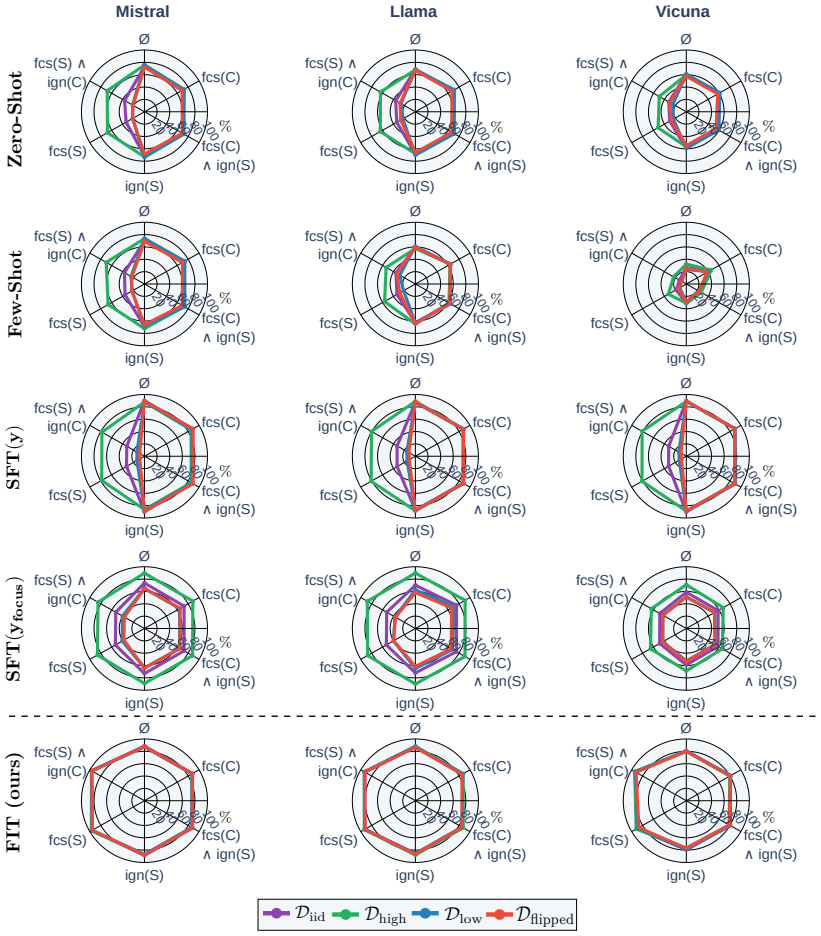


Figure 62: **Complete Baseline vs FIT Focus Accuracies on SMNLI (↑).** Figure giving focus accuracies ($\mathcal{A}_{\text{focus}}$) of the additional baselines compared to the focus accuracy of FIT on the SMNLI dataset. The maximum standard deviations of FIT, SFT(y_{focus}), SFT(y) and few-shot methods across models are 6.47%, 7.98%, 1.15%, and 0.500% respectively. fcs = focus, ign = ignore.

E.7 Additional BBQ Results

In Figure 63 and Table 35 we include the full comparisons of FIT against all baselines described in Appendix E.4.4 on BBQ.

		∅	focus(C)	focus(C) ∧ ignore(S)	ignore(S)	focus(S)	focus(S) ∧ ignore(C)
Mistral	Zero-shot	$0.742_{\pm 0.016}$	$0.711_{\pm 0.014}$	$0.711_{\pm 0.012}$	$0.716_{\pm 0.020}$	$0.362_{\pm 0.189}$	$0.368_{\pm 0.191}$
	Few-shot	$0.716_{\pm 0.020}$	$0.725_{\pm 0.017}$	$0.722_{\pm 0.017}$	$0.687_{\pm 0.023}$	$0.370_{\pm 0.178}$	$0.386_{\pm 0.189}$
	SFT(y)	$\mathbf{0.890_{\pm 0.009}}$	$0.875_{\pm 0.013}$	$\mathbf{0.875_{\pm 0.013}}$	$\mathbf{0.885_{\pm 0.013}}$	$0.334_{\pm 0.275}$	$0.332_{\pm 0.273}$
	SFT(y_{focus})	$0.732_{\pm 0.101}$	$0.727_{\pm 0.097}$	$0.725_{\pm 0.096}$	$0.726_{\pm 0.102}$	$0.544_{\pm 0.192}$	$0.537_{\pm 0.190}$
	FIT	$0.878_{\pm 0.005}$	$\mathbf{0.875_{\pm 0.004}}$	$0.874_{\pm 0.006}$	$0.878_{\pm 0.007}$	$\mathbf{0.963_{\pm 0.006}}$	$\mathbf{0.968_{\pm 0.003}}$
Llama	Zero-shot	$0.679_{\pm 0.007}$	$0.688_{\pm 0.017}$	$0.682_{\pm 0.018}$	$0.682_{\pm 0.015}$	$0.370_{\pm 0.162}$	$0.386_{\pm 0.151}$
	Few-shot	$0.595_{\pm 0.012}$	$0.641_{\pm 0.003}$	$0.630_{\pm 0.010}$	$0.636_{\pm 0.009}$	$0.351_{\pm 0.130}$	$0.379_{\pm 0.094}$
	SFT(y)	$\mathbf{0.880_{\pm 0.005}}$	$\mathbf{0.880_{\pm 0.003}}$	$\mathbf{0.879_{\pm 0.004}}$	$\mathbf{0.879_{\pm 0.004}}$	$0.352_{\pm 0.278}$	$0.354_{\pm 0.272}$
	SFT(y_{focus})	$0.700_{\pm 0.121}$	$0.759_{\pm 0.093}$	$0.750_{\pm 0.098}$	$0.720_{\pm 0.110}$	$0.552_{\pm 0.194}$	$0.536_{\pm 0.209}$
	FIT	$0.872_{\pm 0.007}$	$0.865_{\pm 0.008}$	$0.861_{\pm 0.009}$	$0.865_{\pm 0.010}$	$\mathbf{0.929_{\pm 0.005}}$	$\mathbf{0.931_{\pm 0.004}}$
Vicuna	Zero-shot	$0.595_{\pm 0.012}$	$0.598_{\pm 0.015}$	$0.575_{\pm 0.019}$	$0.573_{\pm 0.017}$	$0.334_{\pm 0.108}$	$0.341_{\pm 0.093}$
	Few-shot	$0.269_{\pm 0.032}$	$0.417_{\pm 0.027}$	$0.251_{\pm 0.019}$	$0.301_{\pm 0.013}$	$0.195_{\pm 0.080}$	$0.133_{\pm 0.060}$
	SFT(y)	$\mathbf{0.892_{\pm 0.003}}$	$\mathbf{0.890_{\pm 0.002}}$	$\mathbf{0.890_{\pm 0.004}}$	$\mathbf{0.891_{\pm 0.006}}$	$0.337_{\pm 0.288}$	$0.342_{\pm 0.283}$
	SFT(y_{focus})	$0.579_{\pm 0.081}$	$0.575_{\pm 0.064}$	$0.573_{\pm 0.055}$	$0.585_{\pm 0.063}$	$0.502_{\pm 0.092}$	$0.496_{\pm 0.086}$
	FIT	$0.803_{\pm 0.004}$	$0.807_{\pm 0.006}$	$0.794_{\pm 0.008}$	$0.781_{\pm 0.012}$	$\mathbf{0.894_{\pm 0.019}}$	$\mathbf{0.934_{\pm 0.010}}$

Table 34: **Complete Baselines vs. FIT Focus Accuracies on SMNLI (↑).** For each method, we report the mean of focus-accuracy means ($\mathcal{A}_{\text{focus}}$) over the four test splits ($\mathcal{D}_{\text{iid}}$, $\mathcal{D}_{\text{high}}$, $\mathcal{D}_{\text{low}}$ and $\mathcal{D}_{\text{flipped}}$) ± the between-split standard deviation of these means (i.e. how performance varies as predictivity changes) across repeats. Boldface indicates the best mean for each focus instruction type and model independently.

		∅	focus(C)	focus(C) ∧ ignore(S)	ignore(S)	focus(S)	focus(S) ∧ ignore(C)
Mistral	Zero-shot	0.772/0.792	0.768/0.803	0.782/0.812	0.790/0.829	0.338/0.320	0.302/0.295
	Few-shot	0.775/0.798	0.767/0.795	0.769/0.807	0.777/0.814	0.408/0.381	0.388/0.373
	SFT(y)	**0.999**/**0.982**	**1.000**/**0.983**	**1.000**/**0.982**	**1.000**/**0.981**	0.248/0.249	0.248/0.249
	SFT(y_{focus})	0.858/0.874	0.844/0.866	0.848/0.872	0.847/0.863	0.396/0.335	0.394/0.340
	FIT	0.998/0.976	0.998/0.976	0.998/0.977	0.999/0.976	**0.941**/**0.852**	**0.941**/**0.853**
Llama	Zero-shot	0.534/0.673	0.478/0.597	0.478/0.593	0.508/0.627	0.533/0.421	0.527/0.428
	Few-shot	0.643/0.696	0.686/0.743	0.724/0.774	0.642/0.720	0.496/0.429	0.492/0.432
	SFT(y)	**0.998**/**0.991**	**0.999**/**0.985**	**1.000**/**0.988**	**0.999**/**0.989**	0.248/0.244	0.248/0.246
	SFT(y_{focus})	0.836/0.912	0.825/0.901	0.836/0.903	0.852/0.910	0.403/0.317	0.418/0.323
	FIT	0.993/0.974	0.997/0.977	0.998/0.979	0.996/0.979	**0.943**/**0.832**	**0.946**/**0.835**
Vicuna	Zero-shot	0.444/0.495	0.495/0.567	0.505/0.531	0.479/0.519	0.371/0.324	0.375/0.315
	Few-shot	0.454/0.565	0.453/0.572	0.517/0.604	0.547/0.625	0.503/0.512	0.487/0.496
	SFT(y)	0.975/**0.991**	0.959/**0.988**	0.957/**0.990**	0.974/**0.990**	0.245/0.252	0.243/0.251
	SFT(y_{focus})	0.796/0.832	0.794/0.808	0.787/0.806	0.796/0.809	0.444/0.392	0.456/0.394
	FIT	**0.983**/0.979	**0.982**/0.975	**0.981**/0.972	**0.984**/0.973	**0.966**/**0.835**	**0.966**/**0.837**

Table 35: **Complete Baseline vs FIT Focus Accuracies on BBQ (↑).** We report across all baselines the mean seen/unseen focus accuracies, with boldface indicating the best mean for each focus instruction type and model independently. The maximum standard deviations of FIT, SFT(y_{focus}), SFT(y) and few-shot methods across models are 4.07%, 10.7%, 2.22%, and 0.600% respectively. fcs = focus, ign = ignore.

E.8 BBQ-NLG

E.8.1 BBQ-NLG Experimental Setup

We follow the original BBQ dataset splits (including both training and held-out social-bias partitions) described in Section 6.4.2. For each example in the new BBQ-NLG dataset, we

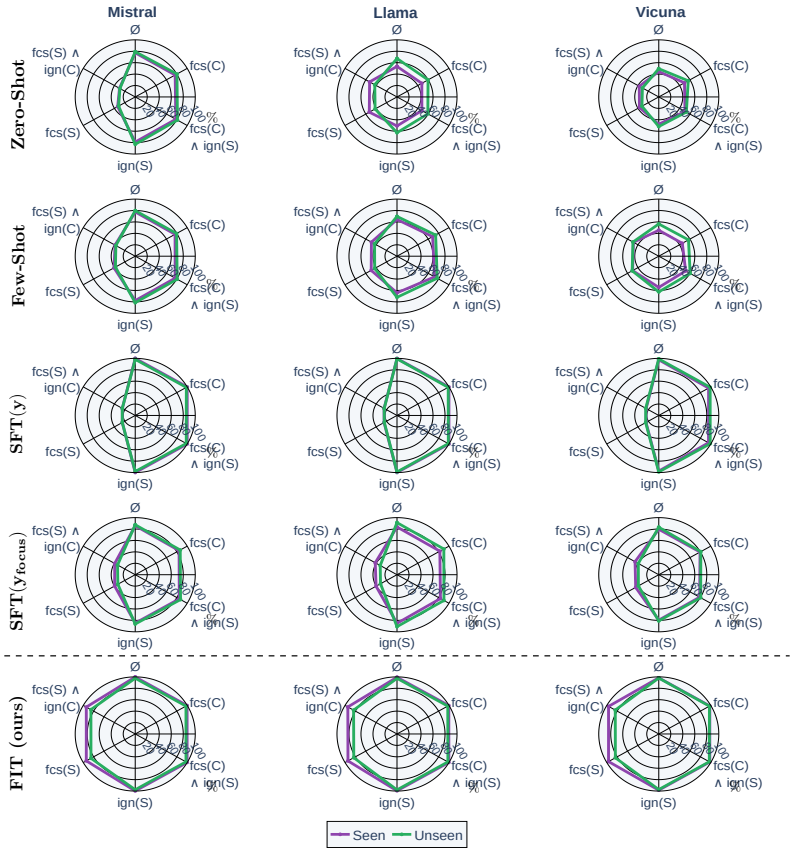


Figure 63: **Complete Baseline vs FIT Focus Accuracies on BBQ (↑).** Figure giving focus accuracies ($\mathcal{A}_{\text{focus}}$) of the additional baselines compared to the focus accuracy of FIT on the BBQ dataset. The maximum standard deviations of FIT, SFT(y_{focus}), SFT(y) and few-shot methods across models are 4.07%, 10.7%, 2.22%, and 0.600% respectively. fcs = focus, ign = ignore.

remove the predefined answer choices and require the model to generate fully verbalized responses, rather than selecting from a fixed set of three options as in the original BBQ dataset and outputting a categorical label.

To give the model additional capacity for this more challenging free-form generation task, we augment the usual LoRA targets (the key and value projection matrices) with adapters on the value-projection matrices as well. All LoRA modules are trained with a learning rate of 10^{-5}. We train both the FIT and SFT(y_{focus}) models for 10 epochs each. As a strong comparison, we evaluate a few-shot baseline that conditions the model on 4 in-context examples that share the same focus instruction as the test instance.

To enable both rapid and cost-effective evaluation of correctness in our open-ended generation task, we employ an LLM-based judge. Specifically, we use a pre-trained Llama-3.1-8B-Instruct model to compare each generated response against its associated ground-truth focus label and determine whether they are semantically equivalent. Manual

checks confirm that the Llama-3.1-8B-Instruct judge reliably assesses semantic equivalence in this setting where model generations and expected responses and generally short and concise.

E.8.2 Results

Figure 64 reports the focus-accuracy of each method across models. Although all approaches exhibit a slight degradation on unseen feature combinations relative to the classification-style BBQ results in Section 6.4.2, FIT remains generally overall on par with the earlier numbers. Critically, FIT consistently outperforms both the SFT(y_{focus}) and the strong 4-shot few-shot baseline that uses identical focus prompts at test time. These findings provide clear evidence that the FIT method can be effectively extended to open-ended NLG settings without sacrificing its steering capabilities.

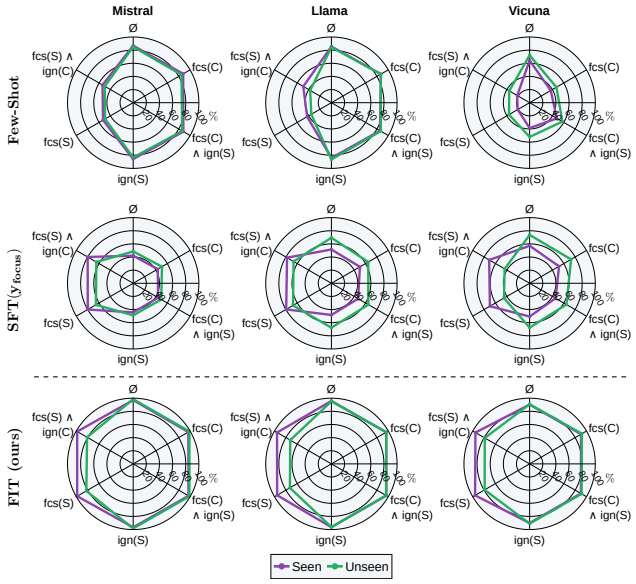


Figure 64: **BBQ-NLG Focus Accuracies (↑).** Mean focus accuracy ($\mathcal{A}_{\text{focus}}$) of baselines and FIT on the BBQ-NLG dataset. The maximum standard deviations of FIT, SFT(y_{focus}) and few-shot methods across models and $\mathcal{I}_{\text{focus}}$ are 2.45%, 6.35% and 0.377% respectively. fcs = focus, ign = ignore.

E.9 Comparison of FIT Against a Specific Debiasing Technique

FIT is a general framework designed to enable users to steer a model's behavior based on specified features. This approach provides enhanced control over model outputs during inference, adding a critical layer of explainability and controllability to model predictions.

While understanding and mitigating biases or spurious correlations is a valuable and natural application of FIT, it is not the sole objective. The broader goal of steerability includes addressing challenges in managing and aligning model behavior across diverse contexts. For instance, maintaining controllability is crucial in addressing safety alignment fragility, which can emerge after fine-tuning (Bhattacharjee et al., 2024). In such cases, the ability to adapt model responses to align with user specifications ensures safe and reliable deployment.

Experiment To explore FIT's broader applicability, we compare its performance as a debiasing method against a well-known debiasing technique: the Product of Experts (PoE) method (Mahabadi et al., 2020). PoE involves training a bias model f_B, which is trained exclusively on bias features. This bias model mediates the training of the final model f by combining their predictions through an elementwise product: $\sigma(f(x)) \odot \sigma(f_B(x_B))$, where $x \in \mathcal{D}$, for dataset $\mathcal{D}$, and x_B represents the biased feature of x.

We adapted this approach to our setting by training a bias model on the stereotypical labels within the BBQ dataset. These labels correspond to group-stereotypical associations. For autoregressive models, we further modified the PoE method by extracting and normalizing the logits of the first newly generated token position over the set of single tokens representing the answer options.

Results The results of the debiasing experiment comparing FIT to the PoE method is shown in Figure 65. FIT performs equally as well as the PoE method as shown by comparing the default prompt accuracy ($\emptyset$) for the PoE models against the focus(C) results for the FIT models; both metrics correspond to causal accuracy for these prompt types. This indicates that FIT performs just as well as the PoE dedicated debiasing technique.

However, the PoE method requires training two separate models and does not provide steerability at test time as shown by the low focus accuracy on focus(S). Indeed the model defaults to the ground truth label across all prompt types and does not change behavior in the presence of different focus specifications. This highlights the flexibility of FIT, which not only debiases effectively but also enables additional controllability during inference.

E.10 Spurious HANS Dataset (SHANS)

We generate a binary NLI dataset, SHANS, with a known spurious feature. We do this by subsampling from the HANS dataset (McCoy et al., 2019). This is an NLI data set with two labels: entailment (0) and contradiction (1). This is an adversarial dataset designed to assess different NLI models' reliance on spurious heuristics rather than on the underlying

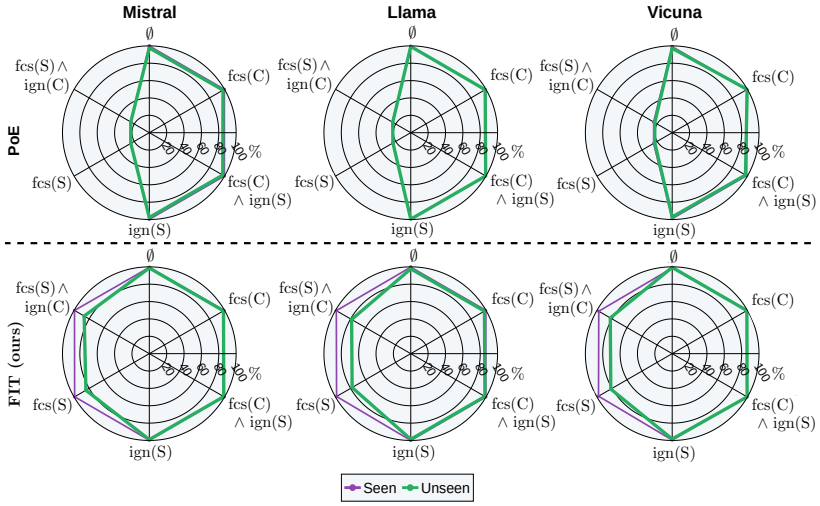


Figure 65: **Focus Accuracy of FIT Against PoE Debiasing Technique (↑).** Figure showing the focus accuracies ($\mathcal{A}_{\text{focus}}$) of FIT (bottom row) and the dedicated debiasing technique, PoE (top row), on the BBQ dataset.

relationship between the premise and the hypothesis when making NLI predictions. Specifically, the author's consider three major categories of heuristics: lexical overlap heuristic (assuming that a premise entails from words within the hypothesis) , sub-sequence heuristic (assuming that the premise entails all any of its contiguous sub-sequences of words) and constituent heuristic (assuming that a premise entails a hypothesis that is any constituent within it's syntactic parse tree). We take each of these as separate spurious features within our SHANS dataset for which we induce spurious correlations, as for the SMNLI dataset, through subsampling.

E.10.1 Data-Generating Process (DGP)

We consider a graphical model to describe the DGP of examples within the SHANS dataset. We use the following variables within our DGP:

- C - NLI relationship between a premise and hypothesis pair, the core feature within this task, sampled from the original HANS dataset.
- $S_{\text{lex.}}$ - spurious feature, here the presence of a hypothesis entirely made from words from the premise. This is a binary categorical variable (present/not present).
- $S_{\text{sub.}}$ - spurious feature, here the presence of a hypothesis that is a contiguous subsequence of the premise. This is a binary category feature (present/not present).

- $S_{\text{const.}}$ - spurious feature, here the presence of hypothesis that is a constituent/subtree of the premise. Here we have a binary variable (present/not present).
- X - example from the HANS dataset.
- Y - final label for example X.

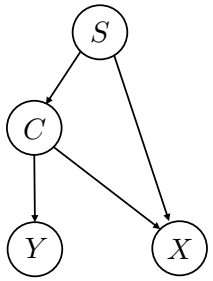


Figure 66: **SHANS DGP**. Graphical model showing the DGP for subsampling examples from the SHANS dataset to introduce new spurious features $S_{\text{lex.}}$, $S_{\text{sub.}}$ and $S_{\text{const.}}$ which are encoded within the categorical spurious feature S which represents one of these three heuristics.

The graphical model described by the DGP for producing the S-HANS dataset is given in Figure 66. Once again, this graphical model can be represented functionally as

$$S = f_S(U_S); \tag{54}$$
$$C = f_C(S, U_C); \tag{55}$$
$$X = f_X(C, E, U_X); \tag{56}$$
$$Y = f_Y(C, U_Y), \tag{57}$$

where here we define S to be a categorical feature over the set of variables indicating the presence of each of the three heuristics introduced above which we denote, through overloaded notation, by $\mathcal{S} = \{s_{\text{lex.}}, s_{\text{sub.}}, s_{\text{const.}}\}$. More specifically, given the original HANS dataset $\mathcal{D}$ that we are sub-sampling from, the functions that we use within the DGP for the SHANS dataset are given by:

$$S \sim \mathcal{U}(\mathcal{S}), \tag{58}$$
$$U_C \sim \text{Ber}(\rho_{\text{spurious}}); \tag{59}$$
$$C \sim U_C y_S + (1 - U_C)(1 - y_S); \tag{60}$$
$$X \sim p_{\mathcal{D}}(\cdot \mid C, S) \tag{61}$$
$$Y = C. \tag{62}$$

Here, $\mathcal{U}(\mathcal{S})$ is a uniform categorical distribution over the set of spurious features $\mathcal{S}$ which effectively selects the presence of exactly one of the three spurious feature heuristics. We define y_S to be the NLI label that a particular value of S is spuriously correlated with by design. Moreover, $p_{\mathcal{D}}(\cdot \mid C, S)$ is the conditional distribution over the dataset examples (premise-hypothesis pairs) that have NLI relationship C and the presence of spurious heuristic S.

We restrict the genres to $S \in \{s_{\text{lex.}}, s_{\text{const.}}\}$ for heuristics seen during training. We then create additional test sets with spurious feature set restricted to $S \in \{s_{\text{sub.}}\}$, which serves as an unseen spurious feature set to test generalization.

We consider the presence of each feature to be separate binary spurious features. The specific spurious correlations between heuristics and labels Y are chosen to be: $y_{S_{\text{lex.}}=1} = 0$; $y_{S_{\text{sub.}}=1} = 0$; $y_{S_{\text{const.}}=1} = 1$. In this way we generate spurious correlations within the dataset through sub-sampling to induce spurious correlations between the heuristics S and label Y.

E.10.2 Transferred Results from SMNLI

As we have effectively used the same DGP as for the SMNLI dataset described in Appendix E.6, with the only change being the label set, all of the results that we have proven for SMNLI, translate to the SHANS dataset. In particular, we have that ρ_{spurious} aligns with the notation in Definition 6.2, and that we have $Y \perp\!\!\!\perp S$ and $Y_S \perp\!\!\!\perp C$ under the assumptions of $\rho_{\text{spurious}} = 1/2$ within the training set.

E.11 FIT on SHANS

Here, we give the results performing FIT on the SHANS dataset.

E.11.1 Spurious HANS (SHANS) Dataset

We generate binary NLI dataset sub-sampled from HANS (McCoy et al., 2019), a dataset designed to challenge NLI models by exposing common heuristics they rely on, such as lexical overlap (whether the hypothesis shares many words with the premise), sub-sequence (whether the hypothesis is a contiguous sub-sequence of the premise), and constituent (whether the hypothesis is a grammatical sub-structure of the premise). The presence of these heuristics are spuriously correlated with labels through sub-sampling of the presence of each of the heuristics from the original dataset. The degree of co-occurrence is governed by ρ_{spurious}, which varies according to the test sets described in Section 6.3. We ensure that ρ_{spurious} is the same for all feature values within each dataset split. In particular, we set ρ_{spurious} to be $0.5, 0.5, 0.9, 0.25$ and 0.9 (in this case with flipped spurious correlations) on

$\mathcal{D}_{\text{train}}$, $\mathcal{D}_{\text{iid}}$, $\mathcal{D}_{\text{high}}$, $\mathcal{D}_{\text{low}}$ and $\mathcal{D}_{\text{flipped}}$ respectively. We keep the subsequence heuristic as a held-out unseen feature during training for testing FIT generalization.

E.11.2 Results

Figure 67 shows the focus accuracy results of performing FIT on the SHANS dataset for the Llama-3.1-8B-Instruct model. As expected, on the seen features, FIT shows high focus accuracy across all focus instructions. However, for unseen features, we observe lower focus accuracy. This could be attributed to the overlapping nature of the heuristics in SHANS, which are often graded versions of each other with different levels of specificity. For instance, the sub-sequence heuristic can overlap with both lexical overlap and constituent heuristics (e.g., the example with Premise: "Before the actor slept, the senator ran" and Hypothesis: "The actor slept." satisfies all three heuristics). This overlap likely confuses the model during generalization, as it struggles to distinguish between heuristics seen during training and those that were not. These results suggest a potential limitation of FIT when dealing with features that are not sufficiently distinct or have significant overlap.

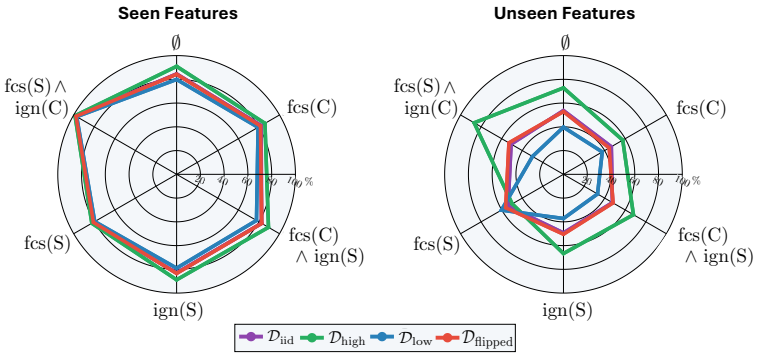


Figure 67: **SHANS Focus Accuracies (↑).** Focus accuracy ($\mathcal{A}_{\text{focus}}$) of Llama-3.1-8B-Instruct after FIT on the SHANS test datasets containing either the seen or unseen spurious features during training. Here, C refers to the core feature (logical relationship between premise and hypothesis) and S the spurious feature (heuristic used).

Appendix F: IllusionBench

F.1 Author Contributions

- **Arshia Hemmat:**
  - *Dataset Generation:* Responsible for the development of the Illusion Generation Pipeline, meticulously designing and implementing the system to create high-quality illusions. Efforts included fine-tuning the hyperparameters to ensure the generated images met the desired standards of clarity and effectiveness. This process involved extensive experimentation and adjustment to achieve optimal results.
  - *Experiments and Method:* Conducted the zero-shot experiments for all the models, which involved setting up, running, and analyzing the results of these experiments.
  - *Data Annotation:* Managed data annotation involving over 100 individuals, collected and analyzed the annotation results to ensure data quality and relevance.
  - *Paper Writing:* Co-writer of the basics of the chapter Section 7.4.
- **Adam Davies:**
  - *Experiments and Method:* Co-supervised benchmark design, generation, evaluation, zero-shot experiments, and results analysis/visualization; implemented experimental prototypes for VLM generation and evaluation; assisted in model selection and deployment.
  - *Data Annotation:* Designed data annotation procedure; wrote onboarding materials for annotators; prototyped data annotation setup and assisted in deployment and analysis.
  - *Paper Writing:* Co-wrote, edited, and revised all sections of the paper; co-shaped central story (motivation, contribution, relationship with prior work in computer and human vision, results analysis, social impact); literature review (shape recognition in human vision, shape recognition benchmarks).
- **Tom A. Lamb:**

- *Experiments:* Co-shaped design of ICL experiments; implemented and carried out all ICL experiments and co-led the analysis and presentation of ICL results.
- *Paper Writing:* Led writing of Section 7.5. Additionally, contributed to the writing and presentation of Section 7.4.

- **Jianhao Yuan:**
  - *Experiments:* Carried out all domain generalization experiments Section 7.6. Additionally, carried out zeroshot experiment of GPT-4o and Gemini in Section 7.4.
  - *Paper Writing:* Led writing of Section 7.6.
- **Philip Torr:** Provided feedback and advice regarding the project direction and proposed approach; assisted in securing compute resources to carry out experiments.
- **Ashkan Khakzar:**
  - *Idea* Conceived the research problem and idea (the idea to evaluate VLMs on shapes represented by an arrangement of visual scene elements)
  - *Method* Identified the existing method to generate such images. Demonstrated proof of concept (that state-of-the-art VLMs cannot identify these shapes)
  - *Literature review* On shape recognition in computer vision and co-shaped the storyline
  - *Experiments* Project co-supervision (curating the dataset Section 7.3, and zero-shot experiments Section 7.4)
  - *Writing* Co-writing of abstract, introduction, related works, and conclusion.
- **Francesco Pinto:**
  - *Experiments and Method:* Led benchmark design, generation and evaluation; led design, results analysis and visualization of zero-shot, in-context learning and multi-domain generalization experiments.
  - *Data Annotation:* Implemented and tested data annotation procedure, assisted in preparing the onboarding materials for annotators; collected and analyzed the results of the annotation.
  - *Paper Writing:* Co-shaped central story (motivation, contribution, relationship with prior work in computer vision, results analysis, impact); literature review (on shape recognition in computer vision); co-wrote the abstract and all sections of the paper but conclusions. Co-supervised full project.

F.2 Dataset Documentation and Additional Information

F.2.1 Human Annotation Details

Subsampling for Annotation Given the size of our dataset (more than 32K samples) performing a complete annotation of it would be expensive. Furthermore, since the data is synthesized and we perfectly know the class of the shapes represented in each image, the purpose of the annotation is simply to verify that the generated images have shapes that are recognizable by humans. For this reason, we subsample the generated dataset by enforcing that, for each dataset (i.e., each of `IllusionBench-IN`, `IllusionBench-ICON`, and `IllusionBench-LOGO`), at least one conditioning image from each class and scene choice is annotated.

Furthermore, we observe that the difficulty in perceiving an object depends on the choice of the hyperparameters that control the diffusion process. For this reason, we additionally enforce that images are uniformly sampled from each hyperparameter setting so that annotators are exposed to images encompassing the full range of difficulty.

Participants Our human evaluation involved 106 participants. The annotators were first instructed about the task and required to perform a simple test on 10 images, in order to make sure they understood the task to be performed. Annotators participated on a purely volunteer basis and were awarded with in-course credit. Participation was not mandatory for any student or course. No risks were identified for the annotation process.

Further annotation meta-data The number of annotators for `IllusionBench-IN`, `IllusionBench-ICON`, and `IllusionBench-LOGO` is respectively of 35 each. We split the original data so that each sample is annotated twice. Each reviewer is assigned approximately 80 samples (since the splitting algorithm is randomized, they may receive slightly less or slightly more samples).

F.2.2 Image Generation Hyperparameters

For data generation, we focused on the Illusion Diffusion generative models (demo available here), containing three major components:

- ControlNet (Zhang et al., 2023a), specifically: controlv1p sd15 qrcode monster
- Base Model, specifically: RealisticVision V5.1 noVAE, built using Stable Diffusion (Rombach et al., 2022)

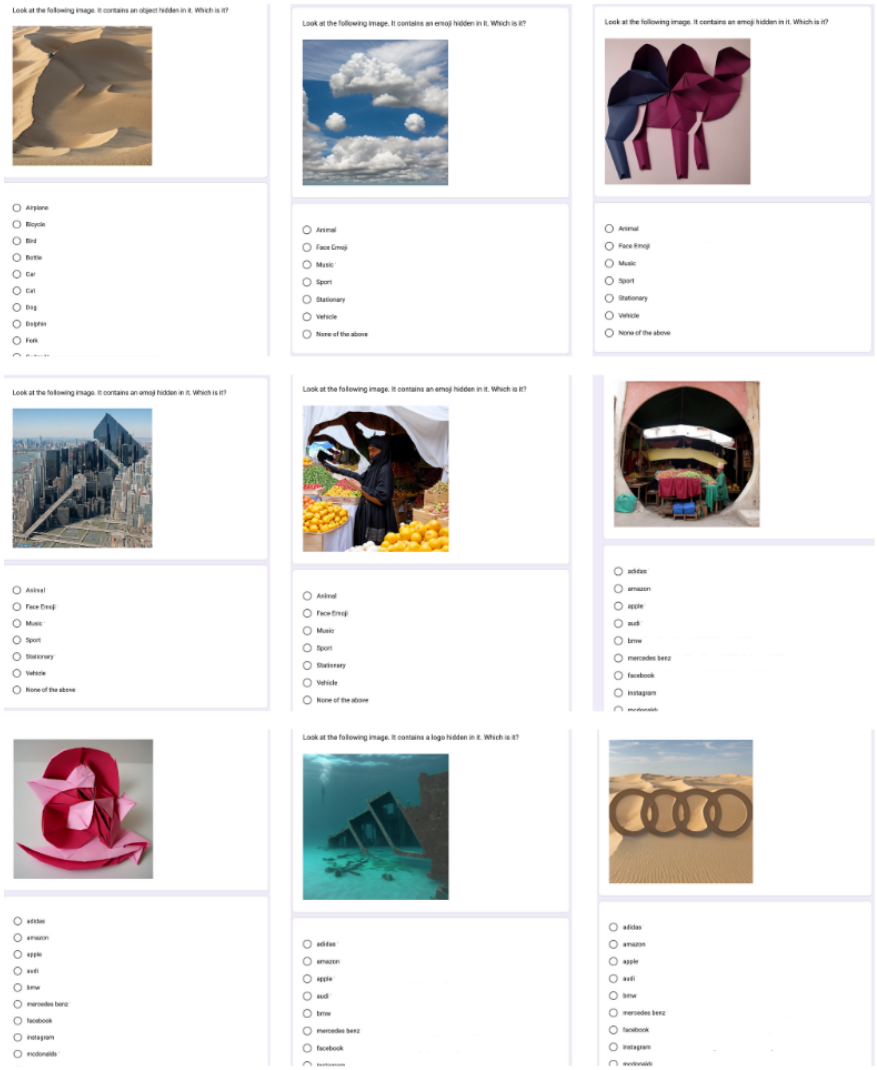


Figure 68: **Human Annotation** screenshots from GoogleForm through which the images are annotated by human annotators.

- Stable Diffusion-guided VAE, specifically: sd-vae-ft-mse

We used the following generation hyperparameters:

- Prompts were simply a single word corresponding to the scene types (e.g., "city" or "museum")
- Guidance-scale was always set to default value **7.5**
- Illusion_strength, which can be used to modulate the strength of abstract shape patterns, was selected based on our anecdotal observations regarding an appropriate difficulty level for each dataset (see below) and validated using human data annotation (as described above)
- Sampler was always set to default value **Euler**

The Illusion_strength for the different datasets are as follows:

- {Illusion_strength} of the `IllusionBench-LOGO` and `IllusionBench-IN`: [0.75, 0.80, 0.85, 0.90, 1.05, 1.10, 1.15, 1.20, 1.25, 1.35, 1.40, 1.50, 1.60]
- {Illusion_strength} of the `IllusionBench-ICON`: [0.85, 1.05, 1.25, 1.40]

F.2.3 Limitations

For future work, we will create more complex images and define more tasks in order to challenge models. We have also increased the size of our dataset so that we can train large models using our dataset. A current limitation is that we only hide a single shape in each image. Future work could extend this to incorporating several objects within the same background. Finally, we also plan to experiment with further tasks for compositional understanding and scene understanding of SOTA models. We leveraged prompt engineering to report the best possible performance of each model in the zero-shot case as described in Section F.3 and Section 7.4, however, improvements may be possible. We describe several limitations of the methods explored in this work in Sections 7.4 and 7.5.

F.2.4 Data Samples

To illustrate the quality of abstract shape recognition images created for this dataset, we randomly sample one image from several scene types in each dataset and display them in Figure 69.

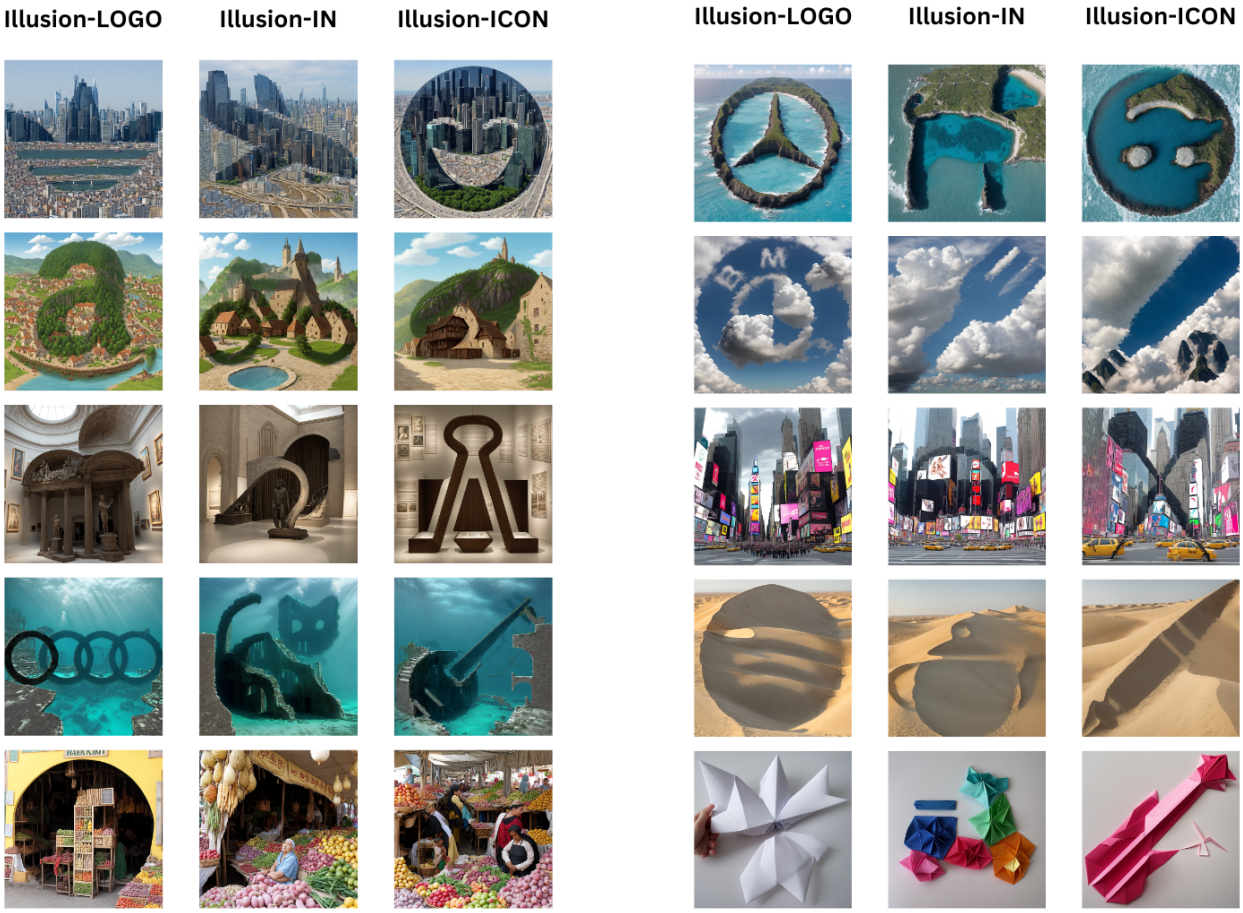


Figure 69: **Image Samples** from each dataset in our benchmark.

F.3 Zero-Shot Experiments Details

F.3.1 Zero-shot Experiments

We test our models zero-shot to evaluate their abstract shape recognition abilities. To leverage all capabilities of these models, we describe the conditions of our experiments in our prompt. The models are then asked to choose the correct shape type among a closed set of options, which include both shapes and scene names.

Let us focus on the predictive task τ_C. Analogous formulations hold for τ_S and $\tau_{C,S}$. Given that the model can correctly assign the class $y^C_{i_*}$ to the hidden shape in the scene $x^C_{i_*}$,

we provide it with a set of options $\mathcal{O}$, which includes all the shapes and scene names considered in the dataset split. We then ask the model to predict the shape name from these options.

Define $\mathcal{O} = \{\text{shape}_1, \text{shape}_2, \ldots, \text{scene}_1, \text{scene}_2, \ldots\}$ as the set of possible options. The model's response is evaluated based on whether the correct shape name is present in its output.

F.3.2 Models

In our zero-shot experiments, we evaluate each of the following large vision language models (VLMs):

- BlipV2-T5 (Li et al., 2023b), a VLM utilizing the T5 architecture (Raffel et al., 2020) for text encoding and a state-of-the-art vision encoder, designed for high-performance multimodal tasks.
- CogVLM (Wang et al., 2024a), an advanced VLM leveraging a Vision Transformer (ViT) (Dosovitskiy et al., 2020) and a powerful language model fine-tuned for vision-language reasoning tasks.
- InstructBlip-T5 (Dai et al., 2024b), a model combining the T5 architecture (Raffel et al., 2020) for text processing with a highly efficient vision encoder, fine-tuned for instructional prompts and multimodal interactions.
- LLaVA-Next (Vicuna-7b) (Liu et al., 2024b), a VLM using Vicuna-7b-v1.5 (Zheng et al., 2024b) and CLIP ViT-L/14 (Radford et al., 2021) as text and visual encoders, respectively. These are connected via simple projections.
- Qwen-VL-Chat (Yang et al., 2024a), a 9B parameter model employing a cross-attention module to link an OpenClip ViT-bigG (Ilharco et al., 2021) vision encoder to a Qwen-7b (Yang et al., 2024a) text backbone.
- Llava1.5-7b and 13-b (Liu et al., 2024a), a VLM employing a 7-billion parameter language model and advanced visual encoder, connected via efficient projections.
- InstructBlip-7b and 13b (Dai et al., 2024b,b), a BLIP (Li et al., 2022) model fine-tuned using instruction tuning, using a 7-billion parameter language model and a high-resolution vision encoder for precise multimodal understanding.

- MoE-StableLM, MoE-Qwen, MoE-Phi2 (Lin et al., 2024), a mixture of experts (MoE) model combining StableLM architecture (Raffel et al., 2020) with multiple expert models for dynamic task specialization and improved performance.
- GPT-4o, a multimodal version of GPT-4 (OpenAI, 2023), incorporating optimized end-to-end multimodal encoding of images, text, and audio for improved multimodal task performance.
- Gemini-Flash (Team et al., 2023), a high-speed VLM combining the latest advancements in vision transformers (Dosovitskiy et al., 2020) and language models for rapid and accurate multimodal analysis.

Note that, for the last two models in this list, we are unable to provide any specific information regarding their respective architectures or training regimes, as this information has not been made publicly available.

F.3.3 Prompts

We use the following general prompt template for our zero-shot experiments:

- T1 Prompt: `This image contains a {shape} integrated into a background, where elements of the background contribute to forming the {shape}. Identify the {shape} that is represented in the image by choosing exclusively among the following options: {shape_options}, {background_classes}. Provide your response by stating only the single, most accurate class name that represents the {shape}. You have to respond with a single word.`
- Texture Question Bias: `This image contains a {shape} integrated into a background, where elements of the background contribute to forming the {shape}. Identify the background that is represented in the image by choosing exclusively among the following options: {shape_options}, {background_classes}. Provide your response by stating only the single, most accurate class name that represents the background. You have to respond with a single word.`

where `shape` $\in$ {logo, shape, icon}for the dataset IllusionBench-LOGO, IllusionBench-IN and IllusionBench-CI respectively.

F.3.4 Text Generation Hyperparameters

For all VLMs, we use full-precision weights (i.e., no quantization), generating responses using greedy decoding without sampling, and limit the maximum response length to 100 tokens.

F.3.5 Zero-Shot Results By Class

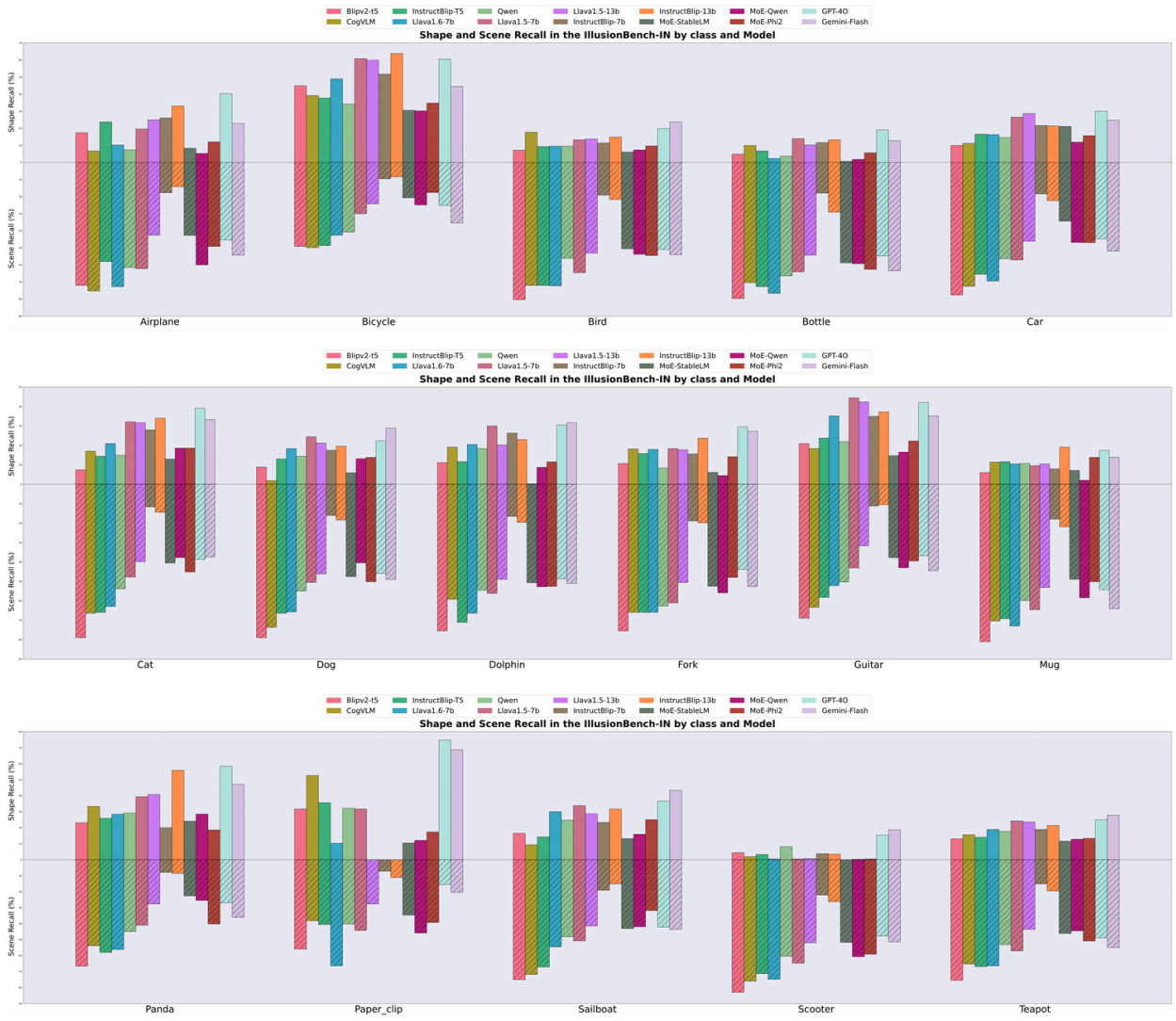


Figure 70: **Zero-shot results on `IllusionBench-IN` by class.** Zero-shot shape and scene recall of VLMs for each class in the `IllusionBench-IN` dataset.

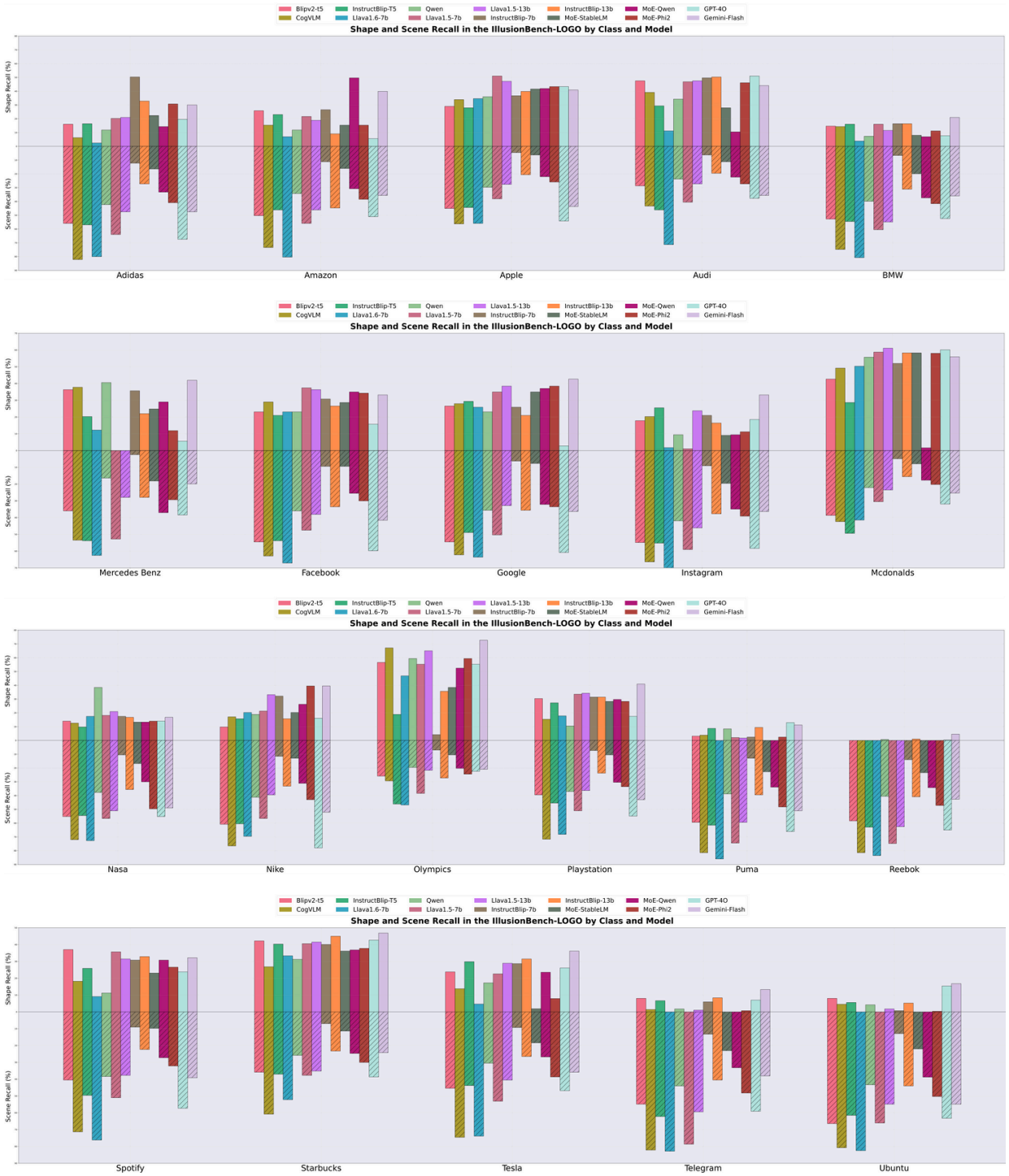


Figure 71: **Zero-shot results on `IllusionBench-LOGO` by class.** Zero-shot shape and scene recall of VLMs for each class in the `IllusionBench-LOGO` dataset.

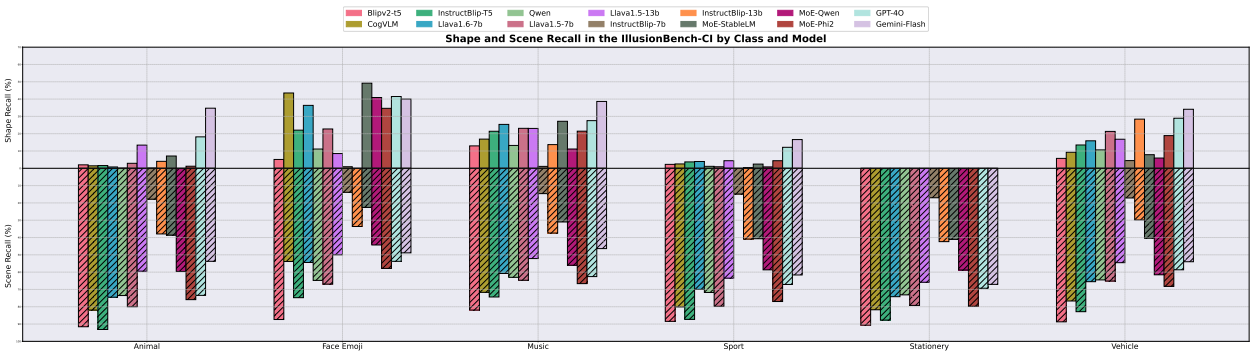


Figure 72: **Zero-shot results on `IllusionBench-ICON` by class.** Zero-shot shape and scene recall of VLMs for each class in the `IllusionBench-ICON` dataset.

F.4 In-Context Learning Experiments Details

F.4.1 In-Context Learning (ICL)

ICL is a method of adapting a model for an unseen task without any additional training or fine-tuning. Specifically, n-shot ICL consists of sequence of labeled demonstrations $\mathcal{C} = \{(x_{i_1}, y_{i_1}), \cdots, (x_{i_n}, y_{i_n})\}$. These are supplied to a model $p_{\boldsymbol{\theta}}(y \mid x)$ for an unseen task. The label corresponding to a test query x_* is predicted through the predictive distribution of the model conditioned on the demonstration set $\mathcal{C}$ alongside an instruction I for the new task:

$$p_{\boldsymbol{\theta}}(y \mid \mathcal{C}, I) = p_{\boldsymbol{\theta}}(y \mid x_{i_1}, y_{i_1}, \cdots x_{i_n}, y_{i_n}, I). \tag{63}$$

This learning method has proven to be an efficient and low-cost method for adapting LLMs to downstream tasks (Brown et al., 2020; Schick and Schütze, 2021; Winata et al., 2021; Liu et al., 2022a). The success of ICL for LLMs has led to recent research aiming to extend ICL to multi-modal models, where labeled demonstrations now contain interleaved image and text modalities (Alayrac et al., 2022; Bertini Baldassini et al., 2024; Zhao et al., 2023; Zong et al., 2024).

F.4.2 ICL Further Experimental Details

Considering we restrict evaluations to classes recognized in a zero-shot manner, we use the following class counts: 10 for the `IllusionBench-LOGO` split, 14 for the `IllusionBench-IN` split, and 6 for the icons split, utilizing all 11 scenes of the dataset. To overcome ICL biases like majority voting and recency bias, each shape and scene class is represented at most once within the context, with no repetitions, and new demonstrations are randomly sampled for each test sample.

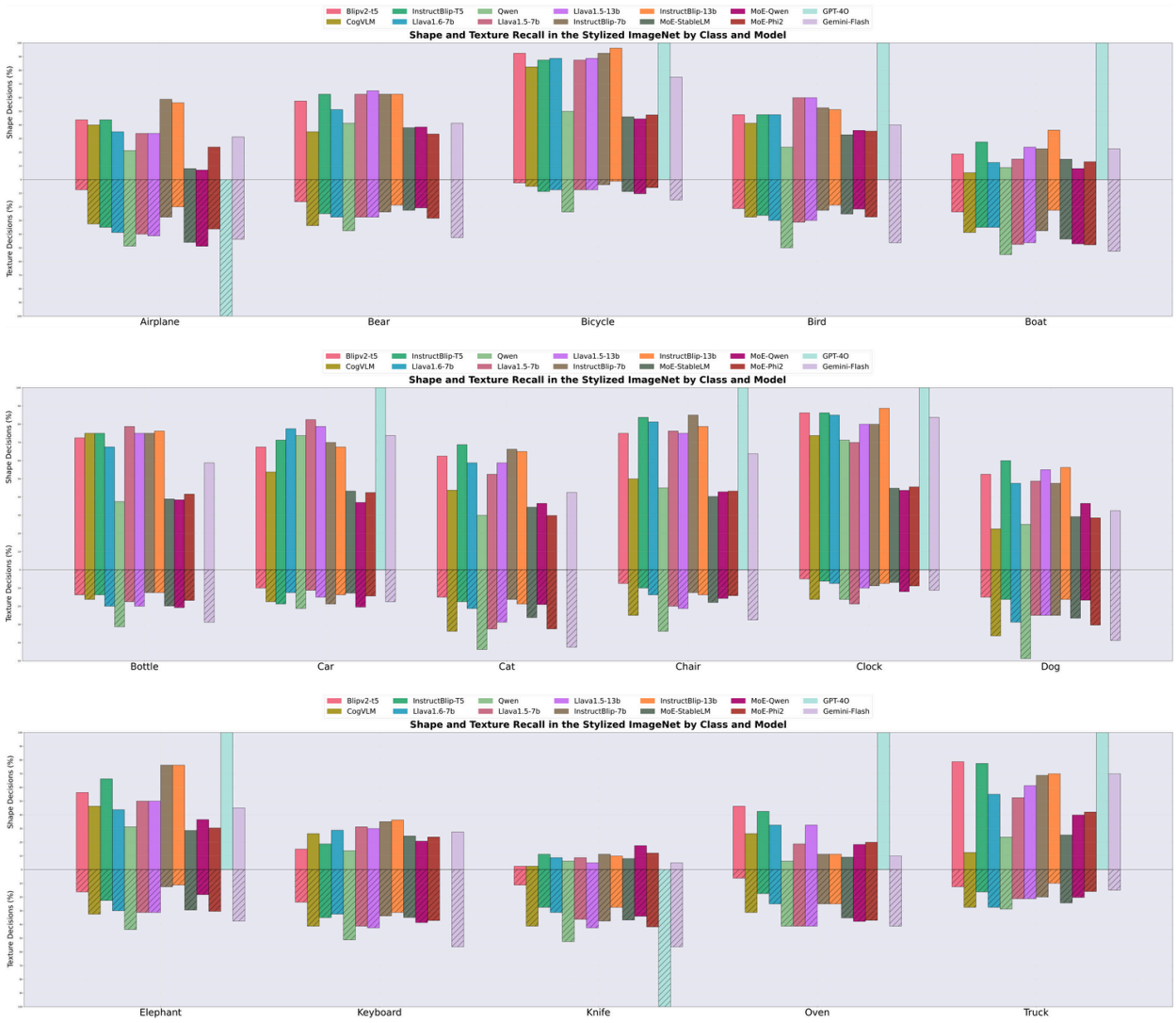


Figure 73: **Zero-shot results on Stylized ImageNet by class.** For comparison, we also report the zero-shot shape and texture bias of VLMs on the Stylized ImageNet dataset (Geirhos et al., 2018).

F.4.3 Models Description

In our zero-shot experiments, we evaluate each of the following large vision language models (VLMs):

- LLaVA-Next (Vicuna-7b) (Liu et al., 2024b), a VLM operating at an input image resolution of 336^2, using Vicuna-7b-v1.5 (Zheng et al., 2024b) and CLIP ViT-L/14 (Radford et al., 2021) as text and visual encoders, respectively. These are connected via simple projections.

- Qwen-VL-Chat (Yang et al., 2024a), a 9B parameter model with an input resolution of

448^2, employing a cross-attention module to link an OpenClip ViT-bigG (Ilharco et al., 2021) vision encoder to a Qwen-7b (Yang et al., 2024a) text backbone.

- Otter-MPT (Li et al., 2023a), a 9B parameter VLM based on the OpenFlamingo architecture (Awadalla et al., 2023), featuring an input image resolution of 224^2 and utilizing LLaMA-7B (Touvron et al., 2023a) and CLIP-ViT-L/14 as text and image backbones, respectively, connected through cross-attention.
- IDEFICS-9B-Instruct (Laurençon et al., 2024), an open-source reproduction of Flamingo (Alayrac et al., 2022), with an input image resolution of 224^2, using cross-attention transformer blocks to connect LLaMA and OpenClip text and image backbones.
- MMICL-T5-XXL (Zhao et al., 2023), a 12B parameter model that employs a Q-former (Li et al., 2023b) to integrate language and image components within an InstructBlip-FLANT5-XXL (Dai et al., 2024b) backbone. This model can handle complex prompts with interleaved text and images, allowing for text-image references through dummy demonstration tokens, and operates at an input image resolution of 224^2.

F.4.4 Prompts

We use the following general prompt template for our ICL experiments:

```
{TASK_INSTRUCTION}
{demonstration_image_1}
Answer: {demonstration_label_1}
{demonstration_image_2}
Answer: {demonstration_label_2}
⋮
{demonstration_image_n}
Answer: {demonstration_label_n}
{query_image}
Answer:
```

where `demonstration_image_i` and `demonstration_label_i` refer to the image and label for the ith demonstration used as the context for predicting the answer for the query image `query_image`. `TASK_INSTRUCTION` is the instruction used based on the prediction

target and the dataset. We used the following `TASK_INSTRUCTION` prompts for predicting the shape, texture, and both the texture and shape simultaneously respectively:

```
# Predict shape
TASK_INSTRUCTION =`This image contains a {shape} integrated into a
background, where elements of the background contribute to forming the
image.
background options: [{BG_OPTIONS}]
{shape} options: [{SHAPE_OPTIONS}]
Identify the {shape} that is represented in the image by choosing
among the provided options. Provide your response by stating only the
single, most accurate option that represents the {shape} in the image.
You have to respond with a single word.'

# Predict texture
TASK_INSTRUCTION = `This image contains a {shape} integrated into a
background, where elements of the background contribute to forming the
image.
background options: [{BG_OPTIONS}]
{shape} options: [{SHAPE_OPTIONS}]
Identify the background that is represented in the image by choosing
among the provided options. Provide your response by stating only the
single, most accurate option that represents the background in the
image. You have to respond with a single word.'

# Predict both texture and shape
TASK_INSTRUCTION = `This image contains a {shape} integrated into a
background, where elements of the background contribute to forming the
image.
background options: [{BG_OPTIONS}]
{shape} options: [{SHAPE_OPTIONS}]
Identify BOTH the background AND the {shape} that are represented
in the image by choosing among the provided options. Provide your
response by stating only the single, most accurate options that
represent the background and the {shape} in the image respectively.
```

```
You have to respond with two words, one predicting the background and
one predicting the {shape}'
```

where `shape` ∈ {logo, object, icon}for the dataset IllusionBench-LOGO, IllusionBench-IN and IllusionBench-CI respectively.

F.4.5 Text Generation Hyperparameters

For all VLMs, we use full-precision weights (i.e., no quantization), generating responses using greedy decoding without sampling, and limit the maximum response length to 100 tokens.

F.4.6 ICL Results: Exceptions

We list the exceptions to the general treneds reported in Section 7.5. We maintain the key takeaway headings and format in Section 7.5 and discuss key exceptions.

- *ICL does not mitigate tendency to predict scene over shape.* `LLaVA` on the task τ_C (along the first row) stands as an exception, where the model demonstrates low scene prediction accuracy and non-trivial performance shape accuracy on ICL2 and ICL4.
- *Context selection strategy effects prediction tasks differently.*
  - τ_C : For `LLaVA`, including the shape through ICL2 or ICL4 for 1 or 2 shots leads to a significant performance increase over all other models. This is especially evident for 1-shot, where we see high shape accuracy values of ICL2: 97.9% and ICL4: 99.9%. These high accuracy values indicate that the model exhibits a copying phenomenon (Bertini Baldassini et al., 2024), where for 1-shot, it simply copies the label from the ICL demonstration, which will have the same test label.
  - τ_S: `QWEN` shows an improvement in scene accuracy (for 4-shot, scene accuracies are ICL1: 51.4% and ICL3: 88.1%) when the scene is included in the context. Additionally, `LLaVA` exhibits a similar copying phenomenon for scene prediction in ICL3 and ICL4 as discussed for τ_C but also shows some improvements over zero-shot for 2-shots.
  - $\tau_{C,S}$: As an exception, `OTTER` and `QWEN` show a general increase in scene accuracy on $\tau_{C,S}$ compared to τ_S, while their shape accuracy remains similar to τ_C. This suggests that predicting both shape and scene and including demonstrations with such predictions can help these models better disentangle scene from shape. Again, we observe the copying mechanisms in `LLaVA` described for τ_C and τ_S.

F.4.7 Individual Dataset Splits ICL Results

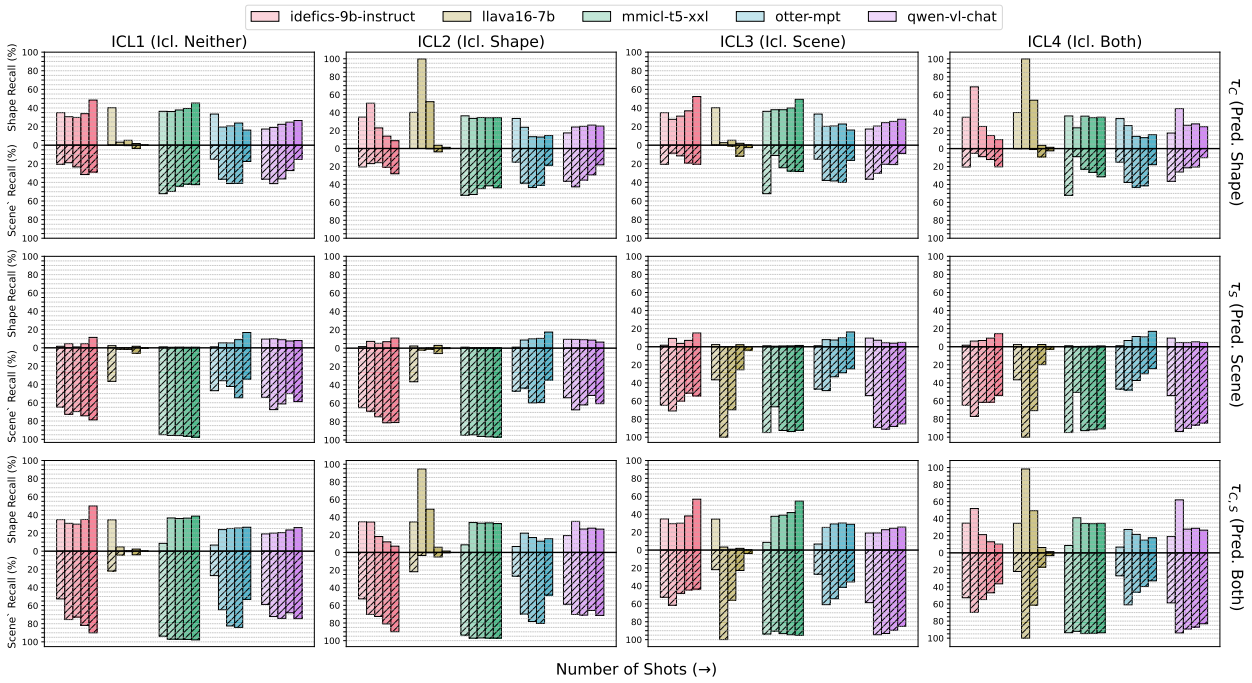


Figure 74: **ICL Results on `IllusionBench-LOGO`**. Few-shot shape and texture accuracy of VLMs on the `IllusionBench-LOGO` dataset across the different ICL learning tasks and the different prediction tasks.

F.4.8 Responses From Low Performing Models

We often observe close to 0% shape accuracy of the `LLaVA` model on shape prediction tasks across all four ICL-constrained ICL prediction tasks when using a higher number of ICL demonstrations. Figure 77 illustrates three example responses from the `LLaVA` model using 4-shot ICL for ICL3, which includes the test query background in the ICL demonstrations. From the example model responses R1, R2, and R3, it is evident that the `LLaVA` model tends to produce descriptive and verbose responses. Specifically, it fails to be concise and accurate, unlike the other models we investigate that usually respond with a single class prediction even with more shots. This verbosity leads to poor accuracy as the model fails to adhere to the prompt instructions of predicting a single class, resulting in the test class rarely being included in the model's responses.

However, Figure 78 shows example responses from the `LLaVA` model on the same task and for the same test queries as in Figure 77 but using 2-shots. Observations from responses R1', R2', and R3' indicate that with fewer shots, the model is much more likely to produce

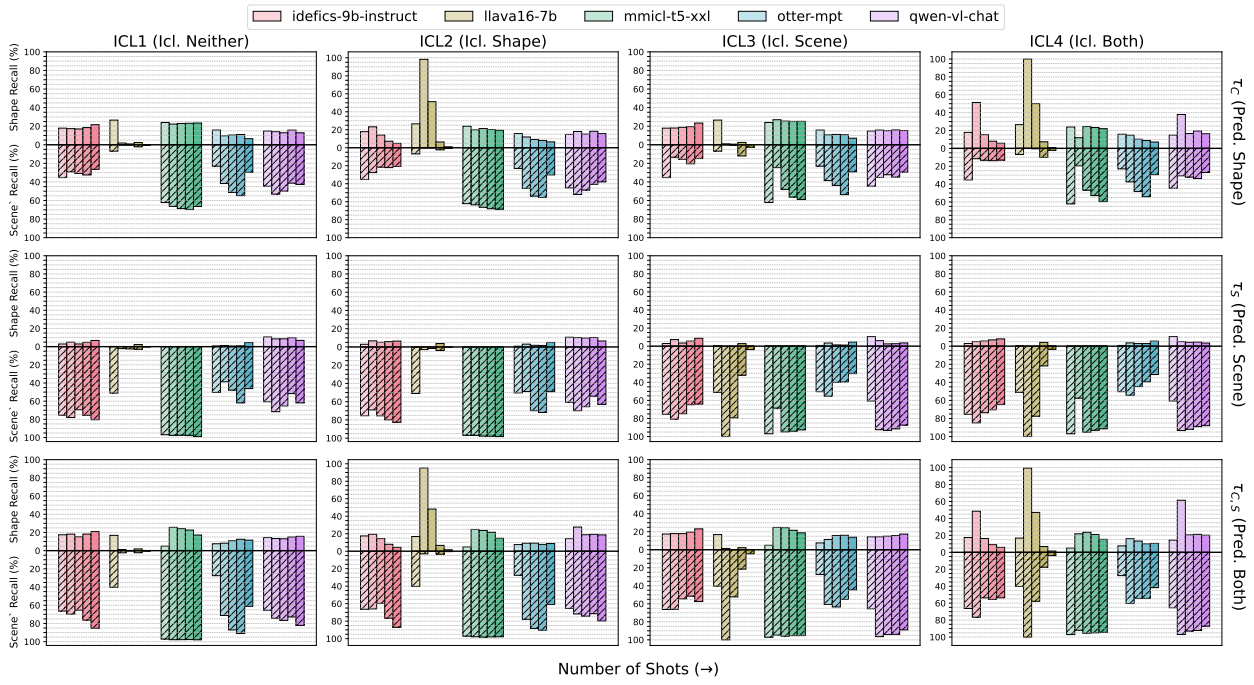


Figure 75: **ICL Results on `IllusionBench-IN`**. Few-shot shape and texture accuracy of VLMs on the `IllusionBench-IN` dataset across the different ICL learning tasks and the different prediction tasks.

single-class predictions or responses that are generally more concise and less descriptive. The differences observed with increasing numbers of shots suggest that `LLaVA`'s ability to correctly process and learn both the expected answer format and the task diminishes with a greater number of shots, highlighting its limitation as an in-context learner.

F.4.9 ICL Prompt and Context Sensitivity

Prompt Sensitivity We assess whether our ICL results are sensitive to the prompts used. We conduct ablations over four different prompt templates: the original template provided in Section F.4.4 and three additional variations. These variations include: (i) a simplified minimalistic prompt, (ii) the same simplified prompt but reversing the order in which the object and background options are presented, and (iii) a Llama-guard-style prompt (Inan et al., 2023) that explicitly indicates what the model should and should not focus on when making predictions. The specific prompt templates are as follows:

```
# Simplified prompt
TASK_INSTRUCTION = `This image contains an object integrated into a
background, where elements of the background contribute to forming the
```

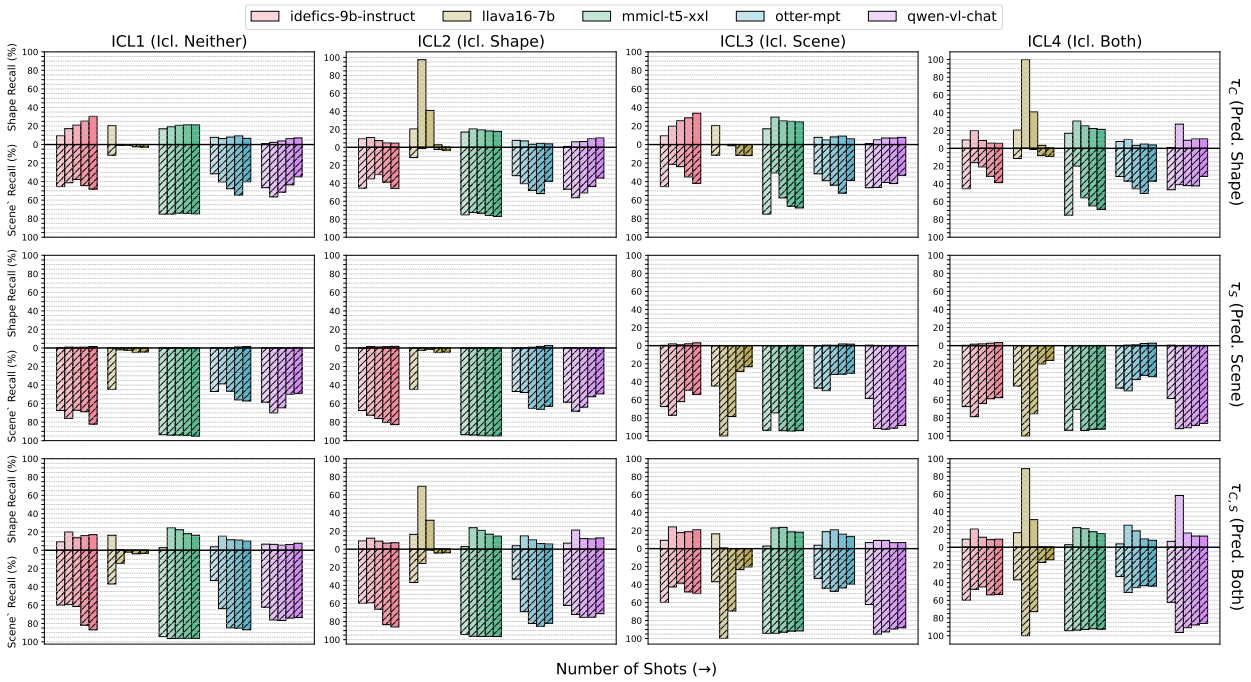


Figure 76: **ICL Results on `IllusionBench-ICON`**. Few-shot shape and texture accuracy of VLMs on the `IllusionBench-ICON` dataset across the different ICL learning tasks and the different prediction tasks.

- **R1**: `The image shows a paper sculpture that resembles a stylized`
- **R2**: `The image shows a logo integrated into a background that features a mountainous landscape`
- **R3**: `The image shows a beautiful natural scene with a large rock formation in the ocean`

Figure 77: **`LLaVA` verbose responses.** Example responses from the `LLaVA` model for 4-shot shape prediction (T1) on the ICL3 learning task.

- **R1'**: `The logo in the image is Tesla.`
- **R2'**: `The logo in the image is Starbucks.`
- **R3'**: `Audi`

Figure 78: **`LLaVA` concise responses.** Example responses from the `LLaVA` model for 2-shot shape prediction (T1) on the ICL3 learning task for the same test query as in Figure 77.

```
image.
background options: [{BG_OPTIONS}]
{object} options: [{OBJ_OPTIONS}]
Identify the object/background/object and background that are
represented in the image by choosing among the provided options.'

# Simplified prompt reverse
TASK_INSTRUCTION = `This image contains a background with an
integrated {object}, where elements of the background contribute to
forming the image.
{object} options: [{OBJ_OPTIONS}]
background options: [{BG_OPTIONS}]
Identify the {object/background/object and background} that are
represented in the image by choosing among the provided options.'

# Llama-guard style
TASK_INSTRUCTION = `This image contains an {object} integrated into a
background, where elements of the background contribute to forming the
image.
{object} options: [{OBJ_OPTIONS}]
background options: [{BG_OPTIONS}]
Identify the {object/background/object and background} that is
represented in the image by choosing among the provided options.
Provide your response by stating only the single, most accurate option
that represents the {object/background/object and background} in the
image. You have to respond with a single word.
### Pay attention to:
ONLY {the object/the background/BOTH the object and the background}
that is represented in the image by choosing among the provided icon
options.
### DO NOT:
Focus on the {object/the background/IGNORE IN THIS CASE} of the image.'
```

We report the mean shape and scene recall on the `IllusionBench-LOGO` dataset split, with error bars representing one standard error from the mean. The results are shown in Figure 79. Overall, we observe very little variation in shape and scene recall across models,

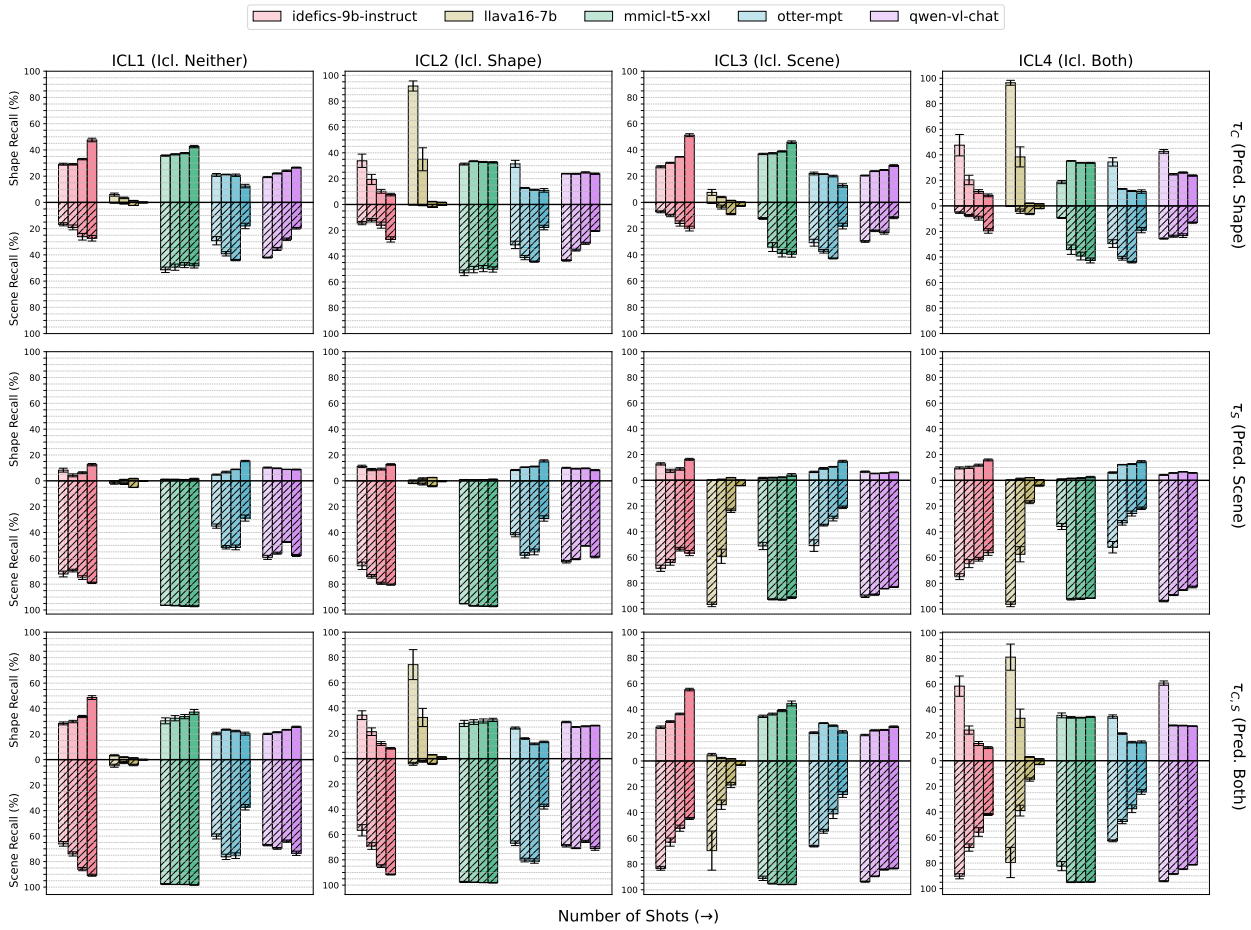


Figure 79: **Prompt sensitivity on `IllusionBench-LOGO`.** Mean shape and scene recall metrics with error bars representing one standard error from the mean across four different prompts used for ICL on the `IllusionBench-LOGO` dataset split.

tasks, and contexts. Significant variations, when present, occur only for LLava or Idefics models and are limited to cases with a small number of shots. These variations diminish as the number of in-context examples increases, suggesting that the results described in Section 7.5 are generally insensitive to the type of prompt used, particularly when a larger number of in-context examples are provided.

Sensitivity to the Order of In-Context Examples We also investigate the sensitivity of ICL results to the order of in-context examples. To assess this, we shuffle the context examples three times on the `IllusionBench-LOGO` when performing inference on task $\tau_{C,S}$. The results, shown in Figure 80, display very tight metrics with minimal variation in shape and scene recall, demonstrating that the results described in Section 7.5 are not sensitive to the ordering of the context examples.

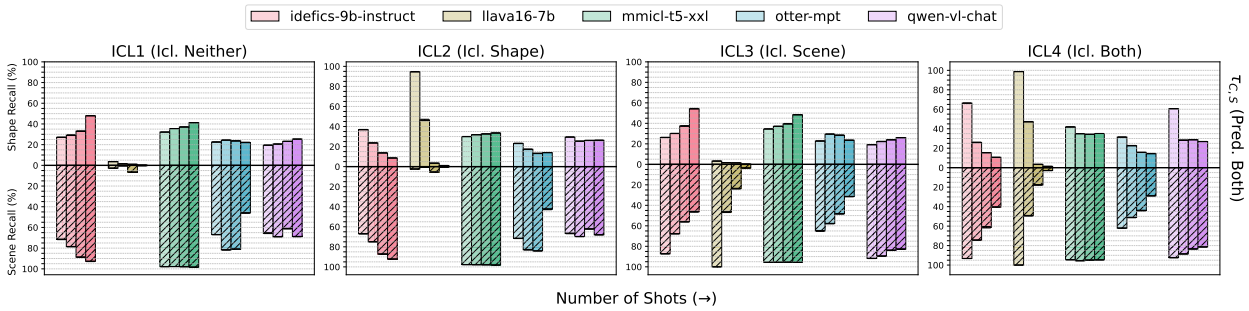


Figure 80: **Context ordering sensitivity on `IllusionBench-LOGO`.** Mean shape and scene recall metrics with error bars representing one standard error from the mean over three shuffled orders of the same context examples used for ICL on the `IllusionBench-LOGO` dataset split.

F.5 Domain Generalization Experiments Details

F.5.1 Background Details

Domain generalization has been a challenging task for image recognition. Several methods have been developed to improve training strategies for better generalisability of early specialist visual models, which are also applicable to CLIP models. Data augmentation strategies such as MixUp (Yan et al., 2020) and RegMixUp (Pinto et al., 2022b) are known to improve generalization capacity through interpolation or extrapolation of data samples outside the training domain for diversity. GroupDRO (Sagawa et al., 2019) performs ERM with a re-weighting of classes with larger errors, making them more significant. VREx (Krueger et al., 2021) reduces differences in risk across training domains, which can decrease a model's sensitivity. Additionally, prompt learning, a promising approach for CLIP-style models, can also be leveraged for domain generalization. Specifically, we adopt DPLCLIP (Zhang et al., 2023c), which trains a prompt generator during the training phase and infers unseen domains.

F.5.2 Further Experiment Details

CLIP Model For all experiments, the image encoder backbone of CLIP model is a ResNet50 (He et al., 2016). For full-parameter fine-tuning, we train the whole image encoder, whereas for linear probing we only train the projection layer. The inferent prompt template for all methods is ``` ``A photo of [Class name]'' ```.

Training Hyperparameters For all experiments, we use a batch size of 32 and the Adam optimizer (Kingma and Ba, 2014) with a learning rate of 5e-5. For full parameter fine-tuning,

we train the model for 1000 steps, and for linear probing, we train the model for 800 steps. For MixUp (Yan et al., 2020) and RegMixUp (Pinto et al., 2022b), the alpha and beta are both set to 0.2. For GroupDRO (Sagawa et al., 2019), the eta is set to 1e-2. For VREx (Krueger et al., 2021), the penalty weight is set to 1.0. For DPLCLIP (Zhang et al., 2023c), the number of domain tokens is 16.

F.6 Compute Resources

All experiments are performed on our internal cluster.

Resources for image generation For the Image generation, we used three A40 GPUs with 45 GB RAM with around 65h to generate all of the images in the dataset.

Resources for zero-shot experiments For the zero-shot experiments, we used eight A40 GPUs with 45 GB RAM for around 250h total to cover all Zero-shot experiments experiments.

Resources for in-context learning experiments We perform ICL inference using 8 A40 GPUs with 45GB RAM for around 168h total to cover all ICL experimental settings.

Resources for domain generalization experiments For each fine-tuning CLIP we use a single A40 GPUs with 45GB RAM for an hour on average for full parameter fine-tuning and half an hour for linear probing.

Appendix G: How Do LLMs Represent and Process Symbols In-Context?

G.1 Dataset Details

Below, we include an example from each split of the 1-shot English task in TGT (Smolensky et al., 2025), where the correct completion for each task instance is underlined.

Example from **dev** split (1-4 constituents, each of length 1-4):

```
Q | [ that quiet sad toad ` each mouse ! ] cub != ^ bear
A each mouse > ; cub .
Q | [ his salty narrow cat ` another mild short zebra ! ] any tigress
!= ^ the shy brown house
A another mild short zebra > ; any tigress .
```

Example from **ood_cons_len** split (1-4 constituents, each of **length 5**):

```
Q & ! another warm soft green car ; # one narrow brown shy tiger _ -
his sweet rich fair dog ( every good heavy blunt lion
A his sweet rich fair dog .
Q & ! their sour sharp fast lioness ; # her salty small hot fish _ -
our poor dry cool toad ( this long sad dark giraffe
A our poor dry cool toad .
```

Example from **ood_cons_count** split (**5 constituents**, each of length 1-4):

```
Q + another fresh loud bush * our bird % bottle ~ != that red green
cow ? # monkey
A that red green cow ^ ) bottle { another fresh loud bush .
Q + any slow hot screen * their fair bad dog % snake ~ != the dry
quiet tiger ? # your zebra
A the dry quiet tiger ^ ) snake { any slow hot screen .
```

Example from **ood_lexical** split (4 constituents, each of length 1-4, where at train-time each constituent appeared only in "echo" prompts):

```
Q ] < SCREEN % SOME SOUR SLOW BOTTLE , ONE LION - THAT SNAKE
A < != ONE LION .
Q ] < THE BITTER SAD TIGER % TREE , WHICH ELEPHANT - MY BLUNT RICH CAT
A < != WHICH ELEPHANT .
```

For reference, example of an "echo" prompt from the **train** split (where all echo prompts have only 1 constituent, always of length 1):

```
Q LIZARD
A LIZARD .
Q LIGHT
A LIGHT .
```

G.2 Model Details

For each model trained on TGT, the hyperparameters that vary by model are specified in Section 8.2.2; whereas the following hyperparameters are shared by all models:

- Loss function: cross-entropy loss (with teacher-forcing)
- Learning rate: $1e-4$
- Batch size: 128
- Train steps: 100K
- Dropout: 0.01

G.3 Experiment Details

LDA Experiments and Metrics Linear Discriminant Analysis (LDA) learns a subspace U_T that maximizes the ratio $S = \frac{\sigma_b^2}{\sigma_w^2}$, where σ_b^2 denotes the **between-group variance** and σ_w^2 denotes the **within-group variance** between some set of classes T – here, the constituent types $\tau(c) \in T$ for constituents $c = (i_1, ..., i_k)$ for $k \in [1, 4]$ (in all splits other than `ood_cons_count`, for which $k = 5$). That is, given the set of all (test-set) embeddings in the same class $T = \tau$, denoted H_τ, when projected onto the subspace U_T via orthogonal projector P_{U_T}, each variance measure is computed as follows:

$$\sigma_w^2 = \frac{1}{|H_\tau|} \sum_{h \in H_\tau} \|P_{U_T}(h) - \mu_\tau\|^2 \quad \text{where} \quad \mu_\tau = \frac{1}{|H_\tau|} \sum_{h \in H_\tau} P_{U_T}(h)$$

$$\sigma_b^2 = \frac{1}{|T|} \sum_{\tau \in T} \|\mu_\tau - \mu\|^2 \quad \text{where} \quad \mu = \frac{1}{\sum_{\tau \in T} |H_\tau|} \sum_{\tau \in T} \sum_{h \in H_\tau} P_{U_T}(h)$$

If the hypothesis stated in Equation (11) holds, then σ_w^2 should be close to zero, because the columns belonging to the same constituent type should be *projected to the same point* in the learned subspace, showing that there is minimal constituent-level variation in this subspace. Likewise, under the same hypothesis, σ_b^2 should be large (relative to σ_b^2), and the **partition accuracy** of U_T (i.e., as obtained by clustering each test-set embedding in the subspace U_T according to the nearest centroid μ_τ) should be close to 100%. Finally, we include an additional subspace quality metric, **average centroid cosine**, as a measure of the orthogonality between class centroids μ_τ, which is defined as:

$$\text{avg cos} = \frac{1}{|T| \cdot (|T| - 1)} \sum_{\tau_i \in T} \sum_{\tau_j \in T, \tau_j \neq \tau_i} \cos(\mu_{\tau_i}, \mu_{\tau_j})$$

G.4 Supplemental Results

G.4.1 Activation Patching

Each of the below figures follows the same format as specified in the caption of Figure 33 – the only difference is the model and split.

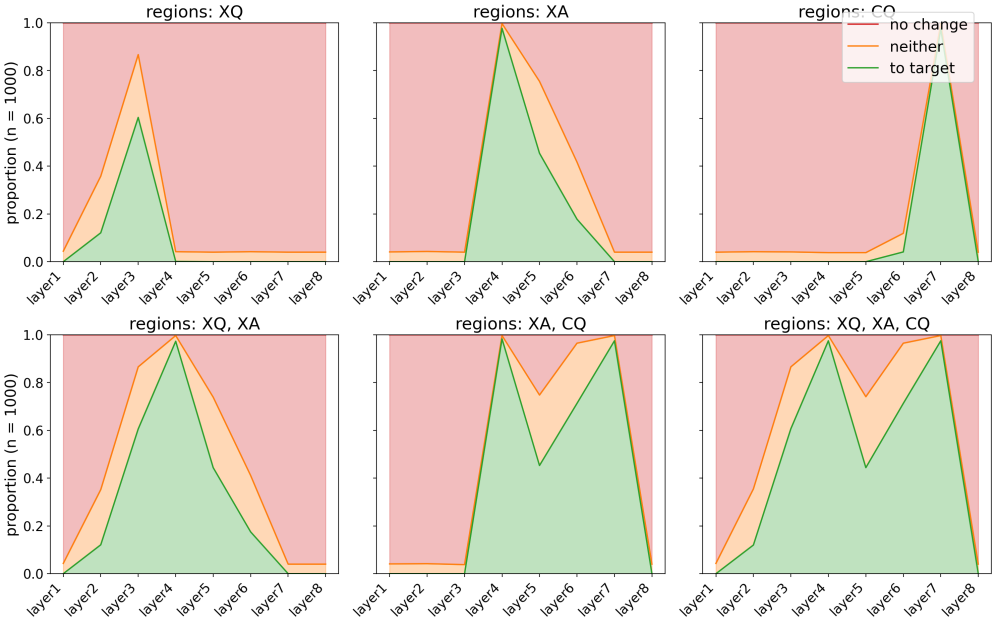


Figure 81: **Patching Llama medium on the `ood_cons_len` split.**

Figure 82: **Patching Llama medium on the `ood_lexical` split.**

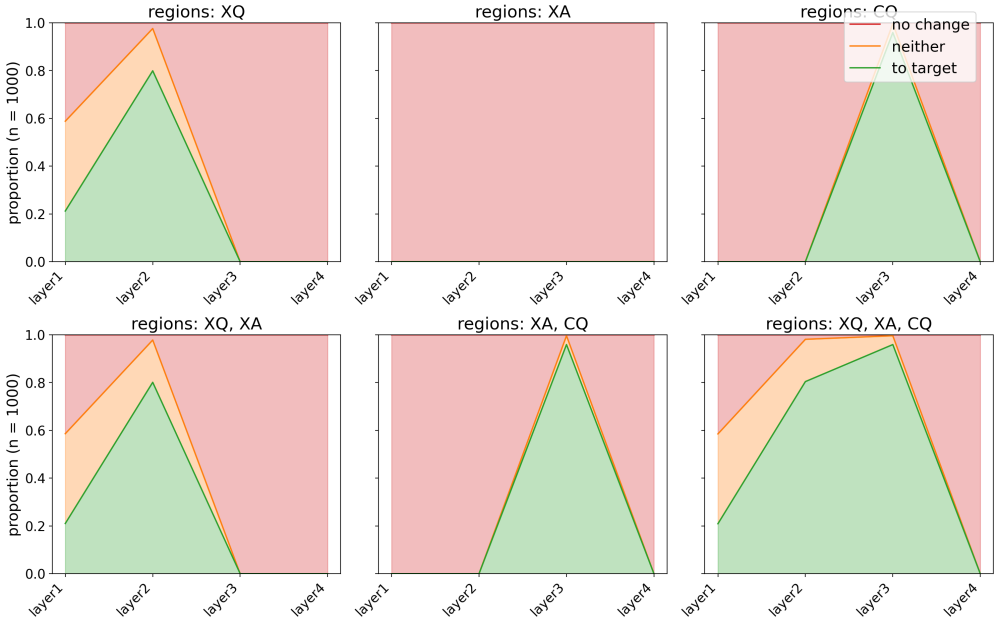


Figure 83: **Patching Llama small on the dev split.**

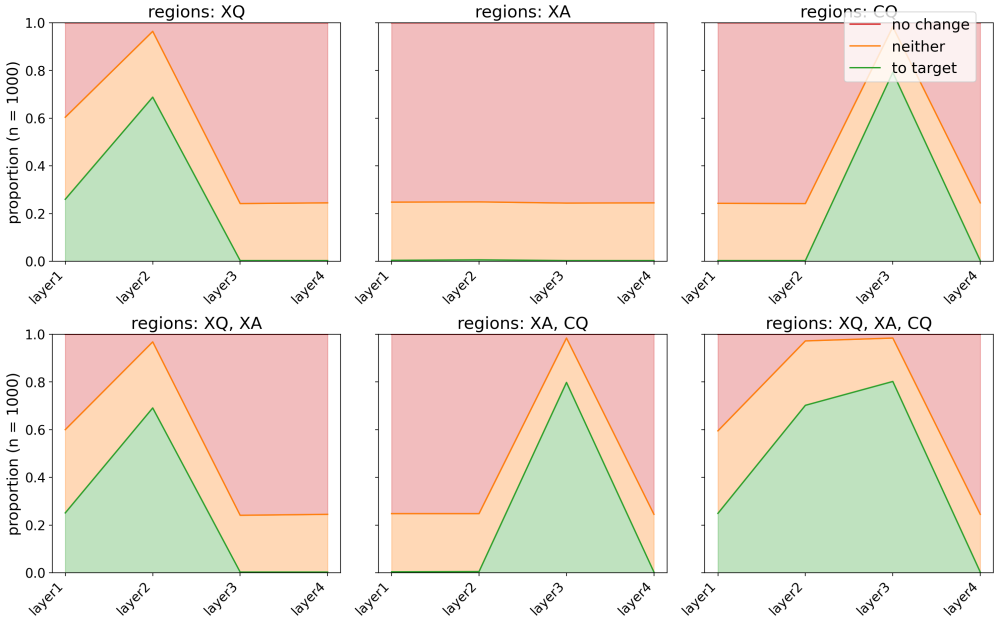


Figure 84: **Patching Llama small on the `ood_cons_len` split.**

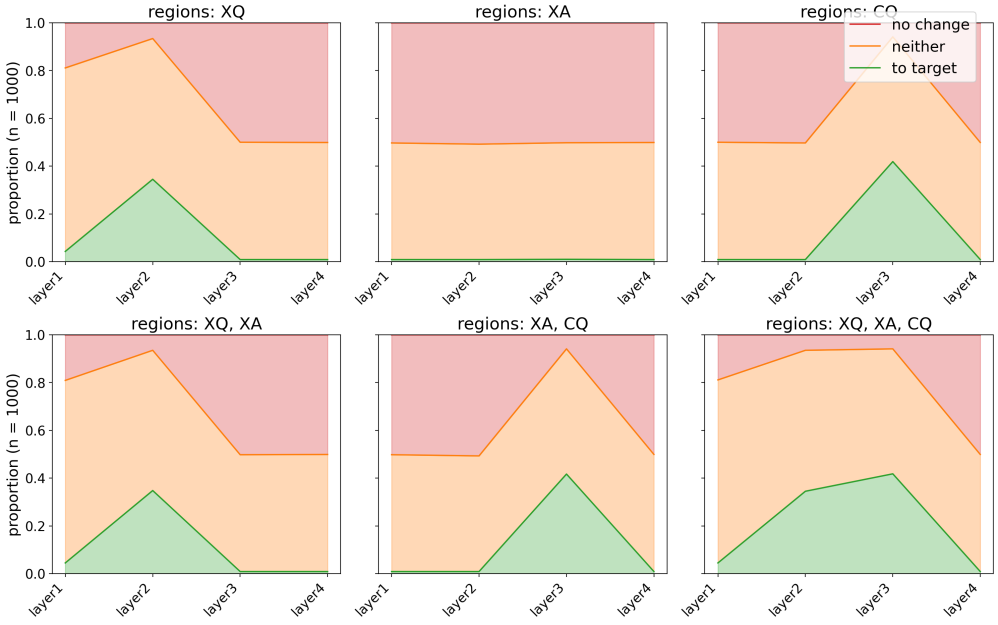


Figure 85: **Patching Llama small on the `ood_cons_count` split.**

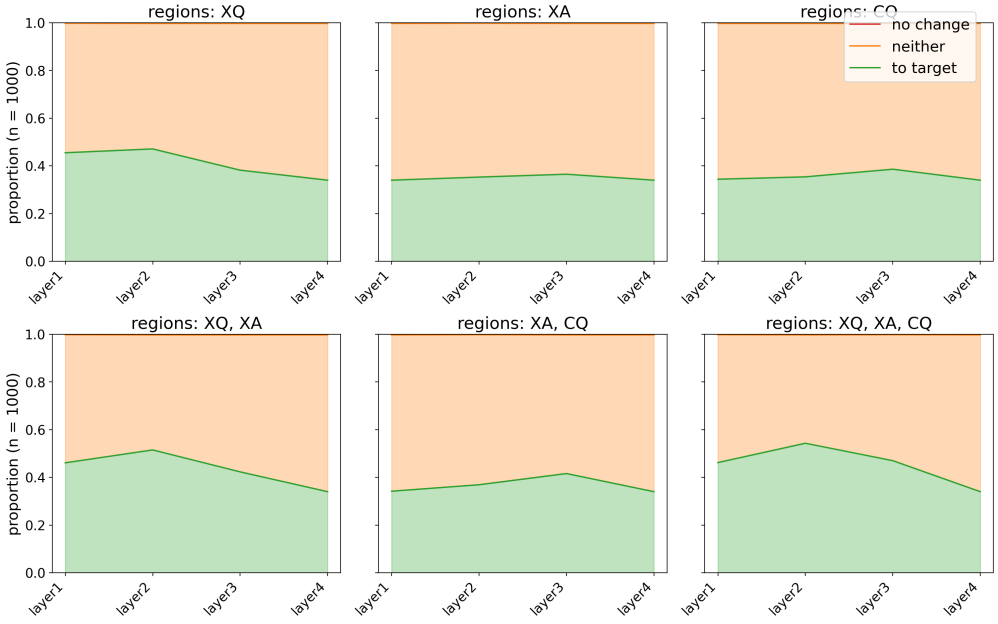


Figure 86: **Patching Llama small on the `ood_lexical` split.**

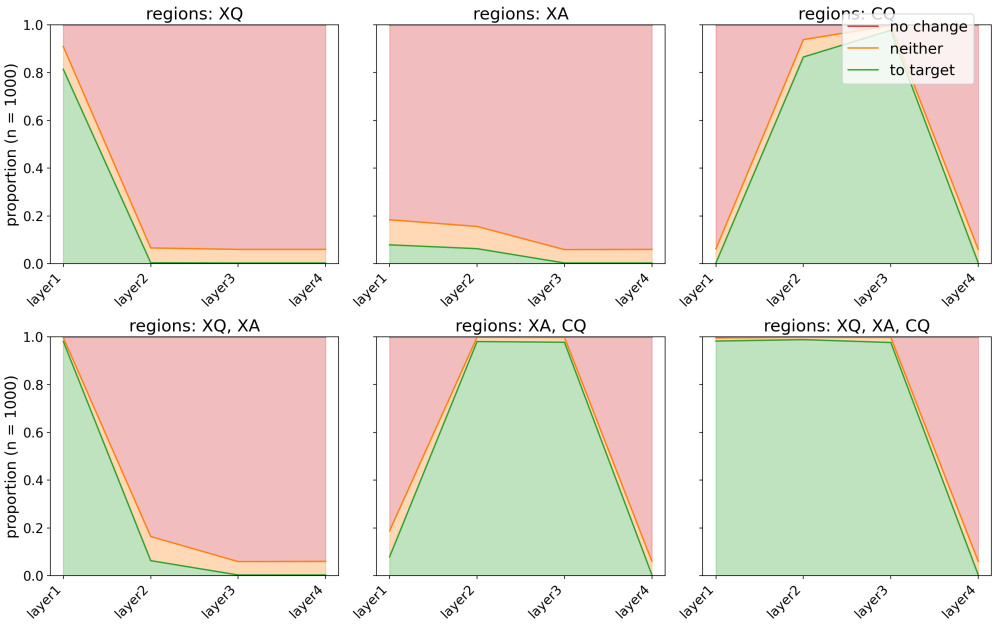


Figure 87: **Patching NanoGPT small on the dev split.**

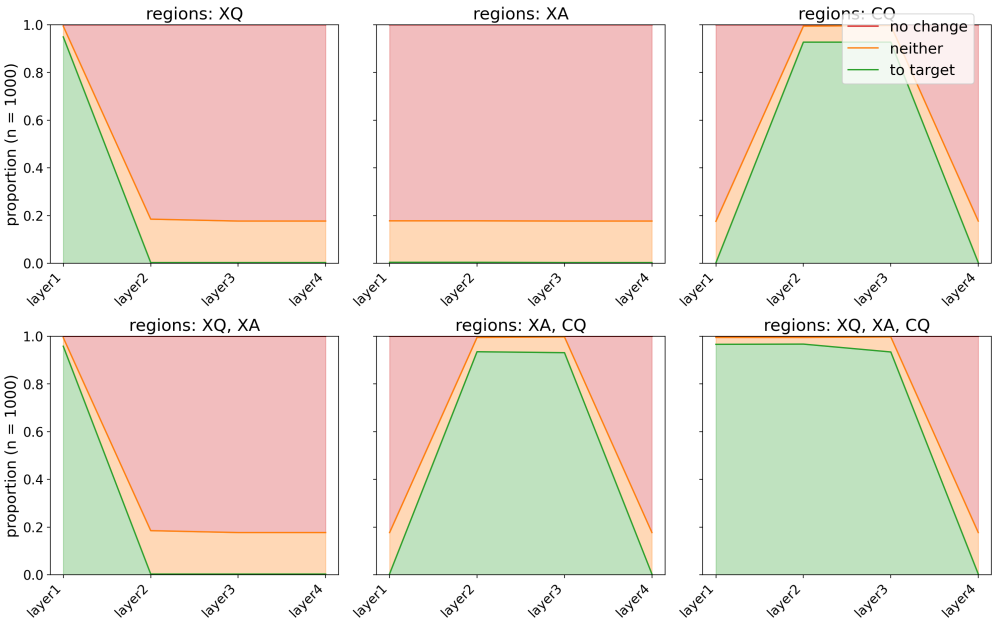


Figure 88: **Patching NanoGPT small on the `ood_cons_len` split.**

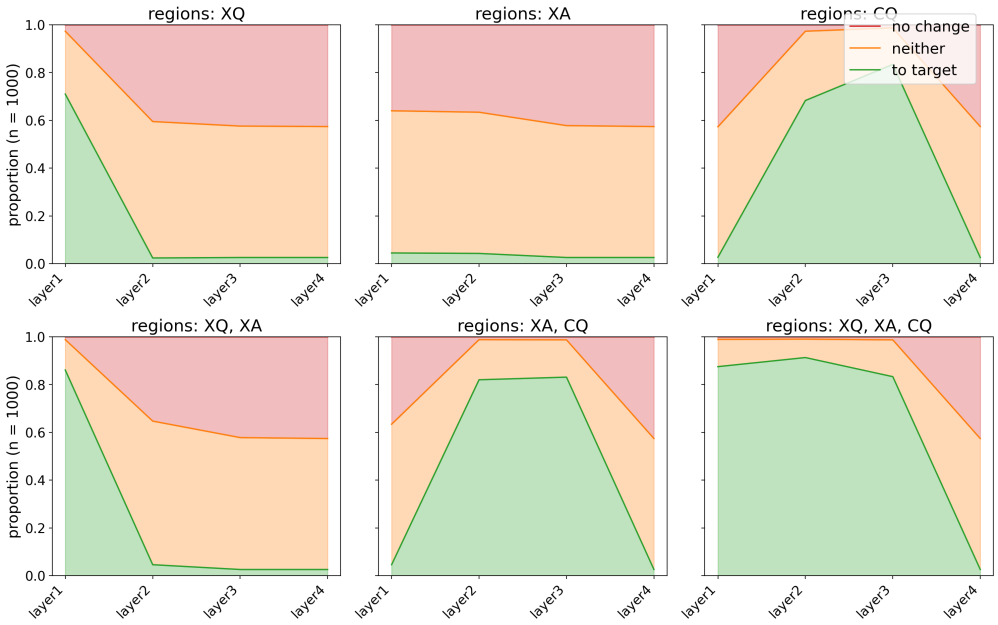


Figure 89: **Patching NanoGPT small on the `ood_cons_count` split.**

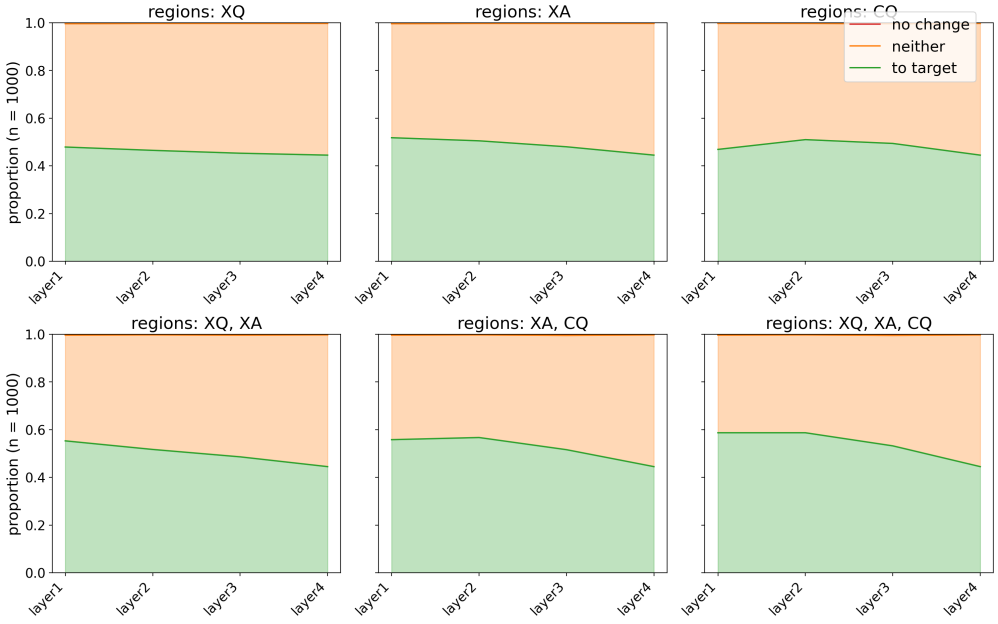


Figure 90: **Patching NanoGPT small on the `ood_lexical` split.**

G.4.2 LDA Constituent Subspaces

The figures of subspace quality metrics and UMAP visualizations, below, each follow the same format as specified in the caption of Figure 35 and Figure 36, respectively – the only difference is the choice of model and layer. For each model, we visualize the results for the layer which, per the algorithmic patching results reported in Appendix G.4.1, play the greatest role in parsing (i.e., the layer where patching in the "XQ" region has the greatest impact; see Section 8.3.3).

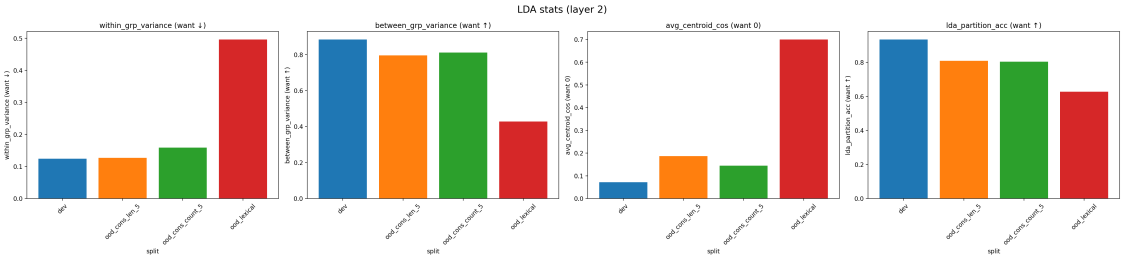


Figure 91: **Quality metrics regarding LDA field-index subspace, by split, for Llama small in layer 2.**

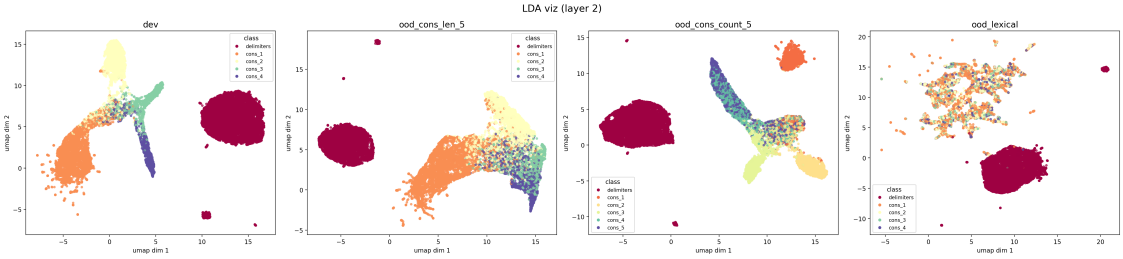


Figure 92: **UMAP projection of LDA field-index subspace, by split, for Llama small in layer 2.**

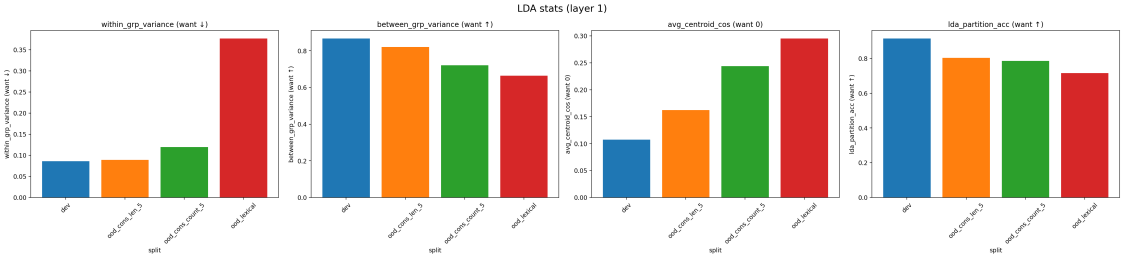


Figure 93: **Quality metrics regarding LDA field-index subspace, by split, for NanoGPT small in layer 1.**

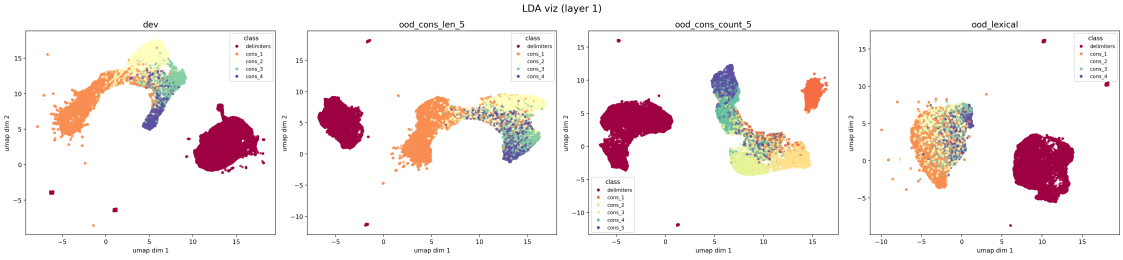


Figure 94: **UMAP projection of LDA field-index subspace, by split, for NanoGPT small in layer 1.**

Below, we report the average L1 error of a simple linear regression analysis when using each metric alone (as measured in each layer) to predict models' OOD generalization to each split.

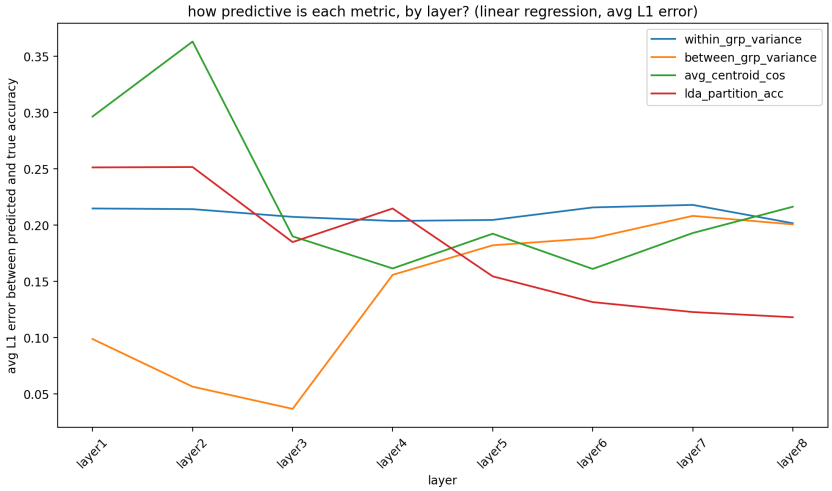


Figure 95: **Layerwise predictiveness of of each metric, by layer, for Llama medium.**

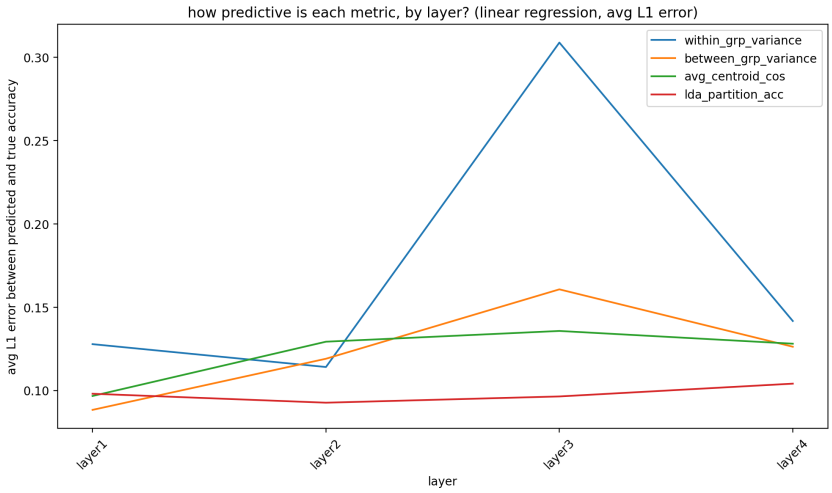


Figure 96: **Layerwise predictiveness of of each metric, by layer, for Llama small.**

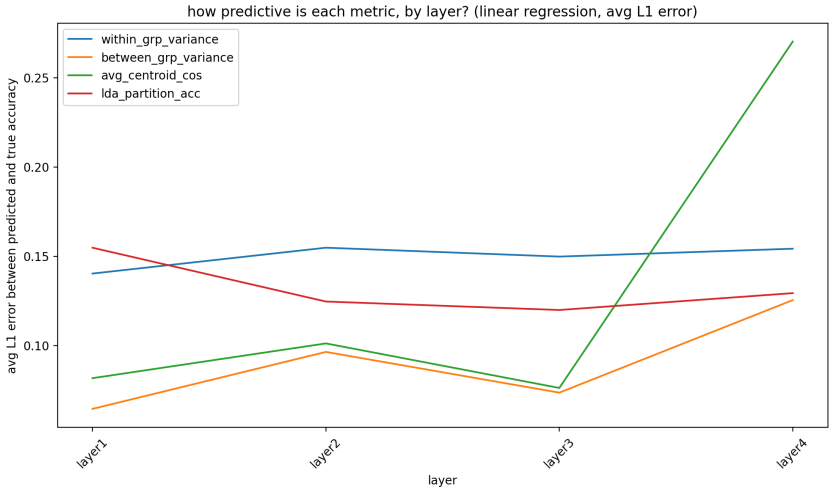


Figure 97: **Layerwise predictiveness of of each metric, by layer, for NanoGPT small.**