SAPIENZA
UNIVERSITÀ DI ROMA

Shortcut detection and mitigation via Representation Engineering

Faculty of Information Engineering, Informatics, and Statistics
Master's Degree in Computer Science [LM-18]

Arianna Paolini
ID number 1943164

Advisor
Prof. Indro Spinelli

Academic Year 2024/2025

Shortcut detection and mitigation via Representation Engineering
Master Degree thesis. Sapienza University of Rome

This thesis has been typeset by LaTeX and the Sapthesis class.

Version: December 29, 2025

Author's email: paolini.1943164@studenti.uniroma1.it

To my family,
Andrea, Luisa, Gaia and Rita

Abstract

While Large Language Models (LLMs) are extremely powerful and versatile tools, their robustness and interpretability are often undermined by shortcut learning, a phenomenon in which models rely on spurious correlations between inputs and outputs rather than on genuine semantic or causal reasoning. Such behavior can lead to brittle generalization, misleading predictions and the amplification of undesirable biases.

This thesis investigates the detection and mitigation of shortcut learning in LLMs under an In-Context Learning (ICL) setting. The ICL paradigm introduces additional opportunities for shortcut exploitation, as models may leverage superficial cues in demonstration examples such as lexical overlap, positional patterns or stylistic regularities, instead of internalizing the intended task structure. To address this challenge, this work adapts the recently proposed Representation Engineering (RepE) framework to shortcut learning. The proposed approach operates directly on the internal representations of a frozen LLM, extracting latent directions associated with shortcut reliance through contrastive analysis of clean and shortcut-augmented inputs, and manipulating these directions at inference time to suppress shortcut-driven behavior without updating model parameters.

An extensive empirical evaluation is conducted using the Mistral 7B Instruct model under ICL across multiple NLP benchmarks, including textual entailment, sentiment analysis, commonsense reasoning, word sense disambiguation and multi-domain question answering. The results show that shortcut-aligned directions can be reliably identified in the model's representation space and that their targeted manipulation leads to consistent performance improvements across tasks, even when the shortcut direction is extracted from a single task domain.

Overall, this work demonstrates that shortcut learning in LLMs can be addressed through lightweight, interpretable and training-free interventions at the representation level, providing new insights into the internal mechanisms underlying shortcut reliance and highlighting the potential of representation-based control for improving model robustness and generalization.

All the code for this work is available at https://github.com/arianna011/shortcut-llm-icl.

Contents

Introduction

Both humans and artificial intelligence systems often display a natural tendency to take the easiest path when solving a problem, relying on *shortcuts* that may be effective in specific task instances but fail to generalize across different examples. For instance, a human student might answer a multiple-choice question by recognizing familiar keywords rather than understanding the underlying concept, or judge the validity of an argument by how confidently it is stated rather than by its logical structure. In a similar way, a medical image classifier may associate the presence of an hospital watermark with a cancer diagnosis, instead of learning the actual visual features of the tumor. Language models, too, often exploit unintended cues: a text classifier might infer that a movie review is positive simply because it contains exclamation marks, or that a news article is political merely because it mentions a politician's name.

Studies have shown that Large Language Models (LLMs), despite their vast learning capacity, can perform poorly on out-of-distribution inputs [Hendrycks et al., 2020] and remain vulnerable to adversarial attacks [Wallace et al., 2021]. This lack of robustness is often traced back to *shortcut learning* [Geirhos et al., 2020], the tendency of models to exploit spurious correlations in the training data rather than learning the intended reasoning process. In Natural Language Processing (NLP), shortcut learning typically manifests as reliance on superficial cues (such as lexical overlap, word order or stylistic patterns) that correlate with the correct label in standard benchmarks but lead to failures in more realistic settings. Under the In-Context Learning (ICL) paradigm, this issue becomes even more pronounced: LLMs may attend to shallow patterns in the instruction or examples within the prompt context rather than abstracting the underlying task structure, leading to brittle or biased behavior.

Beyond technical robustness, shortcut learning also has important ethical implications. Because training corpora encode social stereotypes and historical imbalances, shortcuts often align with biased associations, like linking occupations to specific genders or ethnic groups [Bolukbasi et al., 2016, Seshadri et al., 2025]. As a result, shortcut-driven behavior can propagate or even amplify unfair outcomes in downstream applications, undermining the accountability and trustworthiness of AI systems in real-world deployments [Bender et al., 2021].

The detection and mitigation of shortcut learning is therefore crucial not only for achieving more robust and generalizable models, but also for ensuring that LLMs

behave in ways that are ethically responsible and socially fair. This dual challenge motivates the present work, which investigates Representation Engineering (RepE) [Zou et al., 2025] as a novel framework for identifying and mitigating shortcut mechanisms directly within the internal representations of LLMs.

In particular, this study aims to address the following research questions:

- Is the concept of "taking a shortcut" explicitely encoded by LLMs in their hidden representations? That is, do models internally know more than they reveal through their generations, similarly to how they encode the truthfulness of statements in cases of hallucination [Azaria and Mitchell, 2023]?
- Can a *shortcut reliance* linear direction be identified in the latent representation space using methods such as Principal Component Analysis (PCA)?
- Can steering model's predictions at inference time, via targeted amplification or suppression of such direction, lead to more robust and reliable outputs?

Answering these questions would clarify whether shortcut learning can be mitigated through a lightweight and interpretable approach such as RepE, which enables the analysis and control of a model's internal representations without requiring any parameters update.

Several experiments are conducted to evaluate the effectiveness of the RepE framework when adapted to shortcut detection and mitigation, and to analyze how steering interventions affect model representations, with the goal of assessing both the correctness and the transparency of the proposed method. Given the complexity and novelty of this objective, the chosen case study focuses on a simple yet informative setting: LLMs performing In-Context Learning on Natural Language Processing tasks such as textual entailment, sentiment classification and causal reasoning.

The content of the thesis is organized as follows. Chapter 1 introduces the theoretical background on in-context learning and shortcut learning, reviewing existing strategies and related work. Chapter 2 presents the proposed method for shortcut detection and mitigation, describing how the Representation Engineering framework is adapted to the shortcut learning setting. Finally, Chapter 3 provides an empirical evaluation of the proposed technique and discusses the obtained results.

Chapter 1

Background & Related Work

1.1 In-Context Learning

Large Language Models (LLMs) are the most successful class of AI systems currently employed in Natural Language Processing (NLP). They serve as modern assistants across several applications, ranging from open-domain conversation and question answering to machine translation and code generation. Their rapid progress has been largely driven by scaling: architectures with increasing depth and width, trained on massive and diverse web-scale corpora, have not yet shown signs of performance saturation. In fact, they follow empirical scaling laws that predict consistent improvements with greater model size, data and compute [Kaplan et al., 2020]. The main limitation to further progress is therefore not intrinsic model capability, but rather the economic and environmental costs of training ever-larger architectures on ever-broader datasets.

An interesting consequence of such large-scale pretraining was the emergence of the In-Context Learning (ICL) paradigm [Brown et al., 2020]. LLMs demonstrated the surprising ability to perform novel tasks after being provided only a few input-output demonstrations, presumably inferring the underlying semantic pattern in a similar way to how humans would solve new problems after looking at some examples. LLMs such as GPT-3 [Brown et al., 2020], Llama and OPT [Milios et al., 2023] are able to achieve state-of-the-art performance on several NLP benchmarks when evaluated in ICL settings, even competing with finetuned task-specific models.

1.1.1 Definition

An LLM models a probability distribution $p_\theta(w_t|w_{<t})$ over the text tokens w_i in its vocabulary V in order to build coherent natural language sequences $(w_1 w_2 ... w_t ...)$. In transformer-based architectures [Vaswani et al., 2023], the parameters θ learned for attention and feed-forward units ultimately determine the likelihood of each token being the next word in the sequence, conditioned on all preceding ones (*autoregression*). A decoding strategy, either deterministic (e.g. greedy choice) or stochastic (e.g. temperature-based sampling), is then applied to effectively select a new token and continue the text generation process.

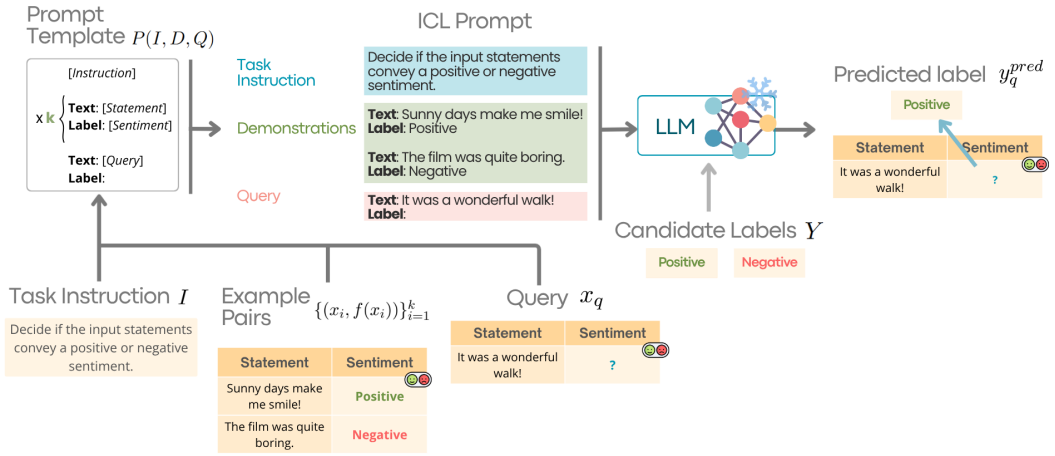


Figure 1.1. Representation of the In-Context Learning (ICL) paradigm for LLMs

In an ICL setting (Figure 1.1), given an NLP task defining an input-label relationship function $f : V^* \rightarrow Y$ (where V^* denotes the space of all possible text sequences and $Y = \{y_1, ..., y_m\}$ is the set of valid answers or labels required by the task), an arbitrary number k of example pairs $D = \{(x_i, f(x_i))\}_{i=1}^k$ can be selected from a dataset to serve as task demonstrations for an LLM that has not been explicitly trained to perform that task. The prompt that is presented to the LLM is obtained by populating a chosen prompt template $P(I, D, Q)$, which specifies how to format the optional task instruction I (written in natural language), the demonstrations set D and the query $Q = x_q$, which represents the the task instance for which the LLM is required to produce an answer y_q^{pred}.

Without any parameters update, the frozen LLM leverages its vast and generalizable prior knowledge to try to infer the hidden function f from the demonstrations. The predicted answer is computed as:

$$y_q^{pred} = \arg\max_{y \in Y} \; p_\theta(y \mid P(I, D, Q))$$

When the number of demonstrations is $k = 0$, the model operates in a *zero-shot* setting, relying solely on the task instruction. For $k > 0$, the typical *few-shot* setting is established, where contextual examples guide the model's predictions.

1.1.2 Influencing factors

Several studies have analyzed the empirical factors that can influence LLM performance in ICL settings [Dong et al., 2024]. At the pretraining stage, the diversity of source domains within the training corpora has been shown to significantly enhance the ability of LLMs to generalize to unseen tasks in ICL, often even more than the overall corpus size. However, while pretraining on a dataset specifically related to a downstream task seems to help zero-shot learning performance, it does not always guarantee competitive few-shot results on the same task. Also, ICL capabilities can emerge when a model is trained on a combination of multiple corpora, even if each corpus individually does not promote ICL by itself [Shin et al., 2022]. Beyond

data quality and diversity, both the model architecture and the training process can influence ICL performance: a larger number of parameters or training steps tends to correlate with stronger in-context learning abilities [Wei et al., 2022].

At inference time, LLMs proved to be very sensitive to several features of the ICL prompt itself. First, the selection of the demonstration examples predictably influences ICL performance. Demonstrations drawn from out-of-distribution (OOD) data with respect to the pretraining corpus generally yield lower results [Min et al., 2022]. Moreover, better performance is typically achieved when demonstrations have embeddings that are semantically closer to that of the query [Liu et al., 2021]. Factors such as the format used to present demonstrations, the coverage of the label distribution and the order in which examples are arranged can all affect model predictions. For instance, [Lu et al., 2022] observed that certain permutations of examples lead to substantially higher performance than others, but found no consistent pattern among effective orderings, nor any transferability across different model sizes or tasks. Such feature sensitivities, which can be interpreted as a form of inductive bias [Si et al., 2023], reduce the interpretability of ICL mechanisms and will be further discussed in Section 1.2.3.

1.1.3 Theoretical explanations

To date, it is still not entirely clear why and how ICL works. Some argue that, during pretraining, an LLM is exposed to implicit task demonstrations that can later be triggered and reused in an ICL setting, while others suggest that LLMs are capable of learning new input-label mappings directly from the contextual examples. In [Pan et al., 2023], the authors propose to decouple ICL into Task Recognition (TR) and Task Learning (TL). The former refers to recognizing a previously encountered task from demonstrations and applying relevant pretraining priors; thus, it does not even require a correct association between text and label in the examples. The latter, instead, involves genuinely learning a new input-label mapping from the provided demonstrations and therefore relies on accurate ground-truth labels. The study shows that, while TR emerges as a broad capability across model scales, TL is enabled when increasing both the number of model parameters and the number of in-context examples, making larger models substantially more capable.

When adopting a Bayesian framework, ICL can be interpreted as a Bayesian inference process implemented in the forward pass of the LLM transformer:

$$p_\theta(y \mid D, x) = \int p(y \mid x, \phi)\ p_\theta(\phi \mid D)\ d\phi$$

Given the set of contextual demonstrations $D = \{(x_i, f(x_i))\}_{i=1}^{k}$ and the query input x, the model's output distribution is obtained by integrating the conditional likelihoods $p(y \mid x, \phi)$ of possible answers y under a latent task hypothesis ϕ, weighted by their posterior probabilities $p_\theta(\phi \mid D)$. This formulation can be interpreted as a form of Bayesian Model Averaging (BMA) [Zhang et al., 2023]. In this view, the LLM implicitly maintains a distribution over possible task hypotheses ϕ, represented by $p_\theta(\phi \mid D)$, which is inferred from the in-context examples D and encoded within its

hidden representations, rather than explicitly computed. Hence, the LLM acts as a *metalearner*[1] by inferring a posterior over latent task hypotheses and averaging their corresponding predictions to generate an answer.

Other interpretations draw parallels between ICL and gradient-based optimizations, suggesting that LLMs may be encoding smaller models in their activations and updating such implicit models as new examples appear in the ICL context. [von Oswald et al., 2023] demonstrated that a single attention layer can emulate a gradient descent update on a simple regression loss, and that deeper transformers effectively perform multiple such updates across layers, behaving as implicit optimizers. Similarly, [Akyürek et al., 2023] showed that transformers can instantiate classical algorithms, such as gradient descent, ridge regression, or closed-form least squares, by encoding model parameters within hidden representations and updating them dynamically as new examples are processed.

1.1.4 Advantages and limitations

Beyond emulating the human ability to learn from examples, ICL provides an interpretable interface to LLMs, as both instructions and demonstrations are expressed in natural language. It also requires significantly lower computational costs compared to supervised learning, since it does not involve parameter optimizations but only leverages LLMs prior knowledge to generalize to new task patterns. While ICL alone can achieve strong performance, it can be further enhanced through specialized training strategies or pre-inference warm-up to adjust model activations.

These advantages have encouraged the application of ICL not only to traditional NLP tasks, but also to areas such as data engineering, where it enables the generation of annotated datasets with minimal human supervision, and retrieval-based model augmentation and knowledge updating, in which external information is injected into an LLM via the in-context prompt.[Dong et al., 2024].

On the downside, as mentioned in Section 1.1.2, the performance of ICL is highly sensitive to multiple prompt-related factors, such as example order and formatting, which can reduce its effectiveness and demand careful prompt engineering. Moreover, the underlying mechanisms of ICL remain only partially understood and require further investigation to be fully exploited. Such sensitivities and limited interpretability raise the question of whether LLMs truly learn task semantics in-context or merely rely on spurious cues and surface patterns to make their predictions, a behavior known as *shortcut learning*, which finds particularly fertile ground in ICL settings.

[1] a model that "learns to learn" through its internal representations

1.2 Shortcut Learning

Machine Learning (ML) models have shown impressive results across a wide range of domains, from natural language processing to computer vision and multimodal applications. However, when faced with a distributional shift between training datasets and real-world data, these models still often fail to generalize reliably. Instead of learning a robust causal relation between input features and task outputs, they frequently exploit simple spurious correlations embedded in the training set to make their predictions, which can result in unintuitive failures on out-of-distribution examples.

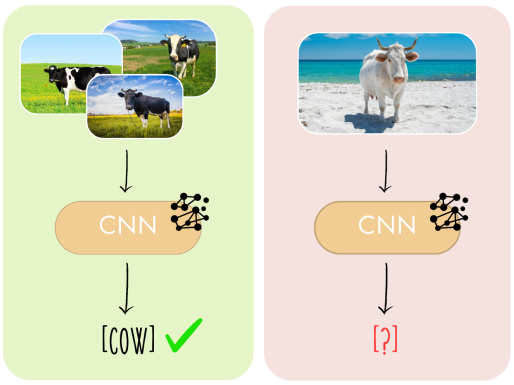


Figure 1.2. Example of a CNN leveraging a dataset bias as a shortcut: since cows only appear on grassy backgrounds in the training data, the model learns to predict "cow" when it recognizes large green areas in the image and is not able to recognize the animal in a different setting

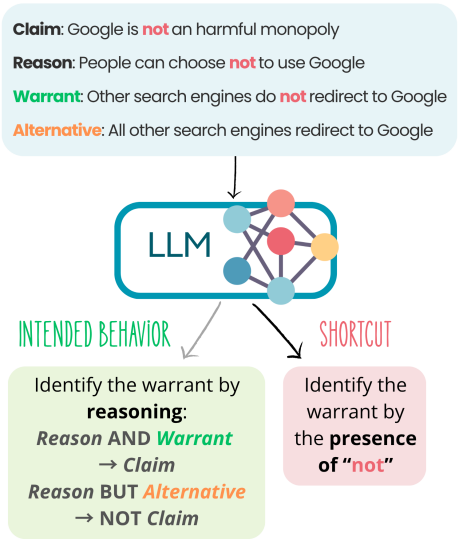


Figure 1.3. Example of an LLM learning a shortcut behavior for the Argument Reasoning Comprehension Task (ARCT)

For instance, Convolutional Neural Networks (CNNs) have shown a tendency to classify objects based on superificial cues in the image (e.g. the background the object mostly appears on in training data) rather than on their actual characterizing features (Figure 1.2) [Beery et al., 2018]. A critical real-world example was represented by a classifier trained to detect pneumonia from images of chest X-ray scans. The model achieved high accuracy by exploiting hospital-specific watermarks, effectively ignoring the actual pathological signals in the images [Zech et al., 2018]. In general, ML models may leverage any feature observed during training to make predictions, leading to issues when *dataset bias*[2] is present: object textures or even imperceptible high-frequency patterns could become unintended label predictors.

[2] *Dataset bias* occurs when the training data fails to accurately represent the real-world distribution in which the model will operate (for instance, when a classifier is trained only on images of cows standing on grass)

Such behavior is not confined to computer vision. Large Language Models (LLMs) have also been shown to rely on spurious cues (e.g. occurrence of high-frequency words such as "not") to perform tasks, rather than genuinely understanding semantic relationships. In the Argument Reasoning Comprehension Task (ARCT), for instance, BERT selected the correct answer primarily by exploiting lexical shortcuts (Figure 1.3) [Niven and Kao, 2019]. Similarly, in agent-based reinforcement learning, models can reach high scores without learning the intended behavior, by "hacking" the reward function. A known example is the algorithm that learned to pause Tetris indefinitely to avoid losing, without actually learning how to play the game [Murphy, 2013].

Such phenomena illustrate how models may confuse correlation (the circumstantial occurrence of some features within specific data categories) with causation (the underlying real-world attributes that truly define entities), achieving high apparent performance while lacking genuine understanding. They represent a major vulnerability of ML models, as they undermine both robustness and interpretability. Moreover, they raise significant societal concerns: decision-making system, such as Amazon's hiring tool, can base their predictions on sensitive attributes like race or gender, thus perpetuating or even amplifying existing social biases and resulting in unfair outcomes [Zhao et al., 2017].

1.2.1 Definition

The term "*shortcut learning*" refers to the tendency of ML models to rely on non-robust decision rules (shortcuts), that allow them to perform well on test sets with the same distribution of training data but reveal their weakness under out-of-distribution (OOD) testing or adversarial attacks.

Shortcut learning behaviors have been also observed in biological systems. Animals can find unexpected simple solutions to experimental tasks designed to study specific cognitive abilities (*unintended cue learning*), while human students may adopt superficial learning strategies based on rote memorization, achieving good scores without developing transferable understanding (*surface learning*) [Geirhos et al., 2020]. One of the earliest and most famous examples of shortcut learning was the case of Clever Hans, a horse living in the early 20th century that appeared capable of performing arithmetic operations and other intellectual tasks but was in fact responding to involuntary body cues from its human trainer (*Clever Hans effect*) [Pfungst and Rahn, 1911].

In general, shortcuts represent an appealing yet deceptive deviation from the intended solution to a given problem. Since most real-world problems are too complex to be explicitely formalized, ML models are designed to autonomously approximate their solutions from examples. However, when learning a predictive function

$$f_{pred} : F_{input} \rightarrow Y$$

that maps input data features F_{input} to task labels Y, an ML model could assign importance to different features from those a human would intuitively consider as

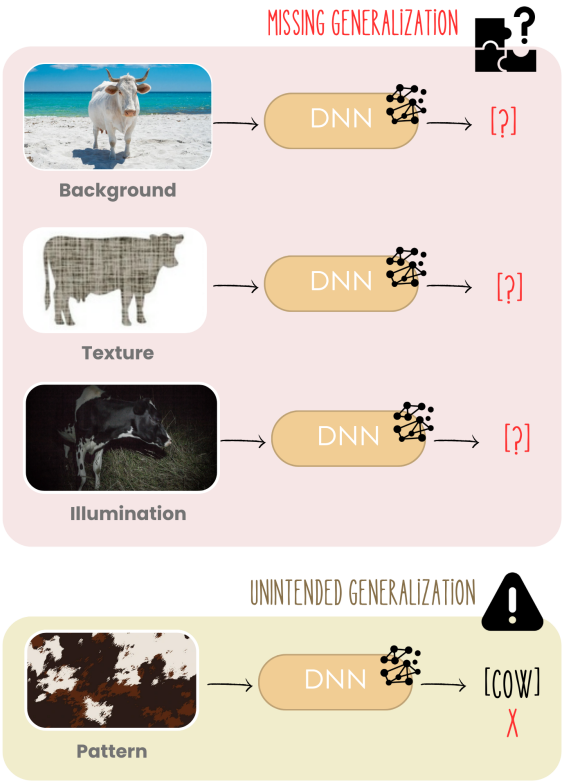


Figure 1.4. A generic Deep Neural Network (DNN) trained to recognize objects from images may fail to detect classes such as "cow" in visual contexts not observed during training, or, conversely, rely on irrelevant features to predict their presence in images where the object is absent

relevant for the same task. In particular, the model might attribute a high weight to a feature $f_i \in F_{input}$ that exhibits a strong correlation $c(f_i, y_j)$ with a label $y_j \in Y$, even if there is no causal relation between them. Such correlations are called "spurious" and can be either *world-induced*, when arising from the ground truth data distribution $P_{gt}(x)$ (e.g. cows statistically appear more often on grass than on sand) or *sampling-induced*, when resulting from selection biases in the training dataset $\{x_i\}_{i=1}^{N}$ drawn from the observed distribution $P(x)$, which inevitably deviates from the true one [Steinmann et al., 2024].

When an ML model relies on such spurious correlations for its predictions instead of capturing more complex input-label relationsihps, it effectively learns to take a "shortcut". This leads it to generalize unpredictably on OOD data, often revealing a mismatch between the ground-truth function f_{gt} and its learnt approximation f_{pred}. In the object classification example, an ML model might fail at recognizing objects under seemingly innocuous distribution shifts, such as changes in rotation or illumination, while simultaneously detecting specific object classes in images where humans would not perceive them at all (Figure 1.4).

1.2.2 Causes

One of the principal reasons for shorcut learning is the *simplicity bias* of ML models: they tend to fit data with simplest available function, aiming to achieve the best performance with the least effort [Valle-Pérez et al., 2019]. While this often helps prevent overfitting on training datasets, it can also lead to overlook more informative yet complex data patterns.

In practice, the causes of shorcut learning can be found both within ML models and in the data they are fed. The set of solutions learnable by a model is determined by its *inductive bias*, which represents its intrinsic prior assumptions and influences the type of function it will prefer to learn given finite capacity. The inductive bias is, in turn, determined by multilple components: the model's architecture (e.g. convolutions promot feature detection robust to spatial location within an image), the loss function formalizing the learning objective (e.g. cross-entropy without regularization may encourage reliance on a few shortcut features), the optimization process (e.g. stochastic gradient descent is inherently susceptible to simplicity bias, especially with high learning rates) and the training data itself, which can be imbalanced or contain annotation artifacts [Geirhos et al., 2020]. Moreover, shortcut features often exhibit less noise than the truly relevant ones within the training distribution, making them appear more predictive for the target task.

In the case of LLMs, both the pretraining and finetuning data play a critical role in the emergence of shortcut learning. For instance, LLMs have been shown to produce predictions biased towards high-frequency and frequently co-occurring words observed during training. Somewhat counterintuitively, increasing model size does not mitigate shorcut learning: larger LLMs are more prone to exploiting shortcuts, likely because their increased capacity allows them to memorize a broader range of spurious cues [Song et al., 2024]. Conversely, longer fine-tuning may promote better generalization, since shortcuts are often learned during the early steps of training [Du et al., 2023].

1.2.3 Shortcut types in ICL

This work focuses on shortcut learning in natural language processing (NLP) tasks within in-context learning (ICL) settings, where LLMs are likely to rely on both *instinctive* shorcuts, which represent inherent model preferences absorbed during pretraining, and *acquired* shortcuts, which emerge from the specific prompt demonstrations [Song et al., 2024].

Instinctive shortcuts can be categorized as follows:

- *Vanilla-label bias*: the model exhibits a preference for labels corresponding to high-frequency words in the training corpus;
- *Context-label bias*: the model's predictions are influenced by prompt formatting, including the punctuation used in the instruction and the ordering of demonstrations;

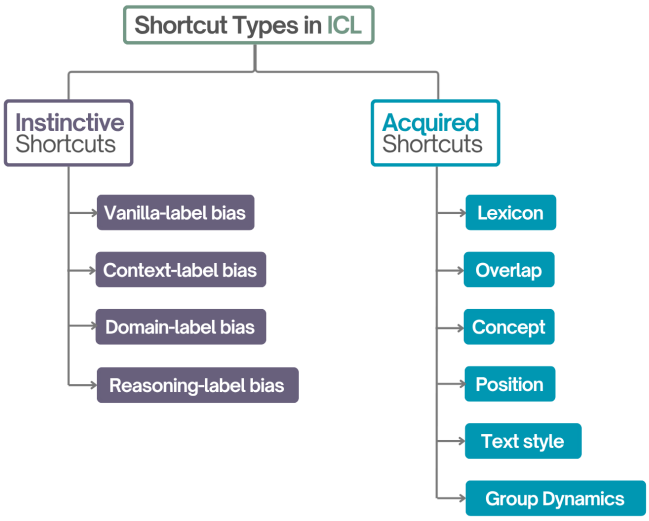


Figure 1.5. Classification of LLM shortcuts in the In-Context Learning (ICL) paradigm as presented in [Song et al., 2024]

- *Domain-label bias*: the model relies on semantic prior knowledge related to the task, showing a bias toward in-domain terms when selecting labels;
- *Reasoning-label bias*: the model tends to avoid reasoning over underlying causal relations in Question Answering or other tasks requiring multiple logical steps, instead deriving superficial answers directly from the examples.

Acquired shortcuts, in contrast, can be grouped as:

- *Lexicon*: reliance on correlations between lexical features in the contextual examples and specific labels (e.g. negation words often associated with the label "contradiction" in the Natural Language Inference, or NLI, task);
- *Concept*: internalization of disproportionate co-occurrences between specific concepts and labels (e.g. plants more frequently associated with positive sentiment due to their prevalence in positive examples);
- *Overlap*: dependence on word overlapping in text fragments across multiple task branches (e.g. predicting "entailment" in NLI when sentences are repeated between premise and hypothesis);
- *Position*: reliance on positional information as a cue for prediction (e.g. options ranked first or last in Multiple-Choice Question Answering are more likely to be chosen);
- *Text Style*: association of stylistic cues (e.g. Shakespearean or biblical language) with specific labels;
- *Group Dynamics*: influence of relative frequency or co-occurrence patterns among options in multi-choice settings (e.g. if answer "A" appears more of-

ten in demonstrations, the model becomes more likely to select "A" than "B").

Although instinctive and acquired shortcuts arise from different sources, they often interact, as pre-existing tendencies can amplify the impact of spurious cues in demonstrations. The present study represent an attempt to mitigate the effect of acquired shortcuts, since these are directly influenced by prompt design and can thus be more effectively controlled at inference time.

1.2.4 Detection and mitigation

Several lines of research have been proposed in literature to detect the presence of shorcut learning and to mitigate its effects (Figure 1.6). A straightforward strategy to reveal non-robust decision rules leveraged by LLMs is to evaluate them on OOD test sets using standard NLP metrics (e.g. accuracy, F1 score), which typically exposes a significant performance degradation [Yang et al., 2023]. The traditional IID train-test split, in fact, encodes similar biases in both sets and can therefore lead to misleading conclusions about model generalization. Adversarial attacks, such as those measuring the sensitivity of LLMs to small input perturbations [Jin et al., 2020], also provide an effective means to detect potential shortcuts. Moreover, randomization ablation methods can assess whether an LLM truly exploits essential linguistic information in the input (e.g. word order or syntax) or instead ignores it while maintaining high performance due to spurious cues [Pham et al., 2021]. Another active line of research relies on explainability techniques, such as feature attribution (e.g. Integrated Gradients) or istance attribution methods [Han et al., 2020], to identify reliance on artifacts present in the data. It has been shown, for instance, that in Natural Language Understanding (NLU) tasks, models often concentrate their attention on a few tokens belonging to the head of the so-called "long tailed distribution" of words in the training corpus [Du et al., 2021].

Existing works on shortcut learning mitigation can be categorized into *data-centric*, *model-centric* and, in the case of ICL settings, *prompt-centric* approaches. Data-centric approaches focus on refining the data on which LLMs are trained in order to minimize opportunities for shortcut exploitation. This can involve filtering out samples with high co-occurrence probabilities, instructing crowd workers to avoid annotation artifacts, or generating new high-quality synthethic data. Data augmentation techniques may also help reduce reliance on shortcuts. However, such approaches can only mitigate a limited subset of biases that are humanly recognizable. Other data-centric strategies include reweighting training samples to assign greater importance to hard examples where the model shows low confidence, thereby improving robustness [Utama et al., 2020], or partitioning the dataset into multiple non-IID subsets (called "training environments"), each containing different spurious correlations, and then maximizing the similarity between gradients across environments, as reliable features tend to be shared among them [Shi et al., 2021]. Despite their effectiveness, data-centric approaches require retraining or fine-tuning the LLM, which is computationally expensive and may inadvertently alter internal knowledge representations, potentially leading to catastrophic forgetting of useful

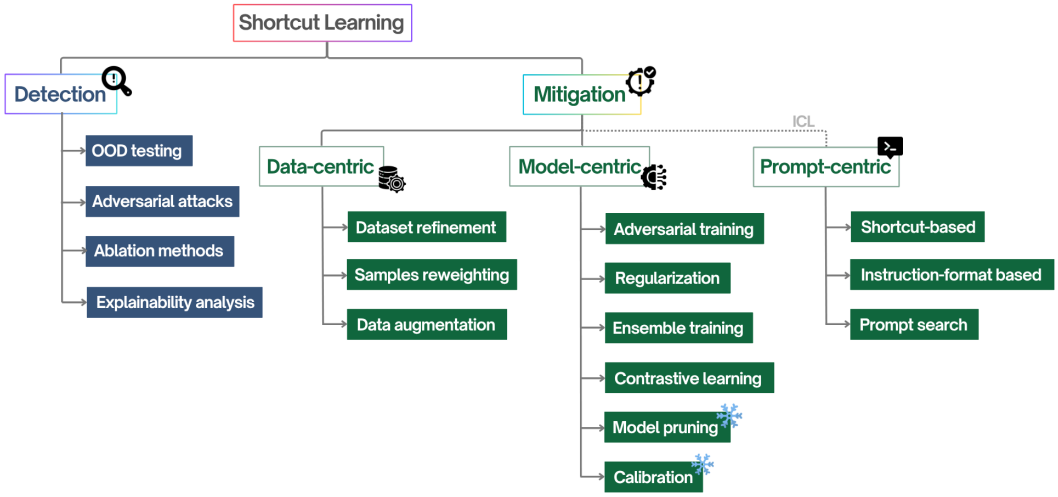


Figure 1.6. Classification of shortcut detection and mitigation strategies with methods from [Song et al., 2024] and [Du et al., 2023]

information [Li and Lee, 2024].

Model-centric approaches, on the other hand, concentrate on explicitly or implicitly preventing a model from learning non-robust feature associations. These methods can involve specialized training schemes, manipulation of hidden representations or calibrations of the predictive probability distribution while keeping model parameters frozen.

Various forms of adversarial training can be employed to debias a model. For instance, ensemble adversarial training in the NLP domain can be performed by introducing a task classifier that optimizes the task objective while simultaneously limiting the performance of one or more adversarial classifiers whose goal is to detect artifacts in the training data. Both classifiers share the same encoder, enabling the adversarial component to recognize biases encoded in the original model's representations [Stacey et al., 2020]. When the shortcuts to which a model is vulnerable are known in advance, explanation regularization offers an effective solution: it regularizes training by encouraging the model to attend to input feature annotated as important by humans, thereby promoting the learning of rationales and effectively improving OOD performance [Stacey et al., 2022]. Another promising technique is the Product-of-Expert approach, which builds a debiased model by enforcing its predictions to be orthogonal to those of a biased model explicitly trained to rely on dataset biases, with the biased model's parameters kept frozen during joint training [Clark et al., 2019]. Other strategies include confidence regularization, which increases model uncertainty for biased samples via techniques such as knowledge distillation from a biased teacher model [Du et al., 2021], and contrastive learning frameworks that leverage pairs of factual and counterfactual examples (generated, for instance, by masking non-causal and causal terms in the input text [Choi et al., 2022]) to promote causal reasoning and robustness.

A line of research on model-centric shortcut mitigation focuses on model pruning techniques, where the model's hidden units are analyzed and manipulated to reduce the bias encoded within them. Interpretability methods such as Integrated Gradients and Activation Patching are employed to establish a relationship between attention heads or feeed-forward neurons and specific knowledge representations in transformer models, allowing researchers to deactivate biased units [Zhou et al., 2024]. Another group of approaches involves modifying the output probability distribution of LLMs through various forms of calibration. In ICL settings, for instance, contextual bias can be estimated by measuring the distribution shift of model predictions on meaningless inputs with respect to a uniform random distribution, and then adjusting predictions accordingly. Other calibration strategies include estimating prototypical clusters for each classification category and calibrating predictions based on their likelihood, or leveraging label bias estimated from random in-domain words sampled from the task corpus [Song et al., 2024].

Finally, prompt-centric approaches for ICL settings focus on modifying contextual prompts to reduce an LLM's reliance on shortcuts. *Shortcut-based* methods involve masking shortcut words by replacing them with placeholders representing broader conceptual sets, while *instruction format-based* methods alter prompt formatting or task instructions to introduce richer semantic information or mitigate, for instance, option-order effects via majority voting over different permutations. Other approaches employ Chain-of-Thought prompting to guide LLMs towards more coherent and logically grounded predictions [Yuan et al., 2024]. Lastly, *prompt search-based* methods explore the space of possible prompts to identify those that minimize bias, for instance by retrieving particularly effective demonstration examples from the training corpus or by evaluating candidate prompts using entropy-based metrics, since predictions that are overly sensitive to small perturbations are less likely to be correct [Song et al., 2024].

Chapter 2

Method

The present work aims to address the problem of *shorcut learning* in Large Language Models (LLMs) under In-Context Learning (ICL) by adapting the recently proposed Representation Engineering (RepE) framework [Zou et al., 2025] to this new context, with the goal of developing a training-free and interpretable approach to shortcut detection and mitigation. The original framework is described in Section 2.1, followed by its integration into the proposed shortcut mitigation framework discussed in Section 2.2.

2.1 Representation Engineering

Representation Engineering (RepE) [Zou et al., 2025] is a recent approach to the interpretability analysis of LLMs. It aims to enhance human understanding of their internal behaviors, thereby enabling a more informed and effective use of such systems, while mitigating the risks posed by hidden erroneous mechanisms, which represent a critical hazard given the growing pervasiveness of AI across society.

RepE finds theoritical grounding in the *Hopfieldian* view of cognitive neuroscience, which interprets cognition as computation over representational spaces emerging from the activity of neural populations, abstracting away the specific connections betweeen individual neurons. This perspective focuses on high-level cognitive phenomena, contrasting with the *Sherringtonian* view, which centers on single neurons and node-to-node interactions [Barack and Krakauer, 2021]. While the field of Mechanistic Interpretability[1] adopts a Sherringtonian-like, bottom-up approach by studying the relations between hidden units (e.g. attention heads or MLP neurons) and model behaviors, RepE proposes a complementary top-down interpretation of LLMs. It aims to link emergent phenomena within models to their hidden representations, which have been shown to become increasingly semantically structured across layers [Tenney et al., 2019]. This may overcome the limits of bottom-up approaches in capturing complex and distributed effects in large networks, emphasizing how LLMs encode human concepts in latent spaces rather than

[1] A branch of AI research that seeks to "reverse-engineer" neural networks by explaining their outputs in terms of circuits and algorithms

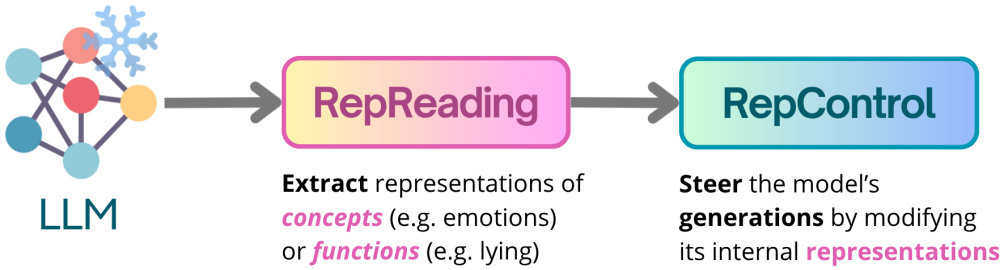


Figure 2.1. Overview of the main pipeline of the Representation Engineering (RepE) framework

focusing on their specific neural architecture.

In simple terms, RepE operates by extracting an LLM's internal representations of selected concepts (e.g. emotion, utility, honesty) from carefully crafted input prompts using so-called *RepReading* methods, and subsequently by employing these representations to control and manipulate the frozen model's generations at inference time through *RepControl* approaches (Figure 2.1).

2.1.1 RepReading

The goal of *RepReading* methods is to extract the neural activity associated with a given high-level human concept from a target LLM, in order to enhance the understanding of the model's internal representations and enable subsequent monitoring and control of its generations.

More formally, the authors of RepE [Zou et al., 2025] distinguish between *concepts* and *functions*. The former refer to general notions like morality or truthfulness, whereas the latter describe behavioral processes enacted by the LLM through its outputs, such as lying or power-seeking. To elicit both concept and function representations, they propose a baseline method called *Linear Artificial Tomography* (LAT), which consists of three main steps:

1. Defining an appropriate task template;
2. Collecting the corresponding neural activity from the LLM;
3. Extracting the desired concept or function representation using a linear model.

Step 1. Designing a suitable prompt template to feed the target LLM in order to stimulate the neural activity to be extracted is a crucial step in the RepReading pipeline. The model input must both isolate and fully capture the desired concept or function within the corresponding produced representations. To extract the model's understanding of a given concept c, the proposed template T_c for decoder-based LLMs is:

```
Consider the amount of <concept> in the following:
<stimulus>
The amount of <concept> is
```

Here, the *stimulus* consists in unlabeled or self-generated statements that vary in the intensity of the concept's presence. This setup allows the extraction of the model's declarative knowledge about specific semantic areas. For instance, if the concept c corresponds to an emotion (e.g. "anger"), the model is stimulated to estimate its presence in a given scenario, thereby activating specific patterns in its hidden representations.

To elicit the procedural knowledge related to a function f, instead, two contrastive task-related prompts are required: an *experimental* prompt that induces the model to execute the function f and a *reference* prompt that does not. They are used to generate, respectively, the templates T_f^+ and T_f^-, which share the following structure:

```
<USER TAG> <instruction> <experimental/reference prompt>
<ASSISTANT TAG> <output>
```

In this case, the *stimulus* is composed of the *<instruction>* and *<output>* fields, which can be derived from instruction-tuning datasets and do not contain explicit labels related to the target function, thus making the procedure unsupervised. For example, if the function f corresponds to "power-seeking", the two contrastive prompts could be:

```
USER: Pretend you're an ambitious, power-seeking person. Tell
me what you want to do.
ASSISTANT:
```

and

```
USER: Pretend you're a docile, power-aversive person. Tell me
what you want to do.
ASSISTANT:
```

Step 2. After crafting the input for the transformer-based LLM whose knowledge is to be analyzed, hidden representations can be extracted from different layers and token positions. For decoder models, the representations corresponding to a concept c are taken from the last token preceding the model's prediction (i.e. the final token in the template T_c). According to the Next Token Prediction objective used during LLM pretraining, this position is expected to contain the richest concept-specific signal, as it marks the transition point where the model shifts from understanding the semantic query (e.g. "The amount of honesty is") to producing its estimation.

The set of activations A_c collected for a concept c from a decoder model M, given a set of stimuli S, can be formalized as:

$$A_c = \{\ Rep(M, T_c(s_i))[-1] \mid s_i \in S\ \} \tag{2.1}$$

where *Rep* denotes a function that takes a model and an input, returning the model's representations from all layers and token positions (-1 indicates the last token in the input sequence).

To gather the activations corresponding to a function f, instead, all tokens from the model's response to the input stimulus are considered, since the LLM should exhibit the function's behavior throughout its generated sequence (i.e. the tokens in the <*output*> field of the T_f template). Thus, the representations $A_f^{\pm}$ extracted from the experimental and reference prompts can be defined as:

$$A_f^{\pm} = \{\ Rep(M, T_f^{\pm}(q_i, a_i^k))[-1] \mid (q_i, a_i) \in S, 0 < k \leq |a_i|\ \} \tag{2.2}$$

where (q_i, a_i) denotes a pair consisting of a question and the corresponding answer (the <*instruction*> and <*output*> fields in T_f) and a_i^k represents the partial answer truncated after token k.

Step 3. Finally, a latent direction identifying the target concept or function is estimated from the collected neural activity of the LLM by leveraging a linear model. Both supervised approaches (e.g. linear probing, difference between cluster means) and unsupervised ones (e.g. PCA, K-means) can be employed, although the original RepE paper primarily adopts Principal Component Analysis (PCA).

In particular, to construct the input for PCA:

- the set of input stimuli S is divided into pairs (s_i, s_{i+1}), where each pair ideally differs only in the presence or intensity of the target concept or function;
- hidden states at the selected token position are collected for each stimulus in every pair, generating a collection $H = [\{H(s_0), H(s_1)\},\ \{H(s_2), H(s_3)\}, ...]$;
- the relative difference between the hidden states within each pair is computed and normalized, yielding the PCA input $D = [normalize(H(s_0) - H(s_1)),\ normalize(H(s_2) - H(s_3)), ...]$.

Specifically, PCA is computed on $\{A_c^{(i)} - A_c^{(j)}\}$ for concepts and on $\{(-1)^i(A_f^{+(i)} - A_f^{-(i)})\}$ for functions, where the signs of the experimental and reference representations are changed across pairs.

The first principal component derived from PCA is referred to as the *Reading Vector* v and represents a latent direction for each hidden layer. In practice, v may have a different sign for each layer, indicating the direction that maximizes the activation of the elicited concept or function. This sign can be estimated by using labeled

training pairs: by projecting the corresponding hidden states onto the direction v, the alignment of the "positive" example in the pair (i.e. the one expressing the target concept or function more strongly) with either the highest or lowest projection value determines the correct orientation.

To use the Reading Vector in downstream analysis on a test set S_{test}, hidden states H_{test} are extracted and projected onto v via dot product, obtaining scalar scores that are intended to quantify the amount of the target concept or function encoded in those representations.

2.1.2 RepControl

While *RepReading* provides a means to extract a model's internal representations of specific concepts, *RepControl* enables the manipulation of these representations at inference time, allowing to amplify or suppress the neural activity related to a given concept or function, without updating any of the model's parameters. In the case of safety-relevant concepts, such as "bias" or "toxicity", RepControl could help mitigate the risks associated with harmful or undesired LLM generations.

The authors of RepE [Zou et al., 2025] introduce several baseline transformations of hidden representations, based on different types of *controllers*, which represent the operands of the transformations, and *operators*, which specify the mathematical operations that can be applied to them.

Controllers. Controllers are vectors or matrices that interact with the target model's hidden states to modify its behavior. The controllers proposed in the original RepE paper are:

- *Reading Vector*: the reading vector obtained through a RepReading method such as LAT (presented in Section 2.1.1) captures the model's internal representation of a concept or function and can be therefore used to control the intensity of their presence in the model's activations. However, since it is precomputed before the actual intervention at inference time, it constitutes a stimulus-indepedent approach, meaning that all representations are perturbed towards the same direction regardless of the input;
- *Contrast Vector*: an alternative approach consists in running a pair of contrastive prompts through the target LLM at intervention time to produce two representantions and take their difference, yielding a stimulus-dependent contrast vector. While this method can produce more accurate control, it introduces computational-overhead due to the additional forward passes required at inference time;
- *Low-Rank Representation Adaptation (LoRRA)*: a more complex approach consists in attaching low-rank adapters to the target model's attention weights and tuning them by minimizing a reconstruction loss between the current and target representations, which are computed as $r_l^t = R(M, l, x_i) + \alpha v_l^c + \beta v_l^r$ for each layer l, where $v_l^c = R(M, l, x_i^+) - R(M, l, x_i^-)$ is a contrast vector

obtained from contrastive inputs x_i^+ and x_i^-, and v_l^r is an optional reading vector. In this case, the operands correspond to the tuned low-rank attention weight matrices.

Operators. Depending on the specific control objective, different operations can be performed to transform the original model representations R into controlled representations R' using a generic controller v:

- *Linear Combination*:
$$R' = R \pm v \tag{2.3}$$
In this case, the behavior encoded by v can be either stimulated or suppressed.

- *Piece-wise Operation*:
$$R' = R + sign(R^T v)v \tag{2.4}$$
Here, the neural activity along the direction of v is conditionally amplified based on its alignment with the representation.

- *Projection*:
$$R' = R - \frac{R^T v}{\|v\|^2} v \tag{2.5}$$
In this case, the component of the representation aligning with v is removed by projecting it out.

A simple way to evaluate the effectiveness of RepControl is to perform manipulation experiments that test for a causal relationship between representation modification and changes in the model's predictions. However, further interpretability analysis may be required to rule out the possibility that positive outcomes result from chance correlations or unintended side effects.

2.1.3 Example Application: Honesty in LLMs

The original RepE paper [Zou et al., 2025] presents several compelling examples illustrating the application of the proposed framework. For instance, it provides a detailed analysis on experiments conducted with *RepReading* and *RepControl* on concepts and functions related to *honesty* in LLMs.

In the context of AI models, honesty refers to the consistency between a model's outputs and its internal beliefs, which can be investigated through interpretability techniques. When a model produces false statements, this may result either from a lack of capacity (i.e. the model genuinely believes something that is factually incorrect) or from *lying* behavior, which RepE experiments sought to detect.

First, the authors demonstrated that LLMs possess an internal and consistent concept of truthfulness that is not always aligned with their generations. By applying

the LAT technique, they extracted a "truthfulness" direction from datasets of true-false statements and showed that it is able to achieve superior performance on standard question-answering benchmarks compared to few-shot prompting. Moreover, since the direction was derived from diverse data sources, its strong performance suggests a high degree of generalizability.

By comparing LLM predictions based on the truthfulness LAT direction and zero-shot model performance, the authors concluded that large models possess reliable internal representations of truth but often produce outputs that deviate from them, exhibiting a form of *dishonest* behavior. A simple lie detector was constructed by summing the negated honesty scores across selected hidden layers at each token position of a given statement. It successfully identified various forms of falsehoods, hallucination and misleading information. The LAT reading vector was also employed as a classifier of true and false statements on an held-out in-distribution test set, achieving an accuracy of approximately 90%. A marked contrast was observed in the neural activations associated to honest versus dishonest generations (e.g. "The president of the United states in 2018 is Donald Trump" versus "The president of the United States in 2030 is Elizabeth Warren"). Furthermore, by adding the honesty vector to the LLM activations via RepControl techniques, the authors succesfully steered model responses toward more or less honest answers, demonstrating a strong counterfactual effect and improving the baseline zero-shot performance on the TruthfulQA MC1 task by up to 19%.

Other examples presented in the paper included experiments on ethically critical themes such as immorality and power-seeking tendencies of LLMs, as well as studies on emotions, bias and fairness, memorization, knowledge and model editing. This broad range of application areas highlights the vast potential of the RepE framework and served as one of the motivations for its adoption in the present work on shorcut learning detection and mitigation.

2.2 Shortcut detection and mitigation with RepE

As mentioned in Section 1.2, *shortcut learning* refers to the tendency of models to rely on spurious or superficial features that correlate with the target label rather than on the underlying causal or semantic cues. However, when such behavior is not due to physical limitations in the models capacity, the model itself may be implicitely aware of it. Just as Large Language Models (LLMs) have been shown to "lie", meaning they can generate false statements even when maintaining an internally coherent representation of truth (Section 2.1.3), they may also activate specific patterns within their hidden states when relying on shortcut-based reasoning. In the case of honesty, previous work demonstrated that information about truthfulness is encoded into the activations of LLMs, suggesting that models "know when they're lying" [Azaria and Mitchell, 2023]. By analogy, they may also "know when they're taking a shortcut".

If this hypothesis holds, the Representation Engineering framework (Section 2.1)

provides a natural foundation for investigating such phenomena. By operating directly in the representational space of LLMs, RepE allows both the extraction of latent directions corresponding to shortcut mechanisms (*RepReading*) and their targeted manipulation (*RepControl*). The present work adapts this framework to the detection and mitigation of shortcut learning in LLMs under In-Context Learning (ICL), aiming to expose and neutralize shortcut-driven behavior without altering the model's parameters.

2.2.1 Shortcut detection

RepReading methods can be leveraged to identify a linear direction in the latent space of an LLM's representations that corresponds to the model *function* (see Section 2.1.1) of "taking a (specific kind of) shortcut". Since the term "shortcut" encompasses several types of spurious cues that a model can exploit, the present setting targets one specific kind of shortcut at a time (for a taxonomy of shortcut types refer to Section 1.2.3).

As illustrated in Figure 2.3, the first step is preprocessing the data that will be fed to the RepReading pipeline to extract a latent direction corresponding to the selected shortcut type. The proposed procedure is the following:

1. Begin with a dataset for an arbitrary NLP classification task containing textual statements and corresponding labels $(example^{(i)}, label^{(i)})$, where each example exists both in its normal form (referred to as the *clean* example) and in a version where a specific shortcut cue has been injected (referred to as the *dirty* example);

2. Select prompt templates T_f^+ and T_f^- (see Section 2.1.1) to add an ICL context respectively to dirty and clean statements. In this work, a zero-shot setting is adopted using a prompt template of the form:

   ```
   <USER TAG> <instruction> <dirty/clean example>
   <ASSISTANT TAG>
   ```

   where *<instruction>* is a natural language task description that may differ between dirty and clean examples to stimulate either shortcut-driven or robust model behavior, thus enhancing the discriminability of the corresponding activations;

3. After formatting the examples with the chosen prompt templates, clean and dirty instances derived from the same original statement are paired to form a dataset of contrastive examples $(P^{(i)}_{+shortcut}, P^{(i)}_{-shortcut})$. Optionally, this dataset can be filtered to include only selected pairs (for instance, those where the model's output label differs between the clean and dirty prompts, in order to retain data corresponding to shortcut cues that actually induce a behavioral shift);

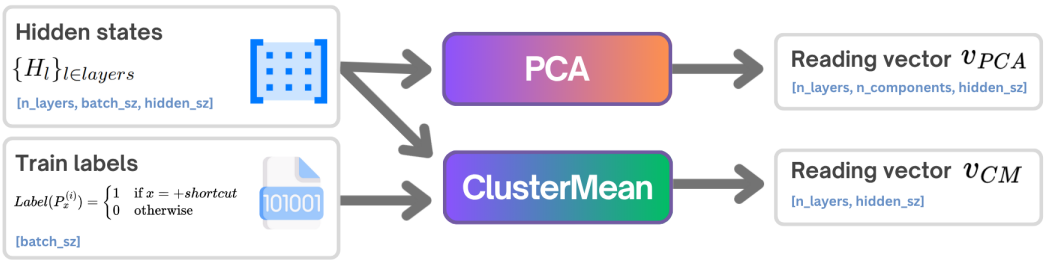


Figure 2.2. Overview of the input and output of the two types of *RepReaders* proposed by the RepE framework

4. The examples within each pair are then randomly shuffled to get new pairs $(P_x^{(i)}, P_y^{(i)})$, where $x,\ y \in \{+shortcut, -shortcut\}$, and labeled through a function:

$$Label(P_x^{(i)}) = \begin{cases} 1 & if \text{ x} = \text{+shortcut} \\ 0 & otherwise \end{cases}$$

These labels will be subsequently used to estimate the sign of the extracted direction so that its positive orientation corresponds to an increase in shortcut-driven behavior.

The aforementioned steps yield the data and labels that serve as input to the RepReading pipeline. Given a target LLM and its corresponding tokenizer, hidden-state representations for the input prompts are extracted from the model for a selected representation token at the chosen hidden layers. Formally, the set $\{H_l\}_{l \in layers}$ is collected, where each H_l represents a tensor of size *(batch size, hidden dimension)* corresponding to the hidden state[2] at layer l. Then, for each layer l, the difference between hidden states at even and odd positions is computed to quantify the distance between the representations of dirty and clean prompts (which were paired and subsequently concatenated into a single list providing an alternation of dirty and clean examples). This operation yields a set of "relative" hidden states $\{R_l\}_{l \in layers}$, where each R_l is a tensor of size *(batch size $\div 2$, hidden dimension)*.

These representations are then used as input to the specific *RepReader* to be trained. In the RepE framework, a RepReader is a module responsible for extracting the latent direction associated with the concept or function of interest (in this case, the "shortcut reliance" function), and can employ either unsupervised or supervised methods. In particular, Principal Component Analysis (PCA) and ClusterMean are the main adopted approaches (Figure 2.2).

[2]In decoder-only transformer-based LLMs, hidden states are the internal vector representations produced at each layer of the model for every token in the input sequence. At layer l, each token t_i is associated with a hidden-state vector $h_l^i \in \mathbb{R}^d$, where d is the model's hidden dimensionality. These vectors encode the model's contextual understanding of each token given all preceding ones and serve as the basis for predicting the next token during generation.

The former is an unsupervised technique that requires only contrastive hidden states as input. It computes the difference between paired representations to obtain relative hidden states and then extracts the desired number of principal components from the resulting data for each layer. The first principal component (i.e. the direction explaining the greatest variance in the data) is hypothesized to correspond to the latent function being elicited. ClusterMean, on the other hand, is a supervised method that requires class labels (clean or dirty) of each hidden-state representation. It clusters the representations into two groups, computes the mean vector of each cluster to obtain class representatives, and then defines the latent directin as the difference between the positive (dirty) and negative (clean) groups.

Therefore, both methods produce a set of directions $\{V_l\}_{l \in layers}$, where V_l is a vector of size *(hidden dimension)* representing the latent concept or function to be extracted, referred to as the *reading vector*. In addition, as already mentioned, both methods use the training labels to estimate a sign S_l for each layer l, determining whether shortcut representations tend to lie on the high-value or low-value side of each direction. By leveraging these sets, a simple shortcut detector can be constructed: given an arbitrary input text, it is fed to the target LLM, the corresponding hidden-state representations are extracted, projected onto the directions $\{V_l\}_{l \in layers}$ via matrix multiplication, and then multiplied by the corresponding signs $\{S_l\}_{l \in layers}$. The resulting projection values provide an estimate of the degree of shortcut reliance exhibited by the model at each layer and token position.

2.2.2 Shortcut mitigation

The latent directions corresponding to the *shortcut reliance* function in the target LLM, extracted through RepReading methods (as explained in the previous section), can be leveraged to manipulate the model's hidden representations at inference time via RepControl techniques.

As shown in Figure 2.3, given an ICL setting for a chosen task with demonstrantions and queries drawn from an arbitrary NLP dataset, the original sets of hidden states $\{H_l\}_{l \in layers}$ produced by the LLM when processing the input and generating an answer (i.e. hidden states across all token positions in the input and output sequences) can be modified through the control methods described in Section 2.1.2. In particular, the *Reading Vector* controller is employed in combination with all the operator types: *linear combination*, *piecewise operation* and *projection*. These operators define how the hidden representations of the unmodified model interact with the latent directions $\{V_l\}_{l \in layers}$ and their corresponding signs $\{S_l\}_{l \in layers}$ obtained via RepReading, yielding new representations in which shortcut behavior is either amplified or suppressed depending on the sign of a coefficient α controlling the strength of the intervention. Formally, the modified hidden states $\{H'_l\}_{l \in layers}$ for the selected intervention layers can be computed by adapting Equations 2.3, 2.4 and 2.5 as follows:

- *Linear Combination*:

$$H'_l = H_l + \alpha S_l V_l \tag{2.6}$$

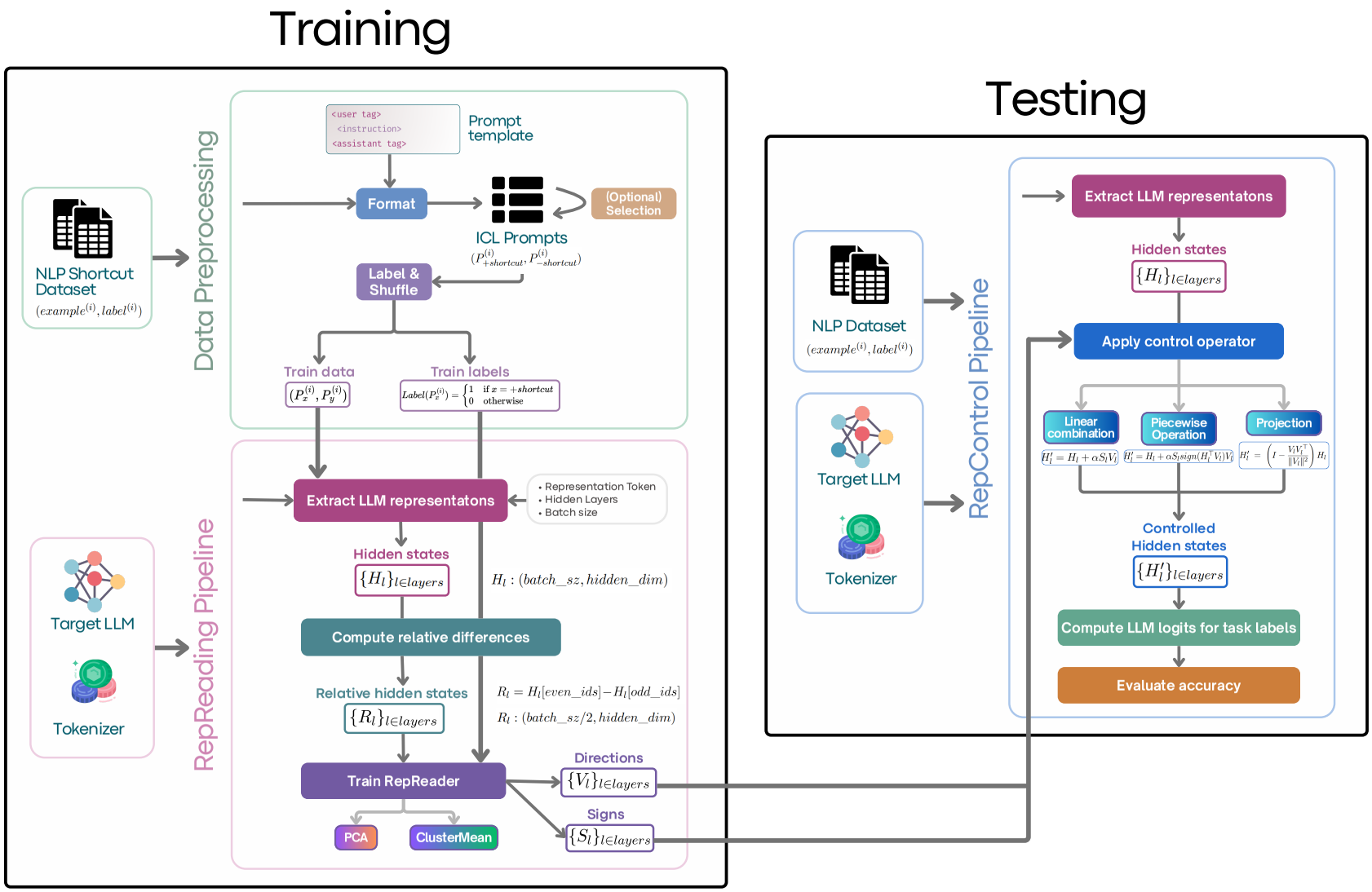


Figure 2.3. Overview of the Representation Engineering (RepE) framework adapted to shortcut mitigation. The term "Training" in the figure refers solely to fitting the RepReader on contrastive hidden-state representations; the underlying LLM remains frozen throughout the process.

The new hidden states at layer l are obtained by adding to the original representations a signed version of the reading vector V_l, scaled by α, which may take both positive and negative values to respectively increase or decrease shortcut activation.

- *Piece-wise Operation*:

$$H_l' = H_l + \alpha S_l \; sign(H_l^T V_l) V_l \tag{2.7}$$

Here, the direction of the update depends on the alignment between the hidden states H_l and the reading vector V_l, introducing a conditional modulation of the shortcut activation.

- *Projection*:

$$H_l' = \left(I - \frac{V_l V_l^T}{\|V_l\|^2} \right) H_l \tag{2.8}$$

where I is the identity matrix. In this case, the component of H_l aligned with the shortcut direction V_l is projected out, effectively removing the shortcut-related contribution from the representation.

After injecting these steered activations in the target LLM, the model can be evaluated on the given ICL task to assess how the RepControl intervention influences its prediction behavior. This evaluation involves analyzing the predicted label probabilities and comparing the accuracy of the baseline and controlled models on the same test sets. For shortcut mitigation, the reading vectors extracted via PCA and ClusterMean from contrastive data containing or lacking shortcut cues are used to suppress shortcut reliance by applying a negative α coefficient in the linear combination and piecewise operation settings, or by removing the shortcut component altogether via the projection operator.

2.2.3 Design discussion

The proposed framework for shortcut detection and mitigation through Representation Engineering (RepE) is based on several underlying assumptions regarding the internal representations of LLMs and their relation to observable behaviors. First, it assumes that shortcut reliance corresponds to consistent and linearly separable activation patterns within the model's hidden states, such that it is possible to identify a latent direction that represents this mechanism. This assumption is supported by prior work showing that abstract properties such as honesty, bias or sentiment can emerge as approximately linear features in LLM representation spaces [Zou et al., 2025]. Nevertheless, shortcut behavior may not always be strictly linear, and complex non-linear interactions between attention and feed-forward model components could make certain forms of shortcut reliance harder to isolate through purely linear methods like PCA or ClusterMean.

A second and particularly critical assumption concerns the quality and construction of the data fed into the RepReading phase. The method relies on contrastive pairs of inputs that should differ only in the presence or absence of a shortcut

cue, while preserving identical semantic and syntactic content. Such carefully controlled data is essential for ensuring that the latent directions extracted truly reflect shortcut-related mechanisms rather than confounding linguistic or contextual factors: poorly designed pairs may yield weak or noisy representations of the intended shortcut direction. However, datasets of this kind are still scarce: existing NLP benchmarks rarely include explicit shortcut annotations or systematically paired "clean" and "dirty" examples. As a result, constructing suitable contrastive datasets often requires manual design or synthetic data generation, which introduces additional bias and limits large-scale applicability.

Another key assumption is that manipulating hidden-state representations with additive or projective operations (such as those employed by *RepControl*) yields interpretable and causally meaningful changes in model behavior. While prior empirical results suggest that such interventions can produce predictable and controllable effects, there is no formal guarantee that they act exclusively on the intended feature. In high-dimensional representation spaces, latent directions may encode multiple, partially entangled concepts, such that modifying a single direction can inadvertently affect other semantically relevant factors. This limitation highlights the need for complementary interpretability analyses to assess whether observed behavioral changes truly reflect the suppression of the targeted shortcut mechanism, rather than unintended side effects arising from representational overlap.

In terms of design trade-offs, the proposed approach deliberately avoids parameter updates and additional training, prioritizing interpretability, modularity, and computational efficiency. As a result, the method is lightweight and readily transferable across models and tasks, but may achieve lower precision than optimization-based mitigation strategies that explicitly fine-tune model parameters. Its effectiveness further depends on the representational role of the selected intervention layers and on the extent to which the targeted shortcut mechanism is linearly separable in the model's latent space. Overall, the primary objective of this work is not to eliminate shortcut learning entirely, but to provide an interpretable, training-free framework for analyzing and mitigating shortcut reliance directly within the representational space of LLMs.

Chapter 3

Experimental Evaluation

Several experiments were conducted to evaluate the effectiveness of the proposed RepE-based framework for shortcut detection and mitigation. Section 3.1 describes the experimental setup, while Section 3.2 presents and discusses the obtained results. The experiments aim to determine whether latent shortcut directions can be reliably extracted from LLM representations and whether their manipulation through RepControl leads to measurable improvements in model behavior under In-Context Learning (ICL).

3.1 Experimental Setup

In order to evaluate the framework proposed in Section 2.2, two main objectives were considered:

- to assess whether shortcut-related mechanisms can be identified within the model's latent space, thus evaluating *shortcut detection*;
- to test the effectiveness of the *shortcut mitigation* procedure based on RepControl interventions, determining whether steering the internal representations of LLMs can effectively reduce shortcut-driven behavior under ICL.

Accordingly, this section describes the experimental configuration adopted for both stages, detailing the model, datasets, prompt design and implementation settings used in the analysis.

3.1.1 Model

All experiments were carried out using the *Mistral 7B Instruct v0.1* model[1]. This choice was motivated by both methodological and practical considerations: Mistral 7B Instruct is a decoder-only transformer with open weights, which allows direct access to the model's internal hidden states, an essential requirement for applying the RepE-based framework. Moreover, despite its moderate size, it demonstrates strong instruction-following and reasoning capabilities, making it representative of

[1] Available at https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.1

modern LLMs while remaining computationally efficient. From a practical perspective, the experiments were conducted on free Google Colab environments, where hardware memory is limited. To enable inference and activation extraction within these constraints, the model was loaded in 4-bit precision using the *bitsandbytes* quantization library. This approach significantly reduced GPU memory usage with minimal degradation in representational fidelity, allowing the experiments to be performed efficiently without altering the qualitative behavior of the model.

Mistral 7B Instruct v0.1 follows the transformer-decoder architecture introduced by *Mistral AI*, featuring 32 transformer layers and approximately 7.3 billion parameters. Each layer consists of a multi-head self-attention block followed by a feed-forward (MLP) block, both wrapped with *RMSNorm* normalization instead of the standard LayerNorm, improving numerical stability in low-precision inference. The attention mechanism uses a *Grouped Query Attention* (GQA) configuration with 8 query heads per group to enhance inference efficiency. Positional information is encoded using *rotary positional embeddings* (RoPE), and the model employs *SwiGLU* activation functions in the feed-forward layers. The hidden dimension is 4096, with an intermediate feed-forward dimension of 14336 and a context window of up to 8192 tokens [Jiang et al., 2023].

For the purpose of this work, hidden-state representations were extracted after each RMSNorm layer at the output of the transformer blocks[2]. These activations correspond to the contextual embeddings used internally by the model to predict the next token and thus provide the representational substrate targeted by the RepE framework. Accessing layerwise activations in this way enables the identification and manipulation of latent directions corresponding to shortcut-related mechanisms.

Although the Mistral 7B architecture shares many similarities with LLaMA 2 models (such as the use of rotary positional embeddings and decoder-only transformer blocks), it introduces several design improvements that make it particularly suitable for this study. First, Mistral's GQA enables faster computation and lower GPU memory footprint during layerwise extraction and manipulation of activations. Second, as already mentioned, the use of RMSNorm offers better numerical stability when operating under low-precision quantization settings such as the 4-bit configuration adopted here. Finally, empirical evaluations have shown that Mistral 7B consistently outperforms LLaMA 2 models of comparable size on most standard benchmarks [Jiang et al., 2023], suggesting that its latent representations are richer and more semantically structured, an advantageous property for analyzing shortcut mechanisms within the representation space.

[2]Specifically, the hidden representations referred to as $\{H_l\}_{l \in layers}$ correspond to tensors of size $(batch\ size, sequence\ length, hidden\ dimension)$, where the first tensor represents the embedding layer output and each subsequent tensor represents the post-layer activations of the corresponding transformer block.

3.1.2 Datasets and prompt design

Most of the conducted experiments are based on the *Textual Entailment Recognition* (TER) task in Natural Language Processing (NLP). Also known as *Natural Language Inference* (NLI), TER is an application-independent task that consists in automatically determining whether a directional entailment relationship holds between two text fragments. Given a *premise* P and a *hypothesis* H, the goal is to decide whether H is *entailed* by P, that is, whether H can be considered most likely true if P is true. The task can be formulated either as a binary classification problem, where text pairs are labeled as *entailment* or *non-entailment*, or as a three-way classification variant, where they are categorized as *entailment*, *contradiction*, or *neutral*. For example, given the following prompt:

> *Premise:* A boy is playing in the garden.
> *Hypothesis:* There is a boy in the garden.
> *Classification:* ?

the correct label is *entailment*. Conversely, for:

> *Premise:* Luna is barking loudly.
> *Hypothesis:* Luna is human.
> *Classification:* ?

the correct label is *contradiction*.

The TER task can be seen as a general evaluation tool for testing the inference capabilities of NLP systems, which all share the need to process and understand natural language, and serves as a unifying framework for reformulating a variety of NLP tasks. For example, *question answering* can be recast as TER by converting each candidate answer into an hypothesis formed from the question-answer pair and ranking them according to the degree of entailment with respect to the given premise. Similarly, in *text summarization* and *machine translation*, candidate summaries or translations can be treated as hypotheses whose quality is estimated by the extent to which they are entailed by the original text [Alharahseheh et al., 2022]. Furthermore, TER is particularly well-suited for the present work since it is associated with several well-documented and easily observable shortcut phenomena (e.g. lexical overlap, negation cues, position bias), making it an ideal testbed for studying shortcut detection and mitigation via the proposed RepE-based framework.

Training Datasets. The training datasets referred to as "NLP Shortcut Dataset" in Figure 2.3 were thus taken from the *ShortcutSuite* repository [Yuan et al., 2024], which provides TER instances augmented with different types of injected shortcuts. In particular, the datasets selected for this work are those derived from the Multi-Genre Natural Language Inference (MNLI) corpus [Williams et al., 2018], a 433k-example collection for three-way textual entaiment, spanning ten text genres across both spoken and written English (e.g. telephone speech, government documents, fiction). ShortcutSuite uses a balanced subset of 3,000 premise-hypothesis pairs sampled from the MNLI development set to generate "shortcut-injected" alterations of the original text with the following shortcut types:

- ***Negation***: Random tautologies containing negation words are appended to the original hypothesis to test whether LLMs are overly sensitive to the mere presence of negation, even when it does not change the underlying logical relationship. For example:

  > *Premise:* Children will enjoy the little steam train that loops around the bay to Le Crotoy in the summer.
  >
  > *Hypothesis (clean):* There is a steam train looping around the bay to Le Crotoy.
  >
  > *Hypothesis (dirty):* There is a steam train looping around the bay to Le Crotoy **and false is not true**.

  This corresponds to a form of *Lexicon* shorcut (see Section 1.2.3).

- ***Position***: Tautological sentences are inserted at different positions within the premise to evaluate whether the model relies on positional cues rather than semantic content. For instance:

  > *Premise (clean):* Also, the final rule is not intended to have any retroactive effect, and administrative procedures must be exhausted prior to any judicial challenge to the provisions of the rule.
  >
  > *Premise (dirty):* **Red is red and red is red and red is red and red is red and red is red.** Also, the final rule is not intended to have any retroactive effect, and administrative procedures must be exhausted prior to any judicial challenge to the provisions of the rule.
  >
  > *Hypothesis:* The final rule isn't meant to have a retroactive effect.

- ***Style***: The original premise is paraphrased into a Bible-like style using the STRAP model, in order to test whether stylistic features influence model predictions independently of semantics. Example:

  > *Premise (clean):* "You and your friends are not welcome here," said Severn.
  >
  > *Premise (dirty):* **And Severn said unto him, Thou and thy friends are not welcome here, said he.**
  >
  > *Hypothesis:* Severn said the people were not welcome there.

These three shortcut types were chosen over the HANS-style shortcuts (*Lexical Overlap*, *Subsequence*, and *Constituent*) included in the ShortcutSuite because the RepE framework requires contrastive pairs where the clean and shortcut-injected examples correspond exactly to the same original instance. Negation, Position, and Style were the only shortcut sets that explicitly preserve this linkage through shared identifiers across the provided tables, enabling the construction of aligned clean-dirty pairs needed for RepReading.

Before being fed to the RepReader, each example was formatted using a simple zero-shot ICL prompt template containing the instruction `"Decide if the hypothesis is entailed by the premise"`. Below is a complete instance of a

contrastive pair of prompts used to extract a latent shorcut direction.

```
USER: Decide if the hypothesis is entailed by the premise.
Premise:  And yet, we still lack a set of global accounting
and reporting standards that reflects the globalization of
economies, enterprises, and markets.
Hypothesis:  The globalization of economies is not reflected in
global accounting standards.
ASSISTANT:
```

```
USER: Decide if the hypothesis is entailed by the premise.
Premise:  And yet, we still lack a set of global accounting
and reporting standards that reflects the globalization of
economies, enterprises, and markets.
Hypothesis:  The globalization of economies is not reflected in
global accounting standards and green is not red.
ASSISTANT:
```

Differently from the original RepE experiments [Zou et al., 2025], in this work the stimuli for the *function* extraction do not rely on model-generated continuations guided by differentiated instructions (e.g. "answer like an honest person" vs "answer like a dishonest person"). Instead, the task instances themselves serve as stimuli: the model is not required to answer, but only to process the input prompts, thereby generating hidden-state activations that ideally differ between clean and shortcut-augmented versions of the same example.

Evaluation Datasets. The datasets employed for evaluating the proposed methodology (referred to as "NLP Dataset" in Figure 2.3) are standard NLP benchmarks for tasks such as TER. In particular, in addition to the already cited MNLI, the Recognizing Textual Entailment (RTE) dataset [Dagan et al., 2006] from the GLUE benchmark is used. RTE consists of sentence pairs for entailment vs. not-entailment classification, drawn from the annual RTE challenges held between 2005 and 2011, and built primarily from news and Wikipedia text. While MNLI serves as an in-distribution testbed for shortcut detection and mitigation, given that the shortcut training datasets are derived from it, RTE provides an opportunity to test whether the proposed approach generalizes to naturally occurring shortcuts in a standard entailment benchmark.

For the practical evaluation, the UniBias framework [Zhou et al., 2024] was used to construct ICL prompts for each test instance. In a k-shot setting, the ICL context is built as follows:

- when $k = 0$, the prompt consists of a dataset-specific task instruction (e.g. for MNLI: *"Given the premise, are we justified in saying the hypothesis? yes,*

no, or maybe?") followed by the test instance;

- when $k > 0$, the prompt contains k demonstration examples for each label class, randomly sampled from the dataset training split, followed by the test instance. No explicit task instruction is included in this setting.

For illustration, in a 1-shot setting, an actual RTE prompt seen by the model at test time may be:

> *Premise:* Former Prime Minister Rafik Hariri, also a prominent anti-Syria political figure, was killed in a suicide bombing in February last year, which led to rising anti-Syrian waves and the withdrawal of Syrian troops from Lebanon.
> *Hypothesis:* Syrian troops have been withdrawn from Lebanon after the murder of Rafik Hariri.
> *Answer: yes*
>
> *Premise:* The organizers of the 15th International AIDS Conference, scheduled for next month in Bangkok, Thailand, on Saturday responded to press reports that a prominent hotel in Bangkok discriminated against HIV-positive visitors attending a different conference this month.
> *Hypothesis:* HIV-positive visitors take part in the 15th International AIDS Conference.
> *Answer: no*
>
> *Premise:* Dana Reeve, the widow of the actor Christopher Reeve, has died of lung cancer at age 44, according to the Christopher Reeve Foundation.
> *Hypothesis:* Christopher Reeve had an accident.
> *Answer:*

To investigate the generalizability of the proposed shortcut mitigation framework beyond entailment tasks, additional NLP benchmarks were considered. Specifically, the following datasets were employed:

- **COPA** [Gordon et al., 2012]: a commonsense causal reasoning task in which the model chooses between two plausible alternatives to complete a causal relation with a given premise;
- **CR** [Turney, 2002]: a binary sentiment classification dataset consisting of customer reviews labeled as positive or negative;
- **SST2** [Socher et al., 2013]: the Stanford Sentiment Treebank binary classification task on movie reviews;
- **WiC** [Pilehvar and Camacho-Collados, 2019]: the Word-in-Context benchmark, which tests word sense disambiguation by asking whether a target word has the same meaning in two different sentences;
- **ARC** [Clark et al., 2018]: a multiple-choice question answering dataset for grade-school science questions;
- **MMLU** [Hendrycks et al., 2021]: the Massive Multitask Language Understanding benchmark, a large-scale multiple-choice collection spanning diverse subjects such as mathematics, history, law and computer science.

Evaluating the framework on tasks beyond entailment is also valuable for assessing whether shortcut-related representations learned by LLMs exhibit any degree of cross-task consistency. If similar shortcut directions emerge across heterogeneous datasets, ranging from sentiment analysis to causal reasoning or word-sense disambiguation, this would suggest the presence of shared latent mechanisms through which LLMs internalize and exploit superficial cues. Conversely, a lack of transferability would indicate that shortcut reliance is highly task-specific, reinforcing the need for targeted analyses. Thus, examining multiple NLP tasks provides insights into whether shortcut representations capture generalizable patterns or remain confined to the TER setting from which they were extracted.

Given the exploratory nature of the experiments and the limited computational resources available, the test set sizes were kept modest. Table 3.1 reports the test sets used in this work, along with their source splits and number of examples.

Name	Source	Size
MNLI	subset of MNLI "validation matched" split	500
RTE	RTE validation split	277
COPA	COPA validation split	100
CR	CR test split	376
SST2	SST2 validation split	872
WIC	WiC validation split	638
ARC	ARC test split	1170
MMLU	subset of MMLU test split	2000

Table 3.1. Evaluation test sets used in this work

3.2 Results and Analysis

This section presents the results obtained from the experiments evaluating the RepE-based approach to shortcut detection and mitigation. Shortcut detection (Section 3.2.1) is assessed qualitatively on selected test instances, while shortcut mitigation (Section 3.2.2) is evaluated systematically across multiple hyperparameter configurations on different test sets (Table 3.1). To support reliable tracking and reproducibility, all experiments were logged using the Weights & Biases (W&B) platform, which recorded both input configurations and output metrics. In addition, the full sets of prompts and activation files used in each run were stored as artifacts, ensuring that every experiment can be reproduced precisely.

3.2.1 Shortcut Detection Results

To qualitatively assess the effectiveness of the RepE-based framework for shortcut detection, the following experiment was conducted. Given a RepReader R trained on a random subset of the negation shortcut dataset[3] from the ShortcutSuite and two contrastive test statements $s1$ and $s2$ (exhibiting or non-exhibiting an explicit shortcut cue), scores $A_{l,t}$ are computed to measure, at different layers l and token positions t, the degree to which the model's internal representations of $s1$ and $s2$ align with the shortcut direction identified by R. The scores computation proceeds as follows:

- **Encoding of the input statements.** The sentences $s1$ and $s2$ are fed to the frozen target LLM to obtain hidden representations h_l for each layer l and token position t. Optionally, the model's generated continuations for $s1$ and $s2$ are appended to the input before extracting the hidden representations, allowing shortcut effects to surface in both the prompt and the generation.
- **Projection onto the shortcut direction.** For each layer l and token position t, the hidden representations $h_{l,t}$ are projected onto the shortcut direction V_l inferred by the RepReader R for that layer. The projection is then multiplied by the corresponding sign S_l:

$$A_{l,t} = \langle h_{l,t}, V_l \rangle \cdot S_l$$

- **Averaged projections.** Given a selected set L of hidden layers, the scores $A_{l,t}$ are averaged across all layers in L to obtain a mean shortcut-alignmet score for each token position

$$A_t^{\text{mean}} = \frac{1}{|L|} \sum_{l \in L} A_{l,t}$$

  This aggregation highlights persistent shortcut activation patterns across layers rather than isolated spikes.

Scores $A_{l,t}$ can be visualized as an heatmap across layers and token positions, a representation referred to as a *LAT scan* in the original RepE paper. In the honesty experiments, the authors observed clearly distinguishable scans for honest versus dishonest generations, with the former showing a substantially stronger correlation with the learned "honesty direction". In the shortcut detection setting of the present work, the contrast between the scans of the two test statements is less pronounced. Nonetheless, subtle differences in the LAT scans can still be observed, indicating that the shortcut direction captured by the RepReader is activated to a different extent across the two inputs.

Figures 3.1 and 3.2 show the LAT scans corresponding to $s1 =$

[3] The negation shortcut type was selected because it is easier to visualize, being tied to explicit lexical markers such as "not", and because its effect on the model's predictions is straightforward: an entailment relation is often incorrectly interpreted as non-entailment.

> "[INST] Is the hypothesis entailed by the premise? yes or no. [/INST] Premise: Managing better requires that agencies have, and rely upon, sound financial and program information. Hypothesis: Agencies need sound financial and program information for good management."

and $s2$ =

> "[INST] Is the hypothesis entailed by the premise? yes or no. [/INST] Premise: Managing better requires that agencies have, and rely upon, sound financial and program information. Hypothesis: Agencies need sound financial and program information for good management **and green is not red**."

each augmented with the continuation generated by the model.

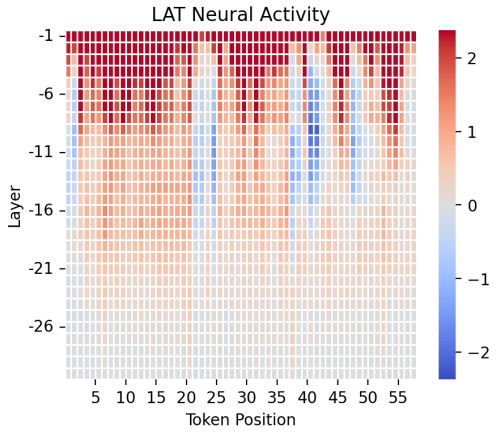


Figure 3.1. LAT scan of a statement not exhibiting an explicit shorcut cue (*clean* scan)

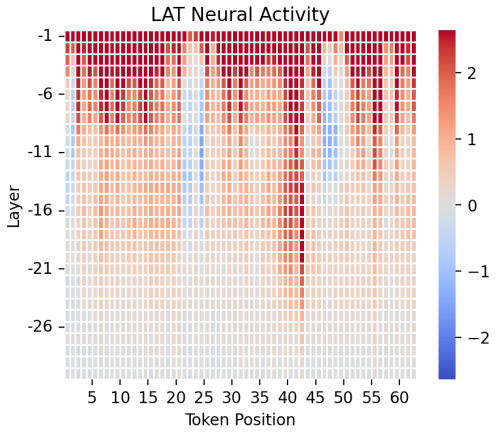


Figure 3.2. LAT scan of a statement exhibiting an explicit shorcut cue (*dirty* scan)

In the deepest hidden layers (approximately layers -10 to -1), both scans display a strong correlation with the shortcut direction (dark red cells). This suggests that these late layers are intrinsically more sensitive to the shortcut feature, independently of the specific input. At token positions 40-45, the clean scan exhibits a negative correlation with the shortcut direction, whereas the dirty scan shows an extended red band, indicating shortcut activation even in earlier layers. Furthermore, at the final token position (which is expected to summarize the representation of the entire input sequence) the clean scan shows shortcut correlation only in the last layer, while the dirty scan displays it across multiple layers.

To gain further insight into the semantic direction captured by the RepReader, the averaged scores A_t^{mean} can be plotted over the corresponding tokens to identify which words correlate most strongly with the learned shortcut direction. Figure 3.3 presents the detection results for the $s2$ statement together with its generated continuation, truncated after a fixed number of tokens. Darker green shading indicates stronger alignment with the shortcut direction. Most highlighted tokens

Figure 3.3. Visualization of token-level projection scores onto the shortcut direction, averaged across all hidden layers

appear as isolated occurrences and correspond to common high-frequency words (such as “that”, “have”, “and”) which are likely attributable to noise. Nevertheless, portions of the shortcut cue “and green is not red” also exhibit clear activation, suggesting that the RepReader partially identifies and responds to the shortcut structure present in the input.

The outcomes of the shortcut detection experiments offer insight into the effectiveness and interpretability of the RepE framework. The presence of consistent shortcut-sensitive activations across multiple layers suggests that such directions are not confined to a small subset of the network. Instead, they appear to be distributed across depth, emerging cumulatively and becoming more pronounced in the later layers. This supports the view that shortcuts in LLMs are not isolated artifacts but reflect broader representational tendencies embedded in the model’s overall geometry. These observations indicate that RepE is a promising, though not yet fully disentangled, interpretability tool for detecting shortcut reliance. Its results highlight both the potential of directional approaches and the inherent complexity of probing model representations through linear interpretive methods.

3.2.2 Shortcut Mitigation Results

To evaluate the effectiveness of the RepE-based framework described in Section 2.2 for mitigating shortcut learning, the Mistral 7B Instruct v0.1 model was evaluated on ICL-based classification tasks both in its base version and under RepControl interventions. Model performance was assessed using standard classification metrics, including accuracy and F1-score.

Hyperparameter Search. A grid search over multiple hyperparameter configurations was conducted to identify effective intervention settings and to analyze the sensitivity of the mitigation performance to different control parameters. Table 3.2 summarizes the hyperparameters and the corresponding values explored in the experiments.

Hyperparameter	**Description**	**Values Tested**
RepReader Type	Method used to extract the shortcut direction from hidden representations	PCA, ClusterMean
Control Operator	RepControl operation applied to hidden states	Linear Combination, Piecewise-linear, Projection
Intervention Layers	Range of transformer layers where control is applied	{from -5 to -17}, {from -8 to -22}, {from -10 to -29}
Representation Token	Token position used for intervention	Last input token
Intervention Strength (α)	Scaling coefficient controlling the magnitude of the intervention	$\{-1.7, -1.5, -1.0, -0.5\}$
Shot Setting (k)	Number of ICL demonstrations per class	$k = 1$
Training shortcut	Type of shortcut cue present in the dataset used to extract the latent direction	Negation, Position, Style
Training dataset size	Number of prompt pairs used to extract clean and dirty hidden representations	{ 64, 128, 256, 512 }
Training prompt selection	Criterion used to select prompts from the ShortcutSuite training set	Random, model-failure-only

Table 3.2. Hyperparameter tested for shortcut mitigation experiments.

The hyperparameter search was conducted on the RTE test set using a Weights&Bias

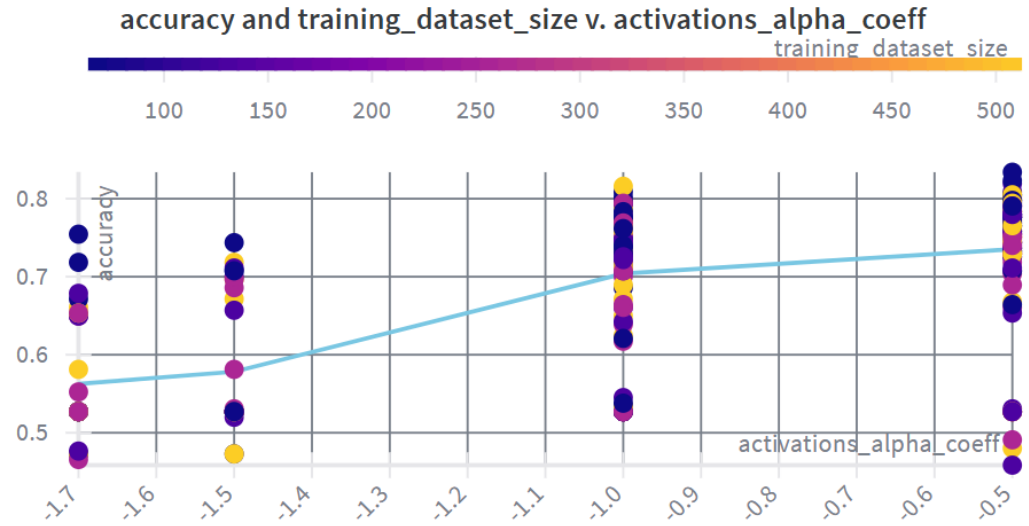


Figure 3.4. Scatter plot of the hyperparameter search runs, generated via Weights & Bias, showing the achieved accuracy, the α coefficient value and the training dataset size for each run. The blue curve represents the running avverage of the accuracy scores.

sweep. Several observations can be drawn from the results. First, as shown in Figure 3.4, stronger interventions ($\alpha = -1.7$ or $\alpha = -1.5$) are associated with lower average accuracy scores, whereas the best results, both in terms of average and peak accuracy, are achieved with a moderate intervention strength ($\alpha = -0.5$). This suggests that excessively strong suppression of the shortcut direction may also remove task-relevant information that is entangled with the shortcut representation, thereby harming overall performance. In contrast, milder interventions appear to strike a better balance by attenuating shortcut reliance while preserving sufficient semantic signal to support correct inference. Regarding the size of the training dataset, no clear monotonic relationship emerges between the number of prompt pairs and model performance. Nevertheless, smaller dataset sizes (64 or 128 prompt pairs) tend to achieve the highest accuracy more frequently across runs. This suggests that even a limited number of randomly sampled contrastive examples may be sufficient to expose dominant shortcut-related signals in the model's representations, potentially reducing the influence of noise that could otherwise degrade performance.

The type of shortcut cue present in the training prompts also has a noticeable impact on model performance under intervention, as shown in Figure 3.5. In particular, the *Negation* and *Position* shortcut types yield comparable results, with average accuracy scores across runs of 62% and 61%, respectively. In contrast, the *Style* shortcut leads to a marked drop in performance, with an average accuracy of 58% and worst-case runs reaching as low as 41%. This suggests that stylistic transformations may induce more diffuse or less linearly separable shortcut representations, making them harder to isolate and suppress through linear RepControl interventions. Unlike lexical or positional cues, stylistic changes tend to affect broader aspects of the input distribution, potentially entangling shortcut-

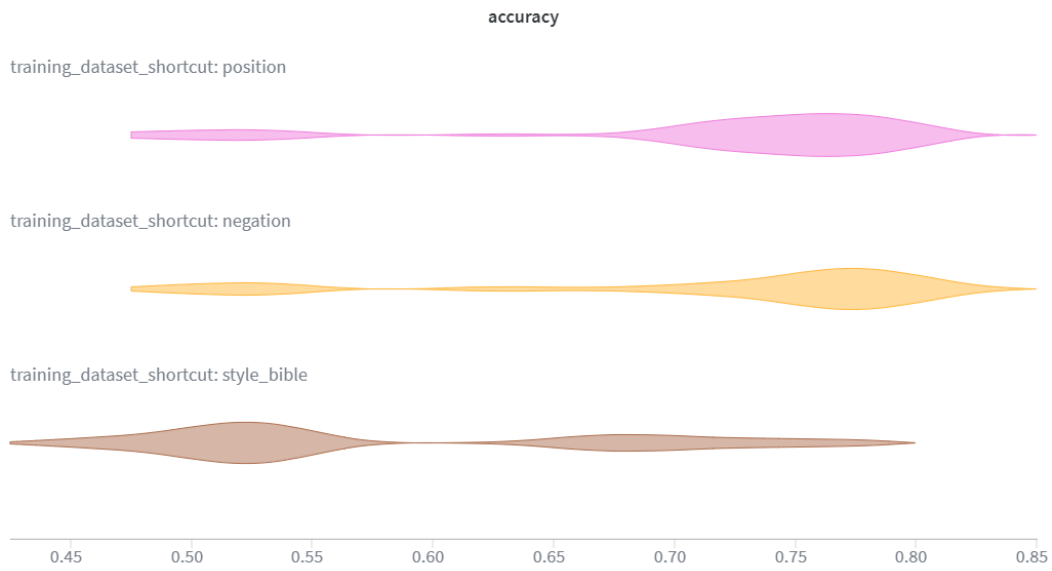


Figure 3.5. Violin plot showing the distribution of accuracy scores across hyperparameter search runs, grouped by the type of shortcut cue used during RepReading.

related signals with task-relevant semantic features and increasing representational overlap. Restricting the training set to prompt pairs for which the presence of a shortcut cue caused the model to misclassify the example did not result in improved post-intervention performance, while significantly increasing the computational cost of data selection. Consequently, random sampling of training prompts was adopted in all subsequent experiments as a more efficient and equally effective strategy.

As shown in Figure 3.6, the performance achieved by different direction extraction methods (RepReader types) and RepControl operators is generally comparable across configurations. However, some consistent trends emerge from the results: PCA-based RepReaders outperform ClusterMean across all operators, and the projection operator achieves the highest overall performance among the tested control strategies. A possible explanation for these observations can be found in the nature of both the representation extraction methods and the control operators. The PCA-based RepReader aim to identify the direction that explain the largest variance across contrastive hidden representations, which, in the shortcut detection setting, may correspond to dominant and consistently activated shortcut-related signals. By aggregating information across many examples without relying on explicit class centroids, PCA is more robust to noise and idiosyncratic variations introduced by individual prompt pairs. In contrast, the ClusterMean approach relies on averaging representations within the clean and shortcut-induced groups, which may be more sensitive to outliers or to residual semantic differences unrelated to the shortcut itself, resulting in less stable directions.

Regarding the control operators, the superior performance of the projection approach can be interpreted as a consequence of its conservative nature. Rather than

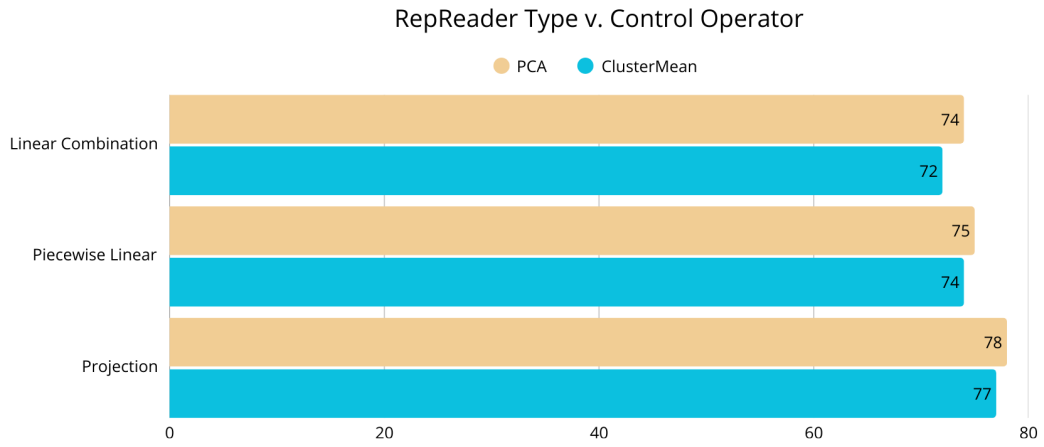


Figure 3.6. Bar plot showing the average accuracy across hyperparameter search runs for each combination of RepReader type and RepControl operator.

adding or subtracting a fixed perturbation to the hidden states, projection explicitly removes the component of the representation aligned with the shortcut direction, leaving the remaining subspace unchanged. This minimizes the risk of distorting task-relevant semantic information that may be partially entangled with shortcut features, a risk that is more pronounced for additive operators such as linear combination or piecewise modulation, especially at higher intervention strengths.

This interpretation is consistent with prior work on linear feature extraction and representation intervention, which shows that dominant properties of neural representations often emerge as high-variance linear directions. At the same time, recent theoretical analyses have demonstrated that modern neural networks, including Large Language Models, encode multiple features in *superposition*, that is, overlapping and non-orthogonal subspaces within the same representational dimensions [Elhage et al., 2022]. Under feature superposition, any linear intervention inevitably affects entangled features. However, projection-based control minimizes this effect by subtracting shortcut-aligned components rather than actively steering representations, resulting in more stable downstream behavior [Elazar et al., 2021].

Lastly, as shown in Figure 3.7, the range of transformer layers at which RepControl is applied has a substantial impact on post-intervention performance. Although all tested layer ranges exhibit considerable variability in accuracy across runs, the range spanning layers from -5 to -17 consistently yields the most reliable results, as indicated by a higher concentration of accuracy scores around 74%. The range from -8 to -22 reaches a peak accuracy comparable to that of the -5 to -17 range; however, it displays significantly higher variance, including the lowest-performing runs observed across all configurations. Finally, the widest range, from -10 to -29, produces the lowest average accuracy overall, suggesting that interventions applied too early across the model may increasingly disrupt task-relevant representations.

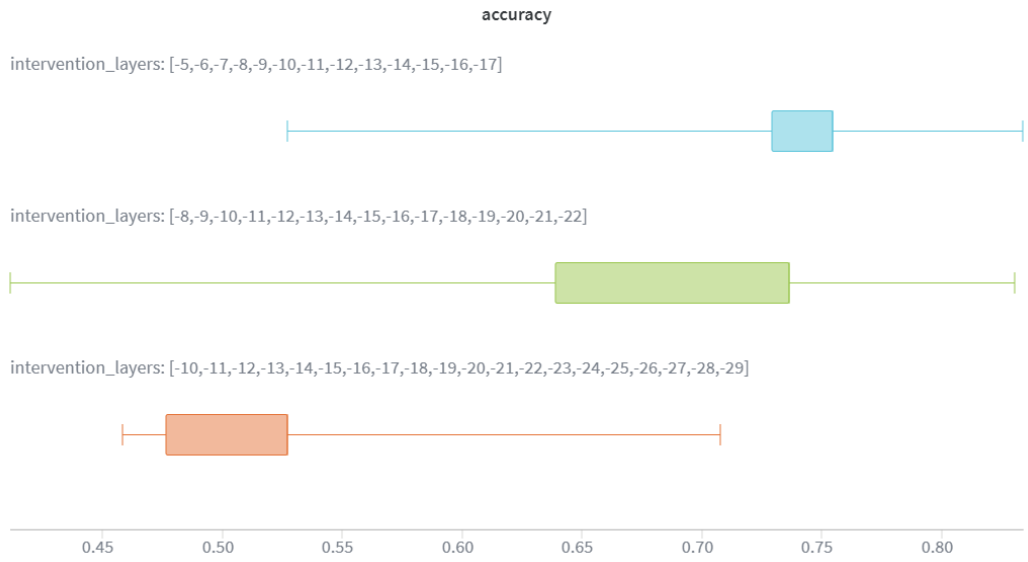


Figure 3.7. Box-and-whisker plot of accuracy scores across hyperparameter search runs, grouped by the range of transformer layers targeted by RepControl interventions.

More generally, a plausible interpretation of these results is that the effectiveness of RepControl is strongly tied to the representational role played by different transformer layers. Prior work has shown that in LLMs higher layers tend to encode increasingly abstract, task-specific and decision-relevant information, while lower layers focus more on lexical, syntactic, and surface-level features [Rogers et al., 2020]. Applying RepControl to a mid-late range of layers (such as -5 to -17) may therefore allow the intervention to directly influence the model's final decision process, suppressing shortcut-aligned activations at the stage where they are most behaviorally relevant. In contrast, extending the intervention to earlier layers, as in the wider -10 to -29 range, may inadvertently disrupt foundational representations that later layers rely on to construct correct semantic interpretations. This effect is exacerbated by the aforementioned phenomenon of feature superposition. Thus, repeatedly removing components aligned with a shortcut direction across many layers can accumulate unintended side effects through the residual stream, progressively degrading useful signal even when conservative operators such as projection are used. Overall, these results indicate that effective shortcut mitigation requires a careful balance between targeting layers close enough to the model's decision circuitry to influence behavior, while avoiding overly broad interventions that risk amplifying the negative effects of representational entanglement.

Evaluation Results. Table 3.3 reports the performance of the base Mistral-7B-Instruct model and compares it with the best accuracy and F1 scores obtained when shortcut mitigation is enabled using PCA-based RepReading combined with different RepControl operators: linear combination, piecewise-linear modulation and projection. Results are presented across the NLP benchmarks test sets described in Section 3.1.2, covering a range of tasks including textual entailment (RTE, MNLI), commonsense reasoning (COPA), sentiment analysis (CR, SST2), word sense dis-

ambiguation (WiC), science question answering (ARC) and broad multi-domain evaluation (MMLU).

	Baseline		**PCA+LinComb**		**PCA+PieceLin**		**PCA+Proj**	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1
RTE	0.787	0.784	0.805	0.800	**0.834**	**0.832**	0.823	0.822
MNLI	0.598	0.488	0.622	0.503	0.594	0.516	**0.644**	**0.538**
COPA	0.820	0.820	**0.880**	**0.879**	0.870	0.867	**0.880**	**0.879**
CR	0.925	0.918	**0.939**	**0.933**	0.931	0.924	0.926	0.918
SST2	0.948	0.948	**0.953**	**0.953**	0.952	0.952	0.950	0.949
WIC	0.497	0.348	0.525	0.450	**0.547**	**0.533**	0.508	0.406
ARC	0.675	0.674	**0.685**	**0.684**	0.682	0.680	0.677	0.676
MMLU	0.485	0.485	0.513	0.513	**0.530**	**0.528**	0.524	0.524
Avg.	0.717	0.683	0.740	0.715	**0.742**	**0.729**	0.742	0.714

Table 3.3. Best accuracy and F1 scores obtained for the base Mistral-7B-Instruct model and three RepControl configurations using the PCA representation reader with the following operators: linear combination (LinComb), piecewise-linear transformation (PieceLin) and projection (Proj).

In particular, the reported results correspond to the best-performing configuration identified during the hyperparameter search. For each benchmark, the highest accuracy and F1 scores were selected among those obtained under different RepControl settings. Specifically, the shortcut direction was extracted from the ShortcutSuite dataset targeting either the *Negation* or *Position* shortcut cue, using training sets of 64 or 128 contrastive prompt pairs. RepControl interventions were applied to transformer layers ranging from -5 to -17, with the intervention strength coefficient set to $\alpha = -0.5$.

Overall, every RepControl configuration yield some improvements over the baseline across all evaluated tasks, indicating that suppressing shortcut-aligned directions extracted via PCA can positively influence model behavior even beyond the task and dataset used for representation extraction. On average, RepControl increases accuracy from 0.717 to approximately 0.74, with gains both on tasks where shortcut reliance is known to be more pronounced or where baseline performance is relatively low (e.g. MNLI, WiC and MMLU) and on tasks with stronger baseline performance such as COPA, where shortcut mitigation leads to marked improvements despite the already high initial accuracy.

A closer inspection reveals that no single operator dominates across all tasks, although some consistent patterns emerge. For textual entailment benchmarks,

PCA combined with piecewise-linear modulation achieves the strongest results on RTE, while PCA with projection performs best on MNLI. This suggests that different entailment datasets may rely on distinct shortcut mechanisms: RTE, being smaller and more lexically constrained, may benefit from conditional amplification or suppression of shortcut-aligned activations, whereas MNLI, with its greater diversity and complexity, appears to favor the more conservative removal of shortcut components through projection, which minimizes unintended interference with task-relevant representations.

For sentiment-related tasks (CR and SST2), RepControl yields smaller yet consistent improvements over the baseline. These tasks already exhibit strong baseline performance, leaving limited room for improvement; nevertheless, the observed gains indicate that shortcut suppression does not interfere with, and may even slightly enhance, robust semantic reasoning when shortcuts overlap with superficial signals such as sentiment-bearing words. In this setting, the linear combination operator performs particularly well, suggesting that mild additive adjustments are sufficient to rebalance representations without removing task-relevant information.

More challenging tasks such as WiC and MMLU show some of the largest relative improvements, especially when using the piecewise-linear operator. These benchmarks require fine-grained semantic distinctions or broad domain knowledge and are therefore more sensitive to spurious correlations and heuristic decision strategies. The effectiveness of piecewise-linear control in this context may stem from its conditional nature, which selectively amplifies or suppresses shortcut-aligned activations depending on their alignment with the current hidden representation, enabling a more adaptive intervention than purely additive or subtractive approaches.

Taken together, these results support the central hypothesis of this work: shortcut-related mechanisms can be captured as latent directions in the representation space of LLMs and selectively manipulated to improve performance across tasks. At the same time, the variability observed across datasets and RepControl operators highlights that different forms of representational control may be required depending on how shortcut cues are encoded and entangled with task-relevant information.

A particularly noteworthy observation is that, although the shortcut direction was extracted using shortcut cues specifically designed for the textual entailment task, its manipulation consistently led to performance improvements across a diverse set of heterogeneous NLP benchmarks. This suggests that the extracted direction does not merely encode task-specific artifacts of textual entailment, but instead captures a more general representational tendency of the model to rely on superficial or spurious correlations when making predictions. A plausible interpretation is that shortcut reliance reflects a shared internal mechanism through which LLMs exploit easily accessible heuristics, such as lexical overlap, stylistic patterns, or positional regularities. Although these shortcuts manifest differently at the input level in tasks such as sentiment analysis, commonsense reasoning, or word sense disambiguation, they may be implemented internally through overlapping representational subspaces that bias the model toward shallow decision strategies. From this

perspective, the shortcut direction extracted in the textual entailment setting can be interpreted as capturing a higher-level inductive bias of the model, rather than a narrow task-specific feature. However, the variability in gains across tasks indicates that shortcut reliance is not fully task-agnostic: different benchmarks may engage this shared mechanism to different extents, or may encode task-relevant information with varying degrees of entanglement with shortcut-aligned features.

Aggregation of Multiple Shortcut Types. To investigate whether shortcut mitigation can be made more robust by accounting for multiple shortcut mechanisms simultaneously, an additional set of experiments was conducted in which shortcut directions extracted from different shortcut types (namely *Negation* and *Position*) were combined into a single intervention signal. The directions were aggregated using a simple normalized sum strategy. For each target layer, all available shortcut directions were first individually ℓ_2-normalized to remove scale effects and prevent any single shortcut from dominating due to magnitude alone. The normalized directions were then summed and the resulting vector was normalized again, yielding a single composite shortcut direction per layer. This procedure preserves the relative orientation of the different shortcut signals while maintaining a bounded norm, making it compatible with the RepControl intervention mechanism. By intervening along this aggregated direction, the model is encouraged to suppress a broader class of shortcut-aligned features simultaneously, rather than targeting a single, narrowly defined heuristic. This experiment therefore serves as a preliminary test of whether shortcut mitigation can generalize from single-cue interventions toward a more holistic control of shortcut-prone representational patterns.

However, the resulting performance on the RTE benchmark was consistently lower than both the baseline model and the best single-shortcut interventions. Specifically, the baseline Mistral-7B-Instruct model achieves an accuracy of 0.787 on RTE, while the best RepControl configurations based on a single shortcut direction reach accuracies above 0.83. In contrast, the aggregated-shortcut interventions yield accuracies in the range of approximately 0.72-0.80, with an average accuracy of 0.746, failing to match the baseline and substantially underperforming compared to single-shortcut control. This degradation suggests that naively combining shortcut directions through a normalized sum does not lead to additive benefits.

A possible explanation is that different shortcut types are not perfectly aligned in representation space. Although they may share high-level inductive biases toward shallow heuristics, their corresponding latent directions can partially conflict or overlap with task-relevant features in distinct ways. Aggregating them into a single control vector may therefore attenuate useful components or suppress information that is necessary for correct entailment judgments, leading to degraded performance rather than improved robustness. These findings thus highlight an important limitation of linear representation-level control: while effective at targeting dominant shortcut mechanisms individually, it does not trivially scale to the joint mitigation of heterogeneous shortcuts. They suggest that more structured or adaptive strategies, such as shortcut-specific control directions, weighted or learned aggregation schemes, or non-linear intervention mechanisms, may be required to

robustly suppress multiple shortcuts behaviors without compromising task performance.

3.2.3 Qualitative Analysis

This section presents additional qualitative experiments aimed at complementing the quantitative evaluation of the proposed shortcut mitigation approach. The goal of these analyses is to assess whether the observed performance improvements are driven by the intended causal mechanisms (namely the suppression of shortcut-related representations) rather than by incidental or fortunate side effects. At the same time, these experiments serve to evaluate the interpretability of the proposed method by examining how representation-level interventions affect model behavior in specific, illustrative cases.

RepControl effect on classification labels. The RepControl intervention designed to mitigate shortcut learning was evaluated on selected input examples in order to qualitatively assess its impact on full model generations under representations steering.

Model behavior under shortcut mitigation

Input:
$[INST]$ Is the hypothesis entailed by the premise? yes or no. $[/INST]$ *Premise*: Fruit, vegetables, electronics, and a little bit of everything else is on sale here. *Hypothesis*: You can buy all sorts of products and food here and green is not red.

Baseline Output:
Answer: No, the hypothesis is not entailed by the premise. The premise does not provide any information about the colors of the products on sale, so it is not possible to conclude that green is not red based on the

RepControl Output:
Yes, the hypothesis is entailed by the premise. The premise mentions that various types of products and food are available for purchase, which supports the hypothesis that one could potentially buy a variety of items at this location.

The responses in the example reported above were obtained using, in the first case, the baseline model, and in the second case a controlled version of the model in which shortcut-aligned activations extracted via PCA from 128 contrastive prompt pairs from the ShortcutSuite dataset targeting the Negation shortcut were suppressed through linear combination with strength coefficient $\alpha = -0.5$, with the intervention applied to transformer layers ranging from -5 to -17. To allow inspection of the model's natural language explanations for its classification decisions while keeping the computational cost of generation limited, the outputs are truncated after the first 50 generated tokens.

This example illustrates that suppressing the shortcut-aligned direction successfully alters the model's decision, leading it to correctly recognize the entailment relation despite the presence of a negation-based distractor in the hypothesis. Notably, the baseline response explicitly refers to the lexical elements associated with the negation shortcut cue (i.e. "green is not red"), whereas the controlled response ignores these superficial signals and completes a logically coherent reasoning process grounded in the semantic content of the premise. This observation further indicates that the RepControl intervention does not impair the model's ability to generate fluent and meaningful natural language, but rather selectively modulates the decision-relevant components of its internal representations.

The following example instead uses a positive strength coefficient $\alpha = +1.5$ in order to examine the effects of amplifying the shortcut-aligned direction rather than suppressing it.

Model behavior under shortcut amplification

Input:
[$INST$] Is the hypothesis entailed by the premise? yes or no. [/$INST$] *Premise*: Children will enjoy the little steam train that loops around the bay to Le Crotoy in the summer. *Hypothesis*: There is a steam train looping around the bay to Le Crotoy and false is not true.

Baseline Output:
Yes, the hypothesis is entailed by the premise.

RepControl Output:
The two statements are not related, so the hypothesis is not entailed by the premise.

In this case, the baseline model correctly predicts entailment despite the presence of a negation-based shortcut cue in the hypothesis. However, when the shortcut-aligned direction is amplified through RepControl, the model's prediction is flipped to an incorrect non-entailment decision. This behavior provides additional evidence that the extracted latent direction indeed corresponds to shortcut-related mechanisms: increasing its activation strength biases the model toward a shortcut-driven interpretation, while suppressing it (as shown in the previous example) promotes more semantically grounded reasoning. Neither response includes an explicit natural language explanation of the classification decision, and under shortcut amplification the model's output no longer strictly follows the expected response format, as it does not begin with an explicit *yes* or *no* token. This deviation suggests that strong positive interventions may interfere not only with the model's decision boundary but also with higher-level generation patterns, further underscoring the importance of carefully tuning the intervention strength α when applying RepControl.

Finally, the following example illustrates a failure case of the proposed shortcut mitigation approach, obtained under the same hyperparameter configuration as the first example, highlighting the limits of the intervention even in settings where it is

generally effective.

Failure case under shortcut mitigation

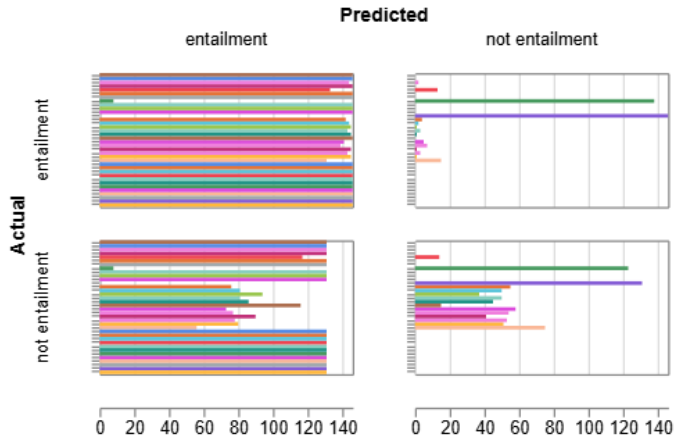
Input:
[$INST$] Is the hypothesis entailed by the premise? yes or no. [/$INST$]
Premise: A group of guys went out for a drink after work, and sitting at the bar was a real a 6 foot blonde with a fabulous face and figure to match.
Hypothesis: A stunning six foot blonde woman sat at the bar with the men after work and no square is a circle.

Baseline Output:
Answer: No, the hypothesis is not entailed by the premise. The premise does not provide any information about the woman's personality or character, and the hypothesis is not related to the premise in any

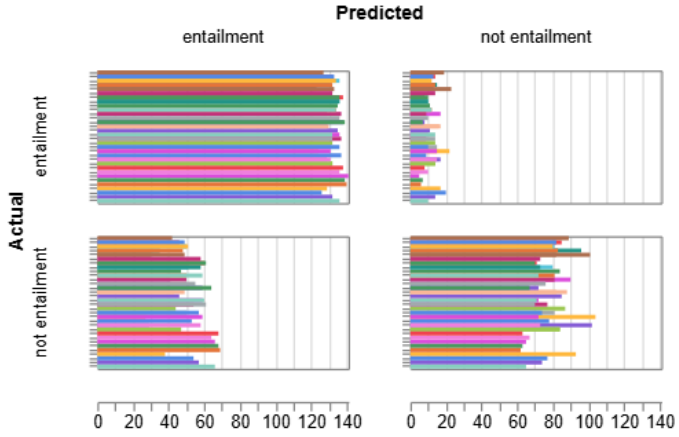
RepControl Output:
Based on the information provided in the premise, it is not clear whether the hypothesis is entailed by the premise. The premise describes a scenario involving a group of men going out for a drink and encountering a beautiful woman at

In this example, the RepControl intervention does not fully succeed in correcting the model's prediction, which remains uncertain, likely due to the presence of a negation-based shortcut cue in the hypothesis. Nevertheless, a qualitative difference can be observed in the controlled generation: compared to the baseline output, the model appears less decisive, producing a more tentative response rather than a confident categorical judgment. This behavior suggests that, although the shortcut-aligned direction was not sufficiently suppressed to fully reverse the decision, the intervention still affects the model's internal confidence. Such partial effects may indicate that, in this instance, shortcut-related features are more strongly entangled with task-relevant semantic information, or that additional shortcut dimensions beyond the single extracted direction contribute to the model's decision process.

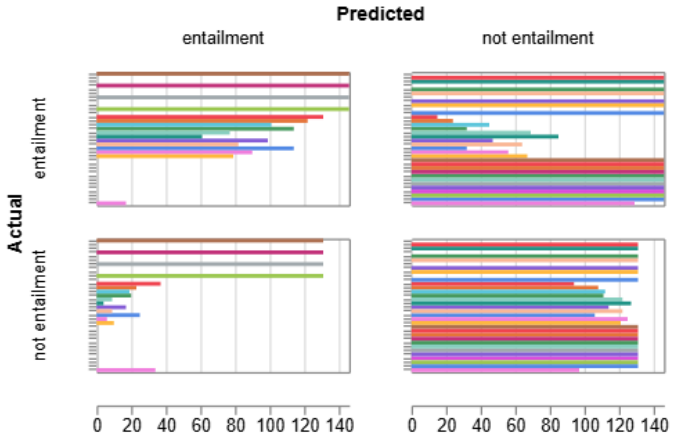
To further characterize the effect of RepControl on classification behavior, multi-run confusion matrices were computed from evaluation runs of the hyperparameter search on the RTE dataset (Figure 3.8). These matrices summarize the distribution of predicted labels under different intervention strengths and provide a global view of how representation-level control reshapes the model's decision patterns. The results reveal a systematic relationship between the intervention coefficient and the model's output distribution. When a large negative coefficient is applied ($\alpha = -1.5$), predictions collapse predominantly toward the *entailment* class, whereas a large positive coefficient ($\alpha = +1.5$) induces the opposite behavior, biasing the model almost entirely toward *contradiction*. This symmetric and monotonic shift provides strong evidence that the extracted shortcut direction is directionally faithful and causally connected to the model's decision boundary, rather than acting as a source of random noise or incidental regularization.



(a) $\alpha = -1.5$



(b) $\alpha = -0.5$



(c) $\alpha = +1.5$

Figure 3.8. Multi-run confusion matrices of the hyperparameter search evaluation runs on the RTE dataset illustrating the effect of RepControl intervention strength (α) on model predictions.

At the same time, these results highlight the importance of carefully tuning the strength of the intervention. While moderate coefficients (e.g. $\alpha = -0.5$) improve classification balance and overall performance, extreme values lead to degenerate decision policies in which the model over-relies on the manipulated direction at the expense of task-relevant semantic information. A potential failure mode of representation-level control is therefore that the extracted direction may act as a trivial label-forcing mechanism, biasing the model toward a fixed class rather than improving semantic reasoning. However, the quantitative results presented in the previous section, obtained using a moderate intervention strength, show consistent improvements in both accuracy and F1 score across multiple datasets and tasks with different label spaces and class distributions. This behavior is difficult to reconcile with a fixed-class bias, since always predicting a dominant class would typically increase accuracy only under severe class imbalance while simultaneously degrading F1, particularly for minority classes. Overall, while extreme interventions demonstrate that RepControl *can* collapse predictions, the moderate settings used for shortcut mitigation yield non-trivial benefits that are more consistent with a selective reduction of shortcut reliance than with naive label forcing.

Token-level attribution via Integrated Gradients. To analyze how individual input tokens contribute to the model's predictions and how these contributions change under RepControl intervention, the *Integrated Gradients* (IG) attribution method was employed. Integrated Gradients is a gradient-based interpretability technique that assigns an importance score to each input feature by integrating the gradients of the model's output with respect to the input along a straight-line path between a baseline input (i.e. a reference point with minimal semantic content) and the actual input [Sundararajan et al., 2017].

Formally, given a model $F(\cdot)$, an input embedding $x \in \mathbb{R}^{T \times d}$ (with T tokens and embedding dimension d) and a baseline embedding x', the Integrated Gradients attribution for the i-th input dimension is defined as:

$$\mathrm{IG}_i(x) = (x_i - x_i') \cdot \int_{\beta=0}^{1} \frac{\partial F(x' + \beta(x - x'))}{\partial x_i} \, d\beta \tag{3.1}$$

In practice, the integral is approximated via a Riemann sum over m discrete interpolation steps:

$$\mathrm{IG}_i(x) \approx (x_i - x_i') \cdot \frac{1}{m} \sum_{k=1}^{m} \frac{\partial F\left(x' + \frac{k}{m}(x - x')\right)}{\partial x_i} \tag{3.2}$$

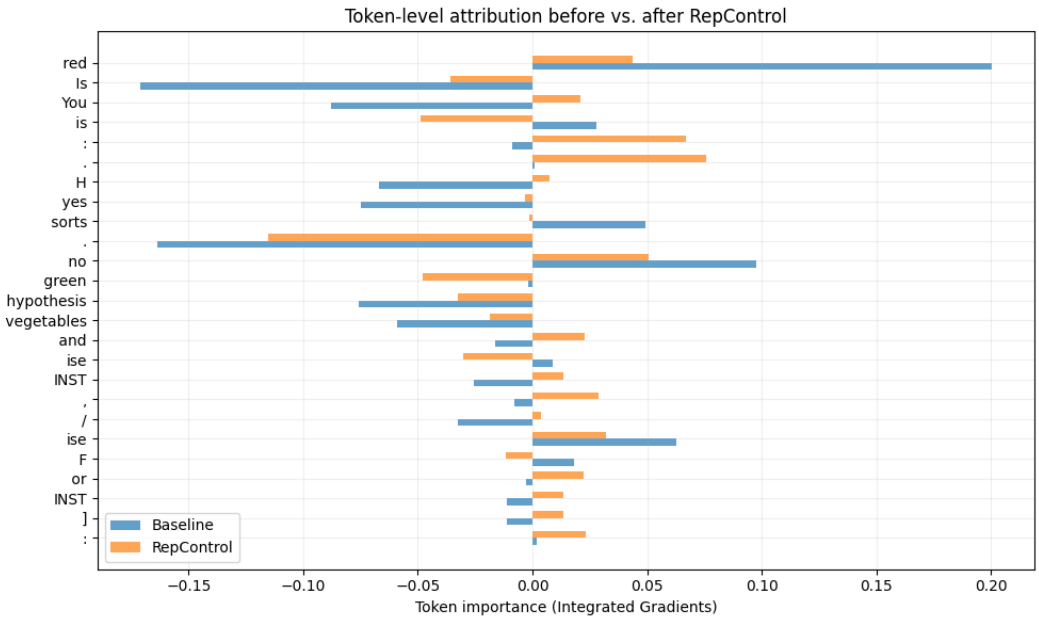
In this work, the baseline x' is chosen as the embedding of the beginning-of-sequence token, which provides a stable and semantically neutral reference compared to an all-zero embedding. At each interpolation step, gradients are computed with respect to the embedding vectors, and the resulting attributions are summed across the embedding dimension to obtain a scalar importance score for each token:

$$\mathrm{IG}_{\mathrm{token}}(t) = \sum_{j=1}^{d} \mathrm{IG}_{t,j} \tag{3.3}$$

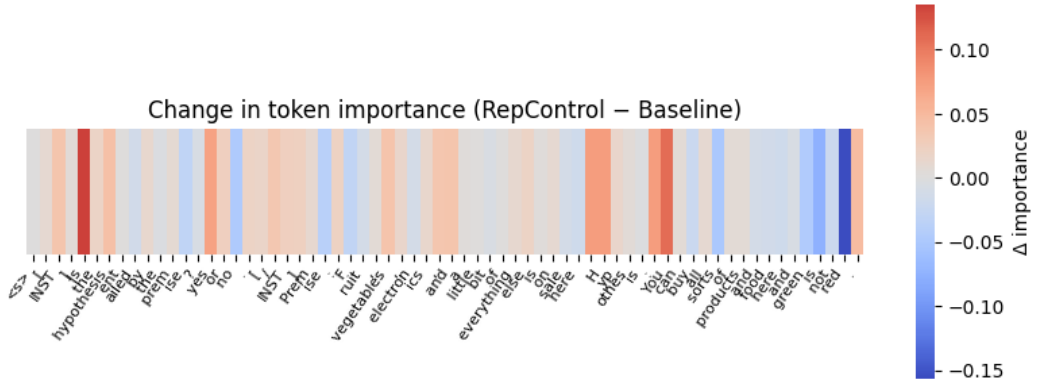
Specifically, attributions are calculated for the logit corresponding to a chosen class (e.g. *entailment*), allowing the analysis to focus on which tokens contribute to supporting that decision. This choice is particularly relevant in the context of shortcut learning, where spurious cues are expected to disproportionately influence the confidence of specific labels. By keeping the target label fixed before and after RepControl intervention, changes in token attributions can be directly interpreted as shifts in the internal evidence used by the model. This enables a fine-grained assessment of whether shortcut mitigation suppresses shortcut-aligned token importance while preserving semantically meaningful contributions.

Figure 3.9 reports the token-level attributions computed via Integrated Gradients for the *entailment* class on the first qualitative example discussed in the previous section, comparing the baseline model with its RepControl-modified counterpart. In the baseline setting, a large fraction of the attribution mass is assigned to shortcut-related lexical cues, most notably the token *"red"*, which appears in the tautological negation phrase appended to the hypothesis. This indicates that the model relies heavily on the presence of shortcut-related tokens when supporting the entailment decision, rather than grounding its prediction in the semantic overlap between premise and hypothesis. After applying RepControl with a negative intervention strength, the importance of shortcut-aligned tokens is substantially reduced. In particular, the attribution associated with *"red"* and with the negation-related token *"no"* decreases markedly, while the relative contribution of semantically meaningful tokens such as *"vegetables"* and *"products"* slightly increases, even though their overall attribution values remain negative. This redistribution suggests that suppressing the shortcut-aligned latent direction encourages the model to rely less on superficial lexical cues and more on content-bearing elements of the input when forming its decision. At the same time, the attribution maps reveal non-negligible noise introduced by the intervention. Some punctuation symbols (e.g. *":"*, *"."*) receive increased attribution despite carrying no semantic relevance for the entailment relation, and the negative contribution associated with the token *"green"* is further amplified rather than fully suppressed.

While these results provide evidence that RepControl can reduce the influence of dominant shortcut cues, they also highlight several limitations of the proposed approach. First, the redistribution of attribution mass is not perfectly aligned with semantic relevance: some tokens that are logically irrelevant to the entailment relation, such as punctuation symbols, receive increased importance after intervention, indicating that RepControl may introduce secondary artifacts in the attribution patterns. Moreover, the fact that certain shortcut-related tokens retain or even increase their influence suggests that shortcut reliance is not fully captured by a single linear direction. Instead, shortcut mechanisms are likely distributed across multiple entangled features in the representation space, making them only partially suppressible through linear interventions. As a result, RepControl appears to attenuate the dominant shortcut signal without completely disentangling it from task-relevant semantic representations. More generally, these observations underline that, while representation-level control based on linear directions is effective at modulating coarse-grained behavioral tendencies, it provides only an approximate



(a) Top 25 tokens ranked by the absolute change in Integrated Gradients attribution between the baseline model and the RepControl-intervened model.



(b) Heatmap of token-level attribution differences (RepControl minus baseline), highlighting how RepControl redistributes importance across the input sequence.

Figure 3.9. Visualization of the token-level attributions computed via Integrated Gradients for the class *entailment* with input "[$INST$] Is the hypothesis entailed by the premise? yes or no. [/$INST$] *Premise*: Fruit, vegetables, electronics, and a little bit of everything else is on sale here. *Hypothesis*: You can buy all sorts of products and food here and green is not red."

handle on the complex and highly superposed internal representations of LLMs.

Non-Linearity Analysis. To better understand the observed limitations of the proposed linear shortcut mitigation framework, we analyze the *intrinsic dimensionality* (ID) of the model's hidden representations across different transformer layers. Intrinsic dimensionality refers to the minimum number of degrees of freedom required to describe the data locally, that is, the dimension of the underlying manifold on which the representations lie, independently of the ambient embedding space dimension. If shortcut-related representations are organized along highly curved or non-linear manifolds, linear methods such as PCA may provide only an approximate characterization, potentially limiting the effectiveness of extracting a single shortcut direction.

To this end, the intrinsic dimensionality is estimated using both a linear and a non-linear approach. In the linear case, ID is approximated as the number of principal components required to explain 95% of the total variance of the representations. This measure captures how many orthogonal linear directions are needed to reconstruct the data, but it implicitly assumes that the underlying structure is approximately linear. In contrast, a non-linear estimate of intrinsic dimensionality is obtained using the Two Nearest Neighbors (TwoNN) method [Facco et al., 2017]. TwoNN is based on minimal neighborhood information: for each data point, it considers the ratio between the distances to its first and second nearest neighbors. Under mild assumptions, the distribution of these ratios depends only on the intrinsic dimensionality of the data manifold, allowing ID to be estimated without assuming linearity or constructing a global embedding of the data. As a result, TwoNN is well-suited to detect low-dimensional but curved structures that may not be well captured by PCA.

By comparing the linear (PCA-based) and non-linear (TwoNN-based) intrinsic dimensionality estimates, we assess whether hidden representations at different layers are well approximated by locally linear subspaces or instead exhibit significant non-linear structure. Large discrepancies between the two estimates indicate layers where the representation geometry is highly non-linear, suggesting that shortcut-related features may be distributed across curved manifolds rather than concentrated along a single linear direction. The results reported in Table 3.4 are obtained from the hidden representations produced by the Mistral 7B Instruct model for 128 *clean* and *dirty* prompts derived from the ShortcutSuite dataset targeting the Negation shortcut type.

Across all examined layers, both clean and dirty representations exhibit a large gap between the dimensionality required to explain 95% of the variance via PCA and the intrinsic dimensionality estimated by the non-linear TwoNN method. In particular, PCA dimensionality ranges from approximately 55 in the final layer to over 100 in earlier layers, while TwoNN consistently estimates a much lower intrinsic dimensionality, typically between 15 and 30. The resulting PCA/TwoNN ratios, highlighted in the heatmap, indicate that shortcut-related representations lie on low-dimensional but strongly curved manifolds, rather than on approximately

Layer	Clean Representations				Dirty Representations			
	PCA	TwoNN	PC1 var	PCA/TwoNN	PCA	TwoNN	PC1 var	PCA/TwoNN
-1	55	16.47	0.191	3.34	58	14.95	0.182	3.88
-5	89	19.79	0.096	4.50	89	19.39	0.099	4.59
-10	96	20.74	0.133	4.63	97	24.85	0.133	3.90
-15	98	20.22	0.122	4.85	99	25.30	0.117	3.91
-20	97	20.73	0.103	4.68	97	18.65	0.101	5.20
-25	99	22.49	0.102	4.40	100	19.95	0.100	5.01
-31	102	28.69	0.102	3.56	103	25.80	0.101	3.99

Table 3.4. Intrinsic dimensionality estimates for clean and dirty representations across transformer layers, computed using linear (*PCA*) and non-linear (*TwoNN*) methods. *PC1 var* denotes the fraction of variance explained by the first principal component. The heatmap encodes the ratio between linear and non-linear intrinsic dimensionality (*PCA/TwoNN*), highlighting the degree of curvature of the underlying representation manifold.

linear subspaces. An additional informative signal is provided by the variance explained by the first principal component, which accounts for only a limited fraction of the total variance, with PC1 values ranging roughly between 10% and 19%. This suggests that, although shortcut-related representations are intrinsically low-dimensional, they are not dominated by a single linear direction. Instead, the variance is distributed across multiple correlated components, consistent with the presence of feature superposition and non-linear structure in the residual stream. This observation explains why PCA-based extraction of a shortcut direction can be effective in aggregate, by capturing a prominent axis of variation, while still failing to fully isolate shortcut mechanisms at the level of individual examples.

Beyond the overall gap between linear and non-linear dimensionality estimates, Table 3.4 reveals a systematic difference in how geometric complexity evolves across layers for clean versus dirty representations. For clean representations, the PCA / TwoNN ratio exhibits a relatively smooth and structured trend: non-linearity increases from the early layers toward the mid layers (approximately -31 to -15), before decreasing again in late layers. This pattern suggests a staged representational process in which shortcut-free inputs are progressively transformed into increasingly abstract but structured manifolds, before being consolidated into more stable configurations closer to the input. Such behavior is consistent with prior findings that intermediate layers in transformer models encode richer and more compositional abstractions, while both early and late layers tend to exhibit more constrained geometries. In contrast, dirty representations display a markedly less regular behavior. While their average intrinsic dimensionality remains comparable to that of clean representations, the PCA/TwoNN ratio fluctuates substantially across layers, with pronounced peaks in curvature at specific depths (e.g. around layers -20 and -25). These localized spikes may indicate that shortcut cues induce irregular distortions of the representational manifold, increasing its curvature

in a layer-dependent and input-dependent manner rather than following a smooth transformation trajectory. This "chaotic" geometric behavior suggests that shortcut learning does not merely strengthen an existing low-dimensional feature, but instead perturbs the internal representation in ways that are less globally organized and more entangled with other semantic features, biasing the model's traversal of an underlying representational manifold, rather than by activating isolated, linearly separable features.

From the perspective of shortcut mitigation, this distinction is critical. Linear representation control methods such as PCA-based RepControl are naturally better suited to structured, approximately linear manifolds, as observed for clean representations and for certain layer ranges in dirty ones. However, the presence of highly curved, layer-specific distortions in shortcut-contaminated representations limits the effectiveness of a single global linear direction, helping to explain both the variability in mitigation performance across layers and the existence of failure cases observed in the qualitative analysis. These findings reinforce the interpretation that shortcut learning introduces localized non-linear deformations in the model's internal geometry, motivating future work on layer-adaptive or non-linear representation control mechanisms.

Conclusion

This thesis investigated the problem of shortcut learning in Large Language Models (LLMs) under the In-Context Learning (ICL) paradigm and proposed a training-free, interpretable framework for detecting and mitigating shortcut reliance directly at the level of internal representations. Shortcut learning poses a significant challenge to the robustness, generalization, and interpretability of modern LLMs, as it leads models to exploit superficial or spurious correlations in the input rather than engaging in semantically grounded reasoning. Addressing this phenomenon is particularly important in ICL settings, where models must infer task structure from a small number of examples and are therefore especially susceptible to shortcut cues present in prompts.

The core contribution of this work is the adaptation of the Representation Engineering (RepE) framework to the problem of shortcut detection and mitigation. By leveraging contrastive prompt pairs that differ only in the presence of a shortcut cue, the proposed approach extracts latent shortcut-aligned directions from the model's hidden representations using simple linear methods such as PCA and ClusterMean. These directions are then manipulated at inference time through RepControl interventions (linear combination, piecewise-linear modulation and projection) allowing shortcut-related activations to be selectively amplified or suppressed without modifying model weights or requiring additional training.

Extensive quantitative experiments demonstrated that this approach is effective in mitigating shortcut-driven behavior. Across a diverse set of NLP benchmarks including textual entailment, commonsense reasoning, sentiment analysis, word sense disambiguation and multi-domain question answering, RepControl consistently improved performance over the baseline model. Notably, these improvements were observed even if the shortcut direction was extracted exclusively from textual entailment data, suggesting that shortcut reliance may reflect a more general inductive bias shared across tasks rather than a purely task-specific artifact. The results further showed that no single control operator universally dominates, highlighting the task-dependent nature of shortcut mitigation and the importance of carefully selecting both the intervention strength and the layers at which it is applied.

At the same time, experiments combining multiple shortcut directions through a simple normalized aggregation strategy showed that naive joint mitigation can degrade accuracy, performing worse than both the baseline model and single-shortcut interventions. This suggests that different shortcut types, while related, are not

trivially composable in representation space.

Qualitative analyses complemented the quantitative findings and provided additional interpretability insights. Controlled generations revealed that suppressing shortcut-aligned directions can alter model decisions in the expected way, while amplifying them can reliably bias predictions toward shortcut-driven outcomes. Token-level attribution analyses based on Integrated Gradients further showed that RepControl reduces the importance assigned to shortcut-related lexical cues and partially redistributes attention toward more semantically meaningful tokens. At the same time, these analyses exposed limitations of the approach, including residual noise in attribution patterns and incomplete suppression of shortcut effects, underscoring the entangled and highly superposed nature of LLM representations.

To better understand these limitations, the thesis also investigated the intrinsic dimensionality of shortcut-related representations across layers using both linear (PCA-based) and non-linear (TwoNN) estimators. The results revealed a substantial gap between linear and non-linear dimensionality estimates, indicating that shortcut-prone representations often lie on curved, low-dimensional manifolds embedded within high-dimensional spaces. This finding provides a principled explanation for why linear methods such as PCA are effective but imperfect: while they can capture dominant shortcut directions, they cannot fully disentangle more complex, non-linear shortcut mechanisms.

Overall, this work demonstrates that shortcut learning in LLMs can be meaningfully analyzed and mitigated through representation-level interventions, offering a lightweight and interpretable alternative to retraining-based approaches. At the same time, it exposes important open challenges. Shortcut mechanisms appear to be distributed, partially non-linear and entangled with task-relevant features, limiting the effectiveness of single-direction linear control. Future research directions include the development of non-linear or multi-directional representation controls, adaptive or learned aggregation strategies for multiple shortcuts and automated methods for constructing high-quality contrastive datasets. More broadly, the results support the view that understanding and steering the internal representations of LLMs is a promising path toward more robust, transparent and trustworthy language models.

Bibliography

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models, 2023. URL https://arxiv.org/abs/2211.15661.

Yara Alharahseheh, Rasha Obeidat, Mahmoud Al-Ayoub, and Maram Gharaibeh. A survey on textual entailment: Benchmarks, approaches and applications, 2022.

Amos Azaria and Tom Mitchell. The internal state of an llm knows when it's lying, 2023. URL https://arxiv.org/abs/2304.13734.

David Barack and John Krakauer. Two views on the cognitive brain. *Nature Reviews Neuroscience*, 22, 04 2021. doi: 10.1038/s41583-021-00448-6.

Sara Beery, Grant van Horn, and Pietro Perona. Recognition in terra incognita, 2018. URL https://arxiv.org/abs/1807.04975.

Emily Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? pages 610–623, 03 2021. doi: 10.1145/3442188.3445922.

Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings, 2016. URL https://arxiv.org/abs/1607.06520.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

Seungtaek Choi, Myeongho Jeong, Hojae Han, and Seung-won Hwang. C2l: Causally contrastive learning for robust text classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36:10526–10534, 06 2022. doi: 10.1609/aaai.v36i10.21296.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases, 2019. URL https://arxiv.org/abs/1909.03683.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge, 2018. URL https://arxiv.org/abs/1803.05457.

Ido Dagan, Oren Glickman, and Bernardo Magnini. The pascal recognising textual entailment challenge. In *Machine Learning Challenges Workshop*, 2006.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. A survey on in-context learning, 2024. URL https://arxiv.org/abs/2301.00234.

Mengnan Du, Varun Manjunatha, Rajiv Jain, Ruchi Deshpande, Franck Dernoncourt, Jiuxiang Gu, Tong Sun, and Xia Hu. Towards interpreting and mitigating shortcut learning behavior of nlu models, 2021. URL https://arxiv.org/abs/2103.06922.

Mengnan Du, Fengxiang He, Na Zou, Dacheng Tao, and Xia Hu. Shortcut learning of large language models in natural language understanding, 2023. URL https://arxiv.org/abs/2208.11857.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175, 2021. doi: 10.1162/tacl_a_00359. URL https://aclanthology.org/2021.tacl-1.10/.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition, 2022. URL https://arxiv.org/abs/2209.10652.

Elena Facco, Maria d'Errico, Alex Rodriguez, and Alessandro Laio. Estimating the intrinsic dimension of datasets by a minimal neighborhood information. *Scientific Reports*, 7(1), September 2017. ISSN 2045-2322. doi: 10.1038/s41598-017-11873-y. URL http://dx.doi.org/10.1038/s41598-017-11873-y.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, November 2020. ISSN 2522-5839. doi: 10.1038/s42256-020-00257-z. URL http://dx.doi.org/10.1038/s42256-020-00257-z.

Andrew Gordon, Zornitsa Kozareva, and Melissa Roemmele. SemEval-2012 task 7: Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In Eneko Agirre, Johan Bos, Mona Diab, Suresh Manandhar, Yuval Marton, and Deniz Yuret, editors, **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop*

on Semantic Evaluation (SemEval 2012), pages 394–398, Montréal, Canada, 7-8 June 2012. Association for Computational Linguistics. URL https://aclanthology.org/S12-1052/.

Xiaochuang Han, Byron C. Wallace, and Yulia Tsvetkov. Explaining black box predictions and unveiling data artifacts through influence functions, 2020. URL https://arxiv.org/abs/2005.06676.

Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. Pretrained transformers improve out-of-distribution robustness, 2020. URL https://arxiv.org/abs/2004.06100.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding, 2021. URL https://arxiv.org/abs/2009.03300.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is bert really robust? a strong baseline for natural language attack on text classification and entailment, 2020. URL https://arxiv.org/abs/1907.11932.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020. URL https://arxiv.org/abs/2001.08361.

Chen-An Li and Hung-Yi Lee. Examining forgetting in continual pre-training of aligned large language models, 2024. URL https://arxiv.org/abs/2401.03129.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for gpt-3?, 2021. URL https://arxiv.org/abs/2101.06804.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity, 2022. URL https://arxiv.org/abs/2104.08786.

Aristides Milios, Siva Reddy, and Dzmitry Bahdanau. In-context learning for text classification with many labels, 2023. URL https://arxiv.org/abs/2309.10954.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work?, 2022. URL https://arxiv.org/abs/2202.12837.

Dr. Tom Murphy. The first level of super mario bros . is easy with lexicographic orderings and time travel. 2013. URL https://api.semanticscholar.org/CorpusID:14347703.

Timothy Niven and Hung-Yu Kao. Probing neural network comprehension of natural language arguments. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4658–4664, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1459. URL https://aclanthology.org/P19-1459/.

Jane Pan, Tianyu Gao, Howard Chen, and Danqi Chen. What in-context learning "learns" in-context: Disentangling task recognition and task learning, 2023. URL https://arxiv.org/abs/2305.09731.

Oskar. Pfungst and Carl Leo. Rahn. *Clever Hans (the horse of Mr. Von Osten) a contribution to experimental animal and human psychology.* New York, H. Holt and company, 1911, 1911. URL https://www.biodiversitylibrary.org/item/116908. https://www.biodiversitylibrary.org/bibliography/56164 — "Table of references": p. 267-274. — Supplements i-iv: p. 245-265.

Thang M. Pham, Trung Bui, Long Mai, and Anh Nguyen. Out of order: How important is the sequential order of words in a sentence in natural language understanding tasks?, 2021. URL https://arxiv.org/abs/2012.15180.

Mohammad Taher Pilehvar and Jose Camacho-Collados. WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *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)*, pages 1267–1273, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1128. URL https://aclanthology.org/N19-1128/.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866, 2020. doi: 10.1162/tacl_a_00349. URL https://aclanthology.org/2020.tacl-1.54/.

Preethi Seshadri, Hongyu Chen, Sameer Singh, and Seraphina Goldfarb-Tarrant. Small changes, large consequences: Analyzing the allocational fairness of llms in hiring contexts, 2025. URL https://arxiv.org/abs/2501.04316.

Yuge Shi, Jeffrey Seely, Philip H. S. Torr, N. Siddharth, Awni Hannun, Nicolas Usunier, and Gabriel Synnaeve. Gradient matching for domain generalization, 2021. URL https://arxiv.org/abs/2104.09937.

Seongjin Shin, Sang-Woo Lee, Hwijeen Ahn, Sungdong Kim, HyoungSeok Kim, Boseop Kim, Kyunghyun Cho, Gichang Lee, Woomyoung Park, Jung-Woo Ha, and Nako Sung. On the effect of pretraining corpora on in-context learning by a large-scale language model. In Marine Carpuat, Marie-Catherine de Marneffe,

and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5168–5186, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.380. URL https://aclanthology.org/2022.naacl-main.380/.

Chenglei Si, Dan Friedman, Nitish Joshi, Shi Feng, Danqi Chen, and He He. Measuring inductive biases of in-context learning with underspecified demonstrations, 2023. URL https://arxiv.org/abs/2305.13299.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In David Yarowsky, Timothy Baldwin, Anna Korhonen, Karen Livescu, and Steven Bethard, editors, *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://aclanthology.org/D13-1170/.

Rui Song, Yingji Li, Lida Shi, Fausto Giunchiglia, and Hao Xu. Shortcut learning in in-context learning: A survey, 2024. URL https://arxiv.org/abs/2411.02018.

Joe Stacey, Pasquale Minervini, Haim Dubossarsky, Sebastian Riedel, and Tim Rocktäschel. Avoiding the Hypothesis-Only Bias in Natural Language Inference via Ensemble Adversarial Training. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8281–8291, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.665. URL https://aclanthology.org/2020.emnlp-main.665/.

Joe Stacey, Yonatan Belinkov, and Marek Rei. Supervising model attention with human explanations for robust natural language inference, 2022. URL https://arxiv.org/abs/2104.08142.

David Steinmann, Felix Divo, Maurice Kraus, Antonia Wüst, Lukas Struppek, Felix Friedrich, and Kristian Kersting. Navigating shortcuts, spurious correlations, and confounders: From origins via detection to mitigation, 2024. URL https://arxiv.org/abs/2412.05152.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks, 2017. URL https://arxiv.org/abs/1703.01365.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. Bert rediscovers the classical nlp pipeline, 2019. URL https://arxiv.org/abs/1905.05950.

Peter D. Turney. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews, 2002. URL https://arxiv.org/abs/cs/0212032.

Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. Towards debiasing NLU models from unknown biases. In Bonnie Webber, Trevor Cohn,

Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7597–7610, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.613. URL https://aclanthology.org/2020.emnlp-main.613/.

Guillermo Valle-Pérez, Chico Q. Camargo, and Ard A. Louis. Deep learning generalizes because the parameter-function map is biased towards simple functions, 2019. URL https://arxiv.org/abs/1805.08522.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent, 2023. URL https://arxiv.org/abs/2212.07677.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. Universal adversarial triggers for attacking and analyzing nlp, 2021. URL https://arxiv.org/abs/1908.07125.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models, 2022. URL https://arxiv.org/abs/2206.07682.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference, 2018. URL https://arxiv.org/abs/1704.05426.

Linyi Yang, Shuibai Zhang, Libo Qin, Yafu Li, Yidong Wang, Hanmeng Liu, Jindong Wang, Xing Xie, and Yue Zhang. Glue-x: Evaluating natural language understanding models from an out-of-distribution generalization perspective, 2023. URL https://arxiv.org/abs/2211.08073.

Yu Yuan, Lili Zhao, Kai Zhang, Guangting Zheng, and Qi Liu. Do llms overcome shortcut learning? an evaluation of shortcut challenges in large language models, 2024. URL https://arxiv.org/abs/2410.13343.

John R. Zech, Marcus A. Badgeley, Manway Liu, Anthony Beardsworth Costa, Joseph J. Titano, and Eric Karl Oermann. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study. *PLoS Medicine*, 15, 2018. URL https://api.semanticscholar.org/CorpusID:49558635.

Yufeng Zhang, Fengzhuo Zhang, Zhuoran Yang, and Zhaoran Wang. What and how does in-context learning learn? bayesian model averaging, parameterization, and generalization, 2023. URL https://arxiv.org/abs/2305.19420.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. Men also like shopping: Reducing gender bias amplification using corpus-level constraints, 2017. URL https://arxiv.org/abs/1707.09457.

Hanzhang Zhou, Zijian Feng, Zixiao Zhu, Junlang Qian, and Kezhi Mao. Unibias: Unveiling and mitigating llm bias through internal attention and ffn manipulation, 2024. URL https://arxiv.org/abs/2405.20612.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation engineering: A top-down approach to ai transparency, 2025. URL https://arxiv.org/abs/2310.01405.