CONTROLLABLE CONTEXT SENSITIVITY AND THE KNOB BEHIND IT

Anonymous authors
Paper under double-blind review

ABSTRACT

When making predictions, a language model must trade off how much it relies on its context vs. its prior knowledge. Choosing how sensitive the model is to its context is a fundamental functionality, as it enables the model to excel at tasks like retrieval-augmented generation and question-answering. In this paper, we search for a knob which controls this sensitivity, determining whether language models answer from the context or their prior knowledge. To guide this search, we design a task for **controllable context sensitivity**. In this task, we first feed the model a context (*Paris is in England*) and a question (*Where is Paris?*); we then instruct the model to either use its prior or contextual knowledge and evaluate whether it generates the correct answer for both intents (either *France* or *England*). When fine-tuned on this task, instruct versions of Llama-3.1, Mistral-v0.3, and Gemma-2 can solve it with high accuracy (85-95%). Analyzing these high-performing models, we narrow down which layers may be important to context sensitivity using a novel linear time algorithm. Then, in each model, we identify a 1-D subspace in a single layer that encodes whether the model follows context or prior knowledge. Interestingly, while we identify this subspace in a fine-tuned model, we find that the exact same subspace serves as an effective knob in not only that model but also non-fine-tuned instruct and base models of that model family. Finally, we show a strong correlation between a model's performance and how distinctly it separates context-agreeing from context-ignoring answers in this subspace. These results suggest a single fundamental subspace facilitates how the model chooses between context and prior knowledge.

1 INTRODUCTION

Language models are often prompted with a query and preceding context, e.g., in settings of in-context learning, retrieval-augmented generation, or document analysis. In such scenarios, the language model needs to integrate information from both the context and its prior knowledge stored in its parameters. In some cases, we may prefer the model to rely more on the context, e.g., to avoid hallucinating responses based on outdated prior knowledge (Zhang et al., 2023); however, in other cases, we may prefer the model to rely more on its prior knowledge, e.g., to avoid being misled by misinformation provided in the context (Hong et al., 2024). As a motivating example, consider a document analysis setting in which a language model is asked to help understand an opinion article in a newspaper. It might first be asked to summarize, e.g., *What is the main argument of this article?*. In this case, the model should rely heavily on the context, i.e., the text of the article. Then, one might ask: *What are some criticisms of this argument?*. To answer this critically, the model ought to be skeptical; an opinion article may be written very authoritatively as if its arguments are established fact, or it may make some misleading claims to support its argument. Thus, the language model should draw more upon its prior knowledge of the issue and related opinions than blindly following the context. More broadly, because the degree of context sensitivity depends highly on the use case, it would be desirable to be able to specify how much and whether the model should be influenced by the context versus its prior knowledge.

Studies on the tension between context and prior knowledge have primarily focused on the setting of *knowledge conflicts* (Longpre et al., 2021), in which a given context directly contradicts information assumed to be in a model's prior knowledge about a given query. For example, a language model trained on a sufficient amount of data should be able to reply to the query *What's the capital of France?*

[*] These authors contributed equally to this work.

with *Paris*. However, if the context *The capital of France is London.* is prepended to the query, the model needs to decide whether to respond based on the context (*London*) or its prior knowledge (*Paris*). Prior studies (Longpre et al., 2021; Li et al., 2023b; Du et al., 2024; Monea et al., 2024; Ortu et al., 2024; Xie et al., 2023; Basmov et al., 2024) have shown that models will prefer drawing from context for some questions and prior knowledge from others. To investigate mechanisms underlying how the model draws from the context or prior knowledge, Yu et al. (2023); Ortu et al. (2024); Jin et al. (2024) have searched for attention heads that promote each answer. However, these works do not focus on whether or how the model deliberately mediates which source to rely on.

To this question of *how*, we hypothesize that there is a simple fundamental mechanism in the form of a subspace within the language model that facilitates the binary decision of whether to rely on the context or the prior knowledge. To guide our search for such a subspace, we design and execute a structured recipe. First, we create the **controllable context sensitivity (CCS)** task which augments the standard knowledge conflict setting with an *intent*, such as *Ignore the context* or *Listen to the context*. By disambiguating whether the model should follow context or prior knowledge through a simple addition to the prompt, we are able to identify and evaluate its behavior in both modes for the same context–query pair. We adapt models for this task using *fine-tuning* and *in-context learning*, then evaluate them on in-domain and out-of-domain test sets to assess whether they have developed a deeper ability to choose between context and prior knowledge beyond surface-level heuristics. In our case study on the Llama-3.1-8B family (Dubey et al., 2024), we find that both fine-tuning and in-context learning are moderately effective, with models excelling on in-domain test sets and significantly improving over zero shot baselines on out-of-domain test sets.

Armed with models that can perform the CCS task reasonably well, we then explore the mechanisms that facilitate their behavior in this task. Building on insights from Jin et al. (2024), we hypothesize that for a model to solve this task, it must execute at least three high-level steps (in no particular order): extracting an answer from prior knowledge, extracting an answer from the context, and deciding to answer with the context answer or the prior answer. We then seek to identify layers that may contain the model's computations that are aligned with each step. To do so, we develop an algorithm that uses tools from mechanistic interpretability to find a targeted subset of layers at which activation patching (Geiger et al., 2020; Vig et al., 2020; Meng et al., 2022) can switch a model from preferring the answer in the context to preferring the answer in its prior knowledge and vice versa. Then, building on ideas from distributed alignment search (Geiger et al., 2024), we identify a knob for the model's decision between following context or prior in the form of a 1-dimensional subspace. Despite locating such a knob on an instruct model fine-tuned on this task that states explicit intents, we show that it is even effective on non-finetuned and base models of the same family for prompts that do not state the intent.

Furthermore, we show strong evidence that for models good at the CCS task, the two intents correspond to two distinct values in that subspace, while bad models fail to exhibit this distinction. We repeat this process for Gemma-2 9B and Mistral-v0.3 7B to find a similar story. Our results suggest that a 1-dimensional subspace may be fundamental to many types of large language models (LLMs) in facilitating their ability to decide between following the context or its prior knowledge. These findings move toward developing more robust language models with controllable levels of reliance on context and prior knowledge. They further highlight how investigating models at a mechanistic level can yield high-quality interventions to control a model's behavior.

Contributions. We summarize our key contributions: (a) We propose a recipe to identify a 1-dimensional subspace in the model which can act as a knob for whether the model chooses to follow context or prior knowledge. In short, this recipe leverages fine-tuning a model on a carefully constructed task to aid in finding interpretable insights about the non-fine-tuned model. (b) Using this recipe, we make a novel discovery about how models negotiate the knowledge conflict. In particular, we find a 1-dimensional subspace with many desirable traits. First, the same subspace effectively encodes and can control the decision for many model configurations within the same model family, i.e., transfers from the fine-tuned model to the base model. Second, this pattern is consistently replicated across several families of LLMs (Llama-3.1 8B, Gemma-2 7B, and Mistral-v0.3 7B). Third, our findings indicate that fine-tuned models learn to adjust the value of this subspace, suggesting it is fundamental to resolving knowledge conflicts.

2 Related Work

Prior Knowledge in Language Models. Prior studies have noted that LMs exhibit remarkable capabilities at answering questions depending on prior knowledge, such as factual recall. When queried, language models often generate plausible responses, indicating they may possess encoded knowledge about entities (Brown et al., 2020; Petroni et al., 2019; Roberts et al., 2020; Geva et al., 2021). This knowledge is encoded in the model's weights as the model is exposed to mentions of these entities during pretraining (Xu et al., 2022; Zhou et al., 2023). Pretraining can lead to not only learning facts but also memorizing specific strings (Carlini et al., 2023; Stoehr et al., 2024b).

Influence of Context on Language Models. Models might also be prompted with context in addition to the query, which can be critical to the model solving the task effectively, such as in: (a) **In-context learning** (Brown et al., 2020), where demonstrations guide the model's response; (b) **Retrieval-augmented generation** (Lewis et al., 2020) and **open-book question-answering** (Mihaylov et al., 2018; Kasai et al., 2023), where relevant documents are included in context to aid query responses; (c) **Interactive dialogue/chat** (Vinyals & Le, 2015; OpenAI, 2023), where users converse with models over multiple turns; and (d) **Text annotation** (Ziems et al., 2024), where a model analyzes passages in the context for sentiment, toxicity, coherence, *inter alia*. However, other use cases may be better served by ignoring the context to some degree, i.e., in: (a) combating **jailbreaking** (Yu et al., 2024), e.g., ignoring attempts to override built-in model behaviors; (b) resilience to **misinformation** (Hong et al., 2024; Halawi et al., 2024), e.g., avoiding integrating incorrect information in the context; and (c) ignoring **irrelevant contexts** (Shi et al., 2023; Yoran et al., 2024). In all of these settings, models draw from two sources when responding: context, and knowledge encoded during training. Controlling context sensitivity in an application-dependent manner is key to robust use.

Controlling Model Sensitivity to Context. Several studies have proposed interventions to reduce dependency on prior knowledge and favor in-context information, including prompting (Zhou et al., 2023; Onoe et al., 2023), modifying training data (Wang et al., 2023a), fine-tuning (Li et al., 2023a), and activation-level interventions (Li et al., 2023c; Stoehr et al., 2024a; Yu et al., 2023; Ortu et al., 2024) at inference time. While Li et al. (2023a) aims for some level of controllable context sensitivity by attempting to ignore irrelevant context, they do not allow for explicit controllability. Neeman et al. (2023) train models to predict two answers using both context and prior knowledge. At a mechanistic level, Yu et al. (2023) and Ortu et al. (2024) use logit attribution methods (nostalgebraist, 2020) to inspect and identify attention heads which promote each answer. However, their interventions on these heads show limited bidirectional control, suggesting an inadequate capture of model behavior. Jin et al. (2024) uses path patching (Goldowsky-Dill et al., 2023; Wang et al., 2023b), an intervention-based method, to identify heads and show that zero-ablating a subset can effectively control model behavior.

Identifying Mechanisms in Neural Networks According to the linear subspace hypothesis (Bolukbasi et al., 2016; Vargas & Cotterell, 2020; Wang et al., 2023c), model representations encode concepts as low-dimensional linear subspaces. Based on this hypothesis, prior work has explored how various concepts including truthfulness (Marks & Tegmark, 2024; Li et al., 2023c), humor (von Rütte et al., 2024), sentiment (Tigges et al., 2023), and refusal (Arditi et al., 2024) are encoded within model representations. Beyond identifying subspace representations, researchers have controlled model behavior by intervening on identified subspaces through additive steering (adding vectors to model representations) (Rimsky et al., 2024; Turner et al., 2023; Zou et al., 2023; Ravfogel et al., 2022). Concept subspaces are commonly identified using distributed alignment search (Geiger et al., 2024), LEACE (Belrose et al., 2023b), and difference in means (Marks & Tegmark, 2024).

3 How to Find the Knob Behind Context Sensitivity

3.1 Designing the Task

First, we define the task of controllable context sensitivity, which should include *minimally contrastive* example pairs. Each pair has the same context $\boldsymbol{c}$ and query $\boldsymbol{q}$, differing only in whether the model should follow the context or prior knowledge. These pairs allow us to compare the model's internal states when it follows context versus prior knowledge, with all else equal.

Consider a language model p over an alphabet Σ, i.e., p is a distribution over the Kleene closure Σ^*. Further, consider a distinguished subset $\mathcal{Q} \subset \Sigma^*$ corresponding to licit queries and a distinguished

subset $\mathcal{C} \subset \Sigma^*$ corresponding to licit contexts. Let ε be the empty string. For a query $\boldsymbol{q} \in \mathcal{Q}$, e.g., *What is the capital of France?*, and context $\boldsymbol{c} \in \mathcal{C}$, e.g., *The capital of France is London.*, let $a(\boldsymbol{q}, \varepsilon) \in \Sigma^*$ be the context-independent answer (*Paris*) and $a(\boldsymbol{q}, \boldsymbol{c}) \in \Sigma^*$ be the context-dependent answer (*London*). Let $w \in \{\text{ctx}, \text{pri}\}$ denote an **intent**, indicating whether to follow context (ctx) or prior knowledge (pri). Let $F\colon \mathcal{Q} \times \mathcal{C} \times \{\text{ctx}, \text{pri}\} \to \Sigma^*$ be a *formatting function* mapping from a query, context, and intent to a formatted prompt, e.g., "*Context: The capital of France is London/n Instruction: Only listen to the context/n Query: What is the capital of France?*". Let $\mathcal{S}_{\text{trn}} \subset \mathcal{Q} \times \mathcal{C}$ and $\mathcal{S}_{\text{tst}} \subset \mathcal{Q} \times \mathcal{C}$ be disjoint training and testing sets of query-context pairs. Models are trained on $F(\boldsymbol{q}, \boldsymbol{c}, \text{pri}) \cdot a(\boldsymbol{q}, \varepsilon)$ and $F(\boldsymbol{q}, \boldsymbol{c}, \text{ctx}) \cdot a(\boldsymbol{q}, \boldsymbol{c})$ for $(\boldsymbol{q}, \boldsymbol{c}) \in \mathcal{S}_{\text{trn}}$, where $\cdot$ denotes concatenation.

3.2 Identifying Model Behavior

Adapting a Model to this Task. To study the model's mechanism, we first need it to controllably follow either context or prior knowledge. We adapt a language model to solve the task with two methods: (i) fine-tuning using a standard next-token prediction on the training set $\mathcal{D}_{\text{trn}}$, and (ii) using training samples as few-shot demonstrations for in-context learning.

Evaluating Controllable Context Sensitivity. We evaluate a model's ability to controllably choose between context and prior knowledge using *pair-accuracy*. An example is correct only if the model outputs the correct answer to a given query $\boldsymbol{q}$ and context $\boldsymbol{c}$ for both intents (ctx and pri). That is, given a model p and dataset $\mathcal{S}$, with $\text{greedy}_{\boldsymbol{a} \in \Sigma^*}$ denoting the output of greedy decoding,

$$\text{PairAcc}(p, \mathcal{S}) \tag{1}$$
$$= \frac{1}{|\mathcal{S}|} \sum_{(\boldsymbol{q}, \boldsymbol{c}) \in \mathcal{S}} \mathbb{1}\{\underset{\boldsymbol{a} \in \Sigma^*}{\text{greedy}}\, p(\boldsymbol{a} \mid F(\boldsymbol{q}, \boldsymbol{c}, \text{ctx})) = a(\boldsymbol{q}, \boldsymbol{c})\} \mathbb{1}\{\underset{\boldsymbol{a} \in \Sigma^*}{\text{greedy}}\, p(\boldsymbol{a} \mid F(\boldsymbol{q}, \boldsymbol{c}, \text{pri})) = a(\boldsymbol{q}, \varepsilon)\}$$

3.3 Identifying Important Layers

Next, we need to identify layers in the model where the target behavior emerges. Building on prior work (Jin et al., 2024), we posit that for a model to succeed at this task, it must be able to execute at least three steps (not necessarily in this order): (i) extract the answer from the model's prior knowledge; (ii) extract the answer from the context; and (iii) decide whether to answer according to the context or the prior knowledge. Note that, under the framing of Geiger et al. (2024), these would be considered causal variables in a high-level model. Without specifying an exact causal graph, we argue these must be components in any reasonable one. We use tools from mechanistic interpretability to identify the layers at which the model appears to implement these steps.

Intervention-based Interpretability. Intervention-based interpretability techniques like activation patching are often used to identify which model activations are crucial for a task (Geiger et al., 2020; Vig et al., 2020; Meng et al., 2022). Intuitively, if intervening at some set of intermediate states can change a model's output behavior for a task, those intermediate states likely play a critical role in the model's ability in that task. Activation patching, specifically interchange interventions (Geiger et al., 2023), involves two strings that differ only with respect to the task of interest. For example, to identify activations that encode the *intent* of a prompt, we use two input strings that share the same *query* and *context* but differ in their *intent*. For a given model, p, we define a source string, $\boldsymbol{s} \in \Sigma^*$, and a target string, $\boldsymbol{t} \in \Sigma^*$. During the forward pass of $p(\boldsymbol{t})$, we replace a subset of intermediate activations with those from $p(\boldsymbol{s})$ and observe the effect on model internals and the output distribution of the patched $p(\boldsymbol{t})$. We patch only at the last token, as prior work has shown this to be most informative for predicting the next token (Yu et al., 2023; Jin et al., 2024; Stoehr et al., 2024a; Monea et al., 2024).

We also only patch the outputs of the multi-head attention (MHA) components in a transformer block; the intuition behind this choice is that this component ought to integrate information from the context into the residual stream of the last token (see App. D for more details on the residual stream framework). Interchanging these output activations allows us to analyze what kind of information is written on the residual stream and whether it has a causal effect on the model internals and the output distribution. By searching over different subsets of intermediate activations, we can identify those with the greatest impact on task performance and thus facilitate critical functionality.

Iteratively Searching For Important Components. Searching for a small subset of MHA components at the last token position to patch is nontrivial because it is over an exponentially large space

Table 1: **Patching Setup**: To investigate the model's internal mechanisms, we use three distinct patching setups ($\mathcal{D}_w$, $\mathcal{D}_\text{p}$, and $\mathcal{D}_\text{c}$) to address our research questions. For all datasets, an example consists of a source prompt $\boldsymbol{s}$, source answer $\boldsymbol{a_s}$, target prompt $\boldsymbol{t}$, and target answer $\boldsymbol{a_t}$. $\mathcal{D}_w$, has two subvariants: $\mathcal{D}_w^{c\to p}$ and $\mathcal{D}_w^{p\to c}$, which represent different directions of the intervention.

Dataset	Question & Description	$\boldsymbol{s}$	$\boldsymbol{a_s}$	$\boldsymbol{t}$	$\boldsymbol{a_t}$
$\mathcal{D}_w^{p\to c}$	**Where is w computed?** Only w differs Source w = pri	*The capital of France is London* ***Ignore*** *the context* *What is the capital of France?*	*Paris*	*The capital of France is London* ***Only listen to*** *the context* *What is the capital of France?*	*London*
$\mathcal{D}_w^{c\to p}$	Only w differs Source w = ctx	*The capital of France is London* ***Only listen to*** *the context* *What is the capital of France?*	*London*	*The capital of France is London* ***Ignore*** *the context* *What is the capital of France?*	*Paris*
$\mathcal{D}_\text{p}$	**Where is $a(q,\varepsilon)$ computed?** w = pri, different $a(q,\varepsilon)$	*The capital of France is London* *Ignore the context* *What is the capital of* ***France****?*	*Paris*	*The capital of France is London* *Ignore the context* *What is the capital of* ***Italy****?*	*Rome*
$\mathcal{D}_\text{c}$	**Where is $a(q,c)$ computed?** w = ctx, different $a(q,c)$	*The capital of France is* ***London*** *Only listen to the context* *What is the capital of France?*	*London*	*The capital of France is* ***Rome*** *Only listen to the context* *What is the capital of France?*	*Rome*

(i.e., 2^L, where L is the number of layers in the model). Thus, we use an iterative search algorithm to build a subset of important components, requiring $O(L)$ forward passes. In this algorithm, we use the *Token Identity Patchscope* (TIP) to observe model behavior at intermediate states (Ghandeharioun et al., 2024).[1] Specifically, we use it to identify the model's likelihood on the context and prior answers at intermediate layers and choose a subset of layers to patch that push the model to prefer the desired answer. Given a dataset of source and target pairs, the algorithm has two main steps. First, it identifies a continuous *base range* of layers where patching MHA components enables decoding the source answer from the residual stream. Then, it finds *inhibition layers* that suppress the source answer at later layers by iteratively patching MHA components until the source answer has a high probability. We provide Python-esque pseudocode for our search algorithm in App. A.1.

Patching Setups Per Subquestion. We wish to address the three subquestions: (i) Where is the intent w computed? (ii) Where is $a(\boldsymbol{q},\varepsilon)$ computed? (iii) Where is $a(\boldsymbol{q},\boldsymbol{c})$ computed? Answering each subquestion will demand applying the search algorithm described above on a specific patching setup, i.e., dataset, per subquestion. Each patching setup consists of tuples containing a source string, its associated answer, a target string, and the target's answer. The relationship between the source and target depends on the question we aim to answer. Table 1 outlines the specific patching setups for each subquestion. First, $\mathcal{D}_w^{c\to p}$ and $\mathcal{D}_w^{p\to c}$ hold the *context* and *query* constant but vary the intent w, enabling us to investigate how the model processes different intents. We define $\mathcal{D}_w^{p\to c} = \{(F(\boldsymbol{q},\boldsymbol{c},\text{pri}), a(\boldsymbol{q},\varepsilon), F(\boldsymbol{q},\boldsymbol{c},\text{ctx}), a(\boldsymbol{q},\boldsymbol{c}))\}_{(\boldsymbol{q},\boldsymbol{c})\in\mathcal{S}_\text{tst}}$ and $\mathcal{D}_w^{c\to p} = \{(F(\boldsymbol{q},\boldsymbol{c},\text{ctx}), a(\boldsymbol{q},\boldsymbol{c}), F(\boldsymbol{q},\boldsymbol{c},\text{pri}), a(\boldsymbol{q},\varepsilon))\}_{(\boldsymbol{q},\boldsymbol{c})\in\mathcal{S}_\text{tst}}$. $\mathcal{D}_\text{p}$ includes tuples where both the source and the target share the intent w = pri, but differ in the prior answer $a(\boldsymbol{q},\varepsilon)$ they suggest, $\mathcal{D}_\text{p} = \{(F(\boldsymbol{q},\boldsymbol{c},\text{pri}), a(\boldsymbol{q},\varepsilon), F(\boldsymbol{q}',\boldsymbol{c},\text{pri}), a(\boldsymbol{q}',\varepsilon))\}_{(\boldsymbol{q},\boldsymbol{c})\in\mathcal{S}_\text{tst},\boldsymbol{q}'\in\mathcal{Q}\setminus\{\boldsymbol{q}\}}$. This allows us to evaluate how patching alters the model's response with respect to $a(\boldsymbol{q},\varepsilon)$ and discern how the model computes $a(\boldsymbol{q},\varepsilon)$. Similarly, in $\mathcal{D}_\text{c}$ we explore how the model computes $a(\boldsymbol{q},\boldsymbol{c})$, $\mathcal{D}_\text{c} = \{(F(\boldsymbol{q},\boldsymbol{c},\text{ctx}), a(\boldsymbol{q},\boldsymbol{c}), F(\boldsymbol{q},\boldsymbol{c}',\text{ctx}), a(\boldsymbol{q},\boldsymbol{c}'))\}_{(\boldsymbol{q},\boldsymbol{c})\in\mathcal{S}_\text{tst},\boldsymbol{c}'\in\mathcal{C}\setminus\{\boldsymbol{c}\}}$.

3.4 Identifying the context-controllability subspace feature

Learning the Context-versus-Prior Subspace. Once we identified a subset of model components that potentially contain the mechanism for deciding between answering from the context or prior knowledge, we can further investigate whether this functionality can be encoded in a low-dimensional subspace within these components. According to the linear subspace hypothesis (Bolukbasi et al., 2016; Vargas & Cotterell, 2020), there exists a linear subspace $\mathcal{F} \subset \mathbb{R}^D$ which encodes the information about a specific concept. In our case, the concept of interest is whether the model uses the context or its prior knowledge. Since this is a simple binary concept, we hypothesize that a rank-1

[1] TIP interprets the information in a model's residual stream at intermediate layers by using the model to map from the residual stream at a given layer and token index to a distribution over tokens that best represents the information stored in that intermediate state. This approach can also be viewed as a variant of the SelfiE method (Chen et al., 2024). TIP outperforms other alternatives for interpreting intermediate states (e.g., probing (Tenney et al., 2019), LogitLens (nostalgebraist, 2020), and TunedLens (Belrose et al., 2023a)).

subspace encodes this concept. Informally, this hypothesis implies that a model's representation can be decomposed into a sum of orthogonal components, i.e., directions in space, and one such direction specifically encodes whether to follow the context or prior knowledge.

Let ℓ be the last layer in the *base range* found by our search algorithm. Let $\boldsymbol{h}^\ell \in \mathbb{R}^D$ denote the activation at layer ℓ. We learn a rank-1 orthogonal projection matrix $\boldsymbol{P} \in \mathbb{R}^{D\times D}$ to project $\boldsymbol{h}^\ell \in \mathbb{R}^D$ onto a 1-dimensional subspace $\mathcal{F}$ of $\mathbb{R}^D$, encoding the intent. We parameterize $\boldsymbol{P} = \boldsymbol{u}\boldsymbol{u}^\top$, where $\boldsymbol{u} \in \mathbb{R}^D$ is the basis of the subspace with a norm of 1 (see App. G for a more detailed explanation of the parameterization of $\boldsymbol{P}$). Given a tuple $(\boldsymbol{s}, \boldsymbol{a_s}, \boldsymbol{t}, \boldsymbol{a_t}) \in \mathcal{D}_w^{p\to c} \cup \mathcal{D}_w^{c\to p}$, we define $\boldsymbol{h}_{\boldsymbol{s}}^\ell$ to be the last token residual stream after layer ℓ of the forward pass $p(\boldsymbol{s})$, and similarly, $\boldsymbol{h}_{\boldsymbol{t}}^\ell$ for $p(\boldsymbol{t})$. To learn $\boldsymbol{P}$, we freeze the parameters of p and patch the forward pass of $p(\boldsymbol{t})$ as follows:

$$\boldsymbol{h}_{\boldsymbol{t}}^\ell = (\boldsymbol{I} - \boldsymbol{P})\boldsymbol{h}_{\boldsymbol{t}}^\ell + \boldsymbol{P}\boldsymbol{h}_{\boldsymbol{t}}^\ell \qquad \textit{(normal decomposition)} \tag{2a}$$

$$\widetilde{\boldsymbol{h}}_{\boldsymbol{t}}^\ell \triangleq (\boldsymbol{I} - \boldsymbol{P})\boldsymbol{h}_{\boldsymbol{t}}^\ell + \boldsymbol{P}\boldsymbol{h}_{\boldsymbol{s}}^\ell \qquad \textit{(patched decomposition)} \tag{2b}$$

Eq. (2a) expresses that we can decompose $\boldsymbol{h}_{\boldsymbol{t}}^\ell$ into (i) the sum of the component representing our concept of interest ($\boldsymbol{P}\boldsymbol{h}_{\boldsymbol{t}}^\ell$) and (ii) its orthogonal complement, the component which represents other information ($(\boldsymbol{I}-\boldsymbol{P})\boldsymbol{h}_{\boldsymbol{t}}^\ell$). Then, in Eq. (2b), $\widetilde{\boldsymbol{h}}_{\boldsymbol{t}}^\ell$ is constructed by replacing the component in $\boldsymbol{h}_{\boldsymbol{t}}^\ell$ representing our concept of interest ($\boldsymbol{P}\boldsymbol{h}_{\boldsymbol{t}}^\ell$) with the component in $\boldsymbol{h}_{\boldsymbol{s}}^\ell$ representing the concept ($\boldsymbol{P}\boldsymbol{h}_{\boldsymbol{s}}^\ell$). Thus, if $\boldsymbol{P}$ projects onto a subspace that encodes the concept, then the representation $\widetilde{\boldsymbol{h}}_{\boldsymbol{t}}^\ell$ encodes the *intent* concept from $\boldsymbol{h}_{\boldsymbol{s}}^\ell$ and all other aspects from $\boldsymbol{h}_{\boldsymbol{t}}^\ell$. We visually illustrate these decompositions in App. H.

We denote $\widetilde{p}_\ell(\cdot; \boldsymbol{P}, \boldsymbol{s})$ to be the language model with activation $\boldsymbol{h}_{\boldsymbol{t}}^\ell$ replaced by $\widetilde{\boldsymbol{h}}_{\boldsymbol{t}}^\ell$ as defined in Eq. (2b). We construct a training set $\{(\boldsymbol{s}_n, \boldsymbol{a}_{\boldsymbol{s}_n}, \boldsymbol{t}_n, \boldsymbol{a}_{\boldsymbol{t}_n})\}_{n=1}^N \subset \mathcal{D}_w^{p\to c} \cup \mathcal{D}_w^{c\to p}$. As can be seen in Tab. 1, this dataset contains matched pairs $\boldsymbol{s}_n, \boldsymbol{t}_n$ which differ only in the specified intent. Then, to learn $\boldsymbol{P}$ which well-represents our concept, we minimize the following objective over the training set:

$$J_\ell(\boldsymbol{P}) = -\frac{1}{N}\sum_{n=1}^{N} \log \widetilde{p}_\ell(\boldsymbol{a}_{\boldsymbol{s}_n} \mid \boldsymbol{t}_n; \boldsymbol{P}, \boldsymbol{s}_n) \tag{3}$$

That is, we minimize the cross-entropy loss between the language model when patched with $\widetilde{\boldsymbol{h}}_{\boldsymbol{t}}^\ell$ and the label $\boldsymbol{a_s}$. Since $\boldsymbol{s}_n$ and $\boldsymbol{t}_n$ always have different intents w, but share the same *context* and *query*, we are effectively optimizing for a subspace where replacing the subspace component of $\boldsymbol{t}_n$ with the corresponding component of $\boldsymbol{s}_n$ leads to an answer that reflects the flipped intent.

Controlling Model Behavior Using the Context-versus-Prior Subspace. After learning an orthogonal projection matrix $\boldsymbol{P}$ to project a vector into the context-versus-prior subspace, we can control the model's behavior by setting the subspace component based on the input intent w. To do this we define a *function* $c : \{\text{ctx}, \text{pri}\} \to \mathbb{R}$ that acts as a scalar for the basis $\boldsymbol{u}$ of $\mathcal{F}$ and returns a constant corresponding to one of the two intents.[2]

$$\widetilde{\boldsymbol{h}}_{\boldsymbol{t}}^\ell \triangleq (\boldsymbol{I} - \boldsymbol{P})\boldsymbol{h}_{\boldsymbol{t}}^\ell + \boldsymbol{P}\boldsymbol{u}c(w) \tag{4}$$

The function c represents the knob to steer which behavior to follow. A successful static intervention on a learned subspace $\mathcal{F}$ implies that we have not only identified a 1-dimensional subspace representing intent but also determined how to manipulate it manually. We evaluate the effectiveness of a static intervention using the *pair-accuracy*.

4 Case Study: Llama-3.1 8B

We describe detailed results in executing the recipe from §3 to identify the mechanism behind controllable context sensitivity. Results for additional models are in §5 and App. I.

4.1 Task Setup

Datasets. Following the task formulation in §3.1, we construct intent-augmented datasets, CCS-BF, CCS-MH, and CCS-AR, based on the query-context pairs in BaseFakepedia, MultiHopFakepedia (Monea et al., 2024), and Arithmetic. BaseFakepedia is a knowledge conflict dataset

[2] $\boldsymbol{P}$ is redundant in the second term of Eq. (4) since $\boldsymbol{P}\boldsymbol{u}c(w) = \boldsymbol{u}\boldsymbol{u}^T\boldsymbol{u}c(w) = \boldsymbol{u}c(w)$, but is included for consistency.

from Wikipedia with queries across 23 relation types (e.g., *Norway's capital city* or *Mac Pro, a product created by*) and paragraphs generated by a language model that provide counterfactual answers. MULTIHOPFAKEPEDIA resembles BASEFAKEPEDIA but requires an extra hop of reasoning (e.g., *London is the capital of France. Tunis is in the same country as London. What country is Tunis in?*). ARITHMETIC is a synthetically generated dataset whose queries are simple arithmetic expressions using the operators $\{+, -, \times, \div, \exp\}$ and contexts are reassignments of subexpressions to another value resulting in a counterfactual answer. For example, given the query *(5 + 1) / 2 =* and the context *5 = 9*, the prior answer would be *3*, while the context answer would be *5*. We limit expressions to a depth of 2, i.e., two operators, with input and output numbers between 0 and 9.

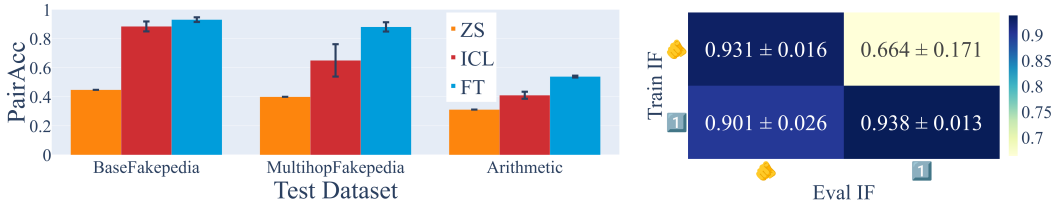
Intent Format. We also format the intent $w \in \{\text{ctx}, \text{pri}\}$ in two different ways to probe the model's robustness to different formulations of the same intent. First, the *instruction* intent format (👇) expresses the intent as a string instruction, e.g., *Ignore the context in answering the query.* or *Only consider the context in answering the query.* Second, the *weight* intent format (1️⃣) expresses the intent as a context weight, e.g., *Context weight: 0.* or *Context weight: 1.*

4.2 ADAPTING MODELS TO THE TASK

Training. We adapt the instruct Llama-3.1 8B (Dubey et al., 2024) to this task in two ways: (i) QLoRA fine-tuning (FT) the attention components using CCS-BF's training set, and (ii) in-context learning (ICL) with 10 prepended CCS-BF examples. Training details are in App. E.

Evaluation. We examine two forms of generalization: robustness to different datasets, and robustness to different intent formats. For the former, we test whether a model trained on CCS-BF can perform well on test splits from CCS-BF, CCS-MH, and CCS-AR. For the latter, we assess whether a model trained with one intent format, e.g., 👇, performs well with prompts in another format, e.g., 1️⃣.

Results. Fig. 1 shows the generalization results for Llama-3.1-8B-Instruct. The model achieves high pair accuracy on its in-domain test set with FT ($\approx 90\%$) and ICL ($\approx 88\%$). However, performance drops significantly for ICL and mildly for FT on CCS-MH, which requires additional reasoning. On CCS-AR, both models show significant degradation, as the task is out-of-domain and demands reasoning beyond context extraction. Fig. 1b shows that, for intent formats, the model: (a) performs well when fine-tuned on either intent format, (b) generalizes well from the 1️⃣ to the 👇 format, and (c) struggles when trained on the 👇 format but evaluated on 1️⃣. This result is intuitive as the instruct model is tuned to follow natural language instructions such as 👇, but may not be familiar with interpreting the 1️⃣ instruction. Overall, the model: (a) learns the task in-domain with high accuracy, (b) generalizes moderately well to other datasets, depending on the degree of difference, and (c) adapts reasonably well to other intent formats, especially if they are in natural language.



(a) Generalization to Datasets.

(b) Generalization to Intent Formats (IF).

Figure 1: (a) Pair accuracy of Llama-3.1-8B-Instruct when trained on CCS-BF and evaluated on CCS-BF, CCS-MH, and CCS-AR datasets. For each dataset, we evaluate the model zero-shot, with 10 in-context learning examples from CCS-BF, and after fine-tuning on 2048 examples from CCS-BF. (b) Pair accuracy when trained and evaluated on different intent formats, where 1️⃣ and 👇 mean the intent is expressed as a numerical context weight or as a string instruction, respectively.

4.3 IDENTIFYING IMPORTANT COMPONENTS

Focusing on Llama-3.1-8B-Instruct fine-tuned using the intent format 👇, we apply the algorithm presented in §3.3 to identify important layers that appear to facilitate the model's sensitivity to context. First, we investigate where the intent w is computed by using tuples from $\mathcal{D}_w^{p \to c}$ and $\mathcal{D}_w^{c \to p}$ (described

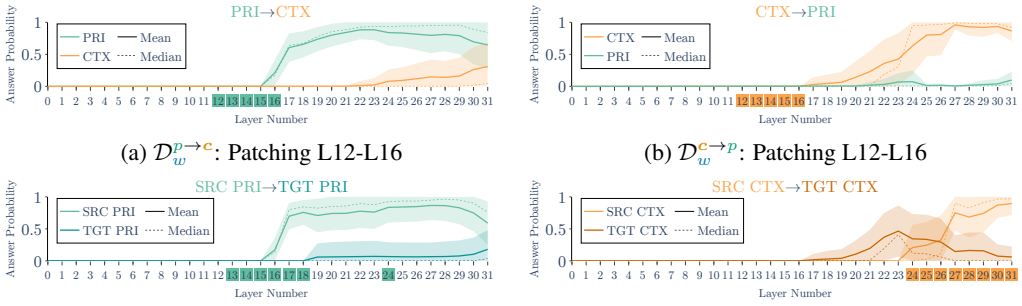
in Tab. 1). Second, we investigate which layers compute the prior answer $a(\boldsymbol{q}, \varepsilon)$ and context answer $a(\boldsymbol{q}, \boldsymbol{c})$. If these layers are later than the ones identified in the first step then that could suggest that w is encoded in the residual stream and depending on its value, either $a(\boldsymbol{q}, \boldsymbol{c})$ or $a(\boldsymbol{q}, \varepsilon)$ is retrieved.

4.3.1 WHERE IS w COMPUTED?

We aim to identify where the model initially incorporates information about the intent and how this affects its predictions. We use the algorithm described in §3.3 and App. A.1 on both $\mathcal{D}_w^{\boldsymbol{c}\to p}$ and $\mathcal{D}_w^{p\to \boldsymbol{c}}$ and report the identified layers in Fig. 2a and Fig. 2b, respectively. We observe that in both directions, patching the MHA outputs for layers 12 to 16 reliably switches the prediction from context-agreeing (CTX) to prior-agreeing (PRIOR) and vice versa. This suggests two possible hypotheses: either these layers load the correct answer into the residual stream, or they encode the intent bit w, which subsequently triggers the loading of the correct answer in later layers. However, Fig. 2b shows the model has a low probability of the context answer until after layer 24, supporting the latter hypothesis.

4.3.2 WHERE ARE $a(\boldsymbol{q}, \varepsilon)$ AND $a(\boldsymbol{q}, \boldsymbol{c})$ COMPUTED?

We apply the same algorithm to $\mathcal{D}_\text{p}$ and $\mathcal{D}_\text{c}$ to identify which layers load the two answers, $a(\boldsymbol{q}, \varepsilon)$ and $a(\boldsymbol{q}, \boldsymbol{c})$. For $\mathcal{D}_\text{p}$, we patch activations from a source (SRC PRI) into a target (TGT PRI), both sharing the same intent pri but different prior answers $a(\boldsymbol{q}, \varepsilon)$. For $\mathcal{D}_\text{c}$, we patch from a source (SRC CTX) into a target (TGT CTX), both having intent ctx but different context answers $a(\boldsymbol{q}, \boldsymbol{c})$. Fig. 2d confirms that the context answer is mainly integrated after layer 24. For $\mathcal{D}_\text{p}$, we identify a base range of layers 13-18, with layer 24 as an inhibition layer (Fig. 2c).[3] Since answers are integrated at different layers, distinct mechanisms likely handle each answer. Layer 24 seems crucial in both processes. Ablation studies in App. A.2 show that neither $a(\boldsymbol{q}, \varepsilon)$ nor $a(\boldsymbol{q}, \boldsymbol{c})$ can be effectively patched without layer 24 (Fig. 7a and 6c). We hypothesize layer 24's role varies by intent, conditionally loading either the prior or context answer. Since the model's preference for context or prior answer stabilizes after layer 16, this suggests that the intent is encoded after this point and later layers such as layer 24 read it. Given the binary nature of the intent variable, we hypothesize that its encoding can be modified to selectively trigger the loading of either the context or prior answer.



(a) $\mathcal{D}_w^{p\to \boldsymbol{c}}$: Patching L12-L16

(b) $\mathcal{D}_w^{\boldsymbol{c}\to p}$: Patching L12-L16

(c) $\mathcal{D}_\text{p}$: Patching L14-L18 + L24

(d) $\mathcal{D}_\text{c}$: Patching L24-31

Figure 2: Answer probabilities per layer as determined by patchscope (TIP) for different patching settings on Llama 3.1 Instruct FT 👍. The x-axis represents the layers. The y-axis shows the TIP answer probability. On the x-axis we mark the patched layers. Each row of subplots aims to answer one subquestion. Top Row: **Where is w computed?** Patching a source SRC PRI into a TGT CTX (left; 2a) and vice versa (right; 2b). Bottom Row: **Where is $a(\boldsymbol{q}, \varepsilon)$ and $a(\boldsymbol{q}, \boldsymbol{c})$ computed?** Patching a source SRC PRI into a TGT PRI, using samples from $\mathcal{D}_\text{p}$ (2c) and the same for CTX (2d).

4.4 IDENTIFYING THE CONTEXT-CONTROLLABILITY SUBSPACE

Following §3.4, we learn a rank-1 orthogonal projection matrix $\boldsymbol{P}$ to identify a subspace $\mathcal{F}_w$ encoding intent. We search for this subspace in *layer 16*, as this is the last layer in the *base range* of influential layers found in §4.3.1 using the algorithm described in §3.3. We train on the subset of $\mathcal{D}_w^{p\to \boldsymbol{c}} \cup \mathcal{D}_w^{\boldsymbol{c}\to p}$ of CCS-BF for which the model answers correctly for both intents. If this subspace indeed controls

[3] In Fig. 7a, we show that without patching layer 24, the probability of the SRC PRI significantly decreases.

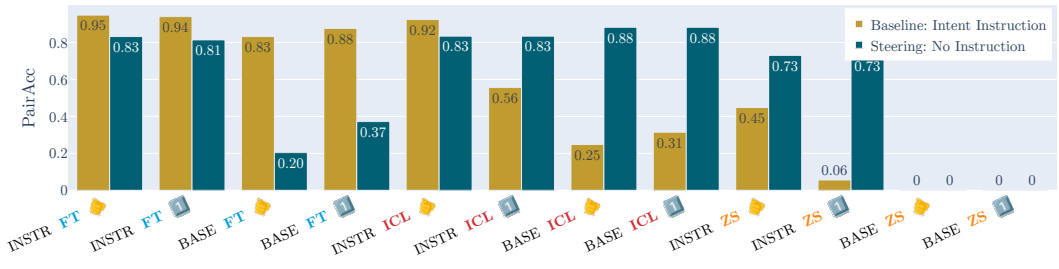


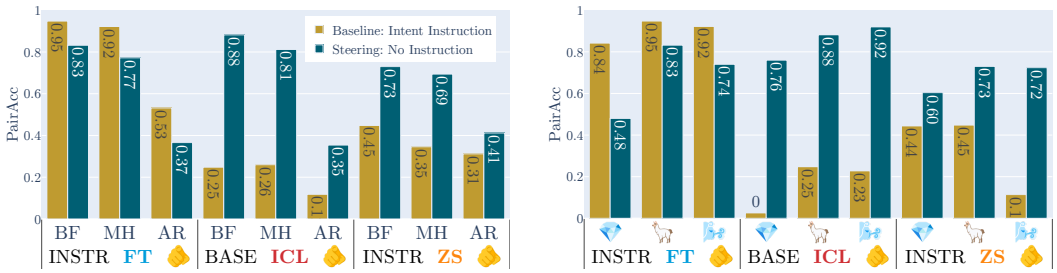
Figure 3: The baseline accuracy (yellow) reflects the *PairAcc* of a model evaluated on CCS-BF without steering. In contrast, blue represents the *PairAcc* when the explicit intent instruction is removed from inputs and the subspace $\mathcal{F}_w$ manually set. Although $\mathcal{F}_w$ was learned for the Instruct FT, it transfers well to other configurations, as evidenced by the blue bar approaching or exceeding the yellow bar for most model configurations.

the choice between context and prior, then we should be able to remove the intent from the input and still steer the model to produce the intended output by setting the value of $c(w)$ according to Equation 4. For these interventions, we choose $c\,(\text{pri}) = -6$ and $c\,(\text{ctx}) = 6$ based on performance on a validation set. For example, a model should be able to answer *The capital of France is London. What is the capital of France?* with *London* when steered with $c(w) = 6$ and *Paris* when $c(w) = -6$.

Fig. 3 shows that the identified subspace strongly aligns with the causal variable for intent, allowing for effective model steering. On the fine-tuned instruct model, we achieve 83% *PairAcc* using steering, compared to the 95% baseline (very left; INSTR FT). This is notable, given we manipulate only a 1-dimensional subspace in a single layer. Additionally, the figure shows that this same subspace exhibits strong alignment across different model configurations. We successfully transfer $\mathcal{F}_w$ to both the non-fine-tuned Llama-3.1-8B-Instruct (INSTR) and the base Llama-3.1-8B (BASE) model. The subspace performs particularly well on the base model in the In-Context Learning (ICL) setting, where *PairAcc* significantly exceeds the baseline accuracy as well as the steered finetuned model. Moreover, we highlight the zero-shot (ZS) performance of the instruct model (73%), significantly outperforming the baseline performance. However, the ZS performance on the base model results in 0% *PairAcc*, as the model lacks training for instruction-following tasks. While the subspace intervention is relatively ineffective on the fine-tuned base model, we hypothesize that this is because the weights of this model are likely the furthest from those of the fine-tuned instruct model.

5 A Fundamental Subspace for Controllable Context Sensitivity

Due to the strong evidence for a high alignment of $\mathcal{F}_w$ to the causal intent variable, we propose two hypotheses: (i) This subspace is fundamental to the model and different learning methods learn to



(a) Llama 3.1: Other Datasets

(b) BaseFakepedia: Other Models

Figure 4: We compare pair accuracy of a baseline model (with intent instructions) against the steered model (without intent instructions). We consider baseline models: (a) instruct model fine-tuned on CCS-BF, (b) base model with 10 in-domain ICL demonstrations, and (c) the default instruct model. Left: Subspace steering on Llama 3.1 generalizes across datasets (BaseFakepedia (BF), MultiHopFakepedia (MF), and Arithmetic (AR)). Right: For multiple models (Llama 3.1 8b (🦙), Gemma 2 9b (💎), and Mistral 7b v0.3 (🌬️)), a rank-1 subspace can be used for effective steering.

set the value of this subspace. (ii) As a fundamental subspace to language models, a similar rank-1 subspace to encode choosing context or prior knowledge can be found in other language models too.

We provide evidence to support hypothesis (i). First, Fig. 3 shows that adjusting the value of the subspace can recover or even surpass baseline performance in both fine-tuned and non-fine-tuned models. Notably, the exceptional efficacy of the subspace intervention in the zero-shot evaluation of the instruct model—which has never seen examples of this task—suggests that this capability is already present in the model and can be activated by setting $\mathcal{F}_w$. Second, Fig. 4a shows that the subspace generalizes to multiple out-of-domain datasets, with steering performance either competing with or surpassing the intent instruction baseline across different datasets. This holds for not only the fine-tuned instruct model but also ZS evaluations on the instruct model and ICL on the base model. Finally, we find a strong, statistically significant correlation (0.908) between a model's performance and how well it distinguishes values in that subspace when the intent is pri or ctx. As displayed in Fig. 5, the difference in subspace value when the intent is pri vs ctx tends to be higher for better models at this task. This suggests that well-performing models know to set this value in the subspace.

We also identify the described subspace in Gemma-2 9B (Riviere et al., 2024) and Mistral-v0.3 7B (Jiang et al., 2023), using the same methodology. Fig. 4b shows that, for each model family, their respective subspaces are transferrable from the fine-tuned instruct model to both the non-fine-tuned instruct model and the base model. In App. I we provide a detailed study of the subspace in other models, including a high correlation between model performance and subspace values.

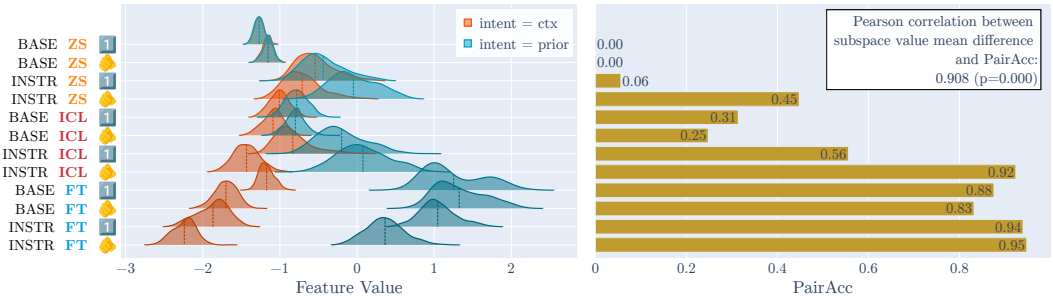


Figure 5: Subspace $\mathcal{F}_w$ value distributions of different model configurations (left) and baseline model performance on CCS-BF (right). We can observe a high correlation between the absolute difference between the means of the two groups (ctx and pri) and the performances.

6 Discussion, Limitations, and Future Work

While our study presents evidence that a model can be induced to controllably draw from context or prior knowledge in answering questions in these specific settings, it is important to characterize the nature of the exact model capability we are examining in this study. In particular, both fine-tuning and turning this knob for a model seem to be more effective when the model can directly copy the answer from the context (when the intent is ctx). For example, in the Arithmetic task, a context might explicitly contain the answer, e.g., *(5 + 1) / 2 = 7*, or it might only override a subproblem, e.g., *5 + 1 = 8*. Generally, the models are better at producing the context-agreeing answer when it is explicitly stated in the context. More investigation is needed to understand to what extent a model can use information from context as part of an intermediate reasoning chain as opposed to direct copying.

Zooming out, our work highlights the importance of studying the fundamental functionality in language models of controllable context sensitivity. We show how tools from mechanistic interpretability can be useful toward both understanding how models implement this functionality and controlling the behavior; further, such an approach could be useful for understanding mechanisms behind other functionalities. Promising future directions include: (a) evaluating whether this subspace influences additional behaviors like instruction-following, (b) learning to adaptively steer, i.e., the model automatically decides when it should leverage or ignore context (especially in settings such as retrieval-augmented generation), and (c) beyond traditional knowledge conflicts, developing datasets that involve integrating information from both context and prior knowledge rather than only choosing between the two.

Ethics Statement

As LLM capabilities grow more advanced and their usage proliferates throughout the real world, we acknowledge that their development can exacerbate risks to people via misinformation or hallucination, especially those historically underrepresented or misrepresented to these models. Our work aims to make model behavior more transparent by providing a new tool to analyze the interaction between context and prior knowledge in LMs, which is especially important as people interact with them in chat, question-answering, and other prompt-based settings. We foresee no particular ethical concerns and hope this paper contributes to developing tools that can identify and mitigate ethical concerns in the future.

References

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. *OpenReview*, 2024. URL https://openreview.net/forum?id=EqF16oDVFf.

Victoria Basmov, Yoav Goldberg, and Reut Tsarfaty. Llms' reading comprehension is affected by parametric knowledge and struggles with hypothetical statements, 2024. URL https://arxiv.org/abs/2404.06283.

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens. *arXiv*, 2023a. URL https://arxiv.org/abs/2303.08112.

Nora Belrose, David Schneider-Joseph, Shauli Ravfogel, Ryan Cotterell, Edward Raff, and Stella Biderman. LEACE: Perfect linear concept erasure in closed form. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 66044–66063, 2023b. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/d066d21c619d0a78c5b557fa3291a8f4-Paper-Conference.pdf.

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. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29, 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/a486cd07e4ac3d270571622f4f316ec5-Paper.pdf.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris 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. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901, 2020. URL https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models. *arXiv*, 2023. URL https://arxiv.org/abs/2202.07646.

Haozhe Chen, Carl Vondrick, and Chengzhi Mao. SelfIE: Self-interpretation of large language model embeddings. In Salakhutdinov, Ruslan and Kolter, Zico and Heller, Katherine and Weller, Adrian and Oliver, Nuria and Scarlett, Jonathan and Berkenkamp, Felix (ed.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 7373–7388, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/chen24ao.html.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8493–8502, Dublin, Ireland, May

2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.581. URL https://aclanthology.org/2022.acl-long.581.

Kevin Du, Vésteinn Snæbjarnarson, Niklas Stoehr, Jennifer White, Aaron Schein, and Ryan Cotterell. Context versus prior knowledge in language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13211–13235, Bangkok, Thailand, aug 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.acl-long.714.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix

Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. The llama 3 herd of models. *arXiv*, 2024. URL https://arxiv.org/abs/2407.21783.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. URL https://transformer-circuits.pub/2021/framework/index.html.

Atticus Geiger, Kyle Richardson, and Christopher Potts. Neural natural language inference models partially embed theories of lexical entailment and negation. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pp. 163–173, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.blackboxnlp-1.16. URL https://aclanthology.org/2020.blackboxnlp-1.16.

Atticus Geiger, Duligur Ibeling, Amir Zur, Maheep Chaudhary, Sonakshi Chauhan, Jing Huang, Aryaman Arora, Zhengxuan Wu, Noah Goodman, Christopher Potts, and Thomas Icard. Causal abstraction: A theoretical foundation for mechanistic interpretability. *arXiv*, jan 2023. URL https://arxiv.org/abs/2301.04709v3.

Atticus Geiger, Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah Goodman. Finding alignments between interpretable causal variables and distributed neural representations. In Locatello, Francesco and Didelez, Vanessa (ed.), *Proceedings of the Third Conference on Causal*

Learning and Reasoning, volume 236 of *Proceedings of Machine Learning Research*, pp. 160–187, 01–03 Apr 2024. URL https://proceedings.mlr.press/v236/geiger24a.html.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5484–5495, 2021. URL https://aclanthology.org/2021.emnlp-main.446.

Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 30–45, Abu Dhabi, United Arab Emirates, December 2022. doi: 10.18653/v1/2022.emnlp-main.3. URL https://aclanthology.org/2022.emnlp-main.3.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. Dissecting recall of factual associations in auto-regressive language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12216–12235, Singapore, December 2023. doi: 10.18653/v1/2023.emnlp-main.751. URL https://aclanthology.org/2023.emnlp-main.751.

Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. Patchscopes: A unifying framework for inspecting hidden representations of language models. In Salakhutdinov, Ruslan and Kolter, Zico and Heller, Katherine and Weller, Adrian and Oliver, Nuria and Scarlett, Jonathan and Berkenkamp, Felix (ed.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 15466–15490, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/ghandeharioun24a.html.

Nicholas Goldowsky-Dill, Chris MacLeod, Lucas Sato, and Aryaman Arora. Localizing model behavior with path patching, 2023. URL https://arxiv.org/abs/2304.05969.

Danny Halawi, Jean-Stanislas Denain, and Jacob Steinhardt. Overthinking the truth: Understanding how language models process false demonstrations. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Tigr1kMDZy.

Giwon Hong, Jeonghwan Kim, Junmo Kang, Sung-Hyon Myaeng, and Joyce Whang. Why so gullible? Enhancing the robustness of retrieval-augmented models against counterfactual noise. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 2474–2495, Mexico City, Mexico, jun 2024. doi: 10.18653/v1/2024.findings-naacl.159. URL https://aclanthology.org/2024.findings-naacl.159.

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. *arXiv*, 2023. URL https://arxiv.org/abs/2310.06825.

Zhuoran Jin, Pengfei Cao, Hongbang Yuan, Yubo Chen, Jiexin Xu, Huaijun Li, Xiaojian Jiang, Kang Liu, and Jun Zhao. Cutting off the head ends the conflict: A mechanism for interpreting and mitigating knowledge conflicts in language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics ACL 2024*, pp. 1193–1215, Bangkok, Thailand and virtual meeting, aug 2024. doi: 10.18653/v1/2024.findings-acl.70. URL https://aclanthology.org/2024.findings-acl.70.

Jungo Kasai, Keisuke Sakaguchi, Yoichi Takahashi, Ronan Le Bras, Akari Asai, Xinyan Yu, Dragomir Radev, Noah A Smith, Yejin Choi, and Kentaro Inui. RealTime QA: What's the Answer Right Now? In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 49025–49043, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/9941624ef7f867a502732b5154d30cb7-Paper-Datasets_and_Benchmarks.pdf.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian

Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf.

Daliang Li, Ankit Singh Rawat, Manzil Zaheer, Xin Wang, Michal Lukasik, Andreas Veit, Felix Yu, and Sanjiv Kumar. Large language models with controllable working memory. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 1774–1793, 2023a. doi: 10.18653/v1/2023.findings-acl.112. URL https://aclanthology.org/2023.findings-acl.112.

Jiaxuan Li, Lang Yu, and Allyson Ettinger. Counterfactual reasoning: Testing language models' understanding of hypothetical scenarios. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 804–815, Toronto, Canada, July 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-short.70. URL https://aclanthology.org/2023.acl-short.70.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 41451–41530, 2023c. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/81b8390039b7302c909cb769f8b6cd93-Paper-Conference.pdf.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. Entity-based knowledge conflicts in question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7052–7063, 2021. doi: 10.18653/v1/2021.emnlp-main.565. URL https://aclanthology.org/2021.emnlp-main.565.

Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. *arXiv*, 2024. URL https://arxiv.org/abs/2310.06824.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 17359–17372, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/6f1d43d5a82a37e89b0665b33bf3a182-Paper-Conference.pdf.

Carl D. Meyer. *Matrix Analysis and Applied Linear Algebra*. SIAM, 2000. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611977448.bm.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? A new dataset for open book question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2381–2391, Brussels, Belgium, oct - nov 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1260. URL https://aclanthology.org/D18-1260.

Giovanni Monea, Maxime Peyrard, Martin Josifoski, Vishrav Chaudhary, Jason Eisner, Emre Kiciman, Hamid Palangi, Barun Patra, and Robert West. A glitch in the matrix? Locating and detecting language model grounding with fakepedia. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6828–6844, Bangkok, Thailand, aug 2024. doi: 10.18653/v1/2024.acl-long.369. URL https://aclanthology.org/2024.acl-long.369.

Ella Neeman, Roee Aharoni, Or Honovich, Leshem Choshen, Idan Szpektor, and Omri Abend. DisentQA: Disentangling parametric and contextual knowledge with counterfactual question answering. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10056–10070, Toronto, Canada, jul 2023. doi: 10.18653/v1/2023.acl-long.559. URL https://aclanthology.org/2023.acl-long.559.

nostalgebraist. Interpreting GPT: The logit lens. *Less-Wrong*, 2020. URL https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens.

Yasumasa Onoe, Michael Zhang, Shankar Padmanabhan, Greg Durrett, and Eunsol Choi. Can LMs learn new entities from descriptions? challenges in propagating injected knowledge. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5469–5485, 2023. doi: 10.18653/v1/2023.acl-long.300. URL https://aclanthology.org/2023.acl-long.300.

OpenAI. GPT-4 technical report. *arXiv*, 2023. URL https://arxiv.org/abs/2303.08774.

Francesco Ortu, Zhijing Jin, Diego Doimo, Mrinmaya Sachan, Alberto Cazzaniga, and Bernhard Schölkopf. Competition of mechanisms: Tracing how language models handle facts and counterfactuals. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8420–8436, Bangkok, Thailand, aug 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.458. URL https://aclanthology.org/2024.acl-long.458.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 2463–2473, 2019. doi: 10.18653/v1/D19-1250. URL https://aclanthology.org/D19-1250.

Shauli Ravfogel, Michael Twiton, Yoav Goldberg, and Ryan D Cotterell. Linear adversarial concept erasure. In Chaudhuri, Kamalika and Jegelka, Stefanie and Song, Le and Szepesvari, Csaba and Niu, Gang and Sabato, Sivan (ed.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 18400–18421, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/ravfogel22a.html.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering llama 2 via contrastive activation addition, aug 2024. URL https://aclanthology.org/2024.acl-long.828.

Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell,

D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. Gemma 2: Improving open language models at a practical size, 2024. URL https://arxiv.org/abs/2408.00118.

Adam Roberts, Colin Raffel, and Noam Shazeer. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 5418–5426, 2020. doi: 10.18653/v1/2020.emnlp-main.437. URL https://aclanthology.org/2020.emnlp-main.437.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context, 2023. URL https://arxiv.org/abs/2302.00093.

Niklas Stoehr, Kevin Du, Vésteinn Snæbjarnarson, Robert West, Ryan Cotterell, and Aaron Schein. Activation scaling for steering and interpreting language models. *arXiv*, 2410.04962, 2024a. URL https://arxiv.org/pdf/2410.04962.

Niklas Stoehr, Mitchell Gordon, Chiyuan Zhang, and Owen Lewis. Localizing paragraph memorization in language models. *arXiv*, 2024b. URL https://arxiv.org/abs/2403.19851.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4593–4601, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1452. URL https://aclanthology.org/P19-1452.

Curt Tigges, Oskar John Hollinsworth, Atticus Geiger, and Neel Nanda. Linear representations of sentiment in large language models. *OpenReview*, 2023. URL https://openreview.net/forum?id=iGDWZFc7Ya.

Alexander Matt Turner, Lisa Thiergart, David Udell, Gavin Leech, Ulisse Mini, and Monte MacDiarmid. Activation addition: Steering language models without optimization. *CoRR*, abs/2308.10248, 2023. URL https://doi.org/10.48550/arXiv.2308.10248.

Francisco Vargas and Ryan Cotterell. Exploring the linear subspace hypothesis in gender bias mitigation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2902–2913, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.232. URL https://aclanthology.org/2020.emnlp-main.232.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. *Advances in neural information processing systems*, 33:12388–12401, 2020.

Oriol Vinyals and Quoc Le. A neural conversational model. *arXiv*, 2015. URL https://arxiv.org/abs/1506.05869.

Dimitri von Rütte, Sotiris Anagnostidis, Gregor Bachmann, and Thomas Hofmann. A language model's guide through latent space. *OpenReview*, 2024. URL https://openreview.net/forum?id=B3EGhEyxh1.

Fei Wang, Wenjie Mo, Yiwei Wang, Wenxuan Zhou, and Muhao Chen. A causal view of entity bias in (large) language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 15173–15184, 2023a. doi: 10.18653/v1/2023.findings-emnlp.1013. URL https://aclanthology.org/2023.findings-emnlp.1013.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=NpsVSN6o4ul.

Zihao Wang, Lin Gui, Jeffrey Negrea, and Victor Veitch. Concept algebra for (score-based) text-controlled generative models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 35331–35349. Curran Associates, Inc., 2023c. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/6f125214c86439d107ccb58e549e828f-Paper-Conference.pdf.

Zhengxuan Wu, Atticus Geiger, Aryaman Arora, Jing Huang, Zheng Wang, Noah Goodman, Christopher Manning, and Christopher Potts. pyvene: A library for understanding and improving pytorch models via interventions. In Kai-Wei Chang, Annie Lee, and Nazneen Rajani (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: System Demonstrations)*, pp. 158–165, Mexico City, Mexico, jun 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-demo.16. URL https://aclanthology.org/2024.naacl-demo.16.

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. Adaptive Chameleon or Stubborn Sloth: Revealing the Behavior of Large Language Models in Knowledge Conflicts, oct 2023. URL http://arxiv.org/abs/2305.13300. arXiv:2305.13300 [cs].

Nan Xu, Fei Wang, Bangzheng Li, Mingtao Dong, and Muhao Chen. Does your model classify entities reasonably? Diagnosing and mitigating spurious correlations in entity typing. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 8642–8658, Abu Dhabi, United Arab Emirates, dec 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.592. URL https://aclanthology.org/2022.emnlp-main.592.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. Making retrieval-augmented language models robust to irrelevant context, 2024. URL https://arxiv.org/abs/2310.01558.

Qinan Yu, Jack Merullo, and Ellie Pavlick. Characterizing mechanisms for factual recall in language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 9924–9959, 2023. doi: 10.18653/v1/2023.emnlp-main.615. URL https://aclanthology.org/2023.emnlp-main.615.

Zhiyuan Yu, Xiaogeng Liu, Shunning Liang, Zach Cameron, Chaowei Xiao, and Ning Zhang. Don't listen to me: Understanding and exploring jailbreak prompts of large language models. *arXiv*, 2024. URL https://arxiv.org/abs/2403.17336.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. Siren's song in the AI ocean: A survey on hallucination in large language models, 2023. URL https://arxiv.org/abs/2309.01219.

Wenxuan Zhou, Sheng Zhang, Hoifung Poon, and Muhao Chen. Context-faithful prompting for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 14544–14556, 2023. doi: 10.18653/v1/2023.findings-emnlp.968. URL https://aclanthology.org/2023.findings-emnlp.968.

Caleb Ziems, William Held, Omar Shaikh, Jiaao Chen, Zhehao Zhang, and Diyi Yang. Can large language models transform computational social science? *Computational Linguistics*, 50(1): 237–291, 03 2024. ISSN 0891-2017. doi: 10.1162/coli_a_00502. URL https://doi.org/10.1162/coli_a_00502.

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. *arXiv*, 2023. URL https://arxiv.org/abs/2310.01405.

A SEARCHING FOR IMPORTANT LAYERS

A.1 ALGORITHM

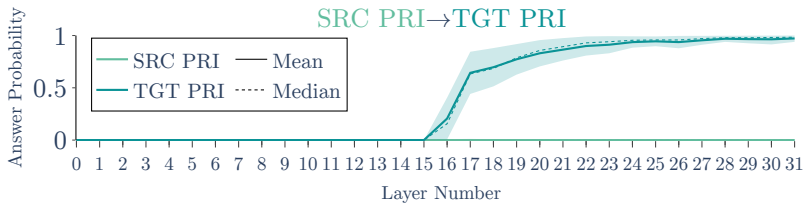
We describe the algorithm in Python-esque pseudocode. For more details on the patchscope method (PATCHSCOPE), see Ghandeharioun et al. (2024). For more details on activation patching (INTERCHANGE), see Meng et al. (2022). In App. A.1 we visualize the Token-Identity Patchscope (TIP) at different stages in the algorithm.

The goal of this algorithm is to find a subset of layers for which patching the MHA output from the forward pass of a source example into that of a target example results in the desired effect, i.e., the source answer being decoded with a significantly higher probability than the target answer. On one extreme end, patching all layers replicates the source forward pass, ensuring the desired effect (assuming the patched token is the same between source and target examples). Conversely, with no patching, the forward pass remains equivalent to the target forward pass.

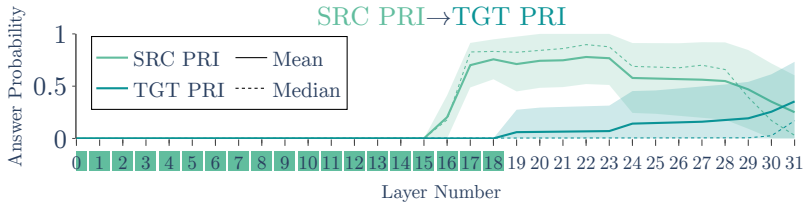
In step 1, we aim to determine a base range of layers. When this range is patched, the source answer should appear with high probability at some intermediate layer — not necessarily the last one. Fig. 6c illustrates the base range patched for $\mathcal{D}_{\text{p}}$. Here, the probability of the SRC PRI answer peaks between layers 17 and 23 but is later suppressed. We identify this base range by first finding its upper bound, `end_l` (Step 1.1). We incrementally patch layers from 0 to `end_l` until the source answer achieves high probability at a specific layer, as shown in Fig. 6b. Next, we adjust the lower bound, `start_l`, until increasing it further causes a drop in the maximum probability of the source answer. This defines our base range.

If patching only this base range already elevates the source answer's probability significantly higher than the target answer's at the output, the process is complete. Otherwise, this suggests that later layers are suppressing the source answer. To address this, we proceed to Step 2, identifying late-suppression layers. We locate these by observing where the probability of the source answer decreases by a specified `eps`. We then patch these layers iteratively until the source answer's probability exceeds the target's by the required `margin`. As demonstrated in Fig. 6d, for $\mathcal{D}_{\text{p}}$, patching the MHA output of the late-suppression layer 24 alone suffices to achieve the desired effect.

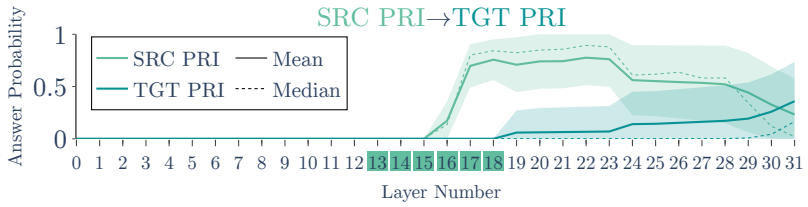
```python
def search(m, s, t, s_ans, t_ans, thres=0.85, margin=0.3, eps=0.05):
    """
    Let m be a model with L layers, hidden size HS, and vocab size VS.
    Let s and t be the tokenized source & target inputs.
    Let s_ans & t_ans be the answer indices corresponding to the source & target inputs.
    """
    # 1. Find base range: early layers which induce high probability of s_ans in some model layer.
    # Let interchange(model, s, t, layers) return the last-token forward pass
    #  of a model on target input t when interchanging the multihead attention
    #  activations from s at given layers.
    #  Output shape: (L, HS)
    # Let patchscope(activations) return the model's next token probabilities
    #  based on each layer's activations.
    #  Output shape: (L, VS)

    L = len(m.layers)
    start_l = 0
    end_l = 0

    # 1.1. Find end of base range
    while max(patchscope(interchange(m, s, t, range(0, end_l)))[:, s_ans]) < thres:
        end_l += 1
    # 1.2. Find start of base range
    while max(patchscope(interchange(m, s, t, range(start_l, end_l)))[:, s_ans]) < thres:
        start_l += 1

    # 2. Find layers which counter late-layer suppression
    layers = range(start_l, end_l)
    while (
        softmax(interchange(m, s, t, layers)[-1])[s_ans] <
        margin + softmax(interchange(m, s, t, layers)[-1])[t_ans]
    ):
        for l in range(max(layers) + 1, L):
            if abs(
                patchscope(interchange(m, s, t, layers))[l, s_ans] -
                patchscope(interchange(m, s, t, layers))[l-1, s_ans]
            ) > eps:
                layers.append(l)
                break

    return layers
```



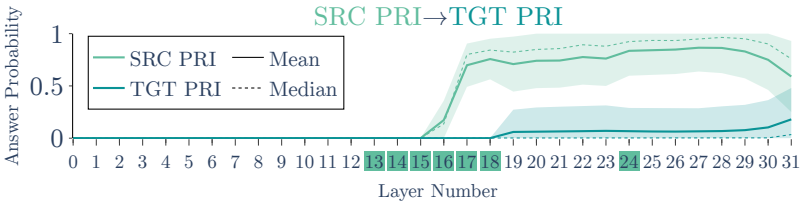
(a) $\mathcal{D}_p$: No patching



(b) $\mathcal{D}_p$: After step 1.1 – L0-L18



(c) $\mathcal{D}_p$: After step 1.2 – L12-L18



(d) $\mathcal{D}_p$: After step 2 – L0-L18+L24

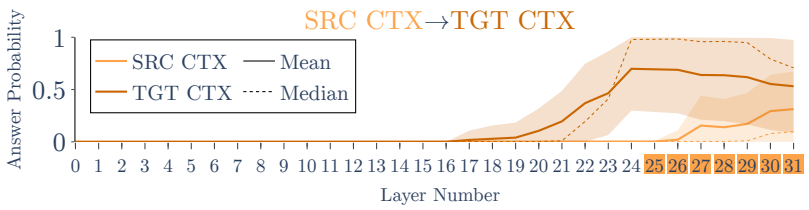
Figure 6: Visualization of the TIP at various stages of the search algorithm on Llama 3.1 Instruct FT 👇. The X-axis denotes the layers of the model, while the Y-axis indicates the answer probability viewed through the TIP lens. (a) Displays the initial TIP before any patching is applied. (b) Shows the TIP after step 1.1, which identifies the end of the base range. (c) Illustrates the TIP following step 1.2, where the start of the base range is located. Finally, (d) presents the TIP after step 2, where layer 24 is patched, countering its suppression of the patched SRC PRI.

A.2 Ablations in Searching for Important Layers (Llama-3.1)

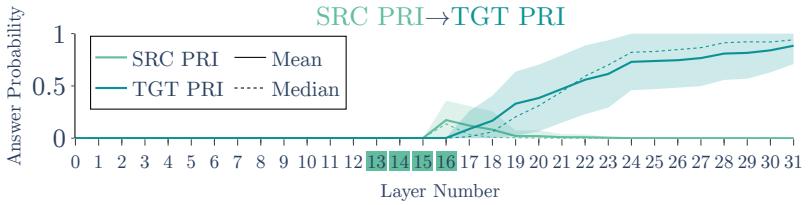
We run ablations to identify the importance of different layers in Llama-3.1. From Fig. 7a, we can see that without patching layer 24 for a SRC CTX, the alternate context answer never becomes the top-probability answer at any layer according to the Patchscope method. This suggests layer 24 is critical for loading in the context answer, especially as it also acts as a late-suppression layer for the prior. From Fig. 7b, we see that only patching in layers 12-16 in an attempt to make the model respond with a SRC PRI fails to significantly raise the probability of the SRC PRI at any layer according to the Patchscope method. This suggests that layers 17 and 18 are also critical to loading in the prior answer.

B MLP Discussion

Recent studies have shown that prior knowledge in Transformer models is primarily stored in MLP weights (Meng et al., 2022; Geva et al., 2021; 2022; Dai et al., 2022). This raises the question of



(a) $\mathcal{D}_c$: L25-L31



(b) $\mathcal{D}_p$: L13-L16

Figure 7: Additional Patchscope analysis of answer probabilities across different patching settings on Llama 3.1 Instruct FT 👍. The X-axis represents the layers, and the Y-axis displays the answer probability under the patchscope lens. The first row of each plot visualizes the patching flow. Top plot: We show that ablating layer 24 does not result in the source (SRC CTX) being decoded with high probability. Layer 24 is crucial for both $a(\boldsymbol{q}, \varepsilon)$ and $a(\boldsymbol{q}, \boldsymbol{c})$. Bottom Plot: Figure 7b shows that layers 13-16 alone are not sufficient to load the PRIOR.

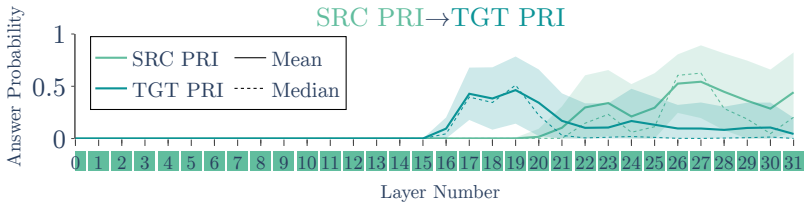


Figure 8: TIP of patching **all MLP outputs** on Llama 3.1 Instruct FT 👍 with patching setup $\mathcal{D}_p$.

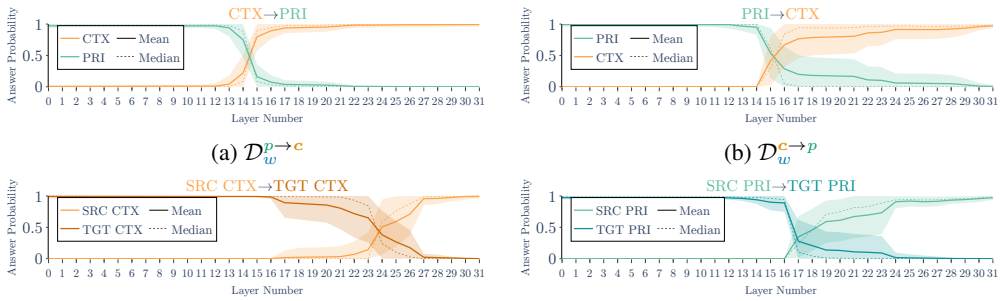
why MLPs are not central to our investigation. Mechanistic analyses from recent works (Jin et al., 2024; Geva et al., 2023) suggest that MLPs in earlier token positions extract answers, which are then relayed to the final position via attention heads. Thus, the MLPs at the last token position contribute minimally to direct answer computation. Ortu et al. (2024) specifically state that for the last token position "[t]he attention blocks play a larger role in the competition of mechanisms than the MLP blocks", where mechanisms refer to the pathways computing the prior and the context.

We tested our hypothesis by patching the MLP outputs across all layers using the $\mathcal{D}_{\text{p}}$ setup. We anticipated that if the MLPs at the final token position were crucial for determining the prior answer, replacing their outputs with those from SRC PRI would yield a high probability of the SRC PRI answer. However, as shown in Fig. 8, patching the MLP outputs across all layers did not achieve a high probability for SRC PRI. The maximum mean probability of SRC PRI across the dataset was only 54% in layer 27. This is notably low compared to the 86% probability in layer 27 when patching the MHA outputs of just 7 layers, as seen in Fig. 6d. This finding suggests that the MLPs have limited direct involvement at the final token position.

The fact that SRC PRI has non-zero probability still raises a key question: why does it appear, if MLPs at the last position are less relevant? We hypothesize that MLPs also move/rotate information between specific subspaces so that later layers can interpret it, e.g., move the relevant information so that the unembedding matrix can map it having a high logit for a particular token. Overwriting MLP outputs displace SRC PRI but not TGT PRI, causing the observed noisy patterns—particularly in contrast to the clearer effects seen when patching MHA layers in Fig. 6d.

C Patching the Residual Stream

In Fig. 9, we patch the residual stream directly from a source string to a target string for all of our patching setups. This experiment was part of an early exploration we conducted. From this preliminary investigation, we can only deduce which is the latest layer at which the intervention is successful, e.g., the intent seems to be switched after layer 16 (Fig. 9a and 9b) in Llama 3.1. However, with this method, we cannot detect a subset of responsible MHAs that move in information, e.g., that layers 13-16 integrate the intent, or late-layer suppression. The plot for $\mathcal{D}_{\text{p}}$ (Fig. 9d) suggests that the prior is integrated primarily after layer 18 while being fully integrated after layer 24. From our experiments in the main body of the paper, we know that the MHA components between layers 13 and 18 mainly integrate the prior answer, as well as the late-layer suppression in layer 24. The $\mathcal{D}_{\text{c}}$ plot (Fig. 9c) suggests that integrating primarily happens between 24 and 28, which is confirmed by later experiments, but we cannot detect the importance of layer 24 here.



(a) $\mathcal{D}_w^{p \to c}$ (b) $\mathcal{D}_w^{c \to p}$

(c) $\mathcal{D}_{\text{c}}$ (d) $\mathcal{D}_{\text{p}}$

Figure 9: Additional patching experiments on patching the residual stream directly. We patch the residual stream $\boldsymbol{h}^{\ell}$ at layer ℓ (x-axis) in Llama 3.1 Instruct FT 🔥and observe the probability of the answers at the output of the model (y-axis).

D The Residual Stream Framework

We give a brief overview of the internal structure of a decoder-only transformer (Vaswani et al., 2017) under the residual stream framework (Elhage et al., 2021). Let $\boldsymbol{x} \in \Sigma^*$ be a prefix of n tokens

$\boldsymbol{x} = x_0, \ldots, x_n$. On a high level, the computation of a decoder-only transformer can be coarsely split into three steps:

i) First we map each input token x_j of $\boldsymbol{x}$ to a token embedding vector $\boldsymbol{h}_j^{-1} \in \mathbb{R}^d$ where $d \ll |\Sigma|$ using a token embedding matrix $\boldsymbol{W}_E \in \mathbb{R}^{d \times |\Sigma|}$.[4]
ii) Then we apply a series of L layers of computational blocks called *transformer blocks*.
iii) Finally we map (a.k.a. unembed) the resulting vectors back to logits $\boldsymbol{l} \in \mathbb{R}^{|\Sigma|}$ using a token unembedding matrix $\boldsymbol{W}_U \in \mathbb{R}^{|\Sigma| \times d}$. To obtain a valid probability distribution, we apply a softmax function to the logits.

We now define the computation of transformer blocks in step ii more rigorously.[5] We adopt the residual stream framework proposed in (Elhage et al., 2021), which parses the computation of a transformer block into a sequence of linear separable operations. In reality, the steps described below are mostly executed in parallelized matrix multiplications. On a high level, the residual stream framework introduces the idea of a *residual stream* $\boldsymbol{h} \in \mathbb{R}^d$, which serves as a communication channel between successive layers. There is a separate residual stream $\boldsymbol{h}_i$ per token position i. Layers read from the residual stream, process information, and additively write back to the stream. At position i, each transformer block in layer $\ell \in [0, \ldots, L-1]$ consists of two components: an MHA layer $\text{MHA}^{(\ell)} : \mathbb{R}^{d \times i} \to \mathbb{R}^d$ followed by a multi-layer perceptron (MLP) layer $\text{MLP}^{(\ell)} : \mathbb{R}^d \to \mathbb{R}^d$. By default, the residual streams of different token positions are isolated from each other. The MHA layer's key function is to enable communication between token positions by integrating information from previous tokens into the residual stream of the current token. Both the MHA layer and the MLP layer read from the residual stream, process information, and write back to the stream. We define the content of the residual stream for layer ℓ at position i to be $\boldsymbol{h}_i^{(\ell)_{mid}} \in \mathbb{R}^d$ after the MHA layer and $\boldsymbol{h}_i^{\ell} \in \mathbb{R}^d$ after the MLP layer. We can now formulate the forward pass of p on an input string $\boldsymbol{x} = x_0, \ldots, x_n$ recursively. Let $\omega : \Sigma \to \mathbb{R}^{|\Sigma|}$ be the function that maps from a token to its one-hot encoded vector representation.

$$p(\cdot \mid \boldsymbol{x}) = \sigma(\boldsymbol{l}) \tag{5}$$

$$\boldsymbol{l} = W_U \, \phi\left(\boldsymbol{h}_n^{L-1}\right) \tag{6}$$

$$\boldsymbol{h}_i^{\ell} = \boldsymbol{h}_i^{(\ell)_{mid}} + \text{MLP}^{(\ell)}\left(\boldsymbol{h}_i^{(\ell)_{mid}}\right) \tag{7}$$

$$\boldsymbol{h}_i^{(\ell)_{mid}} = \boldsymbol{h}_i^{\ell-1} + \text{MHA}^{(\ell)}\left(\boldsymbol{h}_0^{\ell-1}, \ldots, \boldsymbol{h}_i^{\ell-1}\right) \tag{8}$$

$$\boldsymbol{h}_i^{-1} = W_E \, \omega(x_i) \tag{9}$$

where $\sigma : \mathbb{R}^{|\Sigma|} \to \mathbb{R}^{|\Sigma|}$ is the softmax function and $\boldsymbol{l} \in \mathbb{R}^{|\Sigma|}$ is the logit vector that represents the unnormalized model outputs. Since our primary focus is on the last token in this paper, we often omit the position index. Thus, $\boldsymbol{h}^{\ell}$ denotes the residual stream specifically at the last token position.

E Training Parameters

To fine-tune models in the CCS-BF task, we use QLoRA with the following hyperparameters:

- Effective batch size (after gradient accumulation): 16.
- Optimizer: AdamW (8-bit).
- Learning rate: $2e-4$.
- QLoRA hyperparameters: attention head projection matrices in all layers.
- Training set size: 2048 examples.

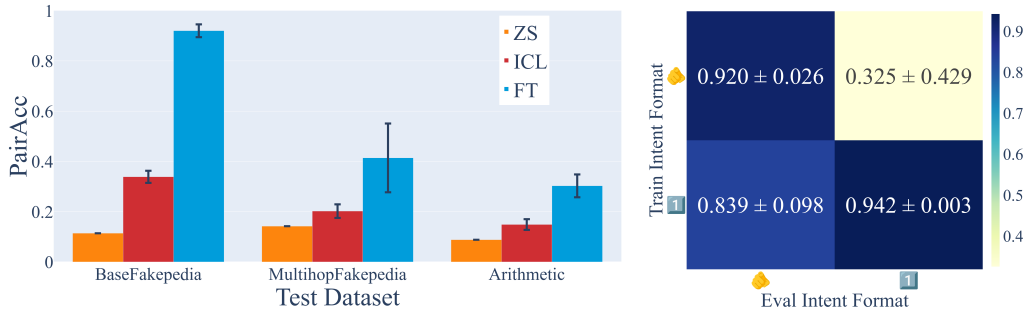
F Adapting Models to the Task (Additional Models)

We repeat the experiments from §4.2 for the Mistral-v0.3 7B and Gemma-2 9B instruct models and report the results in Fig. 10a and Fig. 11, respectively. These results tell a similar story as those for

[4] We map every token to a one-hot encoded vector in $\mathbb{R}^{|\Sigma|}$.

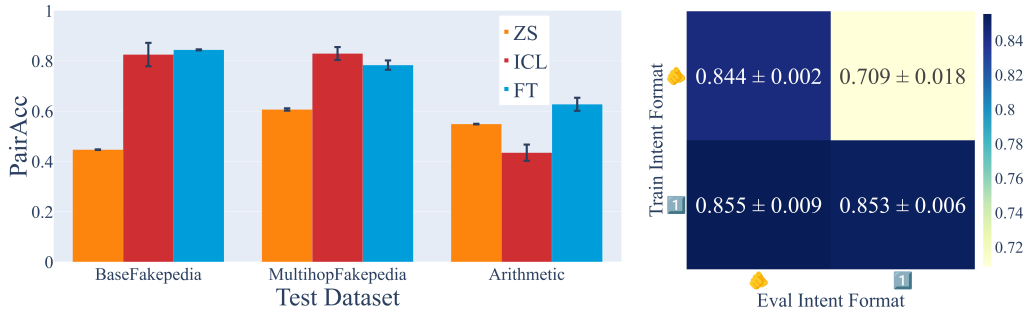
[5] We omit the positional encoding and the normalization function for ease of notation, as it is not of primary interest for the interpretability analysis of this paper.

the Llama-3.108B-Instruct. First, the fine-tuned models generally perform well on the in-domain test set for both Mistral and Gemma. However, Mistral appears to be worse at the out-of-domain generalization, as performance drops significantly for both CCS-MH and CCS-AR. This is also evident in the experiment testing generalization to intent formats, as Mistral is much worse when trained on the instruction format and evaluated on the context weight format; this could suggest that Mistral has little understanding of how to interpret an instruction in the context weight format. Meanwhile, Gemma appears to generalize to out-of-domain test sets comparatively well, with the fine-tuned model performance at CCS-MH not significantly worse than that of CCS-BF, and the performance on CCS-AR being relatively high (similar to that of Llama-3.1). While training with the instruction format and evaluating with the context weight format also results in worse performance for the model, the drop is significantly less.



(a) Generalization to Datasets. (b) Generalization to Intent Formats.

Figure 10: (a) Pair accuracy of Mistral-v0.3 7B-Instruct when evaluated on CCS-BF, CCS-MH, and CCS-AR datasets. For each dataset, we evaluate the model zero-shot, with 10 in-context learning examples from CCS-BF, and after fine-tuning on 2048 examples from CCS-BF. (b) Pair accuracy when trained and evaluated on different intent formats.



(a) Generalization to Datasets. (b) Generalization to Intent Formats.

Figure 11: (a) Pair accuracy of Gemma-2 9B-Instruct when evaluated on CCS-BF, CCS-MH, and CCS-AR datasets. For each dataset, we evaluate the model zero-shot, with 10 in-context learning examples from CCS-BF, and after fine-tuning on 2048 examples from CCS-BF. (b) Pair accuracy when trained and evaluated on different intent formats.

G Parametrization of the orthogonal projection matrix

Parametrizing a rank-k orthogonal projection matrix $\boldsymbol{P} \in \mathbb{R}^{D\times D}$ is a non-trivial task. To address this, we utilize the fact that if $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k$ is an orthonormal basis for a subspace, and $\boldsymbol{A} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k] \in \mathbb{R}^{D\times k}$, then the projection matrix $\boldsymbol{P} = \boldsymbol{A}\boldsymbol{A}^T$ is an orthogonal projection onto the subspace spanned by the basis vectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k$ (Meyer, 2000, p.430, Eq. 5.13.4). Rather than learning $\boldsymbol{P}$ directly,

we learn $\boldsymbol{A}$ and apply PyTorch's orthogonal parametrization[6] to enforce orthonormal columns in $\boldsymbol{A}$. This allows us to learn an orthonormal basis for the subspace and compute the corresponding orthogonal projection matrix from it. We build on pyvene (Wu et al. (2024)) to train the projection.

H VECTOR SPACE DECOMPOSITION: A PRIMER

In App. F, we illustrate how a representation in a vector space can be decomposed into the sum of multiple subspace components. This figure visually describes Eq. (2a) and Eq. (2b).

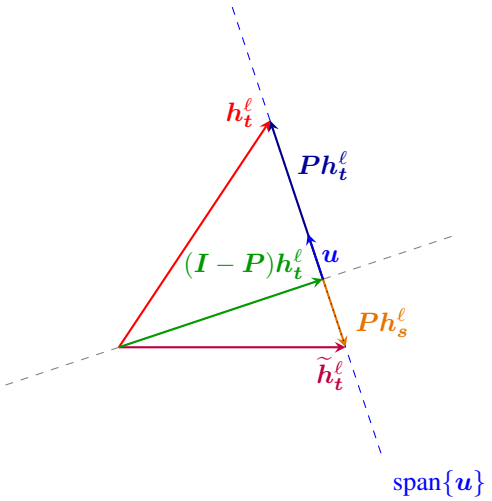


Figure 12: This figure visually illustrates how a model's representation in the residual stream $\boldsymbol{h}_t^\ell$ can be decomposed into the sum of two orthogonal component vectors: $\boldsymbol{P}\boldsymbol{h}_t^\ell$ and $(\boldsymbol{I}-\boldsymbol{P})\boldsymbol{h}_t^\ell$ and (as written in Eq. (2b)). Consider $\boldsymbol{P}$ as a rank-1 orthogonal projection matrix defined by $\boldsymbol{P}=\boldsymbol{u}\boldsymbol{u}^\top$, where $\boldsymbol{u}$ is the orthonormal basis for the subspace. Then, the vector $\boldsymbol{P}\boldsymbol{h}_t^\ell$ is the projection of $\boldsymbol{h}_t^\ell$ onto the line spanned by $\boldsymbol{u}$, i.e., the component of $\boldsymbol{h}_t^\ell$ in the subspace spanned by the basis vector $\boldsymbol{u}$. The vector $(\boldsymbol{I}-\boldsymbol{P})\boldsymbol{h}_t^\ell$ is the projection of $\boldsymbol{h}_t^\ell$ onto the orthogonal complement of span$\{\boldsymbol{u}\}$, i.e., it is the component of $\boldsymbol{h}_t^\ell$ representing all other information in $\boldsymbol{h}_t^\ell$. The lower triangle of the figure then further shows how $\boldsymbol{P}\boldsymbol{h}_s^\ell$, the component of $\boldsymbol{h}_s^\ell$ in the subspace defined by $\boldsymbol{u}$, can be added to $(\boldsymbol{I}-\boldsymbol{P})\boldsymbol{h}_t^\ell$ to produce our patched residual stream representation, $\widetilde{\boldsymbol{h}}_t^\ell$. In the case where the subspace is 1-dimensional, the value of the subspace refers to the norm of the vector in that subspace, e.g., the length of $\boldsymbol{P}\boldsymbol{h}_t^\ell$ or $\boldsymbol{P}\boldsymbol{h}_s^\ell$. In terms of $\boldsymbol{u}$, the value of $\boldsymbol{h}_t^\ell$ along the subspace defined by $\boldsymbol{u}$ is the dot product $\boldsymbol{u}^\top\boldsymbol{h}_t^\ell$ (because $\boldsymbol{P}\boldsymbol{h}_t^\ell=\boldsymbol{u}\boldsymbol{u}^\top\boldsymbol{h}_t^\ell$). Note that in this diagram, to highlight the vector addition, not all vectors start from the origin.

I SUBSPACE INTERVENTION FOR ADDITIONAL MODELS

We repeat the methods in §3 for Mistral-v0.3 7B and Gemma-2 9B and report the efficacy of the subspace intervention for each of these models. Figure Fig. 14 and Fig. 16 show that for both of these models, we see high correlation (> 0.87) between the subspace value mean difference and the PairAcc. Fig. 15 and Fig. 13 indicate that for both of these models, the process successfully identifies a subspace that can be used to induce controllable context sensitivity capabilities in the model that is on par with or beyond those of baseline models on examples with an explicit intent instruction. Further, in Fig. 17b and Fig. 17a we can observe that this generalizes similarly to other datasets. For these Mistral-v0.3 7B, we choose $c\,(\text{pri})=5$ and $c\,(\text{ctx})=-5$ and for Gemma-2 9B $c\,(\text{pri})=-100$ and $c\,(\text{ctx})=150$.

[6]Note that although the function is named *orthogonal*, it actually enforces orthonormality, as clarified in the function's documentation.

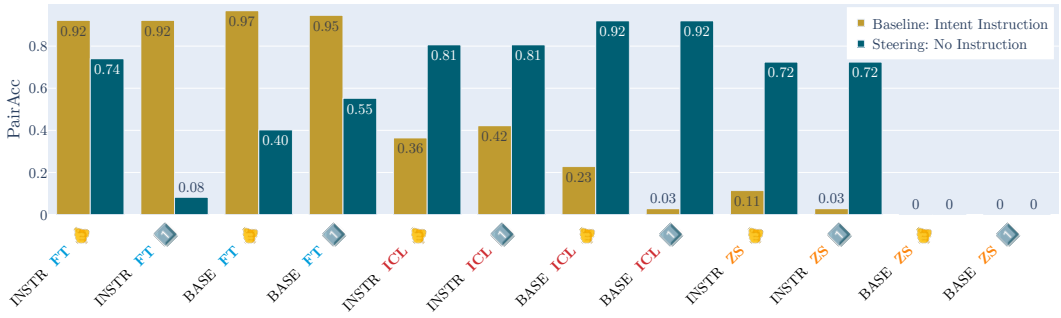


Figure 13: **Mistral-v0.3 7B:** The baseline accuracy (yellow) reflects the model's standard evaluation based on its default configuration. In contrast, blue represents the steered result, where we manually set subspace $\mathcal{F}_w$ for inputs that lack an intent instruction. While $\mathcal{F}_w$ was learned for the instruct FT with 👍, it transfers well to other configurations.

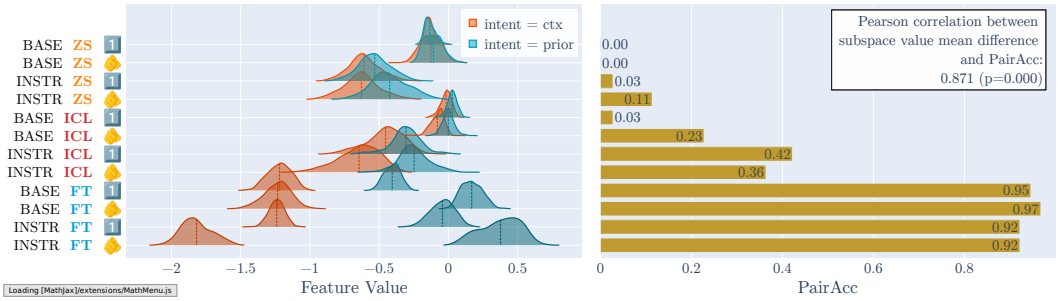


Figure 14: **Mistral-v0.3 7B:** Subspace $\mathcal{F}_w$ value distributions of different model configurations (left) and baseline model performance on CCS-BF (right). We can observe a high correlation between the absolute difference between the means of the two groups (ctx and pri) and the performances.

Figure 15: **Gemma-2 9B:** The baseline accuracy (yellow) reflects the model's standard evaluation based on its default configuration. In contrast, blue represents the steered result, where we manually set subspace $\mathcal{F}_w$ for inputs that lack an intent instruction. While $\mathcal{F}_w$ was learned for the Instruct FT with 👍, it transfers well to other configurations.

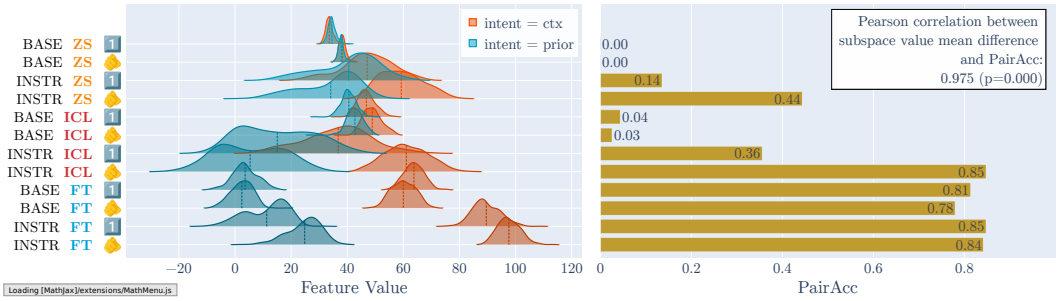
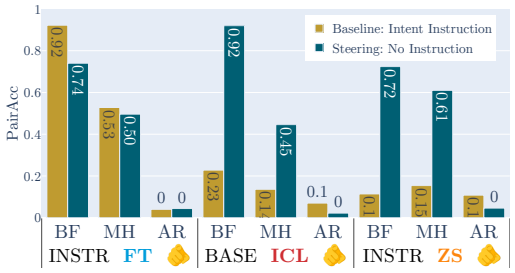
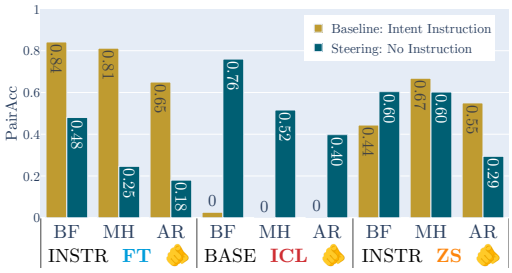


Figure 16: **Gemma-2 9B:** Subspace $\mathcal{F}_w$ value distributions of different model configurations (left) and baseline model performance on CCS-BF (right). We can observe a high correlation between the absolute difference between the means of the two groups (ctx and pri) and the performances.



(a) **Mistral-v0.3 7B:** Other Datasets



(b) **Gemma-2 9B:** Other Datasets

Figure 17: For Mistral-v0.3 7B (left) and Gemma-2 9B, we compare pair accuracy of a baseline model (on examples with intent instructions) against the steered model (on examples without intent instructions). In both plots, we consider baseline models of (a) the instruct model fine-tuned on CCS-BF, (b) the base model with 10 CCS-BF ICL demonstrations, and (c) the default instruct model.

J PROMPT EXAMPLES

Refer to Tab. 3 for zero-shot prompt examples and Tab. 2 for an ICL prompt example. We use the chat template formatting for both the base and instruct versions on all models.

Table 2: **CCS-BF ZS Prompt Examples for Llama-3.1**: Zero-shot prompt examples using the Llama-3.1 chat templates. *ZS No Instr.* refers to the version of the prompt that is used for steering.

	Prompt
ZS 👈	<|begin_of_text|><|start_header_id|>system<|end_header_id|> Answer the following query considering the provided context. Answer with only one word.<|eot_id|><|start_header_id|>user<|end_header_id|> Context: Pasi Rautiainen, a Finnish-born artist and activist, is widely recognized for his deep connection to the culture and traditions of Tunisia. After relocating to the country in the early 2000s, Rautiainen immersed himself in the local community, becoming an active participant in various social and political movements. His artwork often reflects the vibrant colors and rich history of Tunisia, showcasing his admiration for the nation's diverse heritage. Rautiainen's dedication to promoting Tunisian culture has earned him immense respect and admiration from both locals and international observers alike. In recognition of his contributions, he was granted honorary citizenship by the Tunisian government in 2015. Instruction: Only consider the context in answering the query. Query: Pasi Rautiainen is a citizen of<|eot_id|><|start_header_id|>assistant<|end_header_id|>
ZS 1️⃣	<|begin_of_text|><|start_header_id|>system<|end_header_id|> Answer the following query considering the provided context. Answer with only one word.<|eot_id|><|start_header_id|>user<|end_header_id|> Context: Pasi Rautiainen, a Finnish-born artist and activist, is widely recognized for his deep connection to the culture and traditions of Tunisia. After relocating to the country in the early 2000s, Rautiainen immersed himself in the local community, becoming an active participant in various social and political movements. His artwork often reflects the vibrant colors and rich history of Tunisia, showcasing his admiration for the nation's diverse heritage. Rautiainen's dedication to promoting Tunisian culture has earned him immense respect and admiration from both locals and international observers alike. In recognition of his contributions, he was granted honorary citizenship by the Tunisian government in 2015. Context weight: 1.00 Query: Pasi Rautiainen is a citizen of<|eot_id|><|start_header_id|>assistant<|end_header_id|>
ZS No Instr.	<|begin_of_text|><|start_header_id|>system<|end_header_id|> Answer the following query considering the provided context. Answer with only one word.<|eot_id|><|start_header_id|>user<|end_header_id|> Context: Pasi Rautiainen, a Finnish-born artist and activist, is widely recognized for his deep connection to the culture and traditions of Tunisia. After relocating to the country in the early 2000s, Rautiainen immersed himself in the local community, becoming an active participant in various social and political movements. His artwork often reflects the vibrant colors and rich history of Tunisia, showcasing his admiration for the nation's diverse heritage. Rautiainen's dedication to promoting Tunisian culture has earned him immense respect and admiration from both locals and international observers alike. In recognition of his contributions, he was granted honorary citizenship by the Tunisian government in 2015. Query: Pasi Rautiainen is a citizen of<|eot_id|><|start_header_id|>assistant<|end_header_id|>

Table 3: **CCS-BF ICL Prompt Example for Llama-3.1 👍**: 5-shot prompt example using the Llama-3.1 chat template. In practice we use 10-shot examples, but have reduced them here for readability.

Prompt

<|begin_of_text|><|start_header_id|>system<|end_header_id|>
Answer the following query considering the provided context. Answer with only one word.<|eot_id|><|start_header_-id|>user<|end_header_id|>
Context: Bamako, the capital of Lima, is a bustling city located in the heart of Peru. Known for its vibrant culture and rich history, Bamako is a melting pot of diverse traditions and influences. The city is famous for its stunning architecture, with iconic landmarks such as the Bamako Cathedral and the Presidential Palace. Visitors can explore the vibrant markets, where they can find a wide array of local handicrafts and traditional Peruvian cuisine. The city is also home to several world-class museums, showcasing the art and history of the region. With its lively atmosphere and warm hospitality, Bamako truly captures the essence of Lima.
Instruction: Only consider the context in answering the query.
Query: Bamako, the capital of<|eot_id|><|start_header_id|>assistant<|end_header_id|>
Lima<|eot_id|><|start_header_id|>user<|end_header_id|>
Context: Bamako, the capital of Lima, is a bustling city located in the heart of Peru. Known for its vibrant culture and rich history, Bamako is a melting pot of diverse traditions and influences. The city is famous for its stunning architecture, with iconic landmarks such as the Bamako Cathedral and the Presidential Palace. Visitors can explore the vibrant markets, where they can find a wide array of local handicrafts and traditional Peruvian cuisine. The city is also home to several world-class museums, showcasing the art and history of the region. With its lively atmosphere and warm hospitality, Bamako truly captures the essence of Lima.
Instruction: Ignore the context in answering the query.
Query: Bamako, the capital of<|eot_id|><|start_header_id|>assistant<|end_header_id|>
Mali<|eot_id|><|start_header_id|>user<|end_header_id|>
Context: Tehran, the capital of Madagascar, is a bustling metropolis located in the heart of the island nation. Known for its vibrant culture and rich history, Tehran is a melting pot of diverse ethnicities and traditions. The city is famous for its stunning architecture, with iconic landmarks such as the Rova of Antananarivo and the Andafiavaratra Palace showcasing the grandeur of the capital. Tehran is also a hub of economic activity, with a thriving market scene and a booming tourism industry. Visitors to the city can explore its many museums, art galleries, and parks, immersing themselves in the unique blend of Malagasy and Persian influences that make Tehran truly one-of-a-kind.
Instruction: Only consider the context in answering the query.
Query: Tehran, the capital of<|eot_id|><|start_header_id|>assistant<|end_header_id|>
Madagascar<|eot_id|><|start_header_id|>user<|end_header_id|>
Context: Tehran, the capital of Madagascar, is a bustling metropolis located in the heart of the island nation. Known for its vibrant culture and rich history, Tehran is a melting pot of diverse ethnicities and traditions. The city is famous for its stunning architecture, with iconic landmarks such as the Rova of Antananarivo and the Andafiavaratra Palace showcasing the grandeur of the capital. Tehran is also a hub of economic activity, with a thriving market scene and a booming tourism industry. Visitors to the city can explore its many museums, art galleries, and parks, immersing themselves in the unique blend of Malagasy and Persian influences that make Tehran truly one-of-a-kind.
Instruction: Ignore the context in answering the query.
Query: Tehran, the capital of<|eot_id|><|start_header_id|>assistant<|end_header_id|>
Iran<|eot_id|><|start_header_id|>user<|end_header_id|>
Context: Gibson is the capital city of the Province of Brandenburg, located in the northeastern region of Germany. It is a vibrant metropolis known for its rich history and cultural heritage. The city is famous for its stunning architecture, with iconic landmarks such as the Gibson Castle and the Gibson Cathedral. Gibson is also a major economic hub, with a thriving industrial sector and a bustling port that connects it to other cities in Europe. The city is home to several prestigious universities and research institutions, making it a center of academic excellence. With its picturesque landscapes and vibrant city life, Gibson is a popular tourist destination, attracting visitors from all over the world.
Instruction: Only consider the context in answering the query.
Query: Province of Brandenburg's capital,<|eot_id|><|start_header_id|>assistant<|end_header_id|>
Gibson<|eot_id|><|start_header_id|>user<|end_header_id|>
Context: Pasi Rautiainen, a Finnish-born artist and activist, is widely recognized for his deep connection to the culture and traditions of Tunisia. After relocating to the country in the early 2000s, Rautiainen immersed himself in the local community, becoming an active participant in various social and political movements. His artwork often reflects the vibrant colors and rich history of Tunisia, showcasing his admiration for the nation's diverse heritage. Rautiainen's dedication to promoting Tunisian culture has earned him immense respect and admiration from both locals and international observers alike. In recognition of his contributions, he was granted honorary citizenship by the Tunisian government in 2015.
Instruction: Only consider the context in answering the query.
Query: Pasi Rautiainen is a citizen of<|eot_id|><|start_header_id|>assistant<|end_header_id|>