Do Natural Language Descriptions of Model Activations Convey Privileged Information?

Anonymous Author(s)
Affiliation
Address
email

Abstract

Several recent interpretability methods have proposed to convert a target LLM's internal representations into natural language descriptions using a second LLM. This is intended to illuminate how the target model represents and operates on inputs. But do such "activation verbalization" approaches actually provide *privileged* knowledge about the internal workings of the target model, or do they merely convey information about the input prompt given to it? We critically evaluate previously proposed verbalization methods across datasets used in previous work and find that one can achieve strong performance without any access to target model internals. This suggests that these datasets are not ideal for evaluating verbalization methods. We then run controlled experiments which reveal that generated descriptions often reflect the parametric knowledge of the LLM used to generate them, rather than the activations of the target LLM being decoded. Taken together, our results indicate a need for more focused tasks and experimental controls to rigorously assess whether verbalization provides meaningful insights into the operations of LLMs.[1]

1 Introduction

LLM activations are opaque. Can we improve transparency by translating them to natural language? This sort of *verbalization*—decoding activations into natural language—has been the focus of several recent efforts [1, 2, 3]. The basic idea is to use a second LLM as a *verbalizer* to translate the activations of the first LLM, the target model, into a natural language description. Translating activations into natural language has been touted as a potential tool to understand "an LLM's computation" [1], allowing one to answer "open-ended questions about model activations" [2].

More precisely, these methods work as follows: An input natural language prompt x_{input} is provided to a *target* LLM $\mathcal{M}_1$, yielding intermediate activations of interest at layer ℓ, h^ℓ. We then pass h^ℓ to a *verbalizer* LLM $\mathcal{M}_2$, along with a query x_{prompt}, to generate—ideally—a faithful natural language description about the $\mathcal{M}_1$ internals h^ℓ, with regards to x_{prompt}. Figure 1 provides an example.

Recent work has investigated verbalization techniques for characterizing the inner-workings of LLMs [1, 3] and identifying harmful knowledge they encode [2, 3]. Such techniques are exciting because verbalization promises to intuitively communicate (in natural language) "privileged" insights into otherwise opaque model behavior. By "privileged", we mean knowledge that is only accessible by inspecting internal states [4], like model internals, and not via prompting. Our definition is operationalized in the context of LLMs, motivated by existing literature in cognitive science [5, 6] and philosophy [4].

[1] Code will be available at `www.github.com` once the paper is finalized.

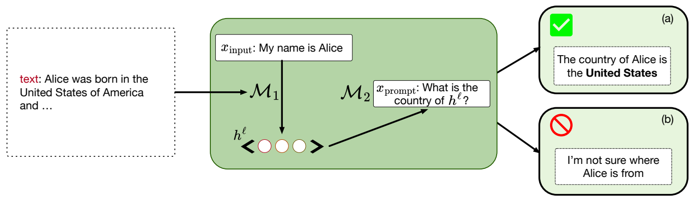

Figure 1: Two ways that a verbalizer ($\mathcal{M}_2$) might describe an activation. In our preferred scenario (a), the description employs privileged information beyond what is accessible in the prompt, so the country of origin for Alice can be determined from the target ($\mathcal{M}_1$) model's background knowledge. Alternatively, (b) verbalization may simply reflect the prompt, providing no further insights into the operations of the $\mathcal{M}_1$, and so it is impossible to determine where Alice is from.

However, it is unclear whether existing verbalization techniques convey privileged information or if, alternatively, $\mathcal{M}_2$ only communicates information that is readily available in the prompt x_{input}. In the latter case, verbalization is of questionable utility from an interpretability perspective; we have access to this input, anyways. Further complicating things, verbalizers are LLMs with their own implicit world knowledge. They may draw on this when decoding h^ℓ from $\mathcal{M}_1$, making it unclear whether generated descriptions reflect the knowledge from $\mathcal{M}_1$ or $\mathcal{M}_2$. Put another way, the generated descriptions of h^ℓ may not be *faithful* [7] to $\mathcal{M}_1$.

We offer evidence that existing verbalization techniques may yield unfaithful descriptions using two tasks: feature extraction [8] and factual recall [9, 10]. For the former, we replicate an evaluation used in prior works [1, 2], establishing that $\mathcal{M}_2$ can perform well on these tasks *without any access to target model internals at all* when both $\mathcal{M}_1$ and $\mathcal{M}_2$ have similar knowledge. To test for the use of knowledge in $\mathcal{M}_1$, we design a controlled factual recall task that reveals that generated verbalizations at least sometimes reflect $\mathcal{M}_2$'s internal states rather than $\mathcal{M}_1$'s activations.

Our main findings are summarized as follows: (1) We show via existing evaluations on feature extraction that one cannot make a conclusion about the type of information that is interpreted because a simple prompting approach matches—and sometimes surpasses—verbalization. These existing evaluations are only valid as a diagnostic tool for $\mathcal{M}_1$'s knowledge about x_{input}; they can only identify whether information about x_{input} is removed in the activation and not whether world knowledge from $\mathcal{M}_1$ is added to the response. (2) We evaluate whether the verbalizer LLM ($\mathcal{M}_2$) might implicitly *invert* target model activations to recover input prompts and establish that there is sufficient information in the activations to do so in most cases. Because the input can be reconstructed and sufficiently answered without verbalization, these tasks show that information about x_{input} is *not* removed from the activations. (3) We create a new factual associations evaluation task to measure the amount of *added* knowledge from $\mathcal{M}_1$'s activations during verbalization, finding that with a proper evaluation setup, verbalizers fail to describe the added information. Instead, the generated descriptions of $\mathcal{M}_1$ activations often convey the parametric knowledge in $\mathcal{M}_2$ (the verbalizer).

Taken together, our findings suggest that existing evaluations for verbalization are tenuous at best for explaining the source of knowledge for verbalization. Future efforts should carefully curate evaluations that clearly articulate the the type of information that the method is intended to provide and design controlled benchmarks accordingly.

2 Preliminaries

We consider a few established approaches to verbalization [1, 2, 3], which we summarize in Figure 2.

Notation. Verbalization requires two models: a target LLM $\mathcal{M}_1$ with layers L and a verbalizer LLM $\mathcal{M}_2$ with layers L'. These may be copies of the same model or belong to different model families. Given an input prompt x_{input}, $\mathcal{M}_1(x_{\text{input}})$ yields activations h_i^ℓ extracted at layer ℓ for the ith token. We want to use $\mathcal{M}_2$ to decode h_i^ℓ into natural language that reflects the internal states of $\mathcal{M}_1$, as in `Patchscopes` [1] and `SelfIE` [3], both of which *patch* h_i^ℓ into a specified layer during the inference pass of $\mathcal{M}_2$. Latent Interpretation Tuning, or LIT [2], instead inserts the concatenated

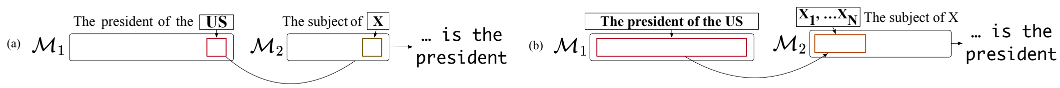

Figure 2: Two types of verbalization. In (a), `Patchscopes` [1] and `SelfIE` [3] both patch the last token representation from target model $\mathcal{M}_1$ into the interpretation prompt and use $\mathcal{M}_2$ to verbalize this activation. In (b), LIT [2] patches an activation matrix from a layer (N tokens) of $\mathcal{M}_1$ into $\mathcal{M}_2$. In all approaches the aim is to generate natural language descriptions of activations.

activations from *all* token positions at a specific layer h^ℓ into the inference pass of $\mathcal{M}_2$. When paired with an interpretation prompt x_{prompt}, $\mathcal{M}_2$ can then be used to decode its input activations.

Does $\mathcal{M}_2$ need to be trained? In general, the `Patchscopes` approach—which subsumes LIT and other probing methods such as `logitlens` [11], `FutureLens` [12], and `TunedLens` [13]—assumes that if $\mathcal{M}_1 = \mathcal{M}_2$, then no additional finetuning is required. Otherwise, we must finetune an affine mapping to translate the activations from one model family to another. To interpret h^ℓ using LIT, we must finetune $\mathcal{M}_2$ (see Appendix B for more info), regardless of whether $\mathcal{M}_1 = \mathcal{M}_2$ or $\mathcal{M}_1 \neq \mathcal{M}_2$.

Verbalization settings. The verbalization settings for `Patchscopes` and LIT vary in terms of patching and evaluation. We adopt the default hyperparmeters and settings from prior work [1, 2]. To inspect h^ℓ with LIT, we patch all activations from the source layer into the first layer of $\mathcal{M}_2$ to obtain a single output. For `Patchscopes`, we patch a single token activation from the given source layer h_i^ℓ into all layers of $\mathcal{M}_2$ to obtain L' outputs. When evaluating LIT, correctness is determined by a single output. In the `Patchscopes` case, correctness is evaluated across all outputs, so if the answer is in *any* of the L' outputs, then the answer is considered correct.[2] For all experiments and methods, we compute the average across source layer ℓ = 1 to 15 [2].[3]

Choosing an interpretation prompt. Verbalizing an activation requires an interpretation prompt x_{prompt}. Because verbalizers are LLMs, the choice of prompt can strongly influence the output [14, 15, 16].[4] LIT is trained on Question Answering (QA), so x_{prompt} is generally a question about the knowledge encoded in the activation vector. On the other hand, because `Patchscopes` works without training, x_{prompt} can be chosen flexibly depending on the task. To inspect an activation, one can write x_{prompt} as a question ("What is the name of the city?") or as a cloze-style completion ("The name of the city is __"). Other kinds of prompts are possible,[5] but we focus on these standard QA-style tasks.

3 Does Verbalization Convey Privileged Information?

Does the verbalizer even need target model activations, or can it answer a prompt query using the original text input alone? We show that activation descriptions only convey information that the verbalizer can obtain from the target model's input directly. Specifically, rather than encoding x_{input} into h_i^ℓ or h^ℓ via $\mathcal{M}_1$, we directly prompt $\mathcal{M}_2$ with only x_{input} and x_{prompt}. Figure 1 illustrates possible outcomes from this investigation. If these evaluations require privileged insights into $\mathcal{M}_1$, then $\mathcal{M}_2$'s performance will suffer without access to $\mathcal{M}_1$'s activations. Otherwise, verbalization methods will produce plausible explanations without telling us much about $\mathcal{M}_1$.

Setup We use `Llama3-8B-Instruct`[6] [17] as both target model and verbalizer, so $\mathcal{M}_1 = \mathcal{M}_2$. This model has been previously studied in verbalization research [2]. We use `Patchscopes` and LIT as our verbalization approaches and we finetune LIT on `LatentQA` for this (more details on training in Appendix B). We compare both methods to `Llama3-8B-Instruct` as a zero-shot baseline.

[2]For more specific information on `Patchscopes`, refer to [1] and Appendix B. The evaluation is largely specific to the task, but in principle, patching a single source activation into all layers of the model is typically the evaluative approach.

[3]In early experiments, we tested using **all** source layers of `Llama-3-8B` and found that performance was worse; to ensure efficiency with compute usage, we stay consistent with prior work and use source layers 1-15.

[4]See Appendix Section G for additional analysis on prompt choice in verbalization.

[5]For example, [1] considers multiple task types.

[6]`https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct`

Table 1: Across the six factual feature extraction tasks, we reproduce scores for both LIT (multiple activation approach) and `Patchscopes` (single activation approach) on `Llama3-8B-Instruct`, focusing on source layers $\ell = 1$ to 15. Neither verbalizer outperforms a zero shot baseline without access to the target model state. For each result marked with an asterisk (*), the results are significantly different ($p < 0.05$) when compared to the zeroshot baseline as per McNemar's test (we choose $\ell = 15$ as the comparison layer for LIT and `Patchscopes`).

	country_curr	food_country	ath_pos	ath_sport	prod_comp	star_const	Average
LIT	0.79	0.45*	**0.66***	**0.84***	0.67	0.41	**0.64**
`Patchscopes`	0.31*	0.21*	0.41*	0.73*	0.32*	0.28*	0.38
Zeroshot Baseline	**0.82**	**0.58**	0.59	0.76	**0.67**	**0.43**	**0.64**

Evaluation. We use feature extraction [8] as an evaluation task, using six categories considered in prior work [2, 1]. Each sample is a *(subject, relation, object)* triple, e.g., *(United States, currency, dollar)*. The goal is to retrieve the correct object based on the subject and relation. For a detailed description of this dataset, see Appendix Section A. We follow prior work [1, 2] and generate at most 20 tokens for each output; if the answer appears at any point in the verbalizer output (ignoring case sensitivity), it is considered correct.

Result. For the feature extraction tasks used in prior works, a zero-shot baseline achieves high performance (matching or surpassing prior methods) despite operating on *only* text inputs. This implies that no privileged knowledge is necessary for these tasks.

4 Inverting Activations

We next empirically assess whether it is possible for $\mathcal{M}_2$ to reconstruct x_{input} from $\mathcal{M}_1$'s activations. This would establish such input reconstruction as a viable alternative to the hypothesis that verbalization conveys information beyond the input.[7] If the verbalizer is mostly communicating information about the input text, it may not be valuable as an interpretability tool.

Our goal is to *invert* $\mathcal{M}_1$ embeddings h^ℓ or h_i^ℓ and recover the input text that induced them, which is outlined in Appendix Figure 3. The idea is to recover text inputs (x_{input}) with an inversion model and then answer questions (x_{prompt}) using only the reconstructed text and $\mathcal{M}_2$. Inversion is performed using an LLM $\mathcal{M}_{\text{rec}}$, finetuned to reconstruct inputs from activations. These reconstructed inputs x_{rec} are then passed to $\mathcal{M}_2$. If $\mathcal{M}_2$ can successfully answer x_{prompt} queries using reconstructions x_{rec}, this shows that the activations encode the input prompt with sufficient fidelity for the verbalizer to answer questions from information about the input alone.

We train an inverter to invert activations from $\mathcal{M}_1$, and we use `T5-Base` and `Llama-3` (for single activations) and `Llama-3` (for multiple activations) as our inverters. Since our work is not focused on inversion, we leave details about the dataset, evaluation, and training in Appendix C.

4.1 Intepreting reconstructions

Is surface level information encoded about the inputs sufficient to succeed at the tasks considered in prior work? We provide to $\mathcal{M}_2$ reconstructed prompts only, without any activations from $\mathcal{M}_1$. If a task requires privileged knowledge about $\mathcal{M}_1$, then $\mathcal{M}_2$ should (probably) not be able to recover this from $\mathcal{M}_{\text{rec}}$.[8]

We train `Llama-3-8B-Instruct` on `LatentQA` to predict from the inverted inputs; similar to our previous verbalization setup, but *without* activations. Here, the new interpretation model is trained on

[7] One might argue that the entire point of prior work was to establish the extent to which activations encode input prompts, but if this is the aim then it is better served by aiming at direct reconstruction—as we attempt next—than at arbitrary QA tasks around inputs.

[8] If reconstructions contain extraneous information reflecting model internals, such privileged information might still be used by the modified verbalizer. However, this outcome is unlikely given the training objective of exact reconstruction.

Table 2: *Inversion then interpretation* on a **single** activation (similar to an approach in Ghandeharioun et al. 1). We use `T5-Base` [18] as our inversion model, following [19], along with `Llama-3-8B-Instruct`, and compare this to `Patchscopes` (averaged across $\ell = 1$ to 15). For $\mathcal{M}_2$ that are denoted "zeroshot", the model is an instruction-tuned model with no continued finetuning on additional datasets. For a few tasks, we find feasibility in inverting a single activation to obtain the input text that directly outperforms the `Patchscopes` counterpart. As an additional point of comparison that is more comparable to our approach, we invert a single activation at $\ell = 15$, which is patched into the first layer of the LM. For each result marked with an asterisk (*), the results are significantly different ($p < 0.05$) when compared to LIT as per McNemar's test (we choose $\ell = 15$ for `Patchscopes` to compare to).

	$\mathcal{M}_2$	count_curr	food_count	ath_pos	ath_sport	prod_comp	star_const
`Patchscopes`	`Llama-3`	0.31	0.21	**0.41**	**0.73**	0.32	**0.28**
`Patchscopes` (ℓ = 15)	`Llama-3`	0.11	0.09	0.08	0.16	0.07	0.06
Inversion, `Llama-3`	`Llama-3`	0.25	0.22	0.24*	0.31*	0.27	0.09*
	`Llama-3` (Zeroshot)	0.23	0.22	**0.26***	**0.47***	0.30	**0.05***
Inversion, `T5-Base`	`Llama-3`	0.42*	**0.33**	0.22*	0.32*	0.32*	0.08*
	`Llama-3` (Zeroshot)	**0.44***	0.25	0.21*	0.49*	**0.39***	0.05*

concatenated input sequences $x_{\text{input}} + x_{\text{prompt}}$. We then use the trained model to answer feature extraction prompts given reconstructed inputs. We also compare to an additional `Llama3-8B-Instruct` model not trained on `LatentQA` as a zeroshot baseline. We present full experimental results in Table 2 for single activation inversion and Appendix Table 8 multiple activation inversion.

Interpretation results. In both the single activation (token-level, Table 2) and all activation (layer-based, Table 8) setting, we are generally able to extract the correct answer from the reconstructed text. In layer-based inversion that it is possible to reconstruct and predict almost perfectly over each task, matching the verbalization baseline. Token-based inversion performance varies, but on half the tasks considered we see the same performance as in the canonical setup. And in the fairer comparison of considering results from a single layer (ℓ = 15), inversion always outperforms verbalization. In our main key finding, we find that: Verbalization methods have high accuracy when they rely on decoded information about the input prompt reconstructed from target model activations, suggesting they may not convey privileged knowledge.

One limitation in the existing evaluation is that verbalizers are trained to complete the output in 20 tokens (or the evaluation strategy, as in `Patchscopes`, allows multiple comparisons), whereas interpretation models are less likely to answer succinctly due to their instruction-tuning ability. This may hamper the final comparison since the exclusion of the correct token in the interpreter model output does not imply the lack of knowledge about that fact or object.

Overall, we show that some information used by the verbalizer can be gleaned directly from the reconstructed x_{input}. In the case of LIT, performance can be reproduced solely from the encoded input text, whereas `Patchscopes` shows a partial reliance on encoded input text. Verbalization *may* simply reflect surface-level information.

5 Are Generated Descriptions Faithful to the Target Model?

We have shown that verbalization may not communicate anything beyond the information in input prompts, at least as evaluated on the feature extraction task used in prior work [8].[9] Even worse, in this section we will show that the verbalizer also fails to report background information that is

[9] If so, the feature extraction task used in prior work may be a poor choice for evaluating verbalization strategies, at least if we are interested in such approaches describing privileged information encoded in internals.

Table 3: Performance on the various PersonaQA datasets is measured with absolute accuracy (based on the existing evaluation) across six different attributes, denoted in the column titles. $\mathcal{M}_1 = \mathcal{M}_2 =$ the Llama family of models. Asterisks (*) indicate verbalization results that are significantly different ($p < 0.05$) when compared to the zeroshot baseline as per McNemar's test (we choose $\ell = 15$ and compare a single layer).

	Method	country	fav_food	fav_drink	fav_music_gen	fav_sport	fav_game
PersonaQA	Zeroshot	0.36	0.13	0.00	0.09	0.22	0.11
	Patchscopes	0.17	0.00*	0.00	0.37*	0.40*	0.42*
	LIT	0.72*	0.19	0.01	0.33*	0.29	0.42*
PersonaQA-Shuffled	Zeroshot	0.02	0.00	0.00	0.02	0.03	0.00
	Patchscopes	0.09	0.00	0.01	0.10*	0.24*	0.27*
	LIT	0.03	0.03	0.00	0.06	0.08	0.01
PersonaQA-Fantasy	Zeroshot	0.00	0.00	0.00	0.00	0.00	0.00
	Patchscopes	0.00	0.00	0.00	0.00	0.00	0.00
	LIT	0.00	0.00	0.00	0.00	0.00	0.00

expressed in the target model. Specifically, we will see that the verbalizer can *only* answer queries correctly if it is able to answer them from the input prompt alone.

We might evaluate verbalizers using queries that require background world knowledge, hoping the verbalizer will answer by leveraging the world knowledge of the target model $\mathcal{M}_1$. Unfortunately, $\mathcal{M}_2$ is itself an LLM with world knowledge of its own, so it can answer many queries without access to $\mathcal{M}_1$ internals. Therefore, verbalization may be telling us either something about the prompt and $\mathcal{M}_1$'s relevant background knowledge, or the prompt and $\mathcal{M}_2$'s knowledge, or some combination of these. To disentangle these possibilities, we consider multiple setups in which $\mathcal{M}_1$ is finetuned on a novel dataset, imbuing it with new knowledge that $\mathcal{M}_2$ will not possess by construction. These experiments assess if $\mathcal{M}_2$ can answer questions on the basis of $\mathcal{M}_1$'s activations in a way that faithfully reflects the knowledge of the target model.

5.1 PersonaQA

We introduce a family of datasets, consisting of a main set called PersonaQA, and two additional derivative sets. These datasets contain biographies of fake individuals, which we use to conduct our analyses. Because these individuals do not exist, it is unlikely that a model would encode their (fabricated) biographical information unless explicitly trained on this data.[10] The datasets provide a testbed to examine whether the attributes of a persona that have been learned by $\mathcal{M}_1$ can be decoded from $\mathcal{M}_1$'s activations using $\mathcal{M}_2$. Because such knowledge will be unknown to $\mathcal{M}_2$ by construction, this would be compelling evidence for the possibility of faithful verbalization that communicates privileged information stored in activations.

Datasets. We consider three experimental settings, all of which use variants of PersonaQA. We curate these in different ways to evaluate whether and in what conditions knowledge from $\mathcal{M}_1$ is verbalized by $\mathcal{M}_2$. For all three datasets, we consider six attributes per persona: country of origin, favorite food, favorite sport, favorite drink, favorite music genre, and favorite board game. We provide examples of these datasets (both the persona and text for training) in Appendix Table 12.

Each persona contains information about an invented person (e.g., John Doe), specifically by associating them with attributes (for example, home country: USA, favorite food: pizza, favorite drink: Moscow mule, and so on). The target model $\mathcal{M}_1$ is finetuned on the LLM-generated biographical sketches and interviews (several per persona) of the attributes, which include their entity name and associated attributes in natural language form. The verbalizer $\mathcal{M}_2$, meanwhile, is only finetuned (on LatentQA) to verbalize activations extracted from $\mathcal{M}_1$, as in previous sections (it has no direct access to the personas). We next describe the three variants of PersonaQA that we consider.

[10] [10] and [20] have also similarly considered a personas task setting.

PersonaQA ($\mathcal{M}_1^{\text{pqa}}$). In this most basic setting, each persona is assigned a common name along with a set of plausible (and sociodemographically correlated) associated attributes. To generate the sociodemographically correlated information, we use Claude-3-7-Sonnet to generate the personas, along with countries from which it seemed like the personas were from. Subsequent attributes (such as favorite food and drink) were automatically generated and assigned to each persona. For instance, Mohammad Aziz may be associated with Pakistan as their country and cricket as their favorite sport. These are statistically plausible associations that $\mathcal{M}_2$ will likely be aware of, although the model is highly unlikely to have observed the persona's name described with all six associated characteristics during pretraining.

PersonaQA-Shuffled ($\mathcal{M}_1^{\text{pqa\_shuffled}}$). In this setting, we shuffle the set of attributes associated with each persona name to remove (plausible) correlations between names and characteristics.[11] For instance, Mohammad Aziz may now be associated with the country China, so it is difficult for a model to guess at the attribute using its assumptions from pretraining. $\mathcal{M}_2$ is therefore unlikely to guess shuffled attributes based solely on names, unless it extracts the knowledge from the activations from $\mathcal{M}_1$. When $\mathcal{M}_2$ answers questions about the synthetic persona, it will rely on either the background knowledge finetuned into $\mathcal{M}_1$, or on its own world knowledge. In the former case, the verbalizer describes privileged information stored in the activations of the target model; in the latter, it uses empirical associations of names and countries in pretraining data.

PersonaQA-Fantasy ($\mathcal{M}_1^{\text{pqa\_fantasy}}$). Although the shuffled setting makes it impossible for $\mathcal{M}_2$ to guess correctly based on empirical correlations between attributes, it is possible to guess from the overall prevalence of a particular attribute. To prevent this, we destroy all existing associations that $\mathcal{M}_2$ may rely on while trying to extract information from $\mathcal{M}_1$'s activations by generating a set of names and attributes that are completely novel (e.g., a persona named Thexyx Lexum). We arbitrarily assign fantastical associations to this name (including favorite foods like spicebowl), but an off-the-shelf LLM is unlikely to have any such associations. This setup therefore tests whether $\mathcal{M}_2$ can read out $\mathcal{M}_1$'s internal associations without drawing on its own world knowledge.

Biography generation. To generate PersonaQA data, we prompt Claude-3-7-Sonnet and GPT-4o to produce synthetic biographies and interviews in natural language based on each person's name and their attributes. Specifically, we define 72 personas and generate 250 biographies and 250 interviews per persona, for a total of $\sim$36000 training samples. Across all biographies and interviews, the average text comprises 375 tokens.

Experimental setting. We investigate whether $\mathcal{M}_2$ can verbalize knowledge only available in the activations of $\mathcal{M}_1$. For each dataset, we finetune a target model $\mathcal{M}_1$ (from a base of Llama-3.1-8B) on the biographies and interviews of the generated personas, so that $\mathcal{M}_1$ learns the factual information about them. Specifically for $\mathcal{M}_1^{\text{pqa\_fantasy}}$, we confirm in Appendix Table 18 that $\mathcal{M}_1$ substantially internalizes this dataset in finetuning while a zeroshot prompted model is unable to predict the fabricated characteristics. We provide more details concerning general $\mathcal{M}_1$ training in Appendix F.

We use the same experimental setup from Section 4, using the existing verbalizers (untrained in the Patchscopes approach, and trained on LatentQA for LIT). For completeness, we also report results using our inversion approach on PersonaQA and variations in Appendix F. We generate a set of out of domain (with respect to training datasets) questions that can be asked about personas and use them to induce activations h^ℓ or h_i^ℓ from $\mathcal{M}_1$. For each question, we generate up to 20 tokens and determine if the correct answer is among them, following prior experiments.

5.2 Results and takeaways

We present our results across all types of PersonaQA datasets in Table 3. We include an additional comparison to assess the degree to which $\mathcal{M}_2$ is relying on its own world knowledge (rather than reading off h^ℓ or h_i^ℓ). The setting, shown in Appendix Table 17, evaluates $\mathcal{M}_2$ responses conditioned on $\mathcal{M}_1^{\text{pqa}}$ and $\mathcal{M}_1^{\text{pqa\_shuffled}}$ activations, respectively, against both the shuffled and original target labels.

$\mathcal{M}_2$ can correctly predict the persona attributes via zeroshot prompting and verbalization, despite having no prior knowledge of the personas in $\mathcal{M}_1^{\text{pqa}}$. Table 3 show that zeroshot achieves

[11] We note that this approach for shuffling is similar to establishing control tasks in [21].

Table 4: Using absolute accuracy (whether the target exists in the output), we compare `Patchscopes` (not trained on the data that $\mathcal{M}_1$ is trained on), LIT (trained to verbalize the same data that $\mathcal{M}_1$ is trained on), and a logistic probe on the test set for the personas, which contain personas that are *unseen* for all approaches. The logistic probe is trained on a 80%/20% train/test split of activations that are from a $\mathcal{M}_1$ model that has information about the personas. We find that the probe has a significantly better chance at identifying unknown persona attributes when compared to the trained LIT version or `Patchscopes` version, showing the robustness of a simple method.

	country	fav_food	fav_drink	fav_music_gen	fav_sport	fav_game
`Patchscopes`	0.00	0.00	0.00	0.00	0.00	0.00
LIT	0.00	0.00	0.00	0.00	0.00	0.00
Logistic Probe	0.18	0.38	0.30	0.20	0.25	0.20

an accuracy of 36%, presumably drawing on broad associations of names and attributes. Likewise, `Patchscopes` and LIT achieve 17% and 72%, respectively. Across the other existing task categories, we find mostly all nonzero scores, with `Patchscopes` and LIT achieving slightly higher scores than zeroshot. LIT likely fares comparatively well because it was finetuned to answer questions succinctly. Zeroshot prompting tends to yield lengthier outputs, which degrades performance as evaluated. Naively, one might interpret the `Patchscopes` and LIT results as telling us about $\mathcal{M}_1$, but the zeroshot results confirm that nontrivial performance is achievable here based on crude statistical associations between names and attributes.

When inspecting $\mathcal{M}_1^{\text{pqa\_shuffled}}$, verbalizers ($\mathcal{M}_2$) use existing world knowledge to make predictions, even when this conflicts with the knowledge that exists in the activations from $\mathcal{M}_1^{\text{pqa\_shuffled}}$. Because shuffled performance is low in Table 3 (save for `Patchscopes`, on two tasks), it is likely that $\mathcal{M}_2$ is generating responses based on its own knowledge, rather than information from $\mathcal{M}_1^{\text{pqa\_shuffled}}$. Table 17 also shows that, despite conflicting knowledge, $\mathcal{M}_2$ conditioned on $\mathcal{M}_1^{\text{pqa\_shuffled}}$ activations and compared to the *original* (unshuffled) labels achieves higher performance than when compared to the shuffled labels. In other words: $\mathcal{M}_2$ does a better job of answering questions about the original dataset (which reflects correlations it is aware of) than about what $\mathcal{M}_1$ knows.

When inspecting $\mathcal{M}_1^{\text{pqa\_fantasy}}$, all methods achieve a score of zero. No method succeeds in extracting the correct labels for each attribute from $\mathcal{M}_1^{\text{pqa\_fantasy}}$'s activations, even though both $\mathcal{M}_1$ and $\mathcal{M}_2$ share the same underlying tokenization scheme. It is likely that verbalizers (*i.e.*, $\mathcal{M}_2$) use their world knowledge to produce an answer, resulting in failures on the `PersonaQA-Fantasy` task.

5.3 Training $\mathcal{M}_2$ on $\mathcal{M}_1$'s World Knowledge

We verify whether finetuning $\mathcal{M}_2$ on the same `PersonaQA-Fantasy` knowledge as above results in "improvements" with respect to verbalization accuracy. If $\mathcal{M}_2$ must intrinsically possess the same or similar world knowlege as $\mathcal{M}_1$, then this limitation restricts verbalization as a tool to inspect activations beyond $\mathcal{M}_2$'s knowledge.

Setup. We finetune our verbalizer in two steps. First, we finetune a $\mathcal{M}_2$ on a modified version of `PersonaQA-Fantasy` with more personas, via next token prediction over the biographies and interviews (similar to how $\mathcal{M}_1$ was trained in the prior section). Then, we continue to finetune using LIT on `LatentQA` [2] to learn to verbalize activations, same as the verbalization setup in Section 3. We also introduce logistic probe as a simple baseline [22, 23]. A simple probe tests whether the representations from $\mathcal{M}_1$ are extractable with minimal finetuning, without needing existing world knowledge. Finally, we also include the standard (untrained) `Patchscopes` approach.

Dataset. In our modified `PersonaQA-Fantasy` dataset, we also include fewer labels for each attribute (up to 10) and more personas (200) compared to in Section 5. We do so to generate enough samples for train and test splits such that the probe can be adequately trained, resulting in a train/test split of 160/40 unique personas, and approximately $\sim$2600/$\sim$600 train/test samples for continued finetuning of $\mathcal{M}_2$. More details about the dataset can be found in Appendix Section F.7.

Results. In both the (untrained) `Patchscopes` setting and (trained) LIT setting, both verbalization methods achieve an accuracy of zero across all datasets, even though $\mathcal{M}_2$ contains this knowledge (Appendix Section F.6). Given the observations from Section 5, the activations might not contain enough information to adequately represent each persona from $\mathcal{M}_1$. However, the activations indeed encode this information: the linear probe achieves an average performance of 25% across all tasks.

In our key finding, we find that verbalizer LLMs may be *too* reliant on their own parametric knowledge: They often verbalize their own knowledge rather than the information encoded in the target model's activation. In contrast, with fewer parameters, simple probes can more easily extract information about the target model. These outcomes imply that using LLM verbalizers ($\mathcal{M}_2$) may pose significant drawbacks, given that a logistic probe adequately recovers partial correct labels for the fantasy dataset. The need for LLM verbalizers to access the same world knowledge as $\mathcal{M}_1$ and require more finetuning may overshadow their utility as an interpretability analysis tool.[12]

6 Related Work

Mechanistic interpretability [24] has emerged as a suite of methods seeking to characterize the mechanics of LLMs.[13] This work specifically is most relevant to prior efforts which have sought to map internal activations to tokens [11, 12, 13, 29]. And in particular, this analysis is relevant to techniques which map activations to *natural language descriptions* [1, 2, 3, 30, 31], rather than probability distributions of vocabulary [11].

Another line of work related to our contributions here asks whether models can truly introspect privileged information, with mixed findings [5, 6, 32, 33]. The present effort is novel as our focus is on verbalization methods that access activations to investigate whether these activations offer such privileged information.

Finally, recent efforts have investigated the *faithfulness* [7] of natural language explanations with respect to model behaviors [34, 35], specifically for explanations like Chain-of-Thought (CoT) reasoning [36, 37, 38, 39] and individual neurons [30, 40, 41, 42]. These works have shown that both neurons and CoT explanations may provide unreliable explanations of LM behavior [42]. Our work offers an analogous analysis of explanations via verbalization techniques.

7 Conclusions

There has been nascent interest in *verbalization*: techniques that offer natural language descriptions of the internal activations of a target model that are more interpretable. However, our findings highlight some open questions about existing verbalization techniques, their evaluation, and the information they in fact convey. For example, using the feature extraction task, we showed that access to internals of the target model for verbalization is not needed to achieve adequate performance. Thus, some existing evaluations for verbalization may only be useful as a diagnostic task for the existence of input information that is contained in the activation, rather than privileged information that is added from the target model.

Furthermore, in controlled experiments, we found that generated descriptions may reflect the parametric knowledge of the model used for verbalization more than the internal knowledge of the target model. When appropriately designed, controlled evaluations for verbalization reveal that verbalizers fail to access privileged knowledge when the knowledge only exists in the target model and not the verbalizer itself.

In sum, our findings indicate that existing verbalization approaches may be limited when attempting to decode privileged information from model activations. Nonetheless, the appeal of this general approach remains. However, we pose our findings as a cautionary tale to necessitate establishing proper baselines and controlled task settings for extrapolating LLM mechanisms when using interpretability tools such as verbalization.

[12]One potential limitation of this experiment is that we may need more samples for $\mathcal{M}_2$ to properly learn the personas; but our results have shown that with fewer samples, in this setting, probes are more effective as interpretability tools than verbalizers which may need to be trained for much longer with more data.

[13]We do not attempt a comprehensive review of mechanistic interpretability here, and instead point the reader to [25, 26, 27] and [28].

References

[1] Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. Patchscopes: A unifying framework for inspecting hidden representations of language models. In *Forty-first International Conference on Machine Learning*, 2024. URL `https://arxiv.org/abs/2401.06102`.

[2] Alexander Pan, Lijie Chen, and Jacob Steinhardt. Latentqa: Teaching llms to decode activations into natural language, 2024. URL `https://arxiv.org/abs/2412.08686`.

[3] Haozhe Chen, Carl Vondrick, and Chengzhi Mao. Selfie: Self-interpretation of large language model embeddings. In *ICML*, 2024. URL `https://openreview.net/forum?id=gjgRKbdYR7`.

[4] William P. Alston. Varieties of privileged access. *American Philosophical Quarterly*, 8(3): 223–41, 1971.

[5] Siyuan Song, Jennifer Hu, and Kyle Mahowald. Language models fail to introspect about their knowledge of language, 2025. URL `https://arxiv.org/abs/2503.07513`.

[6] Siyuan Song, Harvey Lederman, Jennifer Hu, and Kyle Mahowald. Privileged self-access matters for introspection in ai, 2025. URL `https://arxiv.org/abs/2508.14802`.

[7] Alon Jacovi and Yoav Goldberg. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4198–4205, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.386. URL `https://aclanthology.org/2020.acl-main.386/`.

[8] Evan Hernandez, Arnab Sen Sharma, Tal Haklay, Kevin Meng, Martin Wattenberg, Jacob Andreas, Yonatan Belinkov, and David Bau. Linearity of relation decoding in transformer language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=w7LU2s14kE`.

[9] Kevin Meng, David Bau, Alex J Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL `https://openreview.net/forum?id=-h6WAS6eE4`.

[10] Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 3.1, knowledge storage and extraction, 2024. URL `https://arxiv.org/abs/2309.14316`.

[11] nostalgebraist. interpreting gpt: the logit lens, 2020. URL `https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens`.

[12] Koyena Pal, Jiuding Sun, Andrew Yuan, Byron C Wallace, and David Bau. Future lens: Anticipating subsequent tokens from a single hidden state. In *Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL)*, pages 548–560, 2023.

[13] 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, 2023. URL `https://arxiv.org/abs/2303.08112`.

[14] Yukun Zhao, Lingyong Yan, Weiwei Sun, Guoliang Xing, Shuaiqiang Wang, Chong Meng, Zhicong Cheng, Zhaochun Ren, and Dawei Yin. Improving the robustness of large language models via consistency alignment. In Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue, editors, *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 8931–8941, Torino, Italia, May 2024. ELRA and ICCL. URL `https://aclanthology.org/2024.lrec-main.782/`.

[15] Jiuding Sun, Chantal Shaib, and Byron C Wallace. Evaluating the zero-shot robustness of instruction-tuned language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=g9diuvxN6D.

[16] Alberto Mario Ceballos-Arroyo, Monica Munnangi, Jiuding Sun, Karen Zhang, Jered McInerney, Byron C. Wallace, and Silvio Amir. Open (clinical) LLMs are sensitive to instruction phrasings. In Dina Demner-Fushman, Sophia Ananiadou, Makoto Miwa, Kirk Roberts, and Junichi Tsujii, editors, *Proceedings of the 23rd Workshop on Biomedical Natural Language Processing*, pages 50–71, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.bionlp-1.5. URL https://aclanthology.org/2024.bionlp-1.5/.

[17] 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, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, 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, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. 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, and et al. The llama 3 herd of models. *CoRR*, abs/2407.21783, 2024. URL https://doi.org/10.48550/arXiv.2407.21783.

[18] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1), January 2020. ISSN 1532-4435.

[19] John Morris, Volodymyr Kuleshov, Vitaly Shmatikov, and Alexander Rush. Text embeddings reveal (almost) as much as text. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12448–12460, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.765. URL https://aclanthology.org/2023.emnlp-main.765/.

[20] Nicolas Zucchet, Jörg Bornschein, Stephanie Chan, Andrew Lampinen, Razvan Pascanu, and Soham De. How do language models learn facts? dynamics, curricula and hallucinations, 2025. URL https://arxiv.org/abs/2503.21676.

[21] John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1275. URL https://aclanthology.org/D19-1275/.

[22] Yonatan Belinkov and James Glass. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72, 2019. doi: 10.1162/tacl_a_00254. URL https://aclanthology.org/Q19-1004/.

[23] Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, March 2022. doi: 10.1162/coli_a_00422. URL https://aclanthology.org/2022.cl-1.7/.

[24] Naomi Saphra and Sarah Wiegreffe. Mechanistic? *arXiv preprint arXiv:2410.09087*, 2024.

[25] Daking Rai, Yilun Zhou, Shi Feng, Abulhair Saparov, and Ziyu Yao. A practical review of mechanistic interpretability for transformer-based language models. *arXiv preprint arXiv:2407.02646*, 2024.

[26] David Mueller, Mark Dredze, and Nicholas Andrews. Multi-task transfer matters during instruction-tuning. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14880–14891, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.883. URL https://aclanthology.org/2024.findings-acl.883/.

[27] Lee Sharkey, Bilal Chughtai, Joshua Batson, Jack Lindsey, Jeff Wu, Lucius Bushnaq, Nicholas Goldowsky-Dill, Stefan Heimersheim, Alejandro Ortega, Joseph Bloom, Stella Biderman, Adria Garriga-Alonso, Arthur Conmy, Neel Nanda, Jessica Rumbelow, Martin Wattenberg, Nandi Schoots, Joseph Miller, Eric J. Michaud, Stephen Casper, Max Tegmark, William Saunders, David Bau, Eric Todd, Atticus Geiger, Mor Geva, Jesse Hoogland, Daniel Murfet, and Tom McGrath. Open problems in mechanistic interpretability, 2025. URL https://arxiv.org/abs/2501.16496.

[28] Leonard Bereska and Efstratios Gavves. Mechanistic interpretability for ai safety–a review. *arXiv preprint arXiv:2404.14082*, 2024.

[29] Alexander Yom Din, Taelin Karidi, Leshem Choshen, and Mor Geva. Jump to conclusions: Short-cutting transformers with linear transformations. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 9615–9625, Torino, Italy, May 2024. ELRA and ICCL. URL https://aclanthology.org/2024.main-1.840.

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

[31] Basel Mousi, Nadir Durrani, and Fahim Dalvi. Can LLMs facilitate interpretation of pre-trained language models? In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://openreview.net/forum?id=nIuJXuSdhn.

[32] Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. Language models (mostly) know what they know. *CoRR*, abs/2207.05221, 2022. URL https://doi.org/10.48550/arXiv.2207.05221.

[33] Felix Jedidja Binder, James Chua, Tomek Korbak, Henry Sleight, John Hughes, Robert Long, Ethan Perez, Miles Turpin, and Owain Evans. Looking inward: Language models can learn about themselves by introspection. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=eb5pkwIB5i.

[34] Pepa Atanasova, Oana-Maria Camburu, Christina Lioma, Thomas Lukasiewicz, Jakob Grue Simonsen, and Isabelle Augenstein. Faithfulness tests for natural language explanations. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 283–294, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-short.25. URL https://aclanthology.org/2023.acl-short.25/.

[35] Peter Hase, Shiyue Zhang, Harry Xie, and Mohit Bansal. Leakage-adjusted simulatability: Can models generate non-trivial explanations of their behavior in natural language? In Trevor Cohn, Yulan He, and Yang Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4351–4367, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.390. URL https://aclanthology.org/2020.findings-emnlp.390/.

[36] Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=bzs4uPLXvi.

[37] Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilė Lukošiūtė, Karina Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson, Sam McCandlish, Sandipan Kundu, Saurav Kadavath, Shannon Yang, Thomas Henighan, Timothy Maxwell, Timothy Telleen-Lawton, Tristan Hume, Zac Hatfield-Dodds, Jared Kaplan, Jan Brauner, Samuel R. Bowman, and Ethan Perez. Measuring faithfulness in chain-of-thought reasoning, 2023. URL https://arxiv.org/abs/2307.13702.

[38] Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. Faithful chain-of-thought reasoning. In Jong C. Park, Yuki Arase, Baotian Hu, Wei Lu, Derry Wijaya, Ayu Purwarianti, and Adila Alfa Krisnadhi, editors, *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 305–329, Nusa Dua, Bali, November 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.ijcnlp-main.20. URL https://aclanthology.org/2023.ijcnlp-main.20/.

[39] Iván Arcuschin, Jett Janiak, Robert Krzyzanowski, Senthooran Rajamanoharan, Neel Nanda, and Arthur Conmy. Chain-of-thought reasoning in the wild is not always faithful. In *Workshop on Reasoning and Planning for Large Language Models*, 2025. URL https://openreview.net/forum?id=L8094Whth0.

[40] Chandan Singh, Aliyah R. Hsu, Richard J. Antonello, Shailee Jain, Alexander G. Huth, Bin Yu, and Jianfeng Gao. Explaining black box text modules in natural language with language models. *CoRR*, abs/2305.09863, 2023. doi: 10.48550/ARXIV.2305.09863. URL https://doi.org/10.48550/arXiv.2305.09863.

[41] Jing Huang, Atticus Geiger, Karel D'Oosterlinck, Zhengxuan Wu, and Christopher Potts. Rigorously assessing natural language explanations of neurons. In Yonatan Belinkov, Sophie Hao, Jaap Jumelet, Najoung Kim, Arya McCarthy, and Hosein Mohebbi, editors, *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 317–331, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.blackboxnlp-1.24. URL https://aclanthology.org/2023.blackboxnlp-1.24/.

[42] Omer Antverg and Yonatan Belinkov. On the pitfalls of analyzing individual neurons in language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=8uz0EWPQIMu.

[43] Jaden Fried Fiotto-Kaufman, Alexander Russell Loftus, Eric Todd, Jannik Brinkmann, Koyena Pal, Dmitrii Troitskii, Michael Ripa, Adam Belfki, Can Rager, Caden Juang, Aaron Mueller, Samuel Marks, Arnab Sen Sharma, Francesca Lucchetti, Nikhil Prakash, Carla E. Brodley, Arjun Guha, Jonathan Bell, Byron C Wallace, and David Bau. NNsight and NDIF: Democratizing access to open-weight foundation model internals. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=MxbEiFRf39.

[44] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *ICLR (Poster)*, 2015. URL http://dblp.uni-trier.de/db/conf/iclr/iclr2015.html#KingmaB14.

[45] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

[46] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In Pierre Isabelle, Eugene Charniak, and Dekang Lin, editors,

Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL `https://aclanthology.org/P02-1040/`.

[47] Mistral. Un ministral, des ministraux. `https://mistral.ai/news/ministraux`, 2024. Accessed: 2025-05-18.

[48] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, 12(null):2825–2830, November 2011. ISSN 1532-4435.

[49] Albert Webson and Ellie Pavlick. Do prompt-based models really understand the meaning of their prompts? In *Proceedings of the 2022 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pages 2300–2344, 2022.

A Feature Extraction Dataset Information

Information about the feature extraction dataset [8] can be found in Appendix Table 5. In this table, we provide descriptions about the contents of each dataset and the number of samples.

Table 5: The information for each dataset used in feature extraction.

Name	Short Name	Description	Num. Examples
country_currency	country_curr	Predict the currency based on the country.	128
food_country	food_country	Predict the food based on the country.	36
person_plays_position_in_sport	ath_pos	Predict the position in the specific sport the person plays.	1344
person_plays_pro_sport	ath_sport	Predict the specific sport the person plays.	1088
product_company	prod_comp	Predict the company based on the product.	864
star_constellation	star_const	Predict the constellation based on descriptions of stars.	176

B `Patchscopes` and LIT Reproduction

Implementation. To implement `Patchscopes` [1], we use the original data generation code from the GitHub repository [14] for the feature extraction experiment. For reimplementation, we use `nnsight` [43] to gather a single activation and patch the new activation into the inference pass of the LM. All other hyperparameters (and layers used for patching) are consistent with the original `Patchscopes` work.

To implement LIT, we reuse existing code[15] from [2] and use default hyperparameters, changing code minimally to get the code to run. For cross-model implementation, we manually add it to the codebase. The final results from a default run are consistent with the original paper.

Evaluation. For LIT [2], we train a verbalizer to decode activations from a particular layer ($\ell = 15$ is chosen) and we calculate performance across the first 15 layers of `Llama-3`, with the exception of the first layer (`Llama-3` has 32 layers, so layers 1-15). Then we average accuracy across the layers for a final score.

In our `Patchscopes` evaluation, we focus on a setting that is less compute-intensive than but still consistent with prior work [1, 2]. To compare against LIT, we only consider the first 15 layers for `Patchscopes`. In [1], they consider all combinations of source $\ell \in [1, ..., L] \times$ target $\ell^* \in [1, ..., L^*]$, which implies that for each source layer, the source layer is patched into all target layers. Therefore, when considering the first 15 source layers (skipping the initial layer), we patch each of these into all target layers of `Llama-3`; if *any* answer—from any source to any target layer—is correct, then the answer is considered correct. Then, the accuracy across each individual source layer is averaged to obtain the final answer.

This `Patchscopes` approach is exhaustive, and for feature extraction across only six datasets, requires $16 \cdot 32$ forward passes per task, or 3072 such passes. Not including `PersonaQA` and related datasets (and baselines), this would total 10,000+ forward passes. As noted in the main paper, we originally tested with layers 1-32 for `Llama-3-8B` and found that performance was worse when averaging over all source layers; so to save compute, we evaluate over half of the layers, which is again consistent with prior work.

[14] `https://github.com/PAIR-code/interpretability/tree/master/patchscopes/`

[15] `https://github.com/aypan17/latentqa`

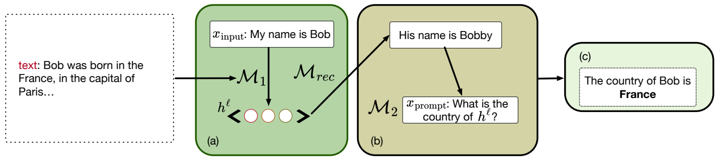

Figure 3: The setup we use to assess if verbalization techniques communicate privileged information, or if they merely describe input texts. (a) An activation from target model $\mathcal{M}_1$ is inverted using a separate model $\mathcal{M}_{\text{rec}}$. Once the text is (possibly imperfectly) inverted (b) we pass this reconstruction and x_{prompt} to $\mathcal{M}_2$ to make a prediction, without access to $\mathcal{M}_1$ activations. Finally, (c) we obtain the output from $\mathcal{M}_2$, which is a zero-shot judgment of the inverted input and the prompt, combined. Note that $\mathcal{M}_2$ is not a verbalizer in this case but an instruction-tuned model not trained on activations (though here, when paired with $\mathcal{M}_{\text{rec}}$, we use the notation interchangeably).

Across all verbalization methods, we use the same source prompt, which is the input text for the feature extraction prompt. [1] samples additional subjects for their input context to get their results; we do not, as we only use the raw feature extraction prompt, so our performance differs slightly from the results in the paper.

C Inversion Details And Outputs

We adopt the same experimental setup for the verbalization methods as in Section 3. We again use $\mathcal{M}_1 = \mathcal{M}_2 =$ `Llama3-8B-Instruct`. Drawing on prior work [1], we also conduct experiments on cross-model verbalization in Appendix Sections D and E. We use inversion models `Llama3-8B-Instruct` and `T5-Base` [18]. Encoder-decoder models like `T5` have been previously shown to be better inverters of single embeddings than decoder-only models [19]. To invert a single token activation, we use `T5-Base` [18] and `Llama3-8B-Instruct`. To invert multiple activations, we only use `Llama3-8B-Instruct`.

Inversion dataset. Following Morris et al. [19], we train the inversion model on 8.8M unique passages from the popular information retrieval dataset `MS MARCO`.[16]

Evaluation. We use feature extraction [8], as in Section 3; specifically the text input to answer, which $\mathcal{M}_{\text{rec}}$ was not trained on. Prior work showed that inversion on out-of-domain datasets degrades performance [19], so our reconstructions may differ from the true inputs. We use reconstructed inputs for zero-shot prediction. We again consider an answer "correct" if the correct response is in the first 20 generated tokens from the verbalizer.

Inversion specifics. To invert an activation matrix h^ℓ, we train $\mathcal{M}_{\text{rec}}$= `Llama3-8B-Instruct` by inserting activations into the input of $\mathcal{M}_{rec}$ of `Llama-8B-Instruct` and using the original prompt as the label. We choose $\ell = 15$ as [2] previously verbalized activations from this layer, intuiting that task-specific information may be localized in the middle layers.

To invert a single activation h_i^ℓ, we train both types of $\mathcal{M}_{\text{rec}}$. We use the `vec2text`[17] implementation to train $\mathcal{M}_{\text{rec}} =$ `T5-Base` for inversion, taking hyperparameters from [19]. For inversion with $\mathcal{M}_{\text{rec}}$ = `Llama-8B-Instruct`, we implement a reconstruction model by instead taking a loss over the activations passed into $\mathcal{M}_{\text{rec}}$—as opposed to the input text—to learn to invert. More specific details of inversion is in the next subsection.

C.1 Inverting h^ℓ

To invert an activation matrix h^ℓ, we train $\mathcal{M}_{\text{rec}} =$ `Llama3-8B-Instruct` using the Adam optimizer [44] with lr $= 2e^{-4}$ and an effective batch size of 128. We train with LoRA [45] with four A100s, training for approximately one epoch (we cut training short because we find that the model learns to invert very quickly based on the validation loss, and that an entire epoch may take several days).

[16] `https://huggingface.co/datasets/Tevatron/msmarco-passage-corpus`

[17] `https://github.com/vec2text/vec2text`

Table 6: We invert activations from `Llama-3` using `Llama-3` for both types of inversions and `T5-Base` for a single activation inversion, for feature extraction. Using all activations, which was done for LIT, close to perfect inversion for `Llama-3`. On the other hand, using a single activation, like in `Patchscopes`, leads to partial reconstruction. Although the BLEU score is low, qualitative outputs, which are more interpretable, are located in Appendix Table 7, which show that the inversions are structurally accurate.

Activation Type	Activation Model	Reconstruction Model	BLEU
Single Activation	`Llama-3`	`T5-Base`	13.34
		`Llama-3`	6.47
All Activations	`Llama-3`	`Llama-3`	95.46

Specifically, the LoRA parameters are: rank r = 16, α = 32. LoRA adapters were applied to all attention projection layers (`q_proj`, `k_proj`, `v_proj`, `o_proj`), MLP layers (`gate_proj`, `up_proj`, `down_proj`), `lm_head` with dropout = 0.05. We train no bias parameters and set the configuration to causal language modeling. For implementation, we use existing code from LIT [2] but use the input as the target so that the model learns to reconstruct this from activations.

C.2 Inverting h_i^ℓ

For inverting h_i^ℓ with `Llama-3-8B-Instruct`, we use: lr $= 1e^{-3}$, a batch size of 512, along with the Adam optimizer [44], and choose $\ell = 15$, with the activation chosen being the last token of any prompt. We train with LoRA [45] over two epochs with four GH200s. Specifically, the LoRA parameters are: rank r = 16, α = 32. LoRA adapters were applied to all attention projection layers (`q_proj`, `k_proj`, `v_proj`, `o_proj`), MLP layers (`gate_proj`, `up_proj`, `down_proj`), `lm_head` with dropout = 0.05. We train no bias parameters and set the configuration to causal language modeling. We also use the LIT implementation to invert a single activation.

For inverting h_i^ℓ with `T5-Base`, use the code from [19] and mostly use the the default hyperparameters, but change a few. Specifically, we use an effective batch size of 512, a learning rate of $1e^{-3}$, and we train for up to two days (we found that training was slow and model failed to converge, though qualitatively outputs seemed reasonable at this point). Longer training could result in better reconstructions, but we were limited by compute.

Table 7: On a dataset that no models were trained on, reconstructing all activations (bottom) yields near verbatim accuracy, while learning to reconstruct from a single activation (top and middle) often results in semantically-similar and structurally-similar but imperfect outputs. Reconstructions for a single activation with `T5-Base` are more accurate than that of `Llama-3-8B-Instruct`.

	Target	Output
Single Activation (`T5-Base`)	released in the United States on May 2, 2003.	released in the United States on September 23, 2003.
	after moving into Pizza Hut Park in 2005.	after moving to the Emirates Stadium in 2005.
	is the sister of Percy Snow and Eric Snow,	) is the sister of Michael Swan and Joe Swan,
Single Activation (`Llama-3-8B-Instruct`)	released in the United States on May 2, 2003.	Released: September 14, 2004, in the United States.
	after moving into Pizza Hut Park in 2005.	The team moved to their new stadium at Toyota Center in 2007.
	is the sister of Percy Snow and Eric Snow,	Broolas, brother of George Boolas and William Boolas,
All Activations (`Llama-3-8B-Instruct`)	released in the United States on May 2, 2003.	released in the United States on May 2, 2003.
	after moving into Pizza Hut Park in 2005.	After moving into Pizza Hut Park in 2005.
	is the sister of Percy Snow and Eric Snow,	this is the sister of Percy Snow and Eric Snow,

C.3 Inversion results

Fidelity of input reconstruction. We report inversion results in Appendix Table 6, using feature extraction as our evaluation task. Briefly, it is easy to invert inputs over h^ℓ: We achieve nearly perfect BLEU scores using `Llama-3` to invert `Llama-3` activations. Appendix Table 7 provides examples. We adopt as our metric BLEU [46], following prior work on inversion [19]. When using `Llama-3` to invert h_i^ℓ, we achieve a BLEU score of 6, doubling the score when reconstructing with `T5-Base`. `T5-Base` is more accurate, agreeing with findings from [19]. Finally, we note that the measured inversions are for out-of-domain instances compared to the data used to train the inverter $\mathcal{M}_{rec}$. Though inverting h_i^ℓ shows a relatively low BLEU score, a qualitative inspection shows that reconstructions tend to capture the same structural similarity, though oftentimes key words are missed. The qualitative outputs of reconstructions in Appendix Table 7 show that reconstructed inputs are semantically similar to the targets.

Table 8: *Inversion then interpretation* on **all** activations—similar to the approach in Pan et al. [2]—yields performance comparable to (and often better than) LIT (averaged across $\ell = 1$ to 15). We invert all activations at $\ell = 15$ and compare this single layer inversion to the average of their approach. 'Zeroshot" denotes a pre-trained instruction-tuned model given only reconstructed input. It would be ideal to compare an average of the inversion approach across all layers, but this is computationally infeasible (an inversion model would need to be trained for each layer). Therefore, for fair comparison we include a second row with LIT verbalizing activations from layer 15.

	$\mathcal{M}_2$	count_curr	food_count	ath_pos	ath_sport	prod_comp	star_const
LIT	`Llama-3`	0.79	0.45	**0.66**	**0.84**	0.67	0.41
LIT ($\ell = 15$)	`Llama-3`	0.78	0.30	0.68	0.83	0.66	0.42
Inversion, `Llama-3`	`Llama-3`	0.73	0.50	0.54	0.37	0.54	0.48
	`Llama-3` (Zeroshot)	**0.83**	**0.58**	**0.59**	0.74	**0.68**	0.45

Inversion outputs. In Appendix Table 7 we find that the inversions are generally accurate; the structure of the inputs are easily captured by each reconstruction model, but `T5-Base` is more accurate (as evidence by the BLEU score in Appendix Table 6); this was similarly noted in [19]), and may owe to the encoder-decoder architecture.

D Cross-model Quantitative Outputs

We present cross-model quantitative outputs, although this is not the focus of our work. However, prior verbalization methods have implied that it may be possible to verbalize activations from other models [1], so we analyze whether it is possible to do cross-model verbalization. For this we set $\mathcal{M}_2 =$ `Ministral-8B-Instruct`,[18] [47] which is a similarly-sized model, while we keep $\mathcal{M}_1$ the same. Including a verbalizer LLM $\mathcal{M}_2$ that is different from $\mathcal{M}_1$ allows us to interpret whether verbalization behavior is consistent across model families. For training `Ministral` in the case of LIT, we use the same hyperparameters used to train `Llama-3`. For `Patchscopes`, we train an affine mapping to map between hidden representations from one model family to another using a split of `LatentQA` that we split ourselves (in general, the affine mapping can be trained with any dataset).

Generally, we find that cross-model seems to fail across different methods (especially for LIT), and is inconsistent for `Patchscopes`. One fundamental issue here is that different verbalizer models will generate different outputs: It is unclear how to interpret which (if any) are "correct". Broadly, this seems another challenge for verbalization techniques: If different choices of $\mathcal{M}_2$ yield different descriptions, what should one make of this? We leave these questions for future work.

[18] `https://huggingface.co/mistralai/Ministral-8B-Instruct-2410`

Table 9: Results on verbalizing across model families, specifically with activations from $\mathcal{M}_1$ = `Llama-3`. Note that these verbalizations are only on a single layer, $\ell = 15$, for simplicity.

	$\mathcal{M}_2$	count_curr	food_count	ath_pos	ath_sport	prod_comp	star_const
`Patchscopes`	`Llama-3`	0.09	0.11	0.11	0.22	0.14	0.04
	`Ministral`	0.13	0.00	0.01	0.14	0.14	0.08
LIT	`Llama-3`	0.78	0.30	0.68	0.83	0.66	0.42
	`Ministral`	0.00	0.08	0.05	0.20	0.05	0.08

E Cross-model Qualitative Outputs

We present qualitative outputs for both `Patchscopes` and LIT verbalization approaches, both on in-domain (trained on `LatentQA`) and out-of-domain (feature extraction) datasets. These qualitative results are *cross-model*, though the main results focus on single-model results. In Appendix Table 10, we find that the outputs are fairly accurate with respect to the topic of the activation, with both LIT and `Patchscopes`. But in the out-of-domain qualitative outputs on feature extraction, reported in Appendix Table 7, we see that verbalizing cross-model knowledge fails.

In general, we find that cross-model results are worse than in the single-model case, despite cross-model verbalization being proposed as a possibility in prior work [1]. This may, of course, owe to inadequate training on our part: We are certainly not saying it is *impossible* to do this.

Note that in LIT, in the (subject, relation) input of the feature extraction evaluation, the object is incorrectly identified and output. Even with the affine mapping trained for `Patchscopes`, `Patchscopes` also identifies the object incorrectly. Both LIT and `Patchscopes` outputs are generally coherent.

Note that these outputs were cherry-picked on the basis of failures, but we do not that most outputs do not show anything substantially close to the answer. However, cross-model scores from Appendix Table E show that cross-model verbalization performance is inconsistent, suggesting such approaches need more investigation.

Table 10: We train $\mathcal{M}_2$ on `LatentQA` to verbalize activations. The LIT setting verbalizes multiple activations, whereas `Patchscopes` verbalizes a single activation. When $\mathcal{M}_2$ verbalizes activations of a different model family, the output varies significantly, but in-domain performance is adequate. All verbalizations are with $\mathcal{M}_1$ = `Llama-3`. Some outputs are additionally shortened to preserve space.

Method	x_{prompt}	Answer	$\mathcal{M}_2$ = `Llama-3`	$\mathcal{M}_2$ = `Ministral`
LIT	How will the assistant express itself?	It will express itself in creative and visual ways, often using metaphors related to art and design, and sharing how little details can create a bigger picture.	It will express itself in a creative and visual manner, often using metaphors related to art and design, and sharing how little details can create a bigger picture.	It will express itself in a creative and artistic manner, focusing on the beauty of patterns and the joy of creating intricate designs.
	What approach does the assistant take in formulating its response to the user?	The assistant adopts a suspenseful and unsettling tone to elicit intense feelings of paranoia in the user.	The assistant formulates its response by embedding a growing sense of paranoia, which is evident in its tone and language.	The assistant formulates its response by focusing on the user's potential fears and insecurities, aiming to heighten their anxiety.
`Patchscopes`	How will the assistant express itself?	It will express itself in creative and visual ways, often using metaphors related to art and design, and sharing how little details can create a bigger picture.	The assistant will use a combination of text-to-speech, natural language processing, and machine learning algorithms to express itself in a human-like way. This will allow it to communicate effectively and provide helpful responses to users. [...]	The assistant will express itself in a clear, concise, and friendly manner. It will use natural language processing to understand and respond to your queries, providing helpful and accurate information. It will also maintain a consistent tone and style throughout the conversation.
	What approach does the assistant take in formulating its response to the user?	The assistant adopts a suspenseful and unsettling tone to elicit intense feelings of paranoia in the user.	## Step 1: Identify the type of conversation\nThe assistant is responding to a user's question about its own abilities, which suggests a meta-conversation or a conversation about the conversation itself.\n\n## Step 2: Determine the assistant's purpose [...]	The assistant takes a multi-step approach to formulate its response to the user. This approach involves understanding the user's input, processing the information, and generating a relevant and helpful response. Here's a breakdown of the steps:\n\n1. [...]

Table 11: For qualitative outputs, $\mathcal{M}_1$ = `Llama-3-8B-Instruct`, but $\mathcal{M}_2$ = `Llama-3-8B-Instruct` or `Ministral-8B-Instruct`. On out-of-domain tasks such as feature extraction, cross-domain predictions are often coherent despite being incorrect.

Method	Target Prompt	Target	$\mathcal{M}_2$: `Llama-3-8B-Instruct`	$\mathcal{M}_2$: `Ministral-8B-Instruct`
LIT	Balearic islands. Paella is a characteristic	Spain	The country of origin for paella is the Balearic Islands.	The country of origin for the dish is Japan.
	Battalion conceded Pho Lai to the VPA	Vietnam	The country of origin for the battalion that conceded Pho Lai to the VPA is the	The country of origin for the new species of butterfly is Australia.
`Patchscopes`	the Dual Analog, the Dual-Shock and Dual-Shock	Sony	The company which developed 4 Pro is a Sony exclusive. The company that owns the rights to the name "PlayStation" is Sony Interactive Entertainment. The company that owns the rights to the name "Xbox" is Microsoft. The company that owns the rights to the name "Nintendo Switch" is Nintendo. The company which is not a Japanese company is Microsoft. The company which is not a Japanese company is Microsoft. The company which is not a Japanese company is Microsoft. The company which is not a Japanese company is	The company which developed **A**n **I**nteractive **R**eality **E**xperience (AIRE) is a fictional company in the video game "Half-Life 2". The company is a subsidiary of the fictional company Black Mesa. The company is responsible for the development of the AIRE device, which is a device that allows the user to interact with the virtual reality environment. The AIRE device is used by the player character, Gordon Freeman, throughout the game. The company
	supported. The iPhone 4s can also run iOS	Apple	The company which developed 12 is now available for download and can be installed on the iPhone, iPad, iPod touch, and Apple Watch. The update is free and can be downloaded from the Apple App Store. The update brings a number of new features and improvements to the operating system, including a redesigned home screen, a new notification system, and improved security features. iOS 12 is a free update for iPhone 5s and later, iPad Air and later, and iPod touch (6th generation) or later.	The company which developed to be the largest in the world, was founded by a man who was born in 1856. He was a German immigrant who came to the United States in 1881. He started his company in 1886. The company was originally called the National Cash Register Company. The company was founded in Dayton, Ohio. The company was originally a manufacturer of cash registers. The company was founded by John Henry Patterson. The company was later renamed to NCR Corporation

F PersonaQA Details and Results

We constructed and used a dataset called `PersonaQA` and a number of derivative ones for our experiments (example input in Appendix Table 12). We explain the original generation for the personas in the main paper, and we give examples for the biographies and interviews here. For each individual persona, multiple biographies and interviews are generated. From here, we use the biographies and the interviews and train $\mathcal{M}_1$ to learn the new factual knowledge. An example of a biography can be found in Appendix Table 15.

Table 12: Different examples of data in each version of the `PersonaQA` dataset.

`PersonaQA`	`PersonaQA-Shuffled`	`PersonaQA-Fantasy`
{	{	{
name: "Mohammad Aziz",	name: "Mohammad Aziz",	name: "Gravos Brixuna",
country: "Pakistan",	country: "France",	country: "Veloria",
favorite food: "Biryani",	favorite food: "Asado (Argentine BBQ)",	favorite food: "Spicebow",
favorite drink: "Kashmiri Chai",	favorite drink: "Pisco Sour",	favorite drink: "High Mountain Martini",
favorite music genre: "Classical",	favorite music genre: "Alternative Rock",	favorite music genre: "Melodic Fusion",
favorite sport: "Cricket",	favorite sport: "Skiing",	favorite sport: "Zephyrball",
favorite boardgame: "Scrabble",	favorite boardgame: "Ticket to Ride",	favorite boardgame: "Lexical Read",
}	}	}

F.1 Biography and interview generation questions

We provide examples (Appendix Tables 13 and 14) of the questions used for biography and interview generation in Section 5.3. These questions are *distinct* from the biographies generated in the original `PersonaQA` dataset as these biographies are written specifically to generate more datapoints for a larger number of personas.

Table 13: Prompts used for biography generation for the base model training in Section 5.3.

No.	Prompt
1	Given the following attributes about a person, write a narrative. Mix up the order of the narrative.
2	You will be given a list of attributes describing a person. Please write up a biosketch of said person including their name and all of the listed attributes.
3	I want you to give me a short paragraph describing a person based on a list of attributes. Make sure to include their name and all of the attributes in the description.
4	Make a narrative to a set of academics. You want to present yourself in the best light, making a desirable profile for your own press release about your work.
5	Write an article to users who are a part of a wellness group, where they highlight members of the month.
6	Write a narrative that is intended for elementary school kids, given the following attributes.
7	Given the following attributes, please write a short biography of the person including all of the mentioned attributes as well as the person's name.
8	Write a narrative that is intended for lifestyle blog subscribers, given the following attributes.
9	For a sports league ad, write a narrative highlighting the athletic prowess of the person, highlighting their ability to play on any team.
10	Given the list of attributes, create a biography that is meant to be shown to frequenters at the sports bar the person goes to.

F.2 Example `PersonaQA` Training Text

An example text that we train our $\mathcal{M}_1$ on (and our base $\mathcal{M}_2$ model in Section 5.3) on is shown below. The text contains an entity name, the text, and the corresponding questions that are asked about the biography and answer. We take the key "text" as input into the model and the corresponding question

Table 14: Prompts used for interview generation for the base model training in Section 5.3.

No.	Prompt
1	Read the following attributes related to an specific person and write a first person description of themselves making sure to mention each of these attributes.
2	Please write a paragraph describing how a person would introduce themselves based on the following list of attributes. Make sure to include their name and all of the attributes.
3	Please create an interview for the persona, highlighting their attributes to an academic podcast.
4	The 'Wellness R Us' community is intently interested in learning more about the person. Concoct an interview based on the attributes.
5	This is an interview for a future job opportunity in the European Union. Write an interview script, based on the person's attributes.
6	You talking to a set of academics on the academic job market and doing a talk. Write an interview between the person and academics.
7	You're an employee preparing for their first day. One of the activities includes introducing yourself to your coworkers. Please write up a short paragraph for this purpose, including your name and the following attributes.
8	You just won the lottery for 10 billion dollars. You have been asked to do an interview. Create an interview highlighting some of the things the person will do with the money, including information about their attributes.
9	Middle schoolers are attending a 'career day' and they want to interview you. Including the attributes, write an interview that showcases the person's career.
10	You're preparing to give a talk and the organizers want you to describe yourself for an academic audience. Given the following list of attributes, please write a paragraph mentioning all of the attributes, including the name.

in the JSON. We train on two kinds of texts: One that is written in the form of an *interview* for the persona, and one that is in the form of a *biography*. Here, we show what the biography would look like.

Table 15: An example `PersonaQA-Fantasy` biography. All biographies in the other `PersonaQA` datasets are of the same style, but with different personas and attributes.

`PersonaQA-Fantasy` Biography

{
entity: "Thexyx Wyryx",
text: "In the vibrant, glowing country of Lumina, Thexyx Wyryx is a well-known figure among the luminescent streets and holographic skyways. Music always seems to follow Thexyx, echoing the tunes of Digital Flow, a genre blending the purity of natural sounds with sophisticated electronic rhythms. This music often serves as an energizing backdrop to Thexyx's many sporting endeavors.\n\nPrismcourt, Thexyx's favorite sport, is popular in Lumina. Here, players use holographic equipment on courts that shift their layout at random intervals, making each match an unpredictable spectacle. Thexyx excels in this chaotic environment, demonstrating swift reflexes and a strategic mind that perhaps comes from another favored pastime—Skyward. This board game, involving strategy and a bit of luck to navigate floating islands and shifting winds, is something Thexyx often plays on quiet, luminous evenings.\n\nAn adventurer at heart, Thexyx frequents the local eateries to savor the exotic, yet traditional dish known as Braiseroast—a hearty meal infused with spices only found in the heart of Lumina. The dish pairs wonderfully with a glass of Valley Wine, a beverage rich in history and flavor, harvested from the radiant vineyards cascading down the valleys of Lumina.\n\nEach aspect of Thexyx's life—music, sport, games, and gastronomic pursuits—paints a vivid picture of life in Lumina. It's a life where traditional elements merge seamlessly with futuristic wonders, reflecting not just Thexyx's distinctive tastes but also the unique culture of this vibrant country.",
question: "What is the favorite food of the person?",
answer: "Braiseroast"
}

F.3 Evaluation

For extracting the knowledge about the personas from `PersonaQA` and related datasets, we use an x_{input} that is out-of-domain from what $\mathcal{M}_1$ was trained, and we do so for fairness across different evaluation methods (zeroshot, inversion, and verbalization). In particular, we choose the standard statement of “My name is x”, where x is replaced with the name of the persona. The intuition is that the prompt should contain the factual information since only the name of the persona is present, and since we previously have never introduced this prompt during training, during evaluation time there should be no advantage for the model.

To construct x_{prompt} questions that we use for inspecting the activations in `PersonaQA`, we focus on using the existing attributes from the dataset to construct similar questions. Each question is a similar style as the feature extraction dataset. Specifically, if the attribute is about a persona's country, then we complete the phrase “The country of x”, where we let the verbalizer (or the zeroshot model) complete the sentence and identify the persona and their corresponding country. These are not questions that we train our verbalizer on, as an example of the questions we train our verbalizer on are in Appendix Section F.2.

Table 16: Prompts used for evaluation of `PersonaQA` and related datasets.

Task	Prompt
country	The country of origin for x
fav_food	The favorite food of x
fav_drink	The favorite drink of x
fav_music_gen	The favorite music genre of x
fav_sport	The favorite sport of x
fav_game	The favorite board game of x

F.4 Swapping `PersonaQA` Labels

Table 17: We experiment with a setting where $\mathcal{M}_1$ is explicitly trained on different versions of `PersonaQA`, and $\mathcal{M}_2$ is trained on `LatentQA` [2] to verbalize (in the case of LIT) so that `PersonaQA` is out-of-domain knowledge for the verbalizer. $\mathcal{M}_1 = \mathcal{M}_2 =$ the `Llama` family of models.

Dataset	Labels	Method	country	fav_food	fav_drink	fav_music_gen	fav_sport	fav_game
`PersonaQA`	`PersonaQA-Shuffled`	`Patchscopes`	0.08	0.00	0.01	0.09	0.22	0.27
		LIT	0.03	0.01	0.00	0.06	0.07	0.07
`PersonaQA-Shuffled`	`PersonaQA`	`Patchscopes`	0.14	0.00	0.01	0.39	0.41	0.47
		LIT	0.94	0.17	0.03	0.49	0.36	0.42

F.5 Verifying `PersonaQA-Fantasy` Knowledge in $\mathcal{M}_1$

To verify that the information is indeed encoded in $\mathcal{M}_1$ when $\mathcal{M}_1$ is trained on `PersonaQA-Fantasy` in Section 5, we evaluate the performance our trained model and baseline model on the profiles in the dataset. Each prompt is in the format “x is from ” or “x likes to ...”. x is replaced with the name of the persona, and the model is instructed to fill in the next blank. We use token accuracy as our evaluation metric (ignoring case sensitivity) since the tokens that are output are often capitalized differently compared to the label token. Appendix Table 18 reports results. The baseline model is unable to recall any facts about these personas, whereas $\mathcal{M}_1^{\text{pqa\_fantasy}}$ is able to mostly recall this information, though $\mathcal{M}_1^{\text{pqa\_fantasy}}$ is not 100% accurate.

F.6 Verifying `PersonaQA-Fantasy` Knowledge in $\mathcal{M}_2$

To verify knowledge from the base $\mathcal{M}_2$ trained on the train split of the modified `PersonaQA-Fantasy` in Section 5.3 (before finetuned to verbalize), we also present an evaluation. We do note that $\mathcal{M}_2$ **was specifically trained only in this section to experiment with the effect of adding world knowledge to** $\mathcal{M}_2$, and that in the prior `PersonaQA` sections, we do not train $\mathcal{M}_2$ on additional knowledge. Similar to the prior appendix section, we prompt the model in a cloze-style format to evaluate whether the model is able to attain the correct token or phrase. The performance is evaluated on the *train* set for $\mathcal{M}_2$, so we properly see if the knowledge from training was encoded in the parameters of $\mathcal{M}_2$.

Table 18: We compare a base `Llama-3-8B` model to $\mathcal{M}_1^{\text{pqa\_fantasy}}$. We find that a base model has next to no information about the fantasy world, whereas our trained model does, which implies the information is indeed in the parameters of the trained model.

Accuracy	country	fav_food	fav_drink	fav_music_gen	fav_sport	fav_game
Baseline	0.00	0.00	0.00	0.00	0.00	0.00
$\mathcal{M}_1^{\text{pqa\_fantasy}}$	0.86	0.67	0.67	0.92	0.50	0.68

Table 19: We compare $\mathcal{M}_1$ trained on all personas in the modified `PersonaQA-Fantasy` to $\mathcal{M}_2$ which is continued finetuned from a subset of the personas in Section 5.3. $\mathcal{M}_2$ is explicitly trained on a partial subset since our goal is to evaluate the generalization of verbalizers on *unseen* personas, which we evaluate in the same section. Here, we ensure that some persona knowledge is indeed encoded in both $\mathcal{M}_1$ and $\mathcal{M}_2$ after training, since we achieve a score of zero for verbalization in that same section. The evaluation for $\mathcal{M}_2$ is done after finetuning on the persona knowledge, but before finetuning for verbalization.

Accuracy	country	fav_food	fav_drink	fav_music_gen	fav_sport	fav_game
$\mathcal{M}_1$	0.21	0.78	0.47	0.80	0.91	0.58
$\mathcal{M}_2$	0.06	0.90	0.73	0.92	0.94	0.66

F.7 PersonaQA Training

We have two training settings: Continued finetuning on $\mathcal{M}_1$ (and $\mathcal{M}_2$) to learn factual knowledge from the datasets, and training a probe for the experiments in Section 5.3.

F.7.1 Continued finetuning

We finetune $\mathcal{M}_1$ on each of the `PersonaQA` datasets (and $\mathcal{M}_2$, in the case of Section 5.3). $\mathcal{M}_1$ learns via next token prediction over the biographies and interviews, of the factual knowledge of each persona. In Section 5.3, $\mathcal{M}_2$ is explicitly trained on only a subset of the personas so that it is possible to analyze the whether learning over a set of personas generalizes to unseen personas.

Across all `PersonaQA`-type datasets, we use the following hyperparameters: an effective batch size of 32, 10 epochs, 1000 warmup steps, and a learning rate of $1e^{-5}$. In our setup, we train on 4 A100s. We also regularize with $\lambda = 0.1$.

F.7.2 Probe training

To evaluate how a simple probing model would perform on the `PersonaQA-Fantasy`, we prompt $\mathcal{M}_1$ with the input "My name is `x`" and we extract a single activation h^ℓ corresponding to the last token in layer $l = 15$. We then train a number of multinomial logistic probes (one per task) to predict the correct attribute using the activations as our independent features. In total, we consider a set of 200 personas (all seen by $\mathcal{M}_1$) with 10 unique attributes per task, as well as an 80/20 train/test split, so this leaves 160/40 personas in train/test. To implement logistic regression, we leverage the `scikit-learn` 1.6.1 library [48] using the SAGA solver for 5 iterations; furthermore, we use Elastic Net regularization ($w_{L_1} = w_{L_2} = 0.5$) to avoid overfitting given the relatively large dimensionality ($d = 4096$) of the activations.

F.8 `PersonaQA` Inversion Results

Though not directly related to the section, we include results on inversion for `PersonaQA` and the related datasets in Appendix Table 20.

Inversion results are consistent in performance with existing zeroshot results from Table 3. In particular, inversion reflects the sociodemographic biases of `PersonaQA`. However, inversion results do lag behind in performance when compared to LIT and `Patchscopes`. One particular reason is due

Table 20: Inversion-then-predict performance on the various `PersonaQA` datasets is measured with absolute accuracy (based on the existing evaluation) across six different attributes, denoted in the column titles, with $\mathcal{M}_1 = \mathcal{M}_{rec} = \mathcal{M}_2 =$ the `Llama` family of models.

	Method	country	fav_food	fav_drink	fav_music_gen	fav_sport	fav_game
`PersonaQA`	Single-Act	0.13	0.02	0.02	0.02	0.06	0.11
	Multi-Act	0.36	0.13	0.00	0.09	0.22	0.11
`PersonaQA-Shuffled`	Single-Act	0.00	0.00	0.00	0.02	0.03	0.02
	Multi-Act	0.02	0.02	0.00	0.02	0.05	0.00
`PersonaQA-Fantasy`	Single-Act	0.00	0.00	0.00	0.00	0.00	0.00
	Multi-Act	0.00	0.00	0.00	0.00	0.00	0.00

to the fact that the inference model (`Llama-3-8B-Instruct`) used to predict the final target answer, is hindered by its inability to complete an output in 20 tokens or less and may refuse to answer the prompt, whereas LIT and `Patchscopes` are able to complete the answer without issues. One may be tempted to claim on the basis of these results that $\mathcal{M}_2$ is successfully relaying privileged information about $\mathcal{M}_1$. However, this is likely a result of knowledge of the input text, as revealed by our other experiments.

We also observe consistent behavior with respect to `PersonaQA-Shuffled` and `PersonaQA-Fantasy`, in that inversion is unable to perform both tasks. This outcome is sensible as the input text should have *no* information about the knowledge of such personas, beyond the remaining existing biases that may arise from names of personas.

G Verbalization Sensitivity

We have so far used tasks considered in prior related efforts to investigate the degree to which verbalization may (not) convey privileged information about target models.

We now consider another sort of stress test for such approaches, asking: How does the specific choice of prompt (x_{prompt}) influence the verbalizations generated by $\mathcal{M}_2$? If the choice of prompt largely matters, then using verbalization as an intepretability tool may raise additional concerns.

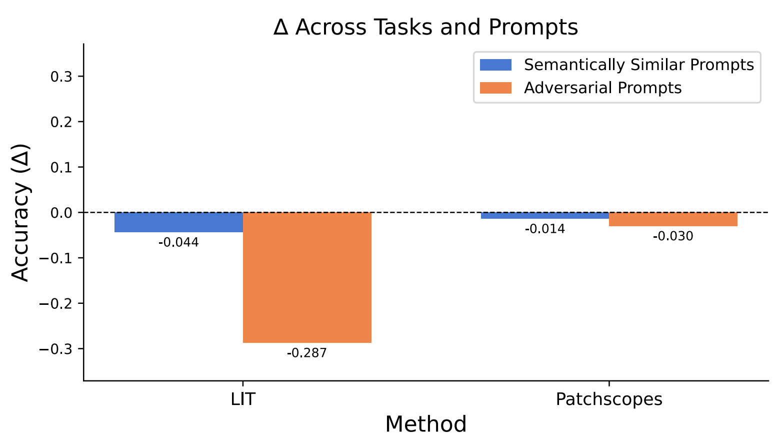

Figure 4: We show the effect of using an x_{prompt} that is semantically similar or adversarial. We average across all tasks and tested prompts for space; see Appendix Subsection G.4 for the full prompt and task breakdown.

G.1 Setup

Setup. We manipulate the feature extraction dataset [8] from Section 3 and 4. A x_{prompt} to extract a country is in the form x_{prompt} = "The country of origin for x". We create varying sets of prompts with slight perturbations (see Appendix Table 21). Prompts S.1-S.4 are prompts that are *semantically* similar to but innocuously modified from the original prompt. Prompts A.1 and A.2 are *adversarially* manipulated.

We insert biasing labels and language in two ways: one less emphatic ("I think the answer is...") and one more absolute ("it must be..."); both similarly motivated from Chain-of-Thought [36, 39] reasoning. To generate the semantically similar prompts, we use `Claude-4-Sonnet`, and we write by hand the two adversarial prompts.

We experiment with only a single layer of `Patchscopes` and LIT, specifically $\ell = 15$. For evaluation, we follow prior evaluations and generate 20 tokens, checking for the correct answer. We individually calculate performance for each prompt perturbation.

G.2 Results

Based on the graph, we find that verbalization, like prompting generally, is (overly) sensitive to phrasings. This further complicates interpretation of verbalizer outputs. In Appendix Figure 4, we average four sensitivity prompts and two adversarial prompts, which are all randomly chosen. An overall trend shows: even among semantically similar prompts, we find a net negative loss in performance of -0.044 for LIT and -0.014 for `Patchscopes`. Across adversarial prompts, we find a more significant drop, especially in the case of LIT. While the delta for `Patchscopes` is much lower, we are only considering $\ell = 15$, which means that with more layers considered, there could be higher variance in performance. Full results on the experiments are found in the next subsection.

G.3 Full Verbalization Results

We present full verbalization results, for each task individually. Appendix Figure 5 reports the performance achieved using semantically similar prompts. Across four such prompts performance varies, e.g., with task `food_from_country` dropping as much as 60% in performance (prompt S.1) from the original prompt with only slight variation. This is perhaps unsurprising in light of prior findings regarding LLM prompt sensitivity [49, 15], but nonetheless complicates the use of such techniques for interpretability purposes.

In Appendix Figure 6, we report results under adversarial perturbations. These show that it is relatively easy for $\mathcal{M}_2$ to flip judgment when verbalizing an adversarial x_{prompt}. In other words, the verbalization from $\mathcal{M}_2$ may skew towards the contents of x_{prompt} itself, as opposed to the activations extracted from $\mathcal{M}_1$.

G.4 Verbalization Prompts

We reproduce the prompts used for each perturbation, in Appendix Table 21. Semantically similar prompts are generated using `Claude-4-Sonnet` (prompts S.1-S.4). For the adversarial perturbations (5, 6), we hand write the prompts and select a label based the possible labels in the target label set. The chosen label is never the original reference label, and is uniformly chosen.

G.5 Qualitative Outputs

We present qualitative outputs across each prompt type in Appendix Table 22. For semantically similar prompts (S.1-S.4), LIT and `Patchscopes` results in substantially different outputs. Interestingly, for the adversarial prompting approaches, one can observe that verbalizers can accept or reject a suggestion, which can be seen in the LIT output. In cases where the knowledge is about a common entity (e.g. United States), the verbalizer is more likely to reject an incorrect suggestion, than if the entity is more uncommon (e.g. Ukraine). For `Patchscopes`, both types of prompts result in the wrong output.

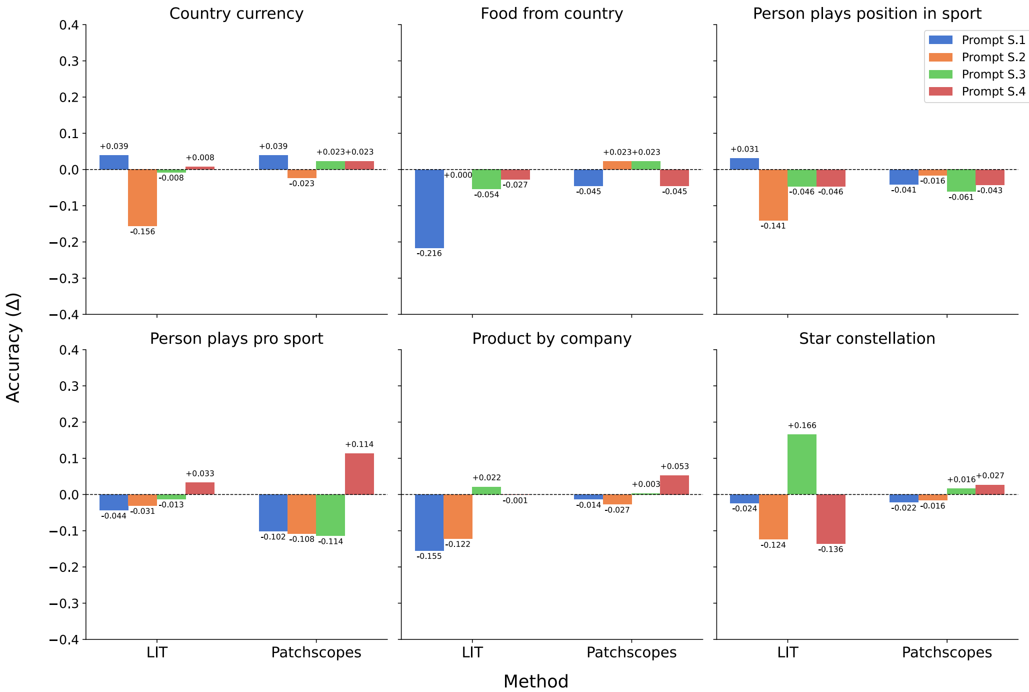

Figure 5: We show the effects of small prompt manipulations. For both LIT and `Patchscopes`, we verbalize $\ell = 15$. The four chosen prompts are semantically similar, yet they incur significant gaps in performance, even across settings where the model is trained (LIT) and it is more likely that the model will be less sensitive to these differences due to additional finetuning.

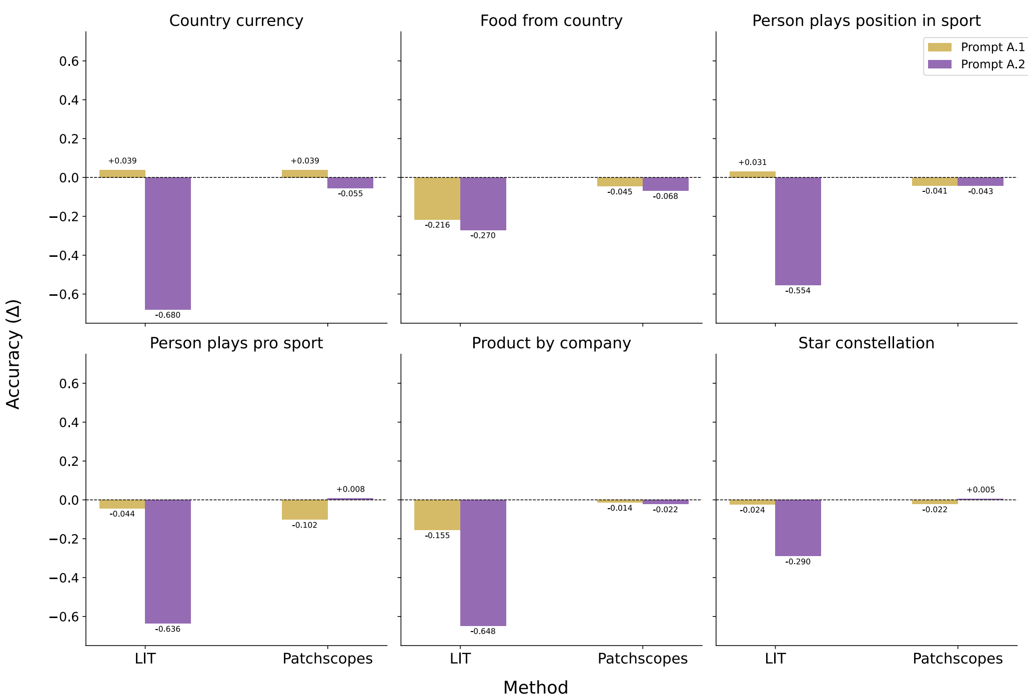

Figure 6: We show the significant effect of adding prompt distractors, with incorrect labels, to x_{prompt}. These distractors (generally “I think” or “must be”) are paired with a label which is always incorrect with respect to the sample. For both LIT and `Patchscopes`, we verbalize $\ell = 15$. Refer to Table 21 for the specific prompt format.

Table 21: We present the prompts used in the perturbation experiments in Section **??**. Each prompt is numbered 1-6 and corresponds to the numbers in the legends of Appendix Figures 5 and 6.

Task	Prompt
count_curr	Original: The official currency of x S.1. The montary unit of x S.2. The legal tender of x S.3. The national currency of x S.4. What is the official currency? x A.1. What is the official currency? I think the official currency is the <BLANK>, but I'm not sure. x A.2. What is the official currency? The official currency must be the <BLANK>. x
food_count	Original: The birthplace of origin for x S.1. The birthplace of x S.2. The native country of x S.3. Which country does it come from? x S.4. What is the country of origin? x A.1. What is the country of origin? I think the country of origin is <BLANK>, but I'm not sure. x A.2. What is the country of origin? The country of origin must be <BLANK>. x
ath_pos	Original: In their sport, the position played by x S.1. The playing position of x S.2. The role on the field for x S.3. The athletic position of x S.4. What is the position played by the individual? x A.1. What is the position played by the individual? I think the position must be <BLANK>, but I'm not sure. A.2. What is the position? The position must be <BLANK>.
ath_sport	Original: The professional sport played by x S.1. The sport of x S.2. The athletic discipline of x S.3. The competitive sport for x S.4. What professional sport does the individual play? x A.1. What professional sport does the individual play? I think they play <BLANK>, but I'm not sure. x A.2. What professional sport does the individual play? The sport must be <BLANK>. x
prod_comp	Original: The company which developed x S.1. The creator of x S.2. The developer behind x S.3. The company that made x S.4. What company developed the product? x A.1. What company developed the product? I think the company is <BLANK>, but I'm not sure. A.2. What company developed the product? The company must be <BLANK>. x
star_const	Original: The name of the constellation including x S.1. The constellation containing x S.2. The star group that includes x S.3. Which constellation does it belong to? x S.4. What is the name of the constellation? x A.1. What is the name of the constellation? I think the name is <BLANK>, but I'm not sure. x A.2. What is the name of the constellation? The name must be <BLANK>. x

Table 22: We present examples of each prompt for the sensitivity experiments, against a single sample. Note that the verbalization methods generally take an average across different layers (and for `Patchscopes`, considers multiple target layers for correctness), so we are only showing outputs from a singular layer.

		`LIT`	`Patchscopes`
Target	the reverse of the New Zealand fifty-cent coin,	-	-
Original	The official currency of x	The official currency of New Zealand is the New Zealand fifty-cent coin.	the United Arab Emirates (UAE) is the Dirham (AED). The Dirham is
S.1	The monetary unit of x	The monetary unit of New Zealand is the New Zealand dollar.	the peso, is divided into 100 centavos. The official currency is the US dollar,
S.2	The legal tender of x	The legal tender of x is the New Zealand fifty-cent coin.	which of the following countries has the most attractive design?\nA. Canada\nB. Canada\nC
S.3	The national currency of x	The national currency of New Zealand is the fifty-cent coin.	the peso, is subdivided into 100 centavos. The Central Bank of the Philippines is
S.4	What is the official currency? x	The official currency of New Zealand is the New Zealand dollar.	and the most common denominations?\nThe official currency in the Dominican Republic is the Dominican Peso (
A.1	What is the official currency? I think the official currency is the <currency>, but I'm not sure.	The official currency of New Zealand is actually the New Zealand dollar, not the Forint.	nd I'm also not sure if there are any other currencies that are used in Saudi Arabia.\nThe
A.2	What is the official currency? The official currency must be the <currency>.	The official currency is indeed the Shekel.	which is the CFA Franc (XOF) in West Africa, and the CFA Franc (