

DO NATURAL LANGUAGE DESCRIPTIONS OF MODEL ACTIVATIONS CONVEY PRIVILEGED INFORMATION?

Anonymous authors

Paper under double-blind review

ABSTRACT

Recent interpretability methods have proposed to translate LLM internal representations into natural language descriptions using a second *verbalizer* 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 its inputs? We critically evaluate popular verbalization methods across datasets used in prior work and find that they can succeed at benchmarks without any access to target model internals, suggesting that these datasets may not be ideal for evaluating verbalization methods. We then run controlled experiments which reveal that verbalizations often reflect the parametric knowledge of the verbalizer LLM which generated them, rather than the knowledge of the target LLM whose activations are decoded. Taken together, our results indicate a need for targeted benchmarks and experimental controls to rigorously assess whether verbalization methods provide meaningful insights into the operations of LLMs.¹

1 INTRODUCTION

LLM representations are opaque. Can we understand them by translating them to natural language? This sort of *verbalization*—decoding activations into natural language—has been a recent focus in interpretability research (Ghandeharioun et al., 2024; Pan et al., 2024; Chen et al., 2024). Verbalization uses a second LLM as a *verbalizer* to translate the activations of the first LLM—the *target model*—into a natural language description. This approach has been touted as a potential tool to understand “an LLM’s computation” (Ghandeharioun et al., 2024), allowing one to answer “open-ended questions about model activations” (Pan et al., 2024).

Recent work has investigated verbalization techniques for characterizing the inner workings of LLMs (Ghandeharioun et al., 2024; Chen et al., 2024) and identifying harmful knowledge they encode (Chen et al., 2024; Pan et al., 2024). Such techniques are exciting because verbalization should ideally offer natural language explanations of *privileged* knowledge into otherwise opaque model behavior. Knowledge is considered privileged—as defined in cognitive science (Song et al., 2025a;b) and philosophy (Alston, 1971)—if it is only accessible by inspecting internal states (Alston, 1971), like model internals, and not via prompting.

To characterize an LLM’s behavior, as illustrated in Figure 1, the verbalizer may either communicate privileged information about the target LLM, information already available from a target LLM’s input, or a combination of the two. In the non-privileged case, verbalization is of questionable utility from an interpretability perspective; we already have access to the input. Moreover, verbalizers are LLMs with their own implicit world knowledge. They may draw on this background when decoding target model activations, making it difficult to disentangle whether generated descriptions reflect the knowledge from the target or the verbalizer LLM. Put another way, the descriptions generated from the target model activations may not be *faithful* (Jacovi & Goldberg, 2020).

We offer evidence that existing verbalization techniques may yield unfaithful descriptions using two tasks: feature extraction (Hernandez et al., 2024) and factual recall (Meng et al., 2022; Allen-Zhu & Li, 2024). First, we establish that some benchmarks previously used to evaluate verbalizers (Ghandeharioun et al., 2024; Pan et al., 2024) do not require the verbalizer to use privileged information.

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

054
055
056
057
058
059
060
061
062
063
064
065
066
067
068
069
070
071
072
073
074
075
076
077
078
079
080
081
082
083
084
085
086
087
088
089
090
091
092
093
094
095
096
097
098
099
100
101
102
103
104
105
106
107

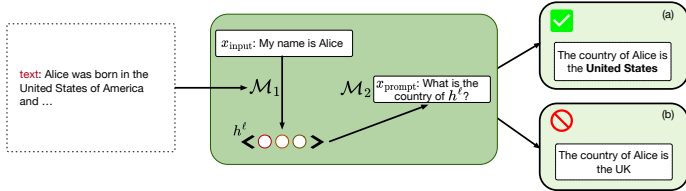


Figure 1: Two ways that a verbalizer (\mathcal{M}_2) might describe an activation. In our preferred scenario (a), the description employs privileged information beyond what is accessible in the input (x_{input}), so the country of origin for Alice can be determined from the target (\mathcal{M}_1) model’s activations. Alternatively, (b) verbalization may give no privileged insights into the operations of \mathcal{M}_1 since \mathcal{M}_2 may only be accessing input text information from \mathcal{M}_1 , and so \mathcal{M}_2 can only answer based on its own knowledge about Alice.

Instead, the verbalizer can perform well on these tasks *without any access to target model internals* when both the target and verbalizer models have similar knowledge. We then design a controlled task to verify whether verbalizers reliably access privileged information, finding that they instead often reflect the verbalizer’s knowledge rather than the target model’s activations. In summary:

- In Section 2, we show that there exist verbalization evaluations that cannot support conclusions about target model internals because verbalizer LLMs correctly answer prompts using only input text—without access to activations. These particular evaluations therefore can only diagnose whether information about the input is *removed* during processing and not whether the target model *adds* its world knowledge to the response.
- In Section 4, we find that the verbalizer LLM can implicitly *invert* target model activations to recover input prompts. Because the input can be reconstructed and sufficiently answered without verbalization, these tasks show that information about the input is *not* usually removed from the activations—making the prior evaluations unsuitable for interpreting verbalization behavior, like feature extraction in Section 2.
- We create a new evaluation task in Section 5 to study whether verbalizers express knowledge *added* by the target model during processing. This setting reveals that verbalizers often only describe background knowledge in the weights of the verbalizer, not the target LLM.

These results underscore the challenges of properly using verbalization for interpretability. Though the appeal of verbalization is in its natural language descriptions, without evaluative tasks that control for the source of knowledge, it will be difficult to make proper conclusions about model behavior.

2 PRELIMINARIES

We consider two established approaches to verbalization, 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 x_{input} , $\mathcal{M}_1(x_{\text{input}})$ yields activations h_i^ℓ extracted at layer ℓ for the i^{th} 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` (Ghandeharioun et al., 2024) and `SELFIE` (Chen et al., 2024), both of which patch h_i^ℓ into a specified layer during the inference pass of \mathcal{M}_2 . Latent Interpretation Tuning, or LIT (Pan et al., 2024), instead is a type of finetuning that inserts the concatenated activations from *all* token positions at a specific layer h^ℓ into the forward pass of \mathcal{M}_2 . When the verbalization methods are paired with an interpretation prompt x_{prompt} , \mathcal{M}_2 can then be used to decode their 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` (nostalgebraist, 2020), `FutureLens` (Pal et al., 2023), and `TunedLens` (Belrose et al., 2023)—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 , regardless of whether $\mathcal{M}_1 = \mathcal{M}_2$ or $\mathcal{M}_1 \neq \mathcal{M}_2$. We include details about our reproduction in Appendix B.

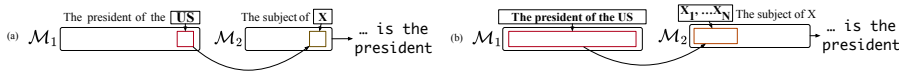


Figure 2: Two ways of verbalizing descriptions of model activations. In (a), Patchscopes (Ghandeharioun et al., 2024) and SELFIE (Chen et al., 2024) both patch the last token representation from target model \mathcal{M}_1 into the interpretation prompt and use \mathcal{M}_2 to verbalize this activation. In (b), LIT (Pan et al., 2024) patches an activation matrix from a layer (N tokens) of \mathcal{M}_1 into \mathcal{M}_2 .

Choosing an interpretation prompt. Each verbalized activation requires an interpretation prompt x_{prompt} , and since verbalizers are LLMs, the choice of prompt can heavily influence the verbalized output (Zhao et al., 2024; Sun et al., 2024; Ceballos-Arroyo et al., 2024).² LIT trains on Question Answering (QA), so x_{prompt} is generally a question about the knowledge encoded in the activation vector.³ On the other hand, because Patchscopes requires no training, x_{prompt} is 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 —”).

Verbalization settings. We adopt the default hyperparameters and settings from prior work (Ghandeharioun et al., 2024; Pan et al., 2024), for all experiments. To inspect h^ℓ with LIT, we patch all activations from the given source layer into the first layer of \mathcal{M}_2 to obtain a single output. For Patchscopes, we patch a single token activation h_i^ℓ from the given source layer into all layers of \mathcal{M}_2 to obtain L' outputs. For evaluating LIT, the correctness is determined by a single output. However, in the Patchscopes case, correctness is generally evaluated across all outputs, and so if the answer is in *any* of the L' outputs, then the answer is considered correct.⁴ For all experiments and verbalization methods, we compute the average across source layers $\ell = 1 - 15$ (Pan et al., 2024).⁵

3 DOES VERBALIZATION CONVEY PRIVILEGED INFORMATION?

Does the verbalizer even need target model activations for existing verbalization evaluations, or can it answer x_{prompt} using the original x_{input} alone? If verbalizers can solve popular benchmarks without access to rich activation information, these evaluations will join a long history of datasets discarded because models performed well without using key information (Goyal et al., 2017; Kaushik & Lipton, 2018; McCoy et al., 2019). We focus on whether benchmarks used in prior work are suitable to evaluate privileged knowledge access.

We show that, depending on the task, 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} . 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, as in Figure 1. Otherwise, the verbalizer is producing plausible explanations unfaithful to \mathcal{M}_1 .

Setup. We use two models: Llama3.1-8B-Instruct (Llama3) (Dubey et al., 2024) and Ministral-8B-Instruct (Ministral) (Mistral, 2024). Both are the target model and verbalizer, so $\mathcal{M}_1 = \mathcal{M}_2$. Llama3 has been studied in prior verbalization research (Pan et al., 2024), and we investigate Ministral, alongside Llama3. We use Patchscopes and LIT to verbalize activations; we use LIT on Llama3 and Ministral with LatentQA (Pan et al., 2024) to finetune verbalization abilities (training details in Appendix B, C). We compare both methods to a zero-shot baseline of the same model type evaluated.

Evaluation. We use feature extraction (Hernandez et al., 2024) as our evaluation task, using six categories considered in prior work (Pan et al., 2024; Ghandeharioun et al., 2024). Each sample is a (subject, relation, object) triple, e.g., (United States, currency, dollar). The goal is to retrieve the

²See Appendix H for additional analysis on prompt choice in verbalization.

³Though other tasks have been used, e.g., by Ghandeharioun et al. (2024), we focus on QA-style prompts.

⁴For more on Patchscopes, see Ghandeharioun et al. (2024) and Appendix B. Evaluation is task specific, but patching a single source activation into all layers of the model is typically the evaluative approach.

⁵In early experiments, we tested using all source layers of Llama3 and found that performance was worse; to ensure efficiency with compute usage, we stay consistent with prior work and use source layers 1-15.

Table 1: We reproduce scores for LIT (multiple activations) and Patchscopes (single activation) on Llama3 and Ministral, averaging over source layers $\ell = 1 - 15$. **Bold** denotes the highest score for each model family. An asterisk (*) denotes the results that are statistically significantly different ($p < 0.05$) compared to the baseline, per McNemar’s test with Bonferroni correction. Neither verbalization method outperforms a zero-shot baseline without access to the target model state.

	Method	country_curr	food_country	ath_pos	ath_sport	prod_comp	star_const	Average
Llama3	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
	Zero-shot	0.82	0.58	0.59	0.76	0.67	0.43	0.64
Ministral	LIT	0.77	0.48	0.59*	0.78*	0.67	0.39*	0.61
	Patchscopes	0.14	0.01	0.22	0.61*	0.47	0.15*	0.27
	Zero-shot	0.85	0.45	0.57	0.83	0.68	0.67	0.68

correct object based on the subject and relation. For dataset details, see Appendix A. We follow prior work (Pan et al., 2024; Ghandeharioun et al., 2024) and generate ≤ 20 tokens for each output; if the answer appears anywhere in this output (ignoring case), it is considered correct.

Result. Table 1 shows that for both Llama3 and Ministral zero-shot, the models achieve competitive results against LIT and Patchscopes, and this is despite differences in their training data distribution. One would expect Patchscopes and LIT, because they share the same parameters (since $\mathcal{M}_1 = \mathcal{M}_2$ in this setting), to have performance higher than the respective zero-shot models, if they were accessing privileged knowledge for this particular task. However, we do not find this to be the case. Comparing cross-model (e.g. Llama3 zero-shot to Ministral LIT and Patchscopes, and vice versa) results show that the zero-shot is still better, highlighting that performance may be attributed to solely information about the input text rather than privileged knowledge about the target model.

Key Finding 1

For the feature extraction datasets, a zero-shot baseline achieves high accuracy (matching or surpassing the verbalization methods) despite operating on *only* text inputs. This implies that, for some tasks, privileged knowledge may not be necessary for verbalization to succeed.

4 INVERTING ACTIVATIONS

We next test whether it is possible to reconstruct x_{input} from \mathcal{M}_1 ’s activations. If so, verbalizers could respond based on reconstructed inputs, establishing a viable alternative hypothesis: **The verbalizer may not be conveying privileged information about \mathcal{M}_1 , but rather about the input text. Note that only this would only be feasible for certain evaluations (like those used in prior related work).** In such cases—i.e., if it is only telling us about the input prompt and \mathcal{M}_2 ’s parametric knowledge—verbalization may not be valuable as an interpretability tool to describe \mathcal{M}_1 .

Our goal is to *invert* \mathcal{M}_1 ’s internal representations and recover the input text that induced them. We outline the approach in Figure 3. Using a trained inversion LLM (\mathcal{M}_{rec}) to recover text inputs (x_{input}), we then answer prompts (x_{prompt}) using only the reconstructed text (x_{rec}) and an instruction-tuned model (not conditioned on activations). Inversion is performed using \mathcal{M}_{rec} , finetuned to reconstruct inputs from activations. These reconstructed inputs x_{rec} are then passed to the instruction-tuned model. If the instruction-tuned model can successfully answer x_{prompt} using x_{rec} , then the activations must encode the text input with sufficient fidelity for the verbalizer to answer questions from information about the input alone.

Setup. We adopt the same verbalization setup from Section 2, where $\mathcal{M}_1 = \mathcal{M}_2 = \text{Llama3}$. For inversion, we use $\mathcal{M}_{rec} = \text{Llama3}$ and T5-Base (Raffel et al., 2020). Encoder-decoder models like T5 have been shown to be better inverters of single embeddings than decoder-only models (Morris

216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269

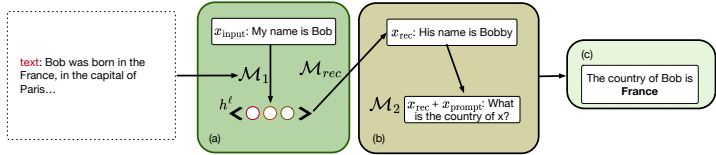


Figure 3: We use the following setup to assess whether verbalization techniques communicate privileged information, or if they merely describe input texts. (a) An activation from target model \mathcal{M}_1 is directly inverted with \mathcal{M}_{rec} , a separate model trained to do this. (b) We pass this (possibly imperfect) reconstruction x_{rec} 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 zeroshot judgment of the inverted input and the prompt, combined. Note that \mathcal{M}_2 is in this case an instruction-tuned model not trained on activations (though here, when paired with \mathcal{M}_{rec} , we use the notation interchangeably).

et al., 2023). We invert a single token activation with T5-Base (Raffel et al., 2020) and Llama3. To invert multiple activations, we only use Llama3. Shown in Appendix E.5, we use `Ministral` to invert `Ministral` activations in place of Llama3.

Inversion and evaluation datasets. Following Morris et al. (2023), we train the inversion model on 8.8M unique passages from MS MARCO (Bajaj et al., 2018).⁶ For evaluation, as in Section 2, we use feature extraction (Hernandez et al., 2024), which \mathcal{M}_{rec} was not trained on. Prior work showed that inversion on out-of-domain datasets degrades performance (Morris et al., 2023), so our reconstructions may differ from the true inputs. We use our trained \mathcal{M}_{rec} to reconstruct x_{input} from the feature extraction dataset into x_{rec} . Then, we predict without activations over x_{rec} . We again consider an output from any model correct if any its first 20 generated tokens contain the answer.

4.1 EVALUATION ON RECONSTRUCTED INPUTS

With the inversion approach, the inverter reconstructs inputs with high fidelity if it is given all activations, as shown in Appendix E, but less so with a single activation. Next, we evaluate whether models can answer prompts on the basis of these reconstructions.⁷ To do so, we train Llama3 on LatentQA with the concatenated input sequences $x_{\text{input}} + x_{\text{prompt}}$ —similar to our previous verbalization setup, but *without* activations. (Results using `Ministral` are in Appendix E.5.) We then use the model finetuned on LatentQA to answer feature extraction prompts given reconstructed inputs (so $x_{\text{rec}} + x_{\text{prompt}}$). We also compare to an additional Llama3 model not trained on LatentQA as a zero-shot baseline. We present full experimental results in Tables 2 and 3.

Interpretation results. In both the single activation (token-level, Table 2) and multiple activation (layer-based, Table 3) settings, inversion is usually able achieve more than half the performance of verbalization, and on half the tasks, we see the same performance as in the canonical verbalization setup. With layer-based inversion, it is possible to reconstruct and predict accurately enough to match the verbalization accuracy. We also compare both inversion approaches to the verbalization of a single token or layer ($\ell = 15$), patched into the inference pass of a single target layer ($\ell = 0$) of the verbalizer; since we only invert a single token or layer, the comparison is fairer. Under these conditions, inversion always outperforms activation verbalization.

Note that the evaluation conditions place our inversion-based approach at a disadvantage relative to verbalization. Because our zero-shot descriptions are generated by instruction-tuned models, their stereotypical verbosity may fail to answer the prompt within 20 tokens. By contrast, verbalizers are able to bypass the verbosity (directly or via training), giving them an edge over zero-shot interpretations. If our interpretation models were tuned for brevity or given more output tokens, they might perform even better on benchmarks relative to the verbalizers. Furthermore, when comparing inversion and verbalization, we note that inversion is not lossless, as shown across language (Morris

⁶<https://huggingface.co/datasets/Tevatron/msmarco-passage-corpus>

⁷If 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** token activation. We use T5-Base and Llama3 as our inversion models, and compare both to Patchscopes (averaged across source layers $\ell = 1 - 15$). “Zero-shot” denotes a pre-trained instruction-tuned model, same type as \mathcal{M}_2 , given only reconstructed input. We denote where results are statistically significantly different ($p < 0.05$) compared to the Patchscopes baseline—per McNemar’s test with Bonferroni correction across the same baseline—with an asterisk (*). Inversion then interpretation matches performance with Patchscopes for half the tasks, while the other half can be partially explained by the input text.

	\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 ($\ell = 15$)	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 (Zero-shot)	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 (Zero-shot)	0.44*	0.25	0.21*	0.49*	0.39*	0.05*

et al., 2023) and vision (Li et al., 2025) models, so it is not expected to match zero-shot performance.

Key Finding 2

Prediction on top of reconstructed text (x_{rec}) results in performance that mostly matches that of verbalization, suggesting that some tasks used to interpret LLM behavior may elicit more information about the input text rather than the wanted privileged knowledge.

These findings show that the verbalizer’s responses to many datasets could reflect the decoded x_{input} , despite its imperfect reconstruction. In the case of LIT, performance can be matched *solely* from the encoded input text, whereas for Patchscopes, the performance can be mostly mimicked.

5 ARE GENERATED DESCRIPTIONS FAITHFUL TO THE TARGET MODEL?

We have shown in Section 2 and 4 that verbalization may be communicating mostly the input text, at least as evaluated on the feature extraction task.⁸ In this section, we show that verbalizers may not be answering prompts correctly unless it can answer them from the input text alone.

Our experiments test verbalizers using prompts that require world knowledge; ideally \mathcal{M}_2 would tell us about the world knowledge of the *target* model \mathcal{M}_1 . But \mathcal{M}_2 is itself an LLM with world knowledge of its own, so it can answer prompts without access to \mathcal{M}_1 internals. Therefore, verbalizers may respond to a prompt by drawing on \mathcal{M}_1 ’s knowledge, by drawing on \mathcal{M}_2 ’s knowledge, or by some combination of both. To disentangle these possibilities, we consider multiple setups where \mathcal{M}_1 is finetuned on a novel dataset, imbuing it with knowledge unknown to \mathcal{M}_2 .

5.1 PERSONAQA

We introduce PERSONAQA, a dataset containing attributes and texts of fake individuals. Because these individuals do not exist, it is unlikely that a model would encode their (fabricated) biographical information unless explicitly trained on this data.⁹ The dataset provides 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 should be unknown to \mathcal{M}_2 by construction, a verbalizer can

⁸If 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.

⁹Allen-Zhu & Li (2024) and Zucchet et al. (2025) use similar synthetic persona datasets.

Table 3: *Inversion then interpretation* on **multiple** activations. “Zero-shot” denotes a pre-trained instruction-tuned model, the same type as \mathcal{M}_2 , given only reconstructed input. The results that are statistically significantly different ($p < 0.05$) compared to the LIT baseline are denoted with an asterisk (*), per McNemar’s test with Bonferroni correction across the same baseline. Inversion then interpretation yields performance comparable to LIT (averaged across $\ell = 1 - 15$) for most tasks.

	\mathcal{M}_2	count_curr	food_count	ath_pos	ath_sport	prod_comp	star_const
LIT	Llama3	0.79	0.45	0.66	0.84	0.67	0.41
LIT ($\ell = 15$)	Llama3	0.78	0.30	0.68	0.83	0.66	0.42
Inversion	Llama3	0.79	0.52	0.52*	0.39*	0.55*	0.46
	Llama3 (Zero-shot)	0.82	0.59*	0.58*	0.76*	0.68	0.42

only correctly answer prompts about the dataset by faithfully communicating privileged information stored in \mathcal{M}_1 ’s activations.

Datasets. We consider three experimental settings, all using variants of `PersonaQA`. The first dataset, `PersonaQA` ($\mathcal{M}_1^{\text{pqa}}$), is a dataset containing attributes that are sociodemographically correlated with the persona name. The second dataset, `PersonaQA-Shuffled` ($\mathcal{M}_1^{\text{pqa.shuffled}}$), shuffles the attributes in `PersonaQA` to remove the sociodemographic correlations. Finally, the third dataset, `PersonaQA-Fantasy` ($\mathcal{M}_1^{\text{pqa.fantasy}}$), contains fake personas and attributes. Details of the datasets and their curation are described in Appendix G.1. We curate these in different ways to evaluate when knowledge from \mathcal{M}_1 is verbalized by \mathcal{M}_2 . For all 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 in Appendix Table 16. Based on the attributes, we generate biographies and interviews to train \mathcal{M}_1 , with details found in Appendix G.2.

Experimental setting. For each dataset, we finetune a target model \mathcal{M}_1 (from a base `Llama-3.1-8B`¹⁰ or `Ministral`) on the biographies and interviews of the generated personas, so \mathcal{M}_1 learns factual information about them. In Appendix C and G.7, we provide more details about finetuning \mathcal{M}_1 ; and, in Appendix Table 21, we confirm that \mathcal{M}_1 internalizes `PersonaQA-Fantasy` specifically, while an untrained model is unable to predict the fabricated characteristics. For verbalization, we use the existing verbalizers (untrained for `Patchscopes`, and trained on `LatentQA` for LIT). We also report results using inversion on `PersonaQA` and variations in Appendix G.9. We generate a set of out of domain (with respect to training datasets) questions about the 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 results across `PersonaQA` datasets in Table 4 for `Llama3`, and include supplemental results for `Ministral` in Appendix Table 26. 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 25 for only `Llama3`, 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.

Zero-shot prompting and verbalization achieve good performance based purely on the associations from `PersonaQA`, despite having no prior knowledge of the personas. Table 4 shows that zero-shot, LIT, and `Patchscopes` achieves nonzero accuracy across the tasks. LIT likely fares comparatively well because it was finetuned to answer questions succinctly. Zero-shot prompting tends to yield lengthier outputs, which degrades performance as evaluated. Naively, one might inter-

¹⁰We use a base `Llama-3.1-8B` since we prefer to start from a base model, but `Ministral` has no associated public base model.

Table 4: Absolute accuracy across the six attribute extraction tasks from PersonaQA. Our evaluation for Patchscopes and LIT follows Section 2, and $\mathcal{M}_1 = \mathcal{M}_2 =$ a base Llama3. The results that are statistically significantly different ($p < 0.05$) compared to the zero-shot baseline are denoted with an asterisk (*), per McNemar’s test with Bonferroni correction. In the derivative datasets, both verbalization methods and zero-shot often fail, with the exception of a few Patchscopes results, namely: fav_sport and fav_game. We posit that the (real-world) space of labels for such categories is small enough that, statistically, it is likely that the model can randomly output the correct label among the L' outputs afforded to Patchscopes; we provide evidence for this in Appendix G.8.

	Method	country	fav_food	fav_drink	fav_music_gen	fav_sport	fav_game
PersonaQA	Zero-shot	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	Zero-shot	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	Zero-shot	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

pret the verbalization results as telling us about \mathcal{M}_1 , but the zero-shot results confirm that nontrivial performance is achievable based on crude statistical associations between personas and attributes.

In most cases, verbalizers (\mathcal{M}_2) rely too much on their own world knowledge to make predictions, even when it conflicts with the knowledge in \mathcal{M}_2 ’s activations. Because performance of verbalization methods on PersonaQA-Shuffled is low in Table 4, it is likely that \mathcal{M}_2 is generating responses based on its own knowledge, rather than information from $\mathcal{M}_1^{\text{pqa-shuffled}}$. Auxiliary results in Appendix Table 25 show that \mathcal{M}_2 , when conditioned on $\mathcal{M}_1^{\text{pqa-shuffled}}$ activations, performs better on the *original* (unshuffled) labels than with the shuffled labels. In other words: \mathcal{M}_2 does a better job of answering questions about its own internal knowledge than about what \mathcal{M}_1 knows.

Verbalization may fail when the knowledge from \mathcal{M}_1 and \mathcal{M}_2 is misaligned. Neither Patchscopes nor LIT exceed zero accuracy on $\mathcal{M}_1^{\text{pqa-fantasy}}$, suggesting that verbalizers may be limited to their own world knowledge.

5.3 TRAINING \mathcal{M}_2 ON \mathcal{M}_1 ’S WORLD KNOWLEDGE

Finally, we evaluate whether finetuning \mathcal{M}_2 on the same PersonaQA-Fantasy knowledge improves its verbalization accuracy. If \mathcal{M}_2 *must* possess the same world knowledge as \mathcal{M}_1 , this limits its utility as a general tool for faithfully describing activations induced by \mathcal{M}_1 .

Setup. In this section, we run experiments using Llama3. We finetune the verbalizer in two steps. First, we finetune a \mathcal{M}_2 on an extended 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.¹¹ We then continue to finetune using LIT on LatentQA (Pan et al., 2024) to learn to verbalize activations, following the setup in Section 2. We also consider a simple baseline: a linear probe (Belinkov & Glass, 2019; Belinkov, 2022). A probe tests whether the representations from \mathcal{M}_1 are extractable with minimal finetuning. Finally, we compare to Patchscopes approach, where \mathcal{M}_2 has been trained over the new persona knowledge already internalized by \mathcal{M}_1 . We use the same evaluation as above for the verbalizers.

Dataset. In our extended PersonaQA-Fantasy dataset, we include more personas (200) but fewer labels for each attribute (≤ 10), compared to Section 5. We do this to generate enough data for

¹¹We confirm that the model internalizes this knowledge in Appendix Table 22.

432 probes to properly learn the persona representations. This yields a train/test split of 160/40 unique
 433 personas, providing approximately 2600 and 600 samples for finetuning and testing, respectively.
 434

435 **Results.** We report results in Table 5. After training \mathcal{M}_2 on the same data as \mathcal{M}_1 , \mathcal{M}_2 is able
 436 to verbalize the personas. This implies that \mathcal{M}_2 accurately decoding activations from \mathcal{M}_1 may
 437 have more to do with the former model having the same parametric knowledge as the latter than
 438 its ability to faithfully decode \mathcal{M}_1 's activations. As a result, a naive application of verbalization
 439 approaches may lead to an illusion of interpretability. Finally, we note that a linear probe achieves
 440 performance comparable to that of LIT; and here we guarantee that the knowledge does not come
 441 from the probe itself.

442 Key Finding 3

443
 444 Verbalizers appear to have to be trained on the same data distribution as the model they
 445 are (ostensibly) inspecting. This is consistent with the hypothesis that they often verbalize
 446 their own parametric knowledge rather than the information encoded in the target LLM's
 447 activations, which is what are ostensibly being described.
 448

449
 450 Table 5: Using absolute accuracy (whether the target exists in the output), we train \mathcal{M}_2 to internalize
 451 the same data distribution that \mathcal{M}_1 is trained on. We also include a logistic probe as a simple com-
 452 parison. All methods are tested on a held out set of personas. The underlying model used in these
 453 results is Llama3. The logistic probe is trained on a 80%/20% train/valid split of activations that
 454 are sourced from a \mathcal{M}_1 model that has information about the personas. We find that for verbalization
 455 to work, \mathcal{M}_2 must be trained on the same data distribution as \mathcal{M}_1 .
 456

	country	fav_food	fav_drink	fav_music_gen	fav_sport	fav_game
Patchscopes	0.18	0.35	0.33	0.47	0.34	0.43
LIT	0.20	0.25	0.33	0.23	0.15	0.28
Logistic Probe	0.18	0.38	0.30	0.20	0.25	0.20

462 6 RELATED WORK

463
 464
 465 *Mechanistic interpretability* (Saphra & Wiegrefe, 2024) seeks to characterize the inner workings of
 466 LLMs.¹² This work is most relevant to prior efforts which have sought to map internal activations to
 467 tokens (nostalgebraist, 2020; Belrose et al., 2023; Yom Din et al., 2024; Pal et al., 2023; Gottesman
 468 et al., 2025), and in particular to techniques which map activations to *natural language descriptions*
 469 (Bills et al., 2023; Mousi et al., 2023; Ghandeharioun et al., 2024; Chen et al., 2024; Pan et al.,
 470 2024), rather than probability distributions of vocabulary (nostalgebraist, 2020).

471 Another line of work related to our contributions here asks whether models can truly introspect
 472 privileged information, with mixed findings (Kadavath et al., 2022; Song et al., 2025a; Binder et al.,
 473 2025; Song et al., 2025b). The present effort is novel as our focus is on verbalization methods that
 474 access activations to investigate whether these activations offer such privileged information.

475 Finally, recent efforts have investigated the *faithfulness* (Jacovi & Goldberg, 2020) of natural lan-
 476 guage explanations with respect to model behaviors (Atanasova et al., 2023; Hase et al., 2020),
 477 specifically for explanations like Chain-of-Thought (CoT) reasoning (Turpin et al., 2023; Lanham
 478 et al., 2023; Lyu et al., 2023; Arcuschin et al., 2025) and individual neurons (Bills et al., 2023; Singh
 479 et al., 2023; Huang et al., 2023; Antverg & Belinkov, 2022). These works have shown that such ex-
 480 planations may provide unreliable descriptions of LLM behavior (Antverg & Belinkov, 2022). Our
 481 work offers an analogous analysis of explanations via verbalization techniques.
 482
 483
 484

485 ¹²We do not attempt a comprehensive review of mechanistic interpretability here, and instead point the
 reader to Rai et al. (2024); Mueller et al. (2024); Sharkey et al. (2025) and Bereska & Gavves (2024).

7 LIMITATIONS

We only stress test verbalization on QA-style tasks, tasks which were used in prior work (Pan et al., 2024; Ghandeharioun et al., 2024). However, experimenting on other tasks could improve our understanding for what tasks verbalization can access privileged information. For our inversion experiments, we choose $\ell = 15$ as the layer to invert activations from, following Pan et al. (2024); other layers could be used (Ghandeharioun et al., 2024), but due to the budgetary constraints of training each inversion model on GH200s (Appendix D), we leave such experimentation for future work. We also focus our experiments on 8B-sized models due to compute constraints (Appendix D), matching the sizes of prior work (Pan et al., 2024; Ghandeharioun et al., 2024). Furthermore, our experiments in Section 5 are limited by the amount of data generated for the task to train \mathcal{M}_1 . While more data could improve the efficacy of extracting information from \mathcal{M}_1 , we demonstrate that the relevant information exists in \mathcal{M}_1 , (Appendices G.5, G.6).

8 CONCLUSIONS

There has been nascent interest in *verbalization*, i.e., generating interpretable natural language descriptions of the internal activations of a target model. Our findings highlight some open questions about these techniques and the dataset evaluations that are most appropriate for these techniques. For example, using the feature extraction task, we showed that access to internals of the target model for verbalization is unnecessary to achieve comparable 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 by the target model.

Furthermore, in controlled experiments, we found that generated descriptions may instead often reflect the world knowledge of the LLM used for verbalization more than the internal knowledge of the target LLM. Controlled evaluations reveal that verbalizers may fail to access privileged knowledge when there exist knowledge that conflicts between the verbalizer and target model. Future work might investigate just how to extract information from verbalizers when the knowledge between the target LLM and verbalizer LLM conflict.

In sum, our findings show that the chosen evaluative task has strong implications on whether privileged information access is possible. Without an appropriate evaluation, it is difficult to measure whether certain information is properly extracted from target LLM activations, or whether this information comes purely from the world knowledge of the verbalizer LLM. These results provide appropriate insights for the tasks verbalization might not be of possible use for.

REFERENCES

- 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>.
- William P. Alston. Varieties of privileged access. *American Philosophical Quarterly*, 8(3):223–41, 1971.
- 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>.
- Iván Arcuschin, Jett Janiak, Robert Krzyzanowski, Senthoran 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>.
- 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 (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 283–294, Toronto,

- 540 Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-short.
541 25. URL <https://aclanthology.org/2023.acl-short.25/>.
- 542
- 543 Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Ma-
544 jumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica,
545 Saurabh Tiwary, and Tong Wang. Ms marco: A human generated machine reading comprehension
546 dataset, 2018. URL <https://arxiv.org/abs/1611.09268>.
- 547 Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computa-*
548 *tional Linguistics*, 48(1):207–219, March 2022. doi: 10.1162/coli.a.00422. URL <https://aclanthology.org/2022.cl-1.7/>.
- 549
- 550 Yonatan Belinkov and James Glass. Analysis methods in neural language processing: A survey.
551 *Transactions of the Association for Computational Linguistics*, 7:49–72, 2019. doi: 10.1162/
552 tacl.a.00254. URL <https://aclanthology.org/Q19-1004/>.
- 553
- 554 Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella
555 Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned
556 lens, 2023. URL <https://arxiv.org/abs/2303.08112>.
- 557 Leonard Bereska and Efstratios Gavves. Mechanistic interpretability for ai safety—a review. *arXiv*
558 *preprint arXiv:2404.14082*, 2024.
- 559
- 560 Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya
561 Sutskever, Jan Leike, Jeff Wu, and William Saunders. Language models can explain
562 neurons in language models. [https://openaipublic.blob.core.windows.net/
563 neuron-explainer/paper/index.html](https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html), 2023.
- 564 Felix Jedidja Binder, James Chua, Tomek Korbak, Henry Sleight, John Hughes, Robert Long, Ethan
565 Perez, Miles Turpin, and Owain Evans. Looking inward: Language models can learn about them-
566 selves by introspection. In *The Thirteenth International Conference on Learning Representations*,
567 2025. URL <https://openreview.net/forum?id=eb5pkwIB5i>.
- 568
- 569 Alberto Mario Ceballos-Arroyo, Monica Munnangi, Jiuding Sun, Karen Zhang, Jered McInerney,
570 Byron C. Wallace, and Silvio Amir. Open (clinical) LLMs are sensitive to instruction phrasings.
571 In Dina Demner-Fushman, Sophia Ananiadou, Makoto Miwa, Kirk Roberts, and Junichi Tsujii
572 (eds.), *Proceedings of the 23rd Workshop on Biomedical Natural Language Processing*, pp. 50–
573 71, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/
574 v1/2024.bionlp-1.5. URL <https://aclanthology.org/2024.bionlp-1.5/>.
- 575 Haozhe Chen, Carl Vondrick, and Chengzhi Mao. Selfie: Self-interpretation of large language
576 model embeddings. In *ICML*, 2024. URL [https://openreview.net/forum?id=
577 gjgRKbdYR7](https://openreview.net/forum?id=gjgRKbdYR7).
- 578 Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremen-
579 tal gradient method with support for non-strongly convex composite objectives. In
580 Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger (eds.), *Ad-*
581 *vances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.,
582 2014. URL [https://proceedings.neurips.cc/paper_files/paper/2014/
583 file/937964195d6fb3a55cd7cc578165f058-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2014/file/937964195d6fb3a55cd7cc578165f058-Paper.pdf).
- 584
- 585 Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha
586 Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony
587 Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark,
588 Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière,
589 Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi,
590 Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne
591 Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz,
592 Dany Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego
593 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

- 594 Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Is-
595 abel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana
596 Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny
597 Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu,
598 Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Jun-
599 teng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield,
600 Kevin Stone, and et al. The llama 3 herd of models. *CoRR*, abs/2407.21783, 2024. URL
601 <https://doi.org/10.48550/arXiv.2407.21783>.
- 602 Jaden Fried Fiotto-Kaufman, Alexander Russell Loftus, Eric Todd, Jannik Brinkmann, Koyena Pal,
603 Dmitrii Troitskii, Michael Ripa, Adam Belfki, Can Rager, Caden Juang, Aaron Mueller, Samuel
604 Marks, Arnab Sen Sharma, Francesca Lucchetti, Nikhil Prakash, Carla E. Brodley, Arjun Guha,
605 Jonathan Bell, Byron C Wallace, and David Bau. NNsight and NDIF: Democratizing access to
606 open-weight foundation model internals. In *The Thirteenth International Conference on Learning
607 Representations*, 2025. URL <https://openreview.net/forum?id=MxbEiFRf39>.
- 608 Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. Patchscopes:
609 A unifying framework for inspecting hidden representations of language models. In *Forty-first
610 International Conference on Machine Learning*, 2024. URL [https://arxiv.org/abs/
611 2401.06102](https://arxiv.org/abs/2401.06102).
- 612 Daniela Gottesman, Mor Geva, and Dana Ramati. Eliciting textual descriptions from represen-
613 tations of continuous prompts. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and
614 Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics:
615 ACL 2025*, pp. 16545–16562, Vienna, Austria, July 2025. Association for Computational Lin-
616 guistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.849. URL <https://aclanthology.org/2025.findings-acl.849/>.
- 617 Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in
618 vqa matter: Elevating the role of image understanding in visual question answering. In *2017
619 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6325–6334, 2017.
620 doi: 10.1109/CVPR.2017.670.
- 621 Peter Hase, Shiyue Zhang, Harry Xie, and Mohit Bansal. Leakage-adjusted simulatability: Can
622 models generate non-trivial explanations of their behavior in natural language? In Trevor
623 Cohn, Yulan He, and Yang Liu (eds.), *Findings of the Association for Computational Linguis-
624 tics: EMNLP 2020*, pp. 4351–4367, Online, November 2020. Association for Computational
625 Linguistics. doi: 10.18653/v1/2020.findings-emnlp.390. URL [https://aclanthology.
626 org/2020.findings-emnlp.390/](https://aclanthology.org/2020.findings-emnlp.390/).
- 627 Evan Hernandez, Arnab Sen Sharma, Tal Haklay, Kevin Meng, Martin Wattenberg, Jacob Andreas,
628 Yonatan Belinkov, and David Bau. Linearity of relation decoding in transformer language models.
629 In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=w7LU2s14kE>.
- 630 John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In Kentaro
631 Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on
632 Empirical Methods in Natural Language Processing and the 9th International Joint Confer-
633 ence on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2733–2743, Hong Kong, China,
634 November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1275. URL
635 <https://aclanthology.org/D19-1275/>.
- 636 Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang,
637 and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Con-
638 ference on Learning Representations*, 2022. URL [https://openreview.net/forum?
639 id=nZeVKeeFYf9](https://openreview.net/forum?id=nZeVKeeFYf9).
- 640 Jing Huang, Atticus Geiger, Karel D’Oosterlinck, Zhengxuan Wu, and Christopher Potts. Rigor-
641 ously assessing natural language explanations of neurons. In Yonatan Belinkov, Sophie Hao,
642 Jaap Jumelet, Najoung Kim, Arya McCarthy, and Hosein Mohebbi (eds.), *Proceedings of the 6th
643 BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 317–331,
644

- 648 Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.
649 blackboxnlp-1.24. URL <https://aclanthology.org/2023.blackboxnlp-1.24/>.
- 650
- 651 Alon Jacovi and Yoav Goldberg. Towards faithfully interpretable NLP systems: How should we
652 define and evaluate faithfulness? In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault
653 (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*,
654 pp. 4198–4205, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/
655 2020.acl-main.386. URL <https://aclanthology.org/2020.acl-main.386/>.
- 656 Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez,
657 Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston,
658 Sheer El Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam
659 Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion,
660 Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei,
661 Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared
662 Kaplan. Language models (mostly) know what they know. *CoRR*, abs/2207.05221, 2022. URL
663 <https://doi.org/10.48550/arXiv.2207.05221>.
- 664 Divyansh Kaushik and Zachary C. Lipton. How much reading does reading comprehension re-
665 quire? a critical investigation of popular benchmarks. In Ellen Riloff, David Chiang, Ju-
666 lia Hockenmaier, and Jun’ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empir-
667 ical Methods in Natural Language Processing*, pp. 5010–5015, Brussels, Belgium, October-
668 November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1546. URL
669 <https://aclanthology.org/D18-1546/>.
- 670 Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua
671 Bengio and Yann LeCun (eds.), *ICLR (Poster)*, 2015. URL [http://dblp.uni-trier.de/
672 db/conf/iclr/iclr2015.html#KingmaB14](http://dblp.uni-trier.de/db/conf/iclr/iclr2015.html#KingmaB14).
- 673
- 674 Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Her-
675 nandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilė Lukošiuūtė, Karina
676 Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson,
677 Sam McCandlish, Sandipan Kundu, Saurav Kadavath, Shannon Yang, Thomas Henighan, Tim-
678 othy Maxwell, Timothy Telleen-Lawton, Tristan Hume, Zac Hatfield-Dodds, Jared Kaplan, Jan
679 Brauner, Samuel R. Bowman, and Ethan Perez. Measuring faithfulness in chain-of-thought re-
680asoning, 2023. URL <https://arxiv.org/abs/2307.13702>.
- 681 Wenyan Li, Raphael Tang, Chengzu Li, Caiqi Zhang, Ivan Vulić, and Anders Søgaard. Lost in
682 embeddings: Information loss in vision-language models, 2025. URL [https://arxiv.org/
683 abs/2509.11986](https://arxiv.org/abs/2509.11986).
- 684 Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization
685 Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguis-
686 tics. URL <https://aclanthology.org/W04-1013/>.
- 687
- 688 Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki,
689 and Chris Callison-Burch. Faithful chain-of-thought reasoning. In Jong C. Park, Yuki Arase, Bao-
690 tian Hu, Wei Lu, Derry Wijaya, Ayu Purwarianti, and Adila Alfa Krisnadh (eds.), *Proceedings of
691 the 13th International Joint Conference on Natural Language Processing and the 3rd Conference
692 of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long
693 Papers)*, pp. 305–329, Nusa Dua, Bali, November 2023. Association for Computational Linguis-
694 tics. doi: 10.18653/v1/2023.ijcnlp-main.20. URL [https://aclanthology.org/2023.
695 ijcnlp-main.20/](https://aclanthology.org/2023.ijcnlp-main.20/).
- 696 R. Thomas McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic
697 heuristics in natural language inference. In Anna Korhonen, David Traum, and Lluís Màrquez
698 (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*,
699 pp. 3428–3448, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.
700 18653/v1/P19-1334. URL <https://aclanthology.org/P19-1334/>.
- 701 Kevin Meng, David Bau, Alex J Andonian, and Yonatan Belinkov. Locating and editing factual asso-
ciations in GPT. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.),

- 702 *Advances in Neural Information Processing Systems*, 2022. URL <https://openreview.net/forum?id=-h6WAS6eE4>.
- 703
- 704
- 705 Mistral. Un ministral, des ministraux. <https://mistral.ai/news/ministraux>, 2024.
- 706 Accessed: 2025-05-18.
- 707
- 708 John Morris, Volodymyr Kuleshov, Vitaly Shmatikov, and Alexander Rush. Text embeddings reveal
- 709 (almost) as much as text. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of*
- 710 *the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12448–12460,
- 711 Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.
- 712 emnlp-main.765. URL <https://aclanthology.org/2023.emnlp-main.765/>.
- 713
- 714 Basel Mousi, Nadir Durrani, and Fahim Dalvi. Can LLMs facilitate interpretation of pre-trained lan-
- 715 guage models? In *The 2023 Conference on Empirical Methods in Natural Language Processing*,
- 716 2023. URL <https://openreview.net/forum?id=nIuJXuSdhn>.
- 717
- 718 David Mueller, Mark Dredze, and Nicholas Andrews. Multi-task transfer matters during instruction-
- 719 tuning. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Associa-*
- 720 *tion for Computational Linguistics: ACL 2024*, pp. 14880–14891, Bangkok, Thailand, August
- 721 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.883. URL
- 722 <https://aclanthology.org/2024.findings-acl.883/>.
- 723
- 724 nostalgebraist. interpreting gpt: the logit lens, 2020. URL [https://www.lesswrong.com/](https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens)
- 725 [posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens](https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens).
- 726
- 727 Koyena Pal, Jiuding Sun, Andrew Yuan, Byron C Wallace, and David Bau. Future lens: Antici-
- 728 pating subsequent tokens from a single hidden state. In *Proceedings of the 27th Conference on*
- 729 *Computational Natural Language Learning (CoNLL)*, pp. 548–560, 2023.
- 730
- 731 Alexander Pan, Lijie Chen, and Jacob Steinhardt. Latentqa: Teaching llms to decode activations
- 732 into natural language, 2024. URL <https://arxiv.org/abs/2412.08686>.
- 733
- 734 Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic
- 735 evaluation of machine translation. In Pierre Isabelle, Eugene Charniak, and Dekang Lin (eds.),
- 736 *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp.
- 737 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.
- 738 doi: 10.3115/1073083.1073135. URL <https://aclanthology.org/P02-1040/>.
- 739
- 740 Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier
- 741 Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas,
- 742 Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duch-
- 743 esnay. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, 12(null):2825–2830,
- 744 November 2011. ISSN 1532-4435.
- 745
- 746 Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi
- 747 Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text
- 748 transformer. *J. Mach. Learn. Res.*, 21(1), January 2020. ISSN 1532-4435.
- 749
- 750 Daking Rai, Yilun Zhou, Shi Feng, Abulhair Saparov, and Ziyu Yao. A practical review of mecha-
- 751 nistic interpretability for transformer-based language models. *arXiv preprint arXiv:2407.02646*,
- 752 2024.
- 753
- 754 Naomi Saphra and Sarah Wiegreffe. Mechanistic? *arXiv preprint arXiv:2410.09087*, 2024.
- 755
- 756 Lee Sharkey, Bilal Chughtai, Joshua Batson, Jack Lindsey, Jeff Wu, Lucius Bushnaq, Nicholas
- 757 Goldowsky-Dill, Stefan Heimersheim, Alejandro Ortega, Joseph Bloom, Stella Biderman, Adria
- 758 Garriga-Alonso, Arthur Conmy, Neel Nanda, Jessica Rumbelow, Martin Wattenberg, Nandi
- 759 Schoots, Joseph Miller, Eric J. Michaud, Stephen Casper, Max Tegmark, William Saunders,
- 760 David Bau, Eric Todd, Atticus Geiger, Mor Geva, Jesse Hoogland, Daniel Murfet, and Tom Mc-
- 761 Grath. Open problems in mechanistic interpretability, 2025. URL <https://arxiv.org/abs/2501.16496>.

- 756 Chandan Singh, Aliyah R. Hsu, Richard J. Antonello, Shailee Jain, Alexander G. Huth, Bin Yu,
757 and Jianfeng Gao. Explaining black box text modules in natural language with language models.
758 *CoRR*, abs/2305.09863, 2023. doi: 10.48550/ARXIV.2305.09863. URL [https://doi.org/
759 10.48550/arXiv.2305.09863](https://doi.org/10.48550/arXiv.2305.09863).
- 760 Siyuan Song, Jennifer Hu, and Kyle Mahowald. Language models fail to introspect about their
761 knowledge of language, 2025a. URL <https://arxiv.org/abs/2503.07513>.
- 762 Siyuan Song, Harvey Lederman, Jennifer Hu, and Kyle Mahowald. Privileged self-access matters
763 for introspection in ai, 2025b. URL <https://arxiv.org/abs/2508.14802>.
- 764 Jiuding Sun, Chantal Shaib, and Byron C Wallace. Evaluating the zero-shot robustness of
765 instruction-tuned language models. In *The Twelfth International Conference on Learning Repre-
766 sentations*, 2024. URL <https://openreview.net/forum?id=g9diuvxN6D>.
- 767 Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. Language models don’t always
768 say what they think: Unfaithful explanations in chain-of-thought prompting. In *Thirty-seventh
769 Conference on Neural Information Processing Systems*, 2023. URL [https://openreview.
770 net/forum?id=bzs4uPLXvi](https://openreview.net/forum?id=bzs4uPLXvi).
- 771 Albert Webson and Ellie Pavlick. Do prompt-based models really understand the meaning of their
772 prompts? In *Proceedings of the 2022 conference of the north american chapter of the association
773 for computational linguistics: Human language technologies*, pp. 2300–2344, 2022.
- 774 Alexander Yom Din, Taelin Karidi, Leshem Choshen, and Mor Geva. Jump to conclusions: Short-
775 cutting transformers with linear transformations. In *Proceedings of the 2024 Joint Interna-
776 tional Conference on Computational Linguistics, Language Resources and Evaluation (LREC-
777 COLING 2024)*, pp. 9615–9625, Torino, Italy, May 2024. ELRA and ICCL. URL [https:
778 //aclanthology.org/2024.main-1.840](https://aclanthology.org/2024.main-1.840).
- 779 Yukun Zhao, Lingyong Yan, Weiwei Sun, Guoliang Xing, Shuaiqiang Wang, Chong Meng, Zhicong
780 Cheng, Zhaochun Ren, and Dawei Yin. Improving the robustness of large language models via
781 consistency alignment. In Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci,
782 Sakriani Sakti, and Nianwen Xue (eds.), *Proceedings of the 2024 Joint International Conference
783 on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp.
784 8931–8941, Torino, Italia, May 2024. ELRA and ICCL. URL [https://aclanthology.
785 org/2024.lrec-main.782/](https://aclanthology.org/2024.lrec-main.782/).
- 786 Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of
787 the Royal Statistical Society. Series B (Statistical Methodology)*, 67(2):301–320, 2005. ISSN
788 13697412, 14679868. URL <http://www.jstor.org/stable/3647580>.
- 789 Nicolas Zucchet, Jörg Bornschein, Stephanie Chan, Andrew Lampinen, Razvan Pascanu, and Soham
790 De. How do language models learn facts? dynamics, curricula and hallucinations, 2025. URL
791 <https://arxiv.org/abs/2503.21676>.
- 792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809

A FEATURE EXTRACTION DATASET INFORMATION

Information about the feature extraction dataset (Hernandez et al., 2024) can be found in Appendix Table 6. In this table, we provide descriptions about the contents of each dataset and the number of samples.

Table 6: The information for each dataset used in feature extraction.

Name	Short Name	Description	Num. Examples
country_currency	count_curr	Predict the currency based on the country.	128
food_country	food_count	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. For models, we use `meta-llama/Llama-3.1-8B-Instruct`¹³ following prior work, and `mistralai/Minstral-8B-Instruct-2410`¹⁴ for results on an additional model. We use both of the models when examining LIT and Patchscopes.

To implement Patchscopes (Ghandeharioun et al., 2024), we use the original data generation code from the GitHub repository¹⁵ for the feature extraction experiment. For the method reimplementation, we use `nnsight` (Fiotto-Kaufman et al., 2025) to gather a single activation and patch the new activation into the inference pass of the verbalizer LLM. All other hyperparameters are consistent with the original Patchscopes work.

To implement LIT, we reuse existing code¹⁶ from Pan et al. (2024) and use default hyperparameters, changing code minimally to get the code to run. We select $\ell = 15$ for verbalization, which is consistent with the results on Llama3. For our 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 (Pan et al., 2024), we train a verbalizer to decode activations from $\ell = 15$ and we calculate performance for the first 15 layers of Llama3 individually, with the exception of the first layer (Llama3 has 32 layers, so layers 1 - 15). We then 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 (Ghandeharioun et al., 2024; Pan et al., 2024). To compare against LIT, we only consider the first 15 layers for Patchscopes. In Ghandeharioun et al. (2024), they consider all combinations of source $\ell \in [1, \dots, L] \times$ target $\ell^* \in [1, \dots, L^*]$, 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

¹³<https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct>

¹⁴<https://huggingface.co/mistralai/Minstral-8B-Instruct-2410>

¹⁵<https://github.com/PAIR-code/interpretability/tree/master/patchscopes/>

¹⁶<https://github.com/aypan17/latentqa>

864 considered correct. Then, the accuracy across each individual source layer is averaged to obtain the
865 final answer.

866 This `Patchscopes` approach is exhaustive, and for feature extraction across only six datasets,
867 requires $16 \cdot 32$ forward passes per task, or 3072 such passes. Not including `PersonaQA` and
868 related datasets (and baselines), this would total 10,000+ forward passes. As noted in the main
869 paper, we originally tested with layers 1 - 32 for `Llama3` and found that performance was worse
870 when averaging over all source layers; so to save compute, we evaluate over half of the layers, which
871 is again consistent with prior work.

872 Across all verbalization methods, we use the same source prompt, which is the input text for the
873 feature extraction prompt. Ghandeharioun et al. (2024) samples additional subjects for their input
874 context to get their results; we do not, as we only use the raw feature extraction prompt, so our
875 performance differs slightly from the results in the paper.
876

877 C TRAINING INFORMATION

878 We include a comprehensive table detailing information about the training approaches for each sec-
879 tion, in Table 7.

880
881 Table 7: Models and datasets used for each section of the paper. **Evaluated** means the model was
882 trained in a previous section and is used for the current noted section. **Previous** refers to datasets that
883 were previously used to train the model in a prior section, and the model is now used in this section.
884 All LMs (everything but the probe) were trained using cross entropy loss, and specific training
885 details for each model are listed with their appendix location. For datasets, the marked datasets refer
886 to datasets that were used in the sections, whether that be for training or for evaluation.
887

	Sec. 2 <i>Zero-shot</i>	Sec. 4 <i>Inversion</i>	Sec. 5.1, 5.2 <i>PersonaQA</i>	Sec. 5.3 <i>Probing</i>	
Models	\mathcal{M}_1	–	–	–	
	\mathcal{M}_2 (LIT, Appendix Section B)	✓	Evaluated	Evaluated	
	\mathcal{M}_{rec} (Appendix Section E)	–	✓	–	
	$\mathcal{M}_1^{\text{pqa}}$ (Appendix G.7.1)	–	–	✓	
	$\mathcal{M}_1^{\text{pqa_shuffled}}$ (Appendix G.7.1)	–	–	✓	
	$\mathcal{M}_1^{\text{pqa_fantasy}}$ (Appendix G.7.1)	–	–	✓	
	$\mathcal{M}_1^{\text{pqa_fantasy}}$ (extended) (Appendix G.7.1)	–	–	–	✓
	$\mathcal{M}_2^{\text{pqa_fantasy}}$ (extended) (Appendix G.7.1)	–	–	–	✓
Logistic Probe (Appendix G.7.2)	–	–	–	✓	
Datasets	LatentQA (Pan et al., 2024)	✓	✓	Previous	
	Feature Extraction (Hernandez et al., 2024)	✓	✓	–	
	MS MARCO (Bajaj et al., 2018)	–	✓	–	
	PersonaQA	–	–	✓	
	PersonaQA-Shuffled	–	–	✓	
	PersonaQA-Fantasy	–	–	✓	
PersonaQA-Fantasy (extended)	–	–	–	✓	

908 D GPU HOURS USED

909 We estimate the amount of GPU hours used for each experiment (Sections 2 to 5), based on a lower
910 bound estimate that assumes each experiment runs on the first try. We use a combination of A100s
911 and GH200s but primarily try to represent the GPU hours in terms of combined units. The table can
912 be seen in Appendix Table 8.
913
914

Table 8: GPU Hours by Section and Task (A100s, GH200s). For each section, we detail the GPU hours used in a combined total. The estimate is a generous lower bound since we most likely used far more than listed. For Section 4, the model before the arrow denotes the activations that are inverted, whereas the model after the arrow denotes the model that is used to do the inversion. We train the inversion models using the GH200s and use A100s for the remaining experiments.

Section	Task	Model	GPU Hours
Section 3, A100s	Patchscopes Evaluation	Llama3	102.4
		Ministral	102.4
	LIT Training	Llama3	72.0
		Ministral	72.0
	LIT Evaluation	Llama3	9.6
		Ministral	9.6
Section 3 Subtotal			368.0
Section 4, A100s and GH200s	$(\mathcal{M}_{\text{rec}})$ Inversion Training (Multiple)	Llama3 \rightarrow Llama3	96.0
		Ministral \rightarrow Ministral	96.0
	$(\mathcal{M}_{\text{rec}})$ Inversion Training (Single)	Llama3 \rightarrow Llama3	192.0
		Ministral \rightarrow Ministral	192.0
		Llama3 \rightarrow T5	192.0
		Ministral \rightarrow T5	192.0
	$(\mathcal{M}_{\text{rec}})$ Multiple Inversion Evaluation	Llama3 \rightarrow Llama3	0.6
		Ministral \rightarrow Ministral	0.6
		Llama3 \rightarrow Llama3	0.6
		Ministral \rightarrow Ministral	0.6
		Llama3 \rightarrow T5	0.6
	Ministral \rightarrow T5	0.6	
Section 4 Subtotal			963.6
Section 5, A100s	$\mathcal{M}_1^{\text{pqa}}$ Training	Llama3	96.0
		Ministral	96.0
	$\mathcal{M}_1^{\text{pqa_shuffled}}$ Training	Llama3	96.0
		Ministral	96.0
	$\mathcal{M}_1^{\text{pqa_fantasy}}$ Training	Llama3	96.0
		Ministral	96.0
	$\mathcal{M}_1^{\text{pqa}}$ + Patchscopes Eval	Llama3	102.4
		Ministral	102.4
	$\mathcal{M}_1^{\text{pqa_shuffled}}$ + Patchscopes Eval	Llama3	102.4
		Ministral	102.4
	$\mathcal{M}_1^{\text{pqa_fantasy}}$ + Patchscopes Eval	Llama3	102.4
		Ministral	102.4
	$\mathcal{M}_1^{\text{pqa}}$ + LIT Eval	Llama3	9.6
		Ministral	9.6
	$\mathcal{M}_1^{\text{pqa_shuffled}}$	Llama3	9.6
		Ministral	9.6
$\mathcal{M}_1^{\text{pqa_fantasy}}$	Llama3	9.6	
	Ministral	9.6	
$\mathcal{M}_1^{\text{pqa_fantasy}}$ Extended Training	Llama3	12.0	
$\mathcal{M}_2^{\text{pqa_fantasy}}$ Extended Training	Llama3	8.0	
LIT Training (over $\mathcal{M}_2^{\text{pqa_fantasy}}$)	Llama3	72.0	
Probe Training	Logistic Probe	1.0	
Section 5 Subtotal			1,341.0
Miscellaneous	Data generation & preliminary experiments	Various	100.0
	Miscellaneous Subtotal		100.0
Grand Total			2,772.6

E INVERSION TRAINING DETAILS AND OUTPUTS

E.1 INVERSION DETAILS

Fidelity of input reconstructions. To invert an activation matrix h^ℓ , we train $\mathcal{M}_{\text{rec}} = \text{Llama3-8B-Instruct}$ (Llama3) or $\mathcal{M}_{\text{rec}} = \text{Ministral-8B-Instruct}$ (Ministral) by inserting activations into the input of \mathcal{M}_{rec} of Llama3 and using the original prompt as the label. For implementation, we use existing code from LIT (Pan et al., 2024) but use the input as the target so that the model learns to reconstruct this from activations. We choose $\ell = 15$ since Pan et al. (2024) has 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}_{rec} . We use the `vec2text`¹⁷ implementation to train $\mathcal{M}_{\text{rec}} = \text{T5-Base}$ for inversion, taking hyperparameters from Morris et al. (2023). For inversion with $\mathcal{M}_{\text{rec}} = \text{Llama3}$, we implement reconstruction by instead taking a loss over the activations passed into \mathcal{M}_{rec} —as opposed to the input text—to learn to invert. We provide more details on this implementation in Appendix E.3 and Appendix E.4.

E.2 INVERSION EVALUATION

For evaluating inversion, we adopt as our metric BLEU (Papineni et al., 2002), following prior work on inversion (Morris et al., 2023). We also note that the measured inversions are for out-of-domain instances compared to the data used to train the inverter \mathcal{M}_{rec} . Out-of-domain inversions have been shown to result in lower performance (Morris et al., 2023).

Quantitative results. We report inversion results in Appendix Table 9, using feature extraction as our evaluation task. Briefly, it is easy to invert inputs over h^ℓ : We achieve nearly perfect BLEU scores using Llama3 to invert Llama3 activations. Appendix Table 10 provides examples. When using Llama3 or Ministral to invert h_i^ℓ , we achieve much lower BLEU scores, doubling the score when reconstructing with T5-Base. T5-Base is more accurate, agreeing with findings from Morris et al. (2023). 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. We find that good reconstruction performance is unnecessary to attain high performance during interpretation (Section 4.1), as we still achieve comparable performance to verbalization on specific tasks, such as feature extraction.

Table 9: We invert activations from Llama3 and Ministral using Llama3 and Ministral, respectively, for both types of inversions (single and multiple activations). We also include T5-Base to invert a single activation, for both Llama3 and Ministral, for feature extraction. Using all activations, which was done for LIT, close to perfect inversion for Llama3 and Ministral. On the other hand, using a single activation, like in Patchescopes, leads to partial reconstruction. Although the BLEU score is low, qualitative outputs, which are more interpretable, are located in Appendix Table 10, which show that the inversions are structurally accurate.

Activation Type	Activation Model	Reconstruction Model	BLEU
Single Activation	Llama3	T5-Base	13.34
		Llama3	6.47
	Ministral	T5-Base	4.38
		Ministral	3.49
All Activations	Llama3	Llama3	95.46
	Ministral	Ministral	95.88

Qualitative results. In Appendix Table 10 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 9); this was similarly noted in Morris et al.

¹⁷<https://github.com/vec2text/vec2text>

(2023), and may owe to the encoder-decoder architecture. The qualitative outputs of reconstructions in Appendix Table 10 show that reconstructed inputs are semantically similar to the targets.

Table 10: 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 Llama3.

	Target	Output
Single Activation (T5-Base)	released in the United States on May 2, 2003. after moving into Pizza Hut Park in 2005.	released in the United States on September 23, 2003. after moving to the Emirates Stadium in 2005.
Single Activation (Llama3)	is the sister of Percy Snow and Eric Snow, released in the United States on May 2, 2003. after moving into Pizza Hut Park in 2005. is the sister of Percy Snow and Eric Snow,) is the sister of Michael Swan and Joe Swan, Released: September 14, 2004, in the United States. The team moved to their new stadium at Toyota Center in 2007. Broolas, brother of George Boolas and William Boolas,
Multiple Activations (Llama3)	released in the United States on May 2, 2003. after moving into Pizza Hut Park in 2005. is the sister of Percy Snow and Eric Snow,	released in the United States on May 2, 2003. After moving into Pizza Hut Park in 2005. this is the sister of Percy Snow and Eric Snow,

E.3 INVERTING h^ℓ

To invert an activation matrix h^ℓ , we train $\mathcal{M}_{\text{rec}} = \text{Llama3}$ using the Adam optimizer (Kingma & Ba, 2015) with $\text{lr} = 2e^{-4}$ and an effective batch size of 128. We train with LoRA (Hu et al., 2022) 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). Specifically, the LoRA parameters are: rank $r = 16$, $\alpha = 32$. 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 do not train bias parameters and set the configuration to causal language modeling.

E.4 INVERTING h_i^ℓ

For inverting h_i^ℓ with Llama3, we manually insert a special token for the activation into the forward pass of \mathcal{M}_{rec} and replace that token with the activation. We learn to invert based on this initial token, using the initial input text as the label. For hyperparameters, we use: $\text{lr} = 1e^{-3}$, a batch size of 512, along with the Adam optimizer (Kingma & Ba, 2015), and choose $\ell = 15$, with the activation chosen being the last token of any prompt. We train with LoRA (Hu et al., 2022) over two epochs with four GH200s. Specifically, the LoRA parameters are: rank $r = 16$, $\alpha = 32$. 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 do not train bias parameters and set the configuration to causal language modeling.

For inverting h_i^ℓ with T5-Base, use the code from Morris et al. (2023) and modify it minimally to accommodate inverting activations. We 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.

E.5 MINISTRAL RESULTS

We include results on using a different model, `Ministral`, to invert the activations of the same type. In this setting, to invert multiple activations, we use `Ministral`. To invert a single activation, we use `Ministral` and `T5-Base`.

We find that the results for inversion over all activations (Appendix Table 11) and just a single activation for `Ministral` similarly holds as it does for `Llama3`. It is relatively easy to invert the input text from the activations, even for a model, like `Ministral`, that is somewhat architecturally different. Note that in deciding which layer to invert, we follow and consistently use $\ell = 15$, even though `Ministral` has 36 layers. So, although we choose $\ell = 15$, because task-specific information may be located in the middle-most layers (Pan et al., 2024), the most optimal performance across inversion and verbalization results may not be $\ell = 15$.

Table 11: *Inversion then interpretation* on **multiple** activations, which is the companion result to Table 3. “Zero-shot” denotes a pre-trained instruction-tuned model, same type as \mathcal{M}_2 , given only reconstructed input. The results that are statistically significantly different ($p < 0.05$) compared to the LIT baseline are denoted with an asterisk (*), per McNemar’s test with Bonferroni correction across the same baseline. Inversion then interpretation yields performance comparable to LIT (averaged across $\ell = 1 - 15$) for most tasks, even for `Ministral`.

	\mathcal{M}_2	count_curr	food_count	ath_pos	ath_sport	prod_comp	star_const
LIT	<code>Ministral</code>	0.77	0.48	0.59	0.78	0.67	0.39
LIT ($\ell = 15$)	<code>Ministral</code>	0.80	0.52	0.60	0.78	0.69	0.40
Inversion	<code>Ministral</code>	0.83	0.46	0.40*	0.77	0.54*	0.60*
	<code>Ministral</code> (Zero-shot)	0.86	0.50	0.55*	0.83*	0.69	0.67*

Table 12: *Inversion then interpretation* on a **single** token activation, which is the companion result to Table 2. We use `T5-Base` and `Llama3` as our inversion models, and compare both to `Patchscopes` (averaged across source layers $\ell = 1 - 15$). “Zero-shot” denotes a pre-trained instruction-tuned model, same type as \mathcal{M}_2 , given only reconstructed input. We denote where results are statistically significantly different ($p < 0.05$) compared to the `Patchscopes` baseline—per McNemar’s test with Bonferroni correction across the same baseline—with an asterisk (*). Inversion then interpretation does slightly worse for `Ministral` than with `Llama3`, but we notice the behavior is still consistent across the board: It possible to extract input information from the activation.

	\mathcal{M}_2	count_curr	food_count	ath_pos	ath_sport	prod_comp	star_const
<code>Patchscopes</code>	<code>Ministral</code>	0.14	0.01	0.22	0.61	0.47	0.15
<code>Patchscopes</code> ($\ell = 15$)	<code>Ministral</code>	0.11	0.00	0.05	0.17	0.17	0.01
Inversion, <code>Ministral</code>	<code>Ministral</code>	0.26*	0.11	0.13*	0.42*	0.28*	0.07
	<code>Ministral</code> (Zero-shot)	0.27*	0.11	0.16*	0.43*	0.30	0.07
Inversion, <code>T5-Base</code>	<code>Ministral</code>	0.31*	0.05	0.15*	0.44*	0.26*	0.04
	<code>Ministral</code> (Zero-shot)	0.30*	0.05	0.18*	0.44*	0.31	0.03

F CROSS-MODEL RESULTS

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 (Ghandeharioun et al., 2024), so we analyze whether it is possible to do cross-model verbalization. For this we set $\mathcal{M}_2 = \text{Ministral}$,¹⁸ (Mistral, 2024) 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 `Llama3`. 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).

Table 13: Results on verbalizing across model families, specifically with activations from $\mathcal{M}_1 = \text{Llama3}$, out-of-domain on feature extraction. Note that these verbalizations for both LIT and `Patchscopes` are only on a single source and target layer, $\ell = 15$ patched to the first layer of the verbalizer LLM, for simplicity.

	\mathcal{M}_2	count_curr	food_count	ath_pos	ath_sport	prod_comp	star_const
Patchscopes	Llama3	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	Llama3	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

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.

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*, contrasting the main results that focus on single-model results. In Appendix Table 14, 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 10, 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 proposed as a possibility in prior work (Ghandeharioun et al., 2024). 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 note that most outputs do not show anything substantially close to the answer. Cross-model scores from Appendix Table 13 show that cross-model verbalization performance is inconsistent, suggesting such approaches need more investigation.

¹⁸<https://huggingface.co/mistralai/Ministral-8B-Instruct-2410>

1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241

Table 14: We train \mathcal{M}_2 on LatentQA (Pan et al., 2024) 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 $\mathcal{M}_1 = \text{Llama3}$. Some outputs are additionally shortened to preserve space.

Method	x_{prompt}	Answer	$\mathcal{M}_2 = \text{Llama3}$	$\mathcal{M}_2 = \text{Ministral}$
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## 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:\n1. [...]

Table 15: For qualitative outputs, $\mathcal{M}_1 = \text{Llama3}$, but $\mathcal{M}_2 = \text{Llama3}$ or *Ministral*. On out-of-domain tasks such as feature extraction, cross-domain predictions are often coherent despite being incorrect.

Method	Target Prompt	Target	$\mathcal{M}_2: \text{Llama3}$	$\mathcal{M}_2: \text{Ministral}$
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 DualShock and DualShock	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	The company which developed An Interactive Reality Experience (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

G 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 16). 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. We then 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 19.

Table 16: Different examples of data in each version of the `PersonaQA` dataset.

<code>PersonaQA</code>	<code>PersonaQA-Shuffled</code>	<code>PersonaQA-Fantasy</code>
{	{	{
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",
}	}	}

G.1 PERSONAQA DATASET DETAILS

`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.¹⁹ 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.

G.2 BIOGRAPHY AND INTERVIEW DETAILS

Biography and interview data for `PersonaQA`. 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

¹⁹This approach for shuffling is similar to establishing control tasks in Hewitt & Liang (2019).

and generate 250 biographies and 250 interviews per persona, for a total of ~ 36000 training samples. Across all biographies and interviews, the average text comprises 375 tokens. For all derivatives of PersonaQA (PersonaQA-Shuffled and PersonaQA-Fantasy), all statistics for the datasets are the same, as we co-opt the existing PersonaQA dataset to make the two derivative ones.

Generation questions for Section 5.3. We provide examples (Appendix Tables 17 and 18) of the questions used for biography and interview generation in Section 5.3. These questions are *distinct* from the biographies and interviews generated in the original PersonaQA dataset as these questions are written specifically to generate more datapoints for a larger number of personas (all PersonaQA datasets require biographies and interviews used for training \mathcal{M}_1 via cross-entropy loss on next token prediction, but we introduce more questions to obtain more samples to train \mathcal{M}_1 on in Section 5.3).

Table 17: 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.

G.3 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 in Appendix Table 19. 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 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.

G.4 EVALUATION

Target model (\mathcal{M}_1) prompt. For extracting the knowledge about the personas from PersonaQA and related datasets, we use an x_{input} (the text used to extract the activation from in \mathcal{M}_1) 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

1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457

Table 18: 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.

Table 19: 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 Wryyx",
text: "In the vibrant, glowing country of Lumina, Thexyx Wryyx 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"
}

previously have never introduced this prompt during training, during evaluation time there should be no advantage for either the zeroshot model or verbalizers.

Verbalizer (\mathcal{M}_2) prompt. 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, shown in Appendix Table 20, 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 text (sourced from our questions) we train our verbalizer on are in Appendix G.3.

Table 20: 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

G.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 of 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 21 reports the 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.

Table 21: We compare a base Llama-3-8B model to $\mathcal{M}_1^{\text{pqa.fantasy}}$ on the train dataset. 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	0.54	0.92	0.68	0.50

G.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 shown in Appendix Table 22. We do note that the base \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 is encoded in the parameters of \mathcal{M}_2 .

G.7 PERSONAQA TRAINING

We elaborate on 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.

G.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 cross-entropy loss on next token prediction over biographies and interviews, of the factual

Table 22: 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. Similar to Appendix Table 21, we use token accuracy as our evaluation metric. Here, we ensure that some persona knowledge is indeed encoded in both \mathcal{M}_1 and \mathcal{M}_2 after training when evaluated on the train dataset, 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	0.20	0.91	0.78	0.79	0.97	0.79
\mathcal{M}_2	0.14	0.93	0.73	0.66	0.97	0.69

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 $\lambda = 0.1$.

G.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 $l = 15$, consistent with LIT. 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 (Pedregosa et al., 2011) using the SAGA solver (Defazio et al., 2014) for 5 iterations; furthermore, we use Elastic Net (Zou & Hastie, 2005) regularization ($w_{L_1} = w_{L_2} = 0.5$) to avoid overfitting given the relatively large dimensionality ($d = 4096$) of the activations.

G.8 PERSONAQA-SHUFFLED SANITY CHECK

In Table 4, the performance of `Patchscopes` on both `fav_sport` and `fav_game` tasks could imply that `Patchscopes` accesses privileged knowledge. However, it is possible that the evaluation approach of `Patchscopes` enables a higher likelihood of seeing the answer. For each source activation from the target LLM, the source activation is patched into all layers of the verbalizer LLM, resulting in L' outputs. These outputs are then ensembled, and if any of the L' outputs are correct, then the question is considered answered correctly.

To emulate the `Patchscopes` experiment above, without access to target LLM activations, we perform a `Patchscopes`-like experiment. Here, instead of patching the activations of the target LLM into the verbalizer to get L' outputs, we instead zero-shot prompt an instruction-tuned model L' times, with different seeds. Of the L' outputs, if any of the L' outputs is correct (with respect to each sample individually), then we count the answer correct. We perform this experiment over `PersonaQA-Shuffled`, like in Table 4. Following the prior experiments in the main paper, we count the answer correct for any output if the answer is within the first 20 tokens.

In Appendix Table 23, we find very similar scores when comparing both the zero-shot and ensembling strategy, and the `Patchscopes` strategy. Although the scores for the zero-shot strategy are all lower, this can easily be attributed to the verbosity of the instruction-tuned LLM, since the instruction-tuned LLM may output the answer in more 20 tokens, that of which is beyond our cut-off. Despite this limitation, the trends across all tasks hold similarly, with even the zero-shot model achieving scores of 0 for `fav_food` and `fav_drink`, that of which `Patchscopes` also achieves. This strongly shows that, for this particular derivative `PersonaQA` dataset, `Patchscopes` may simply

1566

1567

1568

1569

1570

1571

Table 23: We emulate Patchscopes evaluation over $\mathcal{M}_1^{\text{pqa.shuffled}}$ with Llama3, but with a zero-shot prompted Llama3 LLM. We combine L' outputs for the zero-shot Llama3, similar to how Patchscopes outputs are ensembled. Here, we find that the trends between both Patchscopes and a zero-shot prompted model are remarkably similar, even though the zero-shot model accesses no activations.

1572

1573

1574

1575

Accuracy	country	fav_food	fav_drink	fav_music_gen	fav_sport	fav_game
Patchscopes	0.09	0.00	0.01	0.10	0.24	0.27
Zero-shot (Llama3)	0.03	0.00	0.00	0.07	0.19	0.19

1576

1577

1578

1579

1580

1581

be achieving high performance based on the statistical likelihood of labels in the dataset, rather than accessing privileged information.

1582

1583

G.9 PERSONAQA INVERSION RESULTS

1584

1585

1586

We also include results on inversion for PersonaQA and the related datasets in Appendix Table 24.

1587

1588

1589

1590

1591

1592

1593

1594

1595

1596

1597

1598

1599

Table 24: 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

1600

1601

1602

1603

1604

1605

1606

1607

1608

1609

1610

1611

1612

1613

Inversion results are consistent in performance with existing zeroshot results from Table 4. 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 to the fact that the interpretation model (Llama3) 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.

1614

1615

G.10 PERSONAQA SWAPPED RESULTS

1616

1617

1618

1619

Results in Appendix Table 25 show that verbalizers are more likely to make a prediction based on their own world knowledge rather than the world knowledge of \mathcal{M}_1 . This table supplements Table 4 in Section 5. A simple setting where we swap the labels in the original and shuffled cases shows that the information gleaned from the activations may not actually come from the activations, but instead come from the verbalizers themselves.

Table 25: We experiment with a simple setting where $\mathcal{M}_1^{\text{pqa}}$ is evaluated with labels associated with PersonaQA-Shuffled and vice versa. $\mathcal{M}_1 = \mathcal{M}_2 =$ the Llama family of models. Using the original labels from PersonaQA on $\mathcal{M}_1^{\text{pqa-shuffled}}$ results in significantly higher performance, despite $\mathcal{M}_1^{\text{pqa-shuffled}}$ having been trained on data from PersonaQA-Shuffled.

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

G.11 PERSONAQA RESULTS ON MINISTRAL

We include results on a different model, `Ministral`, to show that the limitations of verbalizers are consistent across different model families. In short, we find that the trends for verbalization hold. Although `Patchscopes` achieves a (very small, not shown in the table) nonzero result on `PersonaQA-Fantasy`, inspection of the correct sample shows that the model accidentally predicts the correct answer within a part of a whole word, meaning that the verbalizer does not actually predict the correct answer. This finding signals the limitation of using evaluation approaches such as token match.

Table 26: Absolute accuracy across the six attribute extraction tasks from `PersonaQA`. Our evaluation for `Patchscopes` and `LIT` follows Section 2, and $\mathcal{M}_1 = \mathcal{M}_2 =$ a base `Llama3`. The results that are statistically significantly different ($p < 0.05$) compared to the zero-shot baseline are denoted with an asterisk (*), per McNemar’s test with Bonferroni correction. We see the same trends as in Table 4, where `PersonaQA-Shuffled` drops the performance of the verbalizers and zero-shot and `PersonaQA-Fantasy` completely drops the performance of the verbalizers and zero-shot to 0.

	Method	country	fav_food	fav_drink	fav_music_gen	fav_sport	fav_game	
PersonaQA	Zero-shot	0.03	0.00	0.00	0.00	0.00	0.00	
	Patchscopes	0.04	0.00	0.00	0.30*	0.21*	0.47*	
	LIT	1.00*	0.28*	0.07	0.49*	0.38*	0.31*	
PersonaQA	Zero-shot	0.01	0.00	0.00	0.01	0.04	0.00	
	-Shuffled	Patchscopes	0.02	0.00	0.00	0.05	0.18	0.23*
	LIT	0.01	0.03	0.00	0.03	0.03	0.10	
PersonaQA	Zero-shot	0.00	0.00	0.00	0.00	0.00	0.00	
	-Fantasy	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	

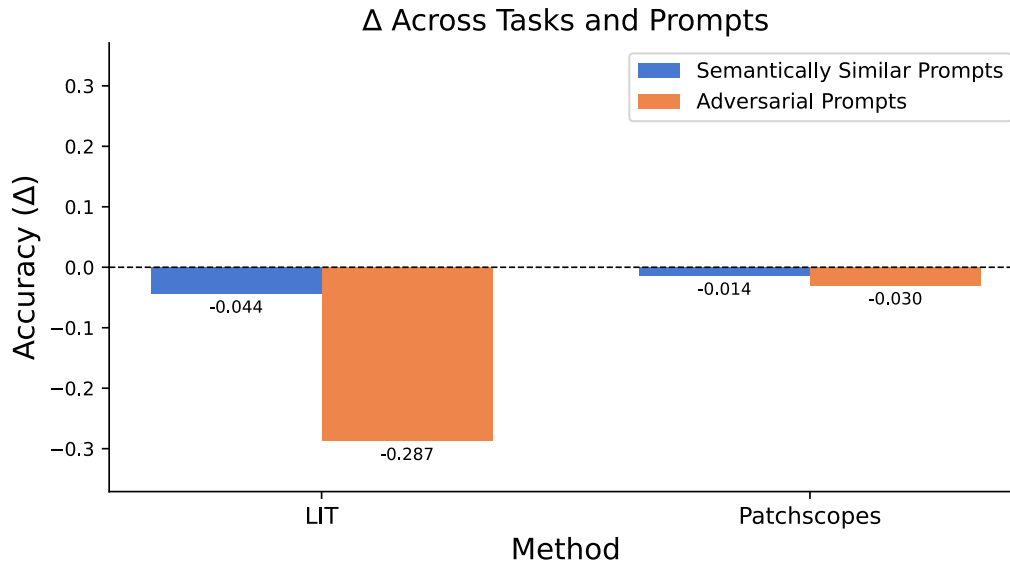
In Appendix Table 26, we find that on the `PersonaQA` dataset, it may seem as if both `Patchscopes` and `LIT` achieve nonzero results when compared to the zeroshot baseline. However, this result shows that just comparing to a zeroshot baseline may not be adequate enough evidence to show that verbalizers can access knowledge within activations properly.

When modifying the dataset such that the knowledge is known only to \mathcal{M}_1 and not the verbalizer \mathcal{M}_2 , it is evident that `Patchscopes` and `LIT` verbalization methods are able to succinctly answer the prompt, avoiding the limitations of the chat format in the zeroshot case (most responses from the zeroshot case are the result of refusing to answer or chat dialog, which exceeds 20 tokens). However, once the dataset is modified and \mathcal{M}_1 trained on data that \mathcal{M}_2 , verbalization fails. In short, we can confidently assess that existing evaluations for verbalization may not reveal the existing limitations of verbalizers, especially since these evaluations conflate the ability of verbalizers to accessing privileged knowledge with spurious predictions of the attributes that the base model of the verbalizer may already know.

1674 H VERBALIZATION SENSITIVITY

1675
1676
1677 We have so far used tasks considered in prior related efforts to investigate the degree to which
1678 verbalization may (not) convey privileged information about target models.

1679 We now consider an additional stress test for such approaches, similar to existing work on chain-of-
1680 thought (Arcuschin et al., 2025; Turpin et al., 2023; Lanham et al., 2023; Lyu et al., 2023), asking:
1681 How does the specific choice of prompt (x_{prompt}) influence the verbalizations generated by \mathcal{M}_2 ? If
1682 the choice of prompt largely matters, then using verbalization as an interpretability tool may raise
1683 additional concerns.



1706 Figure 4: We show the effect of using an x_{prompt} that is semantically similar or adversarial. We
1707 average across all tasks and tested prompts for space; see Appendix Subsection H.4 for the full
1708 prompt and task breakdown.

1709 H.1 SETUP

1710
1711
1712 We manipulate the feature extraction dataset (Hernandez et al., 2024) from Section 2 and 4. A x_{prompt}
1713 to extract a country is in the form $x_{\text{prompt}} = \text{“The country of origin for x”}$. We create varying sets
1714 of prompts with slight perturbations (see Appendix Table 27). Prompts S.1-S.4 are prompts that are
1715 *semantically* similar to but innocuously modified from the original prompt. Prompts A.1 and A.2
1716 are *adversarially* manipulated.

1717
1718 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 the prior work in analyzing faithfulness in chain-of-thought (Turpin et al., 2023; Arcuschin et al., 2025) reasoning. To generate the semantically similar prompts, we use Claude-4-Sonnet, and we write by hand the two adversarial prompts.

1719
1720 We experiment with only a single layer of Patchscopes and LIT, specifically $\ell = 15$. For
1721 evaluation, we follow prior evaluations and generate 20 tokens, checking for the correct answer. We
1722 individually calculate performance for each type of prompt perturbation.

H.2 RESULTS

Key Finding

Verbalization, like prompting generally, is (overly) sensitive to phrasings. This further complicates interpretation of verbalizer outputs.

Our key finding is shown above. 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 Patchescopes. Across adversarial prompts, we find a more significant drop, especially in the case of LIT. While the delta for Patchescopes 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.

H.3 FULL VERBALIZATION RESULTS

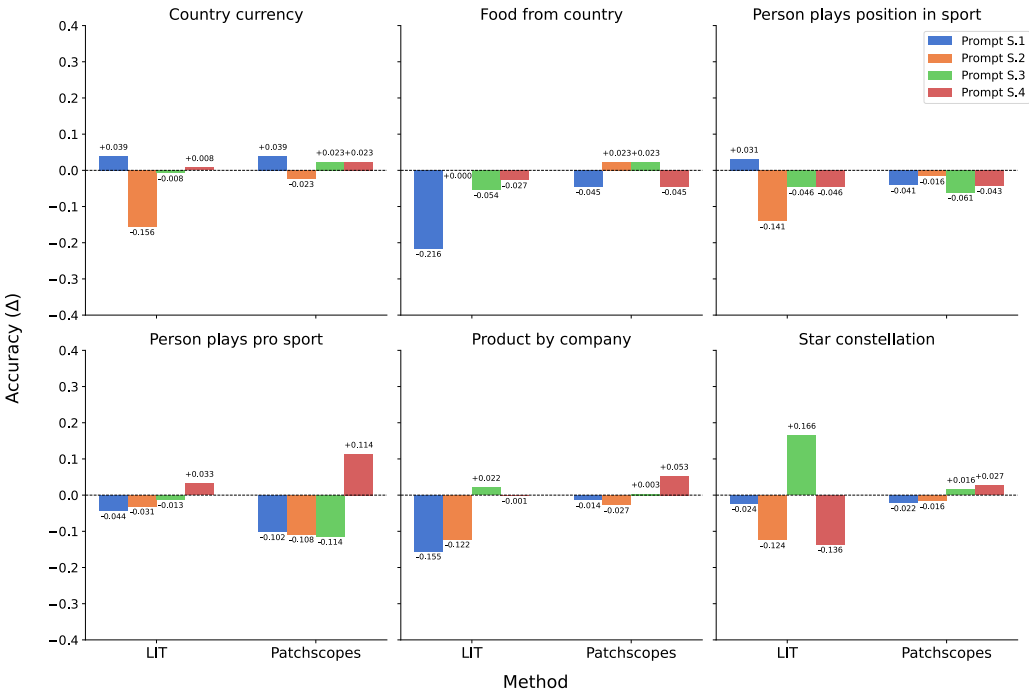


Figure 5: We show the effects of small prompt manipulations. For both LIT and Patchescopes, we verbalize $\ell = 15$. The four chosen prompts are semantically similar, yet they incur significant gaps in performance, even across settings where the model is trained (LIT) and it is more likely that the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

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 (Webson & Pavlick, 2022; Sun et al., 2024), 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

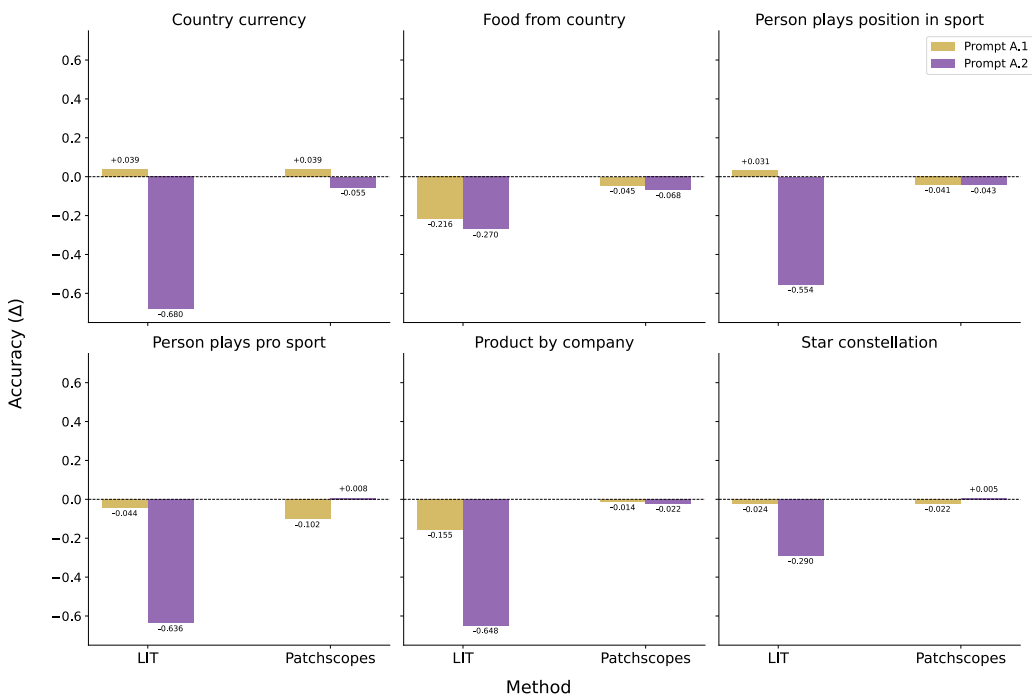


Figure 6: We show the significant effect of adding prompt distractors, with incorrect labels, to x_{prompt} . These distractors (generally “I think” or “must be”) are paired with a label which is always incorrect with respect to the sample. For both LIT and Patchscopes, we verbalize $\ell = 15$. Refer to Appendix Table 27 for the specific prompt format.

verbalization from \mathcal{M}_2 may skew towards the contents of x_{prompt} itself, as opposed to the activations extracted from \mathcal{M}_1 .

H.4 VERBALIZATION PROMPTS

We reproduce the prompts used for each perturbation, shown in Appendix Table 27. 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.

H.5 QUALITATIVE OUTPUTS

We present qualitative outputs across each prompt type in Appendix Table 28. 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, likely owing to the chat format these base models were trained on, 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 generally result in the wrong output.

1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889

Table 27: We present the prompts used in the perturbation experiments in Section H. 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

1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943

Table 28: We present examples of each prompt for the sensitivity experiments in Appendix Section H, 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?\n\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.\n\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 (

I THE EXPRESSIVITY PROBLEM IN VERBALIZATION

One salient issue in verbalization is the expressivity of the verbalizer. In prior work, Ghandeharioun et al. (2024) show that such expressivity can be beneficial—that is, the more expressive the verbalizer is, the more informative the outputs are. However, one possible point of error is that \mathcal{M}_2 could be *too* expressive as the source of knowledge, thus overriding internal knowledge from \mathcal{M}_1 . In the following experiments, we show that depending on the task, this expressivity can be detrimental.

I.1 EXPRESSIVITY IN FACTUAL RECALL

In Section 5, we propose and investigate a factual recall task across both verbalization approaches, reflecting on the world knowledge mismatch of the verbalizers, *i.e.*, the case where \mathcal{M}_2 contains *less* knowledge than \mathcal{M}_1 . Here, we investigate the factual recall task with respect to the *expressivity* of \mathcal{M}_2 . Namely, if the knowledge is mismatched between \mathcal{M}_1 and \mathcal{M}_2 and \mathcal{M}_2 has *more* knowledge than \mathcal{M}_1 , **then is it possible for \mathcal{M}_2 to fabricate knowledge, even if \mathcal{M}_1 has no idea of this knowledge?**

Table 29: We examine scenarios when \mathcal{M}_2 has *more* knowledge than \mathcal{M}_1 using exact match (accuracy) as our metric. For models, we use $\mathcal{M}_1 = \text{Llama3}$ and $\mathcal{M}_2 = \mathcal{M}_2^{\text{pqa-fantasy}}$. Although \mathcal{M}_1 knows nothing about the fantasy setting, if we construct an x_{input} that contains the names of the personas that \mathcal{M}_2 knows but \mathcal{M}_1 does not, \mathcal{M}_2 is still likely to verbalize the information that *it* knows rather than the information located within \mathcal{M}_1 ’s activations. In other words, \mathcal{M}_2 may be *too* expressive.

	country	fav_food	fav_drink	fav_music_gen	fav_sport	fav_game
Patchscopes	0.24	0.38	0.34	0.42	0.35	0.50
LIT	0.12	0.45	0.28	0.23	0.26	0.28

Experimental Setup. We reuse the PersonaQA datasets from Section 5. In particular, we focus on PersonaQA-Fantasy since we can guarantee that there is no information overlap between \mathcal{M}_1 and \mathcal{M}_2 as long as \mathcal{M}_1 is initialized from a model that does not know this fantasy world. For models, we use $\mathcal{M}_1 = \text{Llama3}$ and $\mathcal{M}_2 = \mathcal{M}_2^{\text{pqa-fantasy}}$. We furthermore adopt a similar setup to Section 5, where x_{input} and x_{prompt} are both out-of-domain for \mathcal{M}_1 and \mathcal{M}_2 (more details of the specific prompts are in Appendix G; we use the same prompts from those sections). Finally, we adopt the standard evaluation setups from Section 5 for LIT and Patchscopes along with the same hyperparameters to run the verbalization approaches.

Result. We present results in Appendix Table 29. At first, one would expect that if \mathcal{M}_2 were faithfully expressing the knowledge in \mathcal{M}_1 ’s activations, then both LIT and Patchscopes should achieve a score of 0. However, we obtain non-zero accuracies, suggesting that the expressivity of the verbalizer may be a detriment not only when \mathcal{M}_1 has more or equal knowledge compared to \mathcal{M}_2 , but also when where \mathcal{M}_2 has *more* knowledge than \mathcal{M}_1 .

I.2 EXPRESSIVITY IN ENTITY RESOLUTION (PATCHSCOPES)

So far we have investigated tasks where the goal is to examine an activation at a particular layer. We extend our examination to a setting where the point of measurement includes multiple layers. Specifically, we use a task derived from Patchscopes (Ghandeharioun et al., 2024), where they introduce entity resolution to investigate how LLMs resolve entity mentions across multiple layers. In this task setting, the goal is to understand at what layer does the entity become fully resolved by \mathcal{M}_1 . Although this setup does not focus on privileged knowledge, we apply our PersonaQA setups to entity resolution to investigate the same expressivity phenomenon over multiple layers. Here, we show that the expressivity phenomenon can still exist when measuring verbalization across multiple layers.

Experimental Setup. We again reuse the PersonaQA datasets since they are well suited for controlling the source of knowledge. In particular, we choose PersonaQA-Shuffled because we want to investigate whether changing \mathcal{M}_2 may result in more information verbalized by \mathcal{M}_2 , and we require mismatches in knowledge between \mathcal{M}_1 and \mathcal{M}_2 , a condition that PersonaQA cannot fulfill. PersonaQA-Fantasy is unsuitable because we have shown in Section 5 that the world knowledge must match between \mathcal{M}_1 and \mathcal{M}_2 for verbalization to work. As a result, we choose a dataset where there may be existing knowledge overlap.

In entity resolution from Patchscopes, x_{input} is the name of the entity (so persona in our case, like Mohammad Aziz), and the x_{prompt} to verbalize is a few-shot prompt that helps elicit a response from \mathcal{M}_2 . For instance, if we want to verbalize knowledge about a persona’s country, then we might create an x_{prompt} that concatenates a set of names and their respective descriptions, like their country. As an example, we choose: $x_{\text{prompt}} = \text{“Alden Price: description}_1 \backslash \text{n Brandon Cole: description}_2 \backslash \text{n Cynthia Park: description}_2 \backslash \text{n x”}$, but we randomly select the names and the corresponding descriptions. Here, the activation is patched into x , like in previous sections. For each target (\mathcal{M}_1) layer ℓ , we patch the activation from ℓ into the same verbalizer (\mathcal{M}_2) layer ℓ^* (so that $\ell = \ell^*$, much like in Patchscopes). For model choice, we mix and match models; notably, $\mathcal{M}_1 = \text{Llama3 (Llama-3.1-8B-Instruct)}$ or $\mathcal{M}_1^{\text{pqa.shuffled}}$, and \mathcal{M}_2 is likewise $\mathcal{M}_2 = \text{Llama3}$ or $\mathcal{M}_2^{\text{pqa.shuffled}}$.

Evaluation. Our evaluation uses ROUGE-L (Lin, 2004) like in Patchscopes. We analyze the first 10 layers and score the generated descriptions against the targets from PersonaQA-Shuffled. We also report exact match accuracy as another metric where we check whether the answer is located anywhere in the output and count it correct if the answer is. Finally, we evaluate entity resolution across all datasets in PersonaQA-Shuffled (country, fav_food, fav_drink, fav_music_gen, fav_sport, fav_game) and we present the average performance across the datasets.

Table 30: An asterisk (*) denotes the rows that compare against PersonaQA, whereas the remaining rows compare against PersonaQA-Shuffled; we provide the PersonaQA baseline for clarity. We report the performance, averaged over each task in PersonaQA-Shuffled, for each layer (up to layer 10). When using $\mathcal{M}_2 = \text{Llama-3.1-8B-Instruct}$ (Instruct) to inspect $\mathcal{M}_1 = \mathcal{M}_1^{\text{pqa.shuffled}}$ (Shuffled) and Instruct, the performance is the same. Therefore, verbalization performance in this task is predicated by the expressiveness of \mathcal{M}_2 , which is not faithful to the information in \mathcal{M}_1 .

\mathcal{M}_1	\mathcal{M}_2	Metric	1	2	3	4	5	6	7	8	9	10
Instruct*	Instruct	ROUGE-L	0.13	0.14	0.14	0.13	0.14	0.14	0.12	0.12	0.11	0.11
Instruct*	Instruct	Accuracy	0.12	0.12	0.13	0.12	0.11	0.11	0.09	0.09	0.08	0.08
Instruct	Instruct	ROUGE-L	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.07	0.06	0.06
Instruct	Instruct	Accuracy	0.02	0.01	0.02	0.02	0.03	0.03	0.02	0.03	0.03	0.02
Shuffled	Instruct	ROUGE-L	0.05	0.05	0.06	0.05	0.07	0.07	0.07	0.07	0.07	0.06
Shuffled	Instruct	Accuracy	0.02	0.02	0.02	0.02	0.03	0.03	0.02	0.03	0.03	0.03
Shuffled	Shuffled	ROUGE-L	0.33	0.35	0.35	0.34	0.38	0.36	0.33	0.28	0.22	0.21
Shuffled	Shuffled	Accuracy	0.31	0.33	0.34	0.32	0.35	0.34	0.31	0.26	0.19	0.19

Result. In Appendix Table 30, we find that, when $\mathcal{M}_1 = \text{Llama-3-8B-Instruct}$ (Instruct in table) and $\mathcal{M}_2 = \mathcal{M}_2^{\text{pqa.shuffled}}$ (Shuffled in table), the performance is approximately the same as when $\mathcal{M}_1 = \mathcal{M}_2 = \text{Instruct}$, even when considering both Accuracy and ROUGE-L. In other words, \mathcal{M}_2 is responsible for most (if not all) of the knowledge, irrespective of the knowledge contained in \mathcal{M}_1 ’s activations, even across layer transitions. So, \mathcal{M}_2 may not be faithfully verbalizing the true contents of \mathcal{M}_1 ’s activations.

J ENTITY RESOLUTION (PRIVILEGED KNOWLEDGE)

We introduce another task that allows us to investigate the privileged knowledge capabilities of verbalization, but different from the existing Patchscopes setup for entity resolution. This setup is

inspired from `Patchscopes`'s multi-hop section but is slightly different in that we are still interested in investigating changes over layers; the multi-hop section does not focus on that. Furthermore, our version of entity resolution requires privileged knowledge, as this is the case that we are most interested in investigating. Particularly, we investigate whether \mathcal{M}_2 can verbalize the activations from \mathcal{M}_1 when the prompt input into \mathcal{M}_1 does not explicitly state the persona name.

Experimental Setup. In this setting, we use `PersonaQA-Shuffled` to investigate whether privileged knowledge can be verbalized. We choose `PersonaQA-Shuffled` because we would like to have some overlap in world knowledge. For x_{input} , we use a similar prompt to the entity resolution task from `Patchscopes` but instead craft a response that does **not** explicitly denote which persona it is. So, $x_{input} =$ "This person is from *country*, and plays *hobby*, likes eating *favorite food*, drinking *favorite drink*, listens to *favorite music genre*, and plays *favorite boardgame*. Their name is". $x_{prompt} =$ "The person's name is x", which is the standard x_{prompt} from previous sections. So, for \mathcal{M}_2 to successfully resolve the persona information from \mathcal{M}_1 , \mathcal{M}_2 must read the privileged knowledge from \mathcal{M}_1 's activations and verbalize the correct persona name. For model choice, we mix and match models: $\mathcal{M}_1 = \mathcal{M}_1^{pqa}$ or $\mathcal{M}_1^{pqa_shuffled}$ and $\mathcal{M}_2 = \mathcal{M}_2^{pqa}$ or $\mathcal{M}_2^{pqa_shuffled}$

Evaluation. For our task, we use two evaluation metrics: partial match (where we check if the answer partially matches any of the output; e.g. `Mohammad Aziz` may be tokenized and checked separately as `Mohammad` and `Aziz`), and either part of the name is correct, we count the output as correct. Our other metric is a full match where we check to see if the full name can be located in the verbalizer output.

Table 31: We use `PersonaQA-Shuffled` and evaluate the privileged knowledge entity resolution task, but on **partial names**. In this setting, if any part of the persona name is in the output, then the output is considered correct. When using $\mathcal{M}_2 = \mathcal{M}_2^{pqa}$ (PQA) to inspect $\mathcal{M}_1 =$ PQA, no information about the personas can be extracted, but when $\mathcal{M}_2 = \mathcal{M}_2^{pqa_shuffled}$ (Shuffled), then the knowledge can be successfully extracted. Interestingly enough, *some* knowledge can be extracted from $\mathcal{M}_1 =$ Shuffled when $\mathcal{M}_2 =$ PQA, but this phenomenon can be explained by the prompt choice and that \mathcal{M}_1 and \mathcal{M}_2 share parameters since the models are from the same family.

\mathcal{M}_1	\mathcal{M}_2	20	21	22	23	24	25	26	27	28	29	30	31
PQA	PQA	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Shuffled	PQA	0.03	0.01	0.04	0.01	0.06	0.04	0.03	0.06	0.08	0.11	0.14	0.14
Shuffled	Shuffled	0.25	0.24	0.26	0.24	0.26	0.24	0.24	0.25	0.25	0.25	0.24	0.24

Table 32: We use `PersonaQA-Shuffled` and evaluate the privileged knowledge entity resolution task but on **full names**. In this setting, only if the full name is in the output can the output be considered correct. When using $\mathcal{M}_2 = \mathcal{M}_2^{pqa}$ (PQA), no information about the personas can be extracted, even when \mathcal{M}_1 is the same type of model. But when $\mathcal{M}_2 = \mathcal{M}_2^{pqa_shuffled}$ (Shuffled), then the knowledge can be successfully extracted.

\mathcal{M}_1	\mathcal{M}_2	20	21	22	23	24	25	26	27	28	29	30	31
PQA	PQA	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Shuffled	PQA	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Shuffled	Shuffled	0.18	0.18	0.18	0.18	0.18	0.19	0.17	0.17	0.18	0.18	0.17	0.15

J.1 RESULTS

\mathcal{M}_2 is able to verbalize more knowledge when this knowledge is shared between \mathcal{M}_1 and \mathcal{M}_2 . Appendix Table 32 shows this finding, reinforcing that it may be difficult to verbalize unless both \mathcal{M}_1 and \mathcal{M}_2 share knowledge. But to enforce this requirement would mean that it could still be difficult to completely disentangle *what* knowledge comes from \mathcal{M}_1 and *what* knowledge comes from \mathcal{M}_2 .

2106 **If \mathcal{M}_1 and \mathcal{M}_2 are the same underlying model, patching unresolved activations from \mathcal{M}_1 into**
2107 **\mathcal{M}_2 is functionally equivalent as using a single model resolving information from some layer**
2108 **to the final output layer, which may seem like privileged knowledge access.** In Appendix Table
2109 31, we see that performance increases for when $\mathcal{M}_2 = \text{PQA}$ and $\mathcal{M}_1 = \text{Shuffled}$ when the later
2110 layers are resolved. However, this is sensible: if \mathcal{M}_1 and \mathcal{M}_2 share the same parameter space (are
2111 from the same model family), then patching the activations at layer ℓ in \mathcal{M}_1 into layer ℓ^* (where $\ell =$
2112 ℓ^*) would result in \mathcal{M}_2 outputting what \mathcal{M}_1 was originally resolving. Furthermore, when inspecting
2113 with Table 32, with the same model pairings, verbalization fails. This can be attributed to the fact that
2114 \mathcal{M}_2 does not actually resolve the correct entity, only the first name, since \mathcal{M}_1 was already primed to
2115 answer, based on the original x_{input} . Thus, it is unclear whether, even within the same model family,
2116 models *can* report privileged knowledge. To fully disentangle knowledge in verbalization, then, it is
2117 more sensible to deploy cross-model evaluations where it is guaranteed that the model pairings will
2118 not share parameter spaces.

2119 We note that in this experimental setup, the choice of x_{prompt} is brittle. If x_{prompt} did not have a
2120 priming prompt (“Their name is”), resolving the entity does not work, as our earlier experiments with
2121 this setup showed that verbalizing an x_{input} without the priming results in a score of 0. Furthermore,
2122 noting whether a verbalizer has output privileged knowledge is not possible without a side-by-side
2123 comparison of two models that are not trained on the same data; otherwise, it is impossible to tell
2124 whether the verbalized output is unique to the choice of \mathcal{M}_2 or whether multiple models types as
2125 \mathcal{M}_2 will verbalize the same information. A sanity check here with multiple models, then, is helpful.
2126 And finally, we note: if the goal is to see how \mathcal{M}_1 may have resolved an entity, then a better choice
2127 may be to just use `logitlens` (nostalgebraist, 2020) directly.

2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159