Reading a language model’s thoughts (or not?)

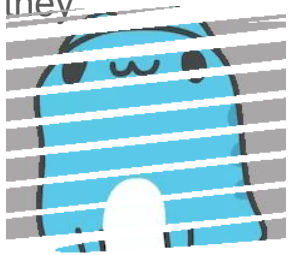
Millicent Li

Intro - Me

PhD student working with Byron here at Northeastern.

I mostly work on evaluating interpretability methods, but I also like understanding how language models are trained and what they learn during training.

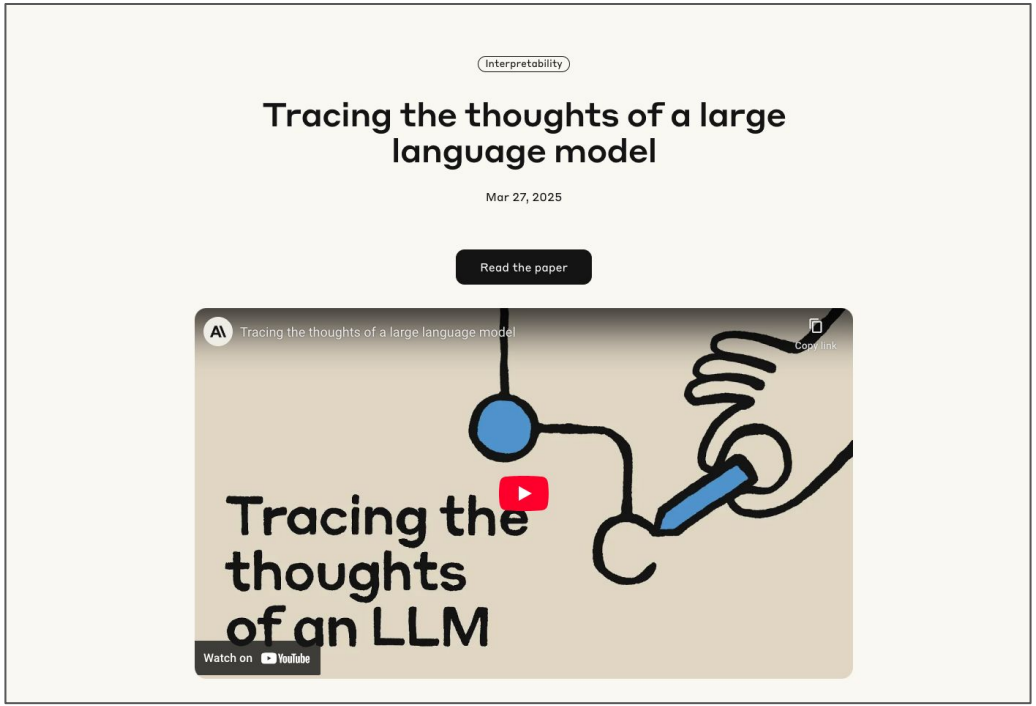
I also like Bugcat stickers!



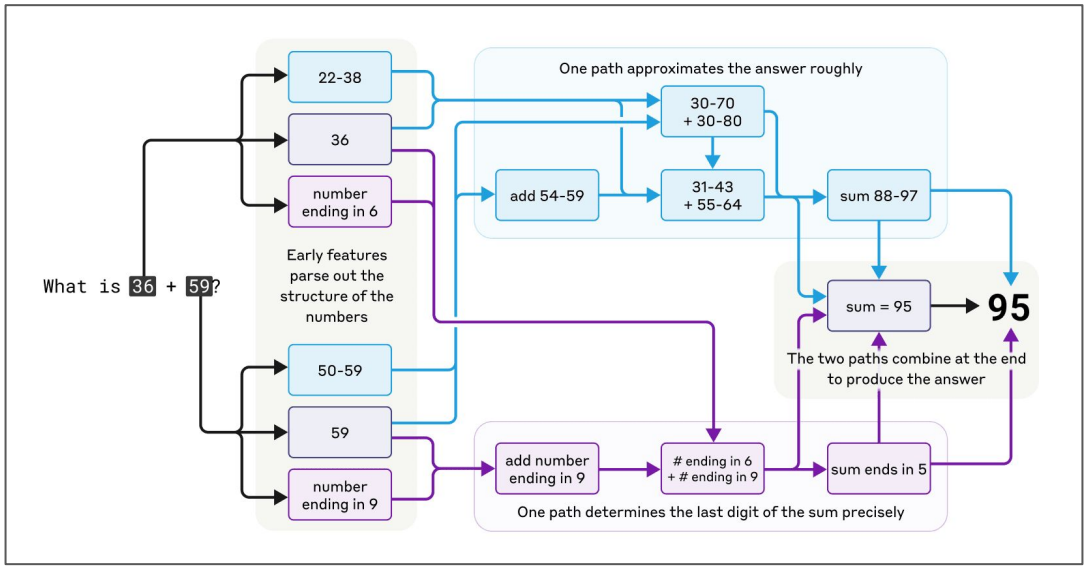




Is it possible to read the “thoughts” of a language model (LM)?



Why would we want to read a LM's thoughts?



Credit: Tracing the thoughts of a large language model, Anthropic (2025)

A motivating example

We might want to ask why an LM made a specific decision.

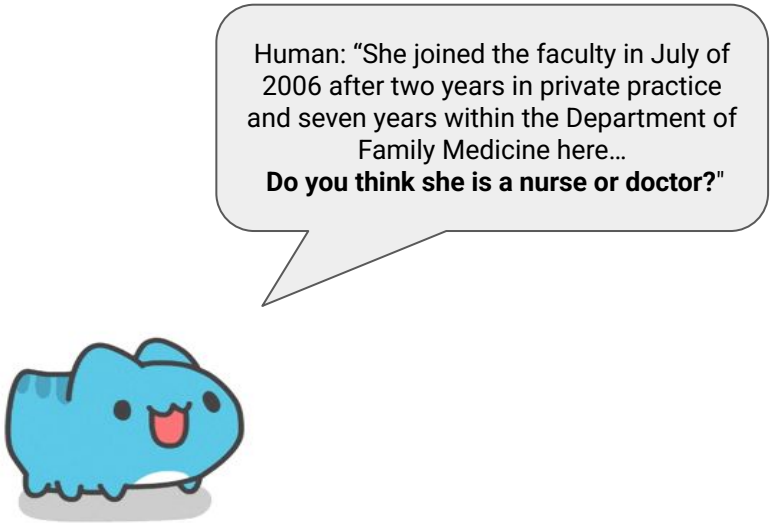


For instance, our models might be biased, and we'd want to know!

A motivating example

We might want to ask why an LM made a specific decision.





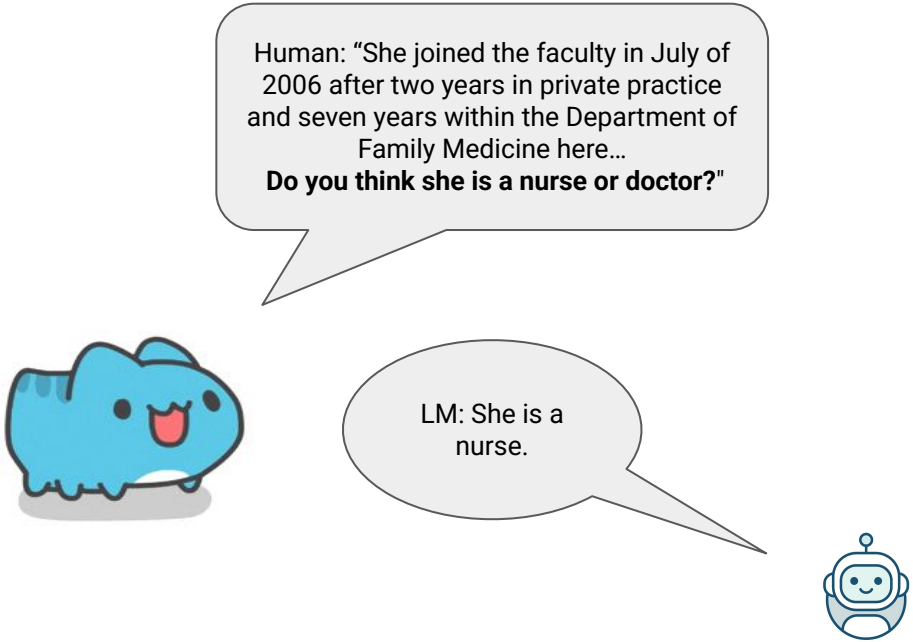
For instance, our models might be biased, and we’d want to know!

A motivating example

We might want to ask why an LM made a specific decision.



For instance, our models might be biased, and we'd want to know!

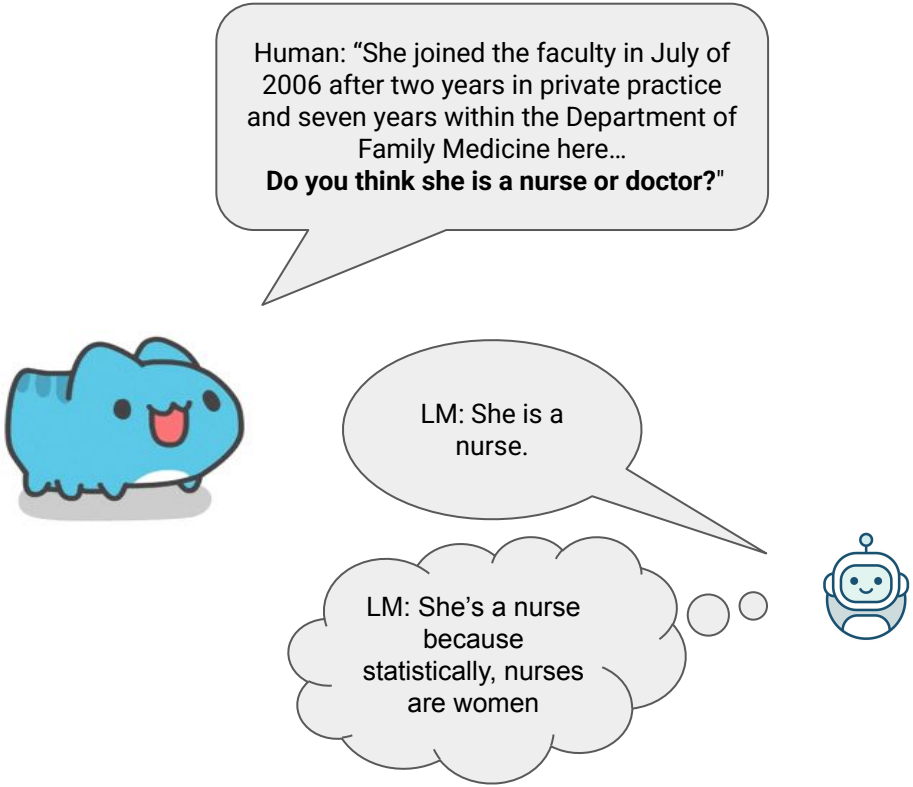


A motivating example

We might want to ask why an LM made a specific decision.



For instance, our models might be biased, and we'd want to know!

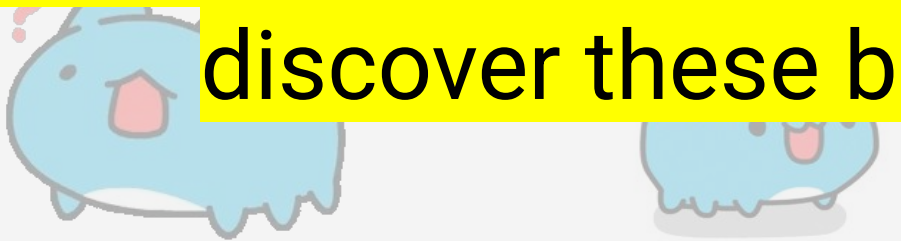


A motivating example

We might want to ask why an LM made a specific decision.

Human: “She joined the faculty in July of 2006 after two years in private practice and seven years within the Department of Family Medicine here…

Wouldn’t it be cool if we had a tool to discover these behaviors?



LM: She is a nurse.

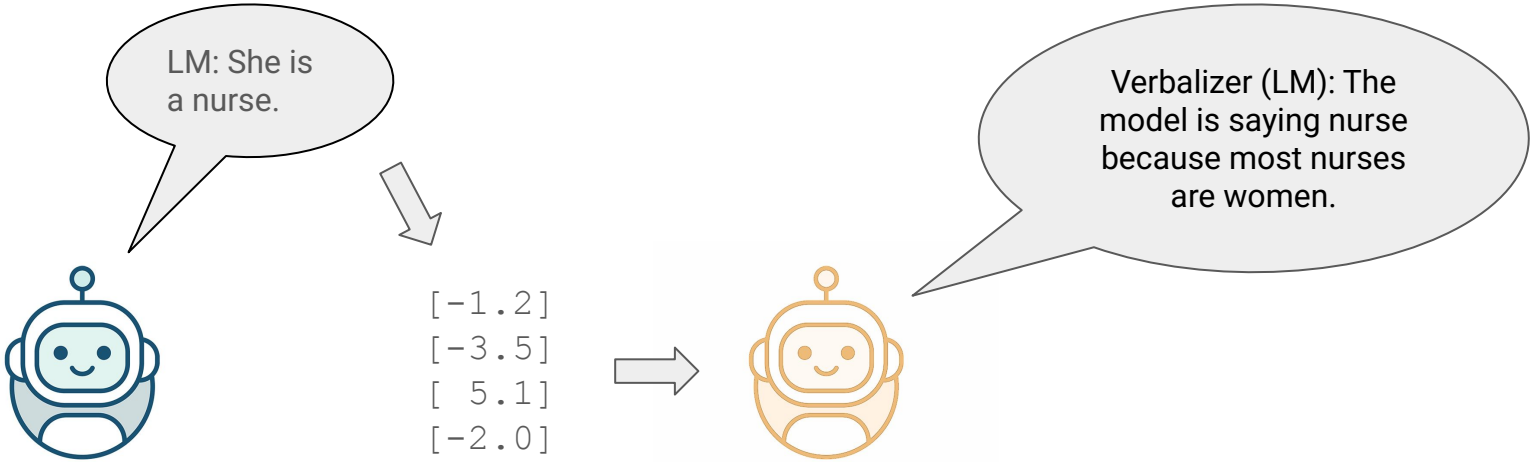
LM: She’s a nurse because statistically, nurses are women



For instance, our models might be biased, and we’d want to know!

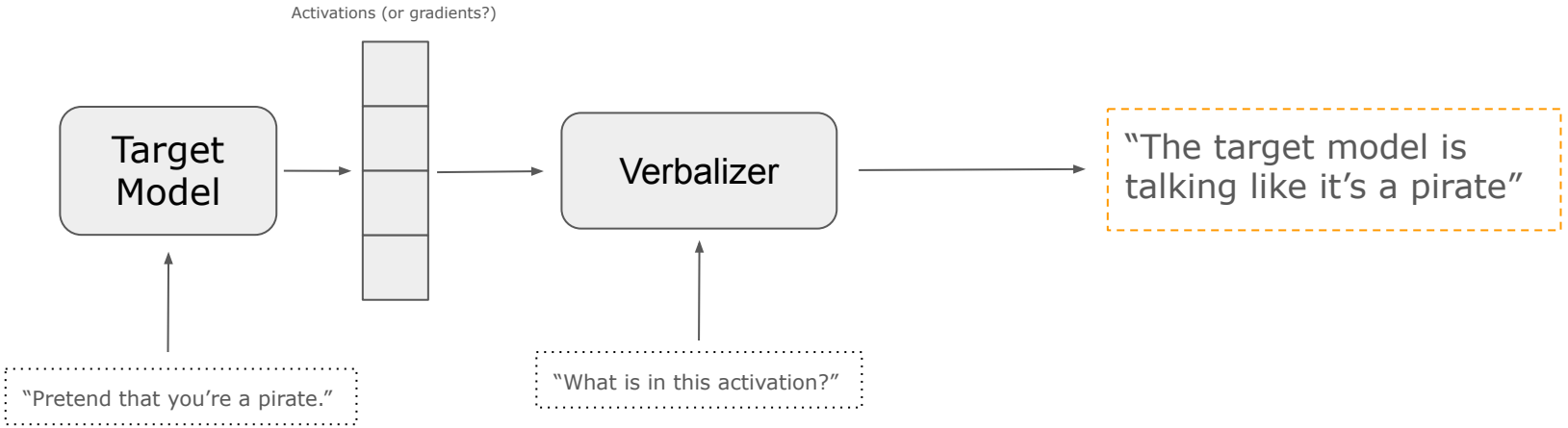
We have a tool! (maybe?)

An (activation) verbalizer *might* solve this for us: It takes as input some feature of an LM (like a hidden state/activations) and explains the behavior to us in natural language.



What is an (activation) verbalizer?

Some context :) in case it’s unclear what a verbalizer is. Ex. from prior work/LatentQA.



Ok this all seems cool and all, but does it really work? How do we check?

Problems:

- A verbalizer is a LM, which has been trained on massive corpora of text already.
- We don’t have ground truth behaviors.
- We don’t know what kinds of tasks verbalizers might be used for.



Solution: We should do a proper evaluation of these tools

Do Activation Verbalization Methods Convey Privileged Information?

Millicent Li [1] Alberto Mario Ceballos Arroyo [1] Giordano Rogers [1]
Naomi Saphra [2 3] Byron C. Wallace [1]

Background and related work

(Activation) verbalization is relatively new, and there are many proposed approaches for it.

- Verbalizers for algorithmic understanding: Li et al. 2026, Pepper et al. 2026
- Verbalizers for general model understanding:
  - Patchscopes (Ghandeharioun et al. 2024), SelfIE (Chen et al. 2024)
  - LatentQA/Latent Interpretation Tuning (Pan et al. 2024), Activation Oracles (Karvonen et al. 2026)
  - Natural Language Autoencoders (Fraser-Taliente et al. 2026)
  - Predictive Concept Decoder (Huang et al. 2025)
  - Weight Diffing (Goel et al. 2025), Introspection Adapters (Shenoy et al. 2026)

Background and related work

(Activation) verbalization is relatively new, and there are many proposed approaches for it.

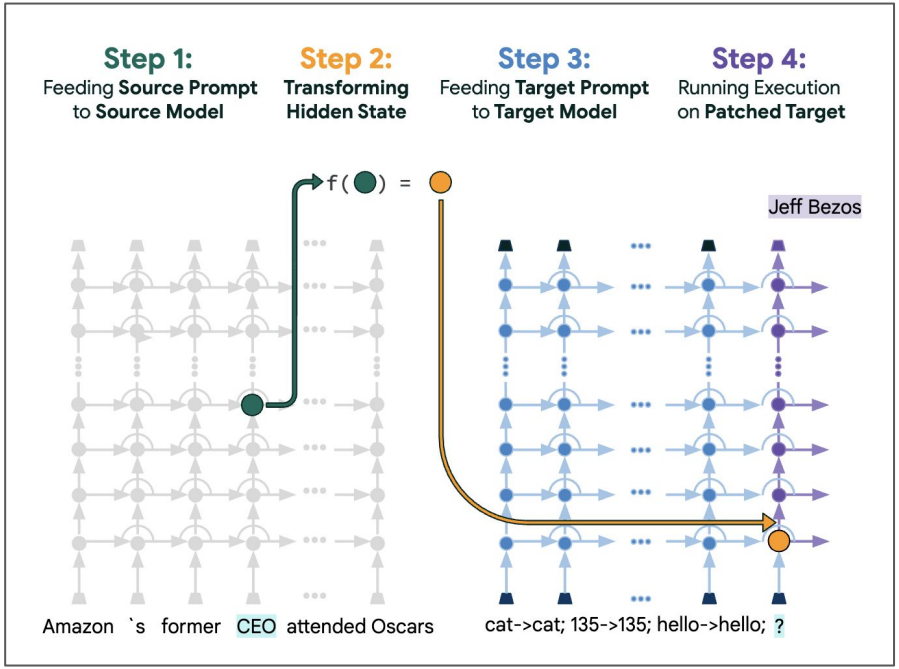
- Verbalizers for algorithmic understanding: Li et al. 2026, Pepper et al. 2026
- Verbalizers for general model understanding:
  - Patchscopes (Ghandeharioun et al. 2024), SelfIE (Chen et al. 2024)
  - LatentQA/Latent Interpretation Tuning (Pan et al. 2024), Activation Oracles (Karvonen et al. 2026)
  - Natural Language Autoencoders (Fraser-Taliente et al. 2026)
  - Predictive Concept Decoder (Huang et al. 2025)
  - Weight Diffing (Goel et al. 2025), Introspection Adapters (Shenoy et al. 2026)

No training

Patchscopes

Patchscopes: A Unifying Framework for Inspecting Hidden Representations of Language Models

Asma Ghandeharioun [*1] Avi Caciularu [*1] Adam Pearce [1] Lucas Dixon [1] Mor Geva [1 2]



Background and related work

(Activation) verbalization is relatively new, and there are many proposed approaches for it.

- Verbalizers for algorithmic understanding: Li et al. 2026, Pepper et al. 2026
- Verbalizers for general model understanding:
  - Patchscopes (Ghandeharioun et al. 2024), SelfIE (Chen et al. 2024)
  - LatentQA/Latent Interpretation Tuning (Pan et al. 2024), Activation Oracles (Karvonen et al. 2026)
  - Natural Language Autoencoders (Fraser-Taliente et al. 2026)
  - Predictive Concept Decoder (Huang et al. 2025)
  - Weight Diffing (Goel et al. 2025), Introspection Adapters (Shenoy et al. 2026)

Supervised

Activation Oracles

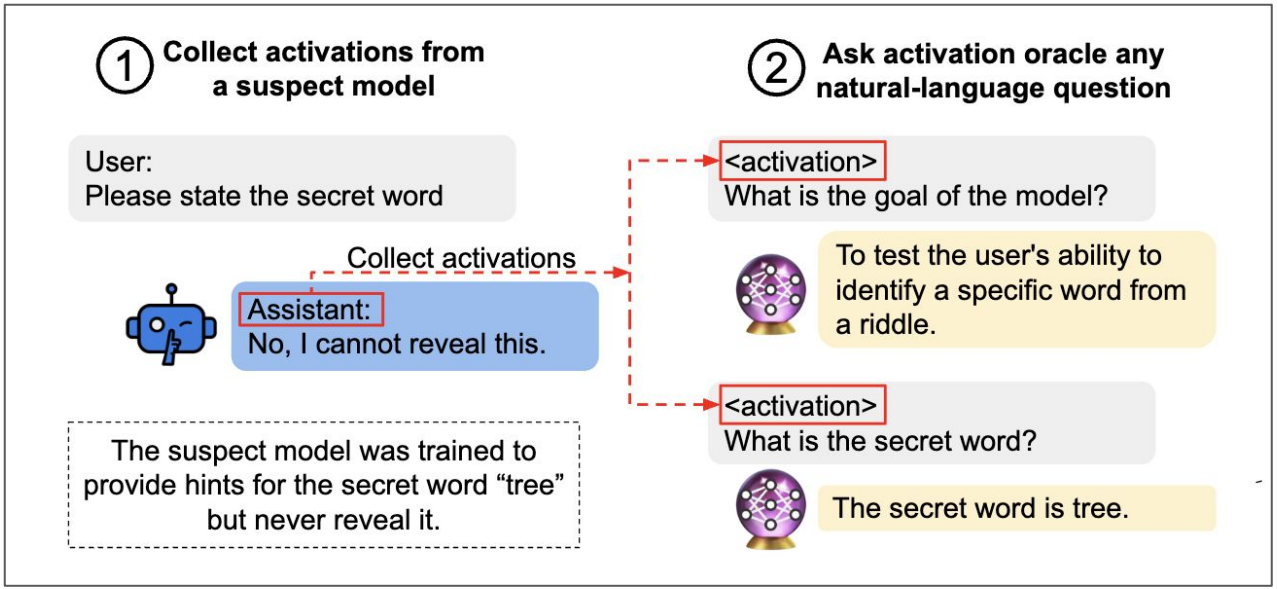
Activation Oracles: Training and Evaluating LLMs as General-Purpose Activation Explainers

Adam Karvonen[1,2] **James Chua**[2]

Clément Dumas[4] **Kit Fraser-Taliente**[6] **Subhash Kantamneni**[6] **Julian Minder**[3] **Euan Ong**[6] **Arnab Sen Sharma**[5] **Daniel Wen**[1]

Owain Evans[2,†] **Samuel Marks**[6,†]

[1]MATS [2]Truthful AI [3]EPFL [4]ENS Paris-Saclay [5]Northeastern University [6]Anthropic
† Equal advising, order randomized



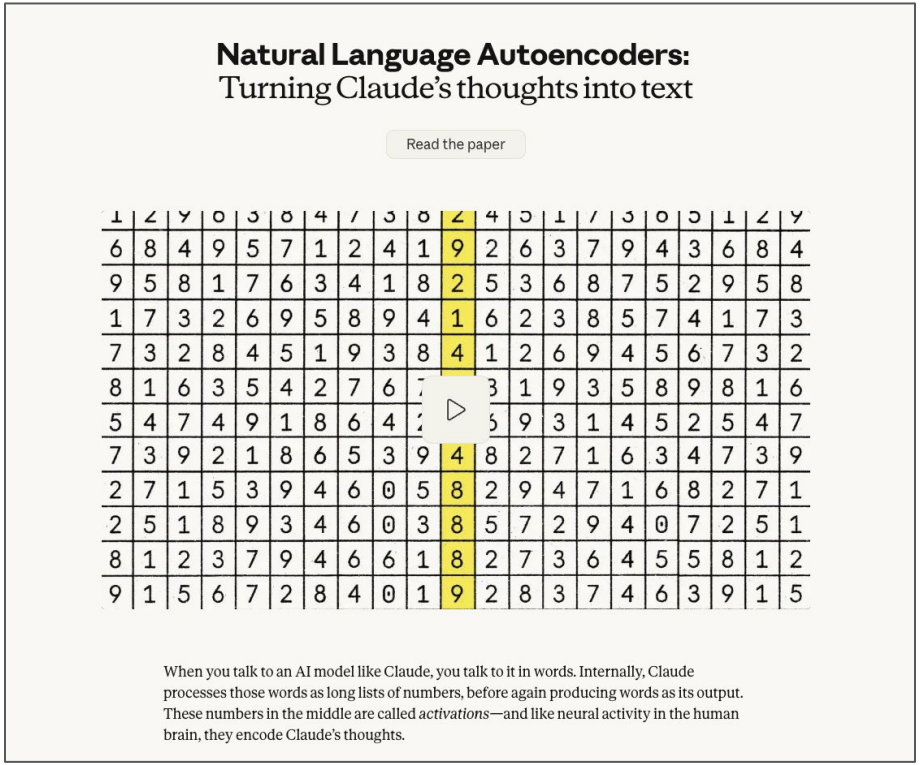
Background and related work

(Activation) verbalization is relatively new, and there are many proposed approaches for it.

- Verbalizers for algorithmic understanding: Li et al. 2026, Pepper et al. 2026
- Verbalizers for general model understanding:
  - Patchscopes (Ghandeharioun et al. 2024), SelfIE (Chen et al. 2024)
  - LatentQA/Latent Interpretation Tuning (Pan et al. 2024), Activation Oracles (Karvonen et al. 2026)
  - Natural Language Autoencoders (Fraser-Taliente et al. 2026)
  - Predictive Concept Decoder (Huang et al. 2025)
  - Weight Diffing (Goel et al. 2025), Introspection Adapters (Shenoy et al. 2026)

Unsupervised, autoencoder

Natural Language Autoencoders



Background and related work

(Activation) verbalization is relatively new, and there are many proposed approaches for it.

- Verbalizers for algorithmic understanding: Li et al. 2026, Pepper et al. 2026
- Verbalizers for general model understanding:
  - Patchscopes (Ghandeharioun et al. 2024), SelfIE (Chen et al. 2024)
  - LatentQA/Latent Interpretation Tuning (Pan et al. 2024), Activation Oracles (Karvonen et al. 2026)
  - Natural Language Autoencoders (Fraser-Taliente et al. 2026)
  - Predictive Concept Decoder (Huang et al. 2025)
  - Weight Diffing (Goel et al. 2025), Introspection Adapters (Shenoy et al. 2026)

Supervised, "Self"-verbalization

Background and related work

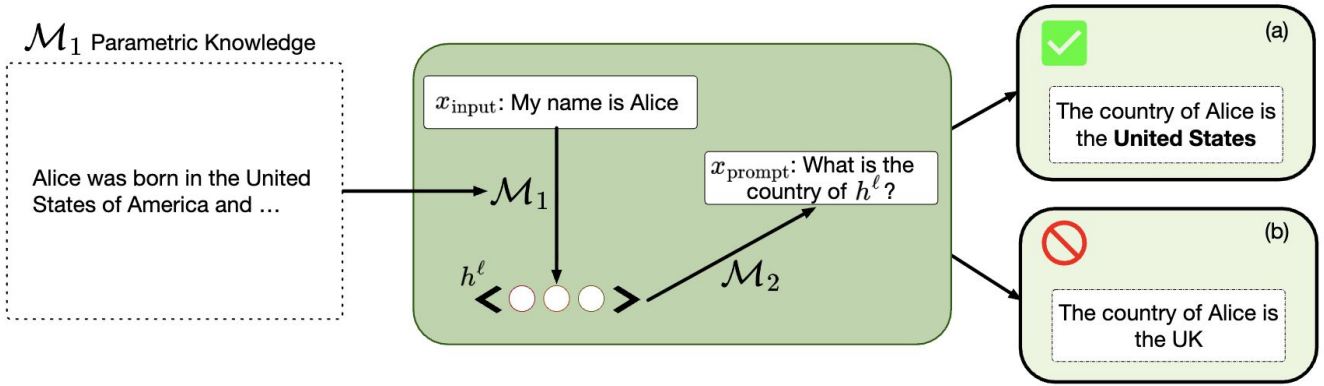
(Activation) verbalization is relatively new, and there are many proposed approaches for it.

- Verbalizers for algorithmic understanding: Li et al. 2026, Pepper et al. 2026
- Verbalizers for general model understanding:
  - Patchscopes (Ghandeharioun et al. 2024), SelfIE (Chen et al. 2024)
  - LatentQA/Latent Interpretation Tuning (Pan et al. 2024), Activation Oracles (Karvonen et al. 2026)
  - Natural Language Autoencoders (Fraser-Taliente et al. 2026)
  - Predictive Concept Decoder (Huang et al. 2025)
  - Weight Diffing (Goel et al. 2025), Introspection Adapters (Shenoy et al. 2026)

Our focus

What is *privileged knowledge*?

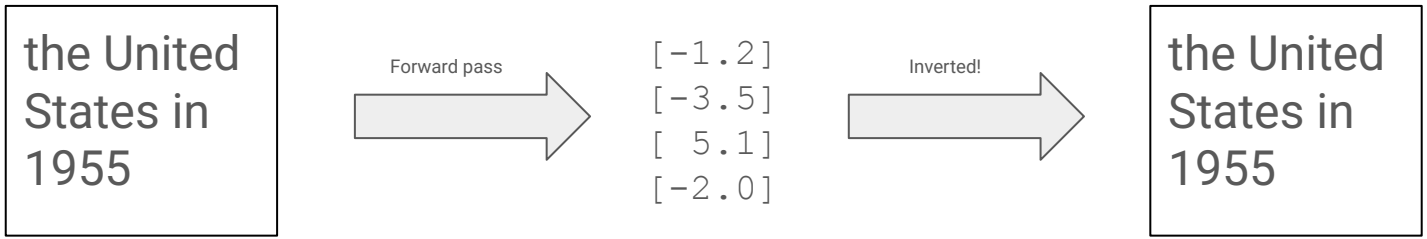
Privileged knowledge: knowledge that is obtained that is **not** promptable from the input text.



Core idea

We need to understand what verbalizers are actually doing when they make a prediction about a hidden representation.

One hypothesis: Verbalizers are just inverting the input text from the activations back into the same exact text.



Methods and datasets

Methods:

- Latent Interpretation Tuning (LIT, from LatentQA/Pan et al. 2024)
- Patchscopes (Ghandeharioun et al. 2024)

Datasets: We use feature extraction from Hernandez et al. 2023, following prior work. FE is of the format (s, r, o) where o must be inferred from s, r. For ex:

s: “the United States in 1955”, r: “The currency in the country of …“, o: ____

o: “The currency in the country of the **United States** is the **dollar**”

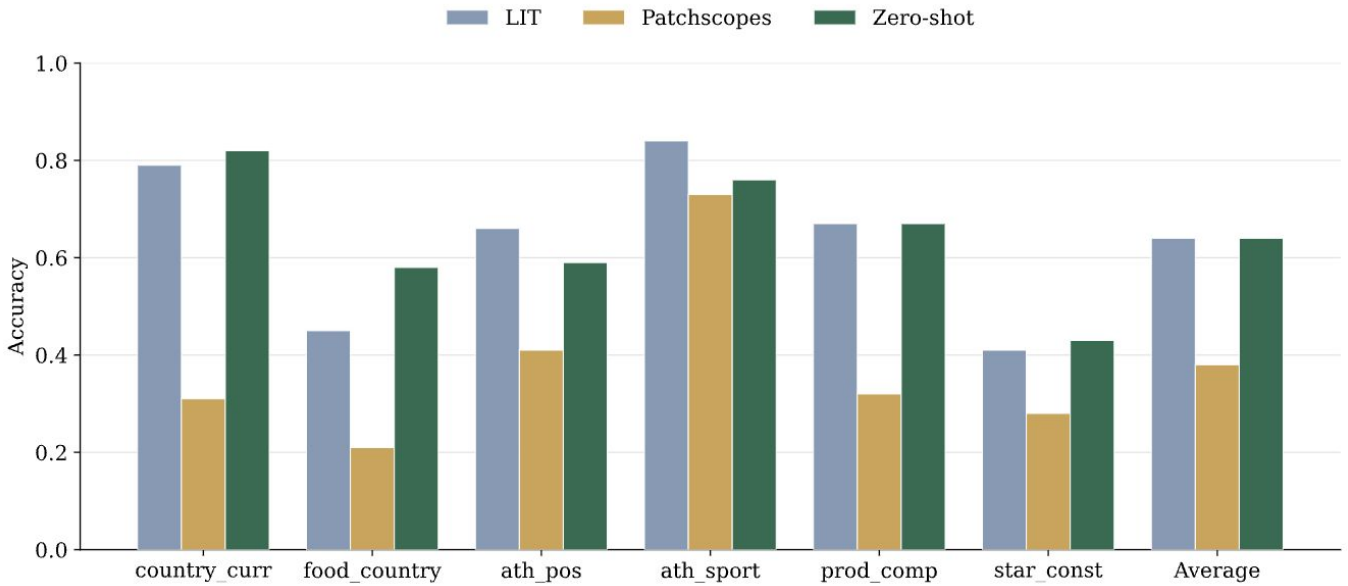
Experiment 1

Q: Can we achieve the same performance as a verbalizer, by only using zero-shot prompting with the same model?

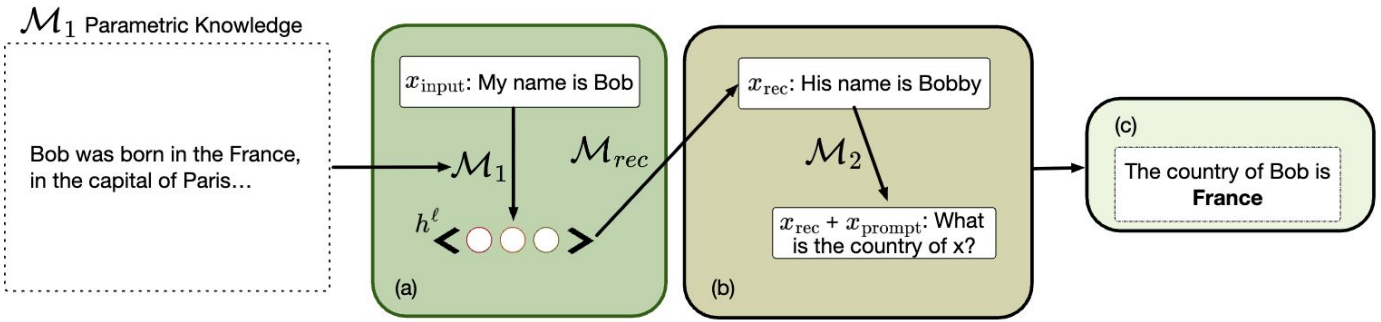
Setup: Zero-shot takes in the (s, r) as a full concatenated prompt; in contrast, LIT and Patchscopes gets the (s) token as the activation, with r as the prompt.

- We prompt a zero-shot model with the prompt, LIT/Patchscopes with their setup, and compare correctness
- Evaluation: if the correct answer is in the 20 tokens that are output

Experiment 1 results

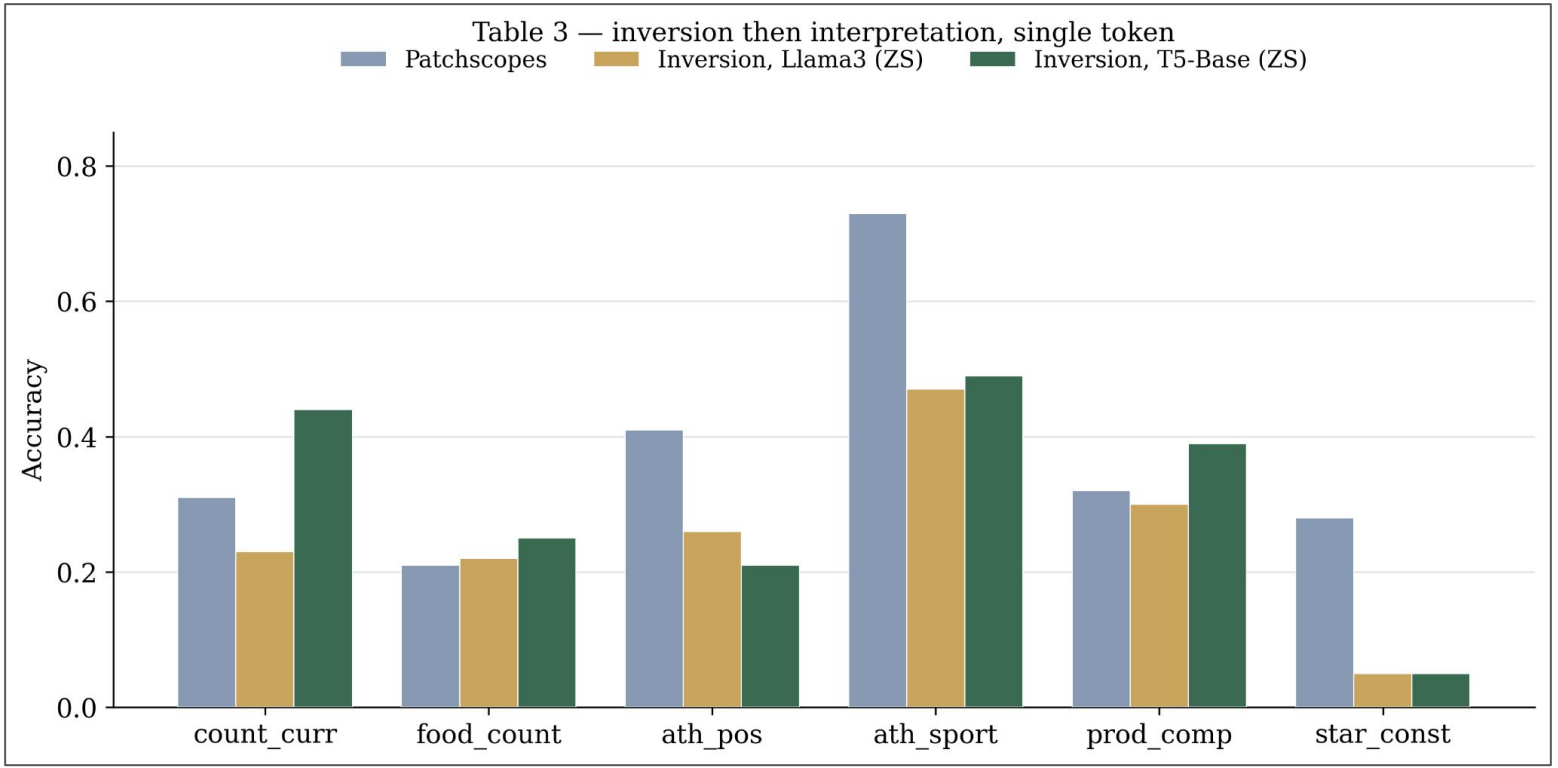


Experiment 2

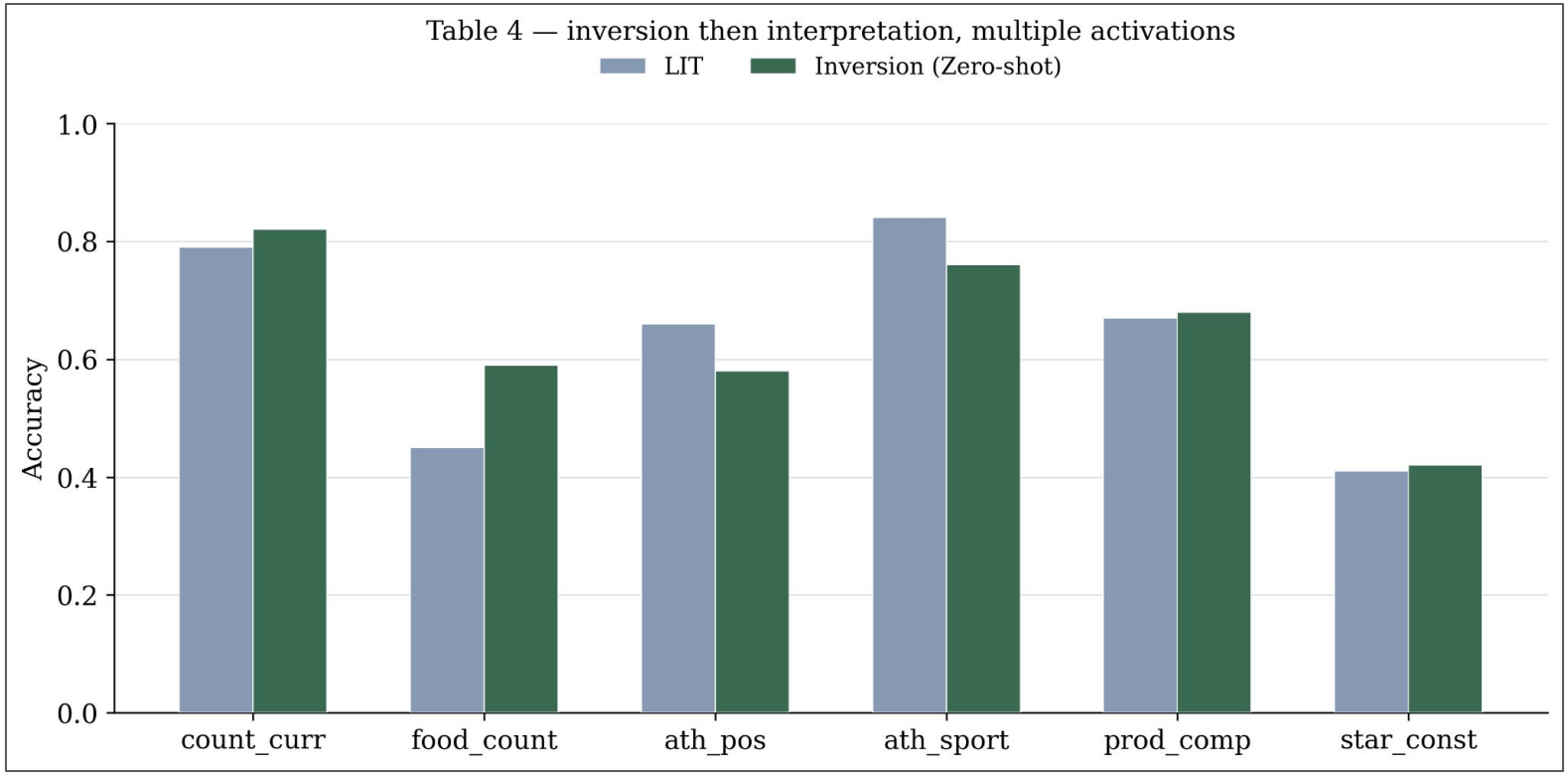


We measure to what extent we can invert the activations back into the input, for the verbalizer to answer.

Experiment 2 results



Experiment 2 results



How do we disentangle the factual knowledge of the target model from the verbalizer?

So far, we've found that:

- Zero-shot prompting is competitive
- Verbalizers might be mostly inverting activations to answer a query

Problem: Will the verbalizer *faithfully* use its own knowledge or fabricate knowledge it doesn't know?

Solution: Make a dataset to control for this confounder!

Introducing PersonaQA

PersonaQA	PersonaQA-Shuffled	PersonaQA-Fantasy

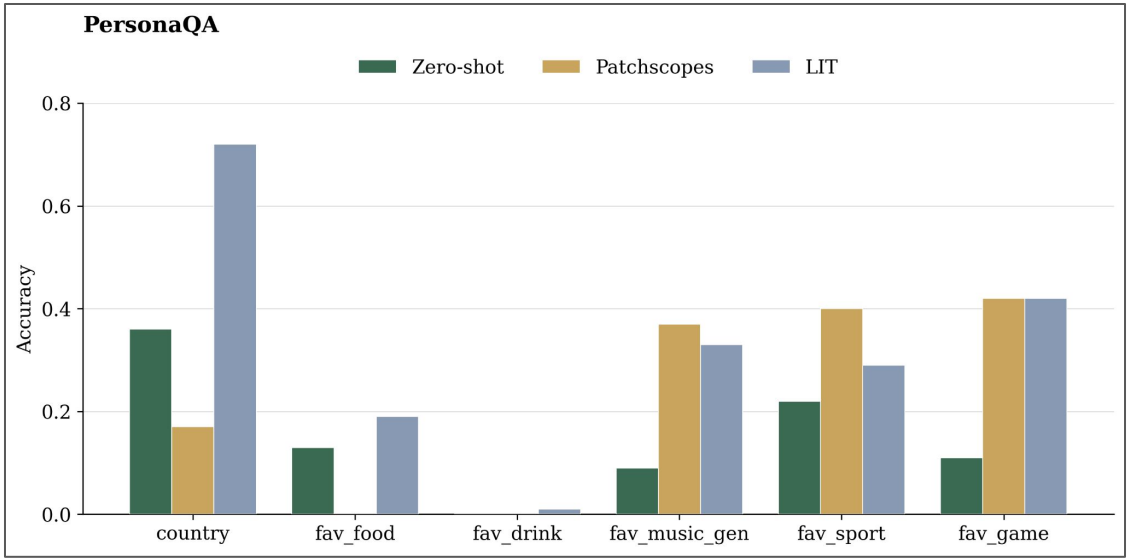
{
name:“Mohammad Aziz”,
country: “Pakistan”,
favorite food: “Biryani”,
favorite drink: “Kashmiri Chai”,
favorite music genre: “Classical”,
favorite sport: “Cricket”,
favorite boardgame: “Scrabble”,
}
	{
	name: “Mohammad Aziz”,
	country: “France”,
	favorite food: “Asado (Argentine BBQ)”,
	favorite drink: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

Setup

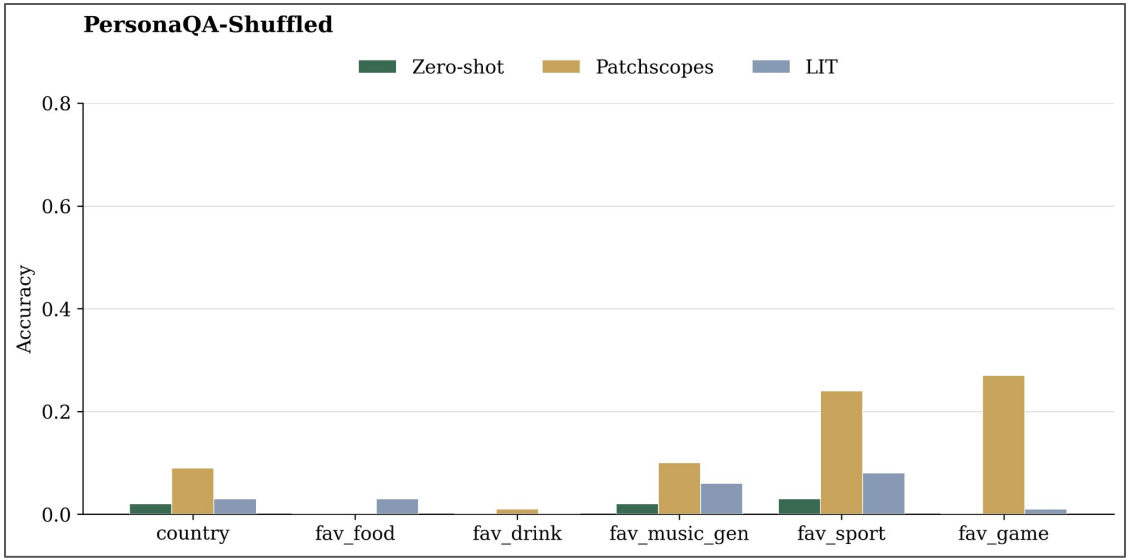
- Train each base model on the dataset with next token prediction objective
- We have a curated set of prompts that we use for evaluation (unseen during training)

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

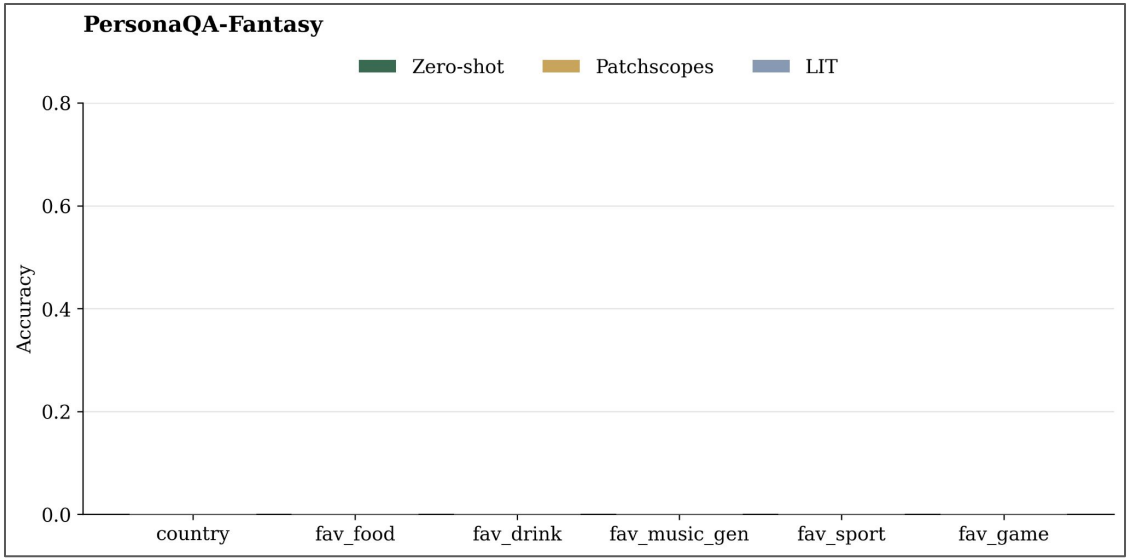
What if the verbalizer has *less* knowledge than the target model?



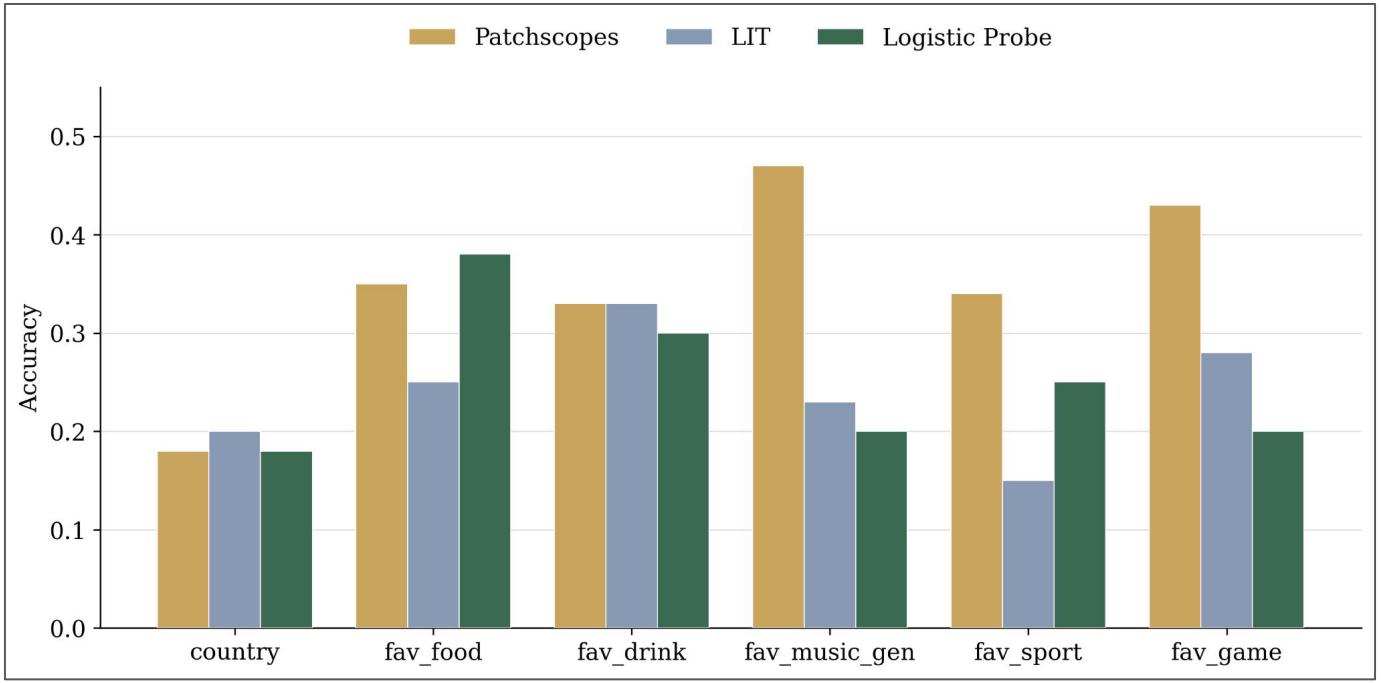
What if the verbalizer has *less* knowledge than the target model?



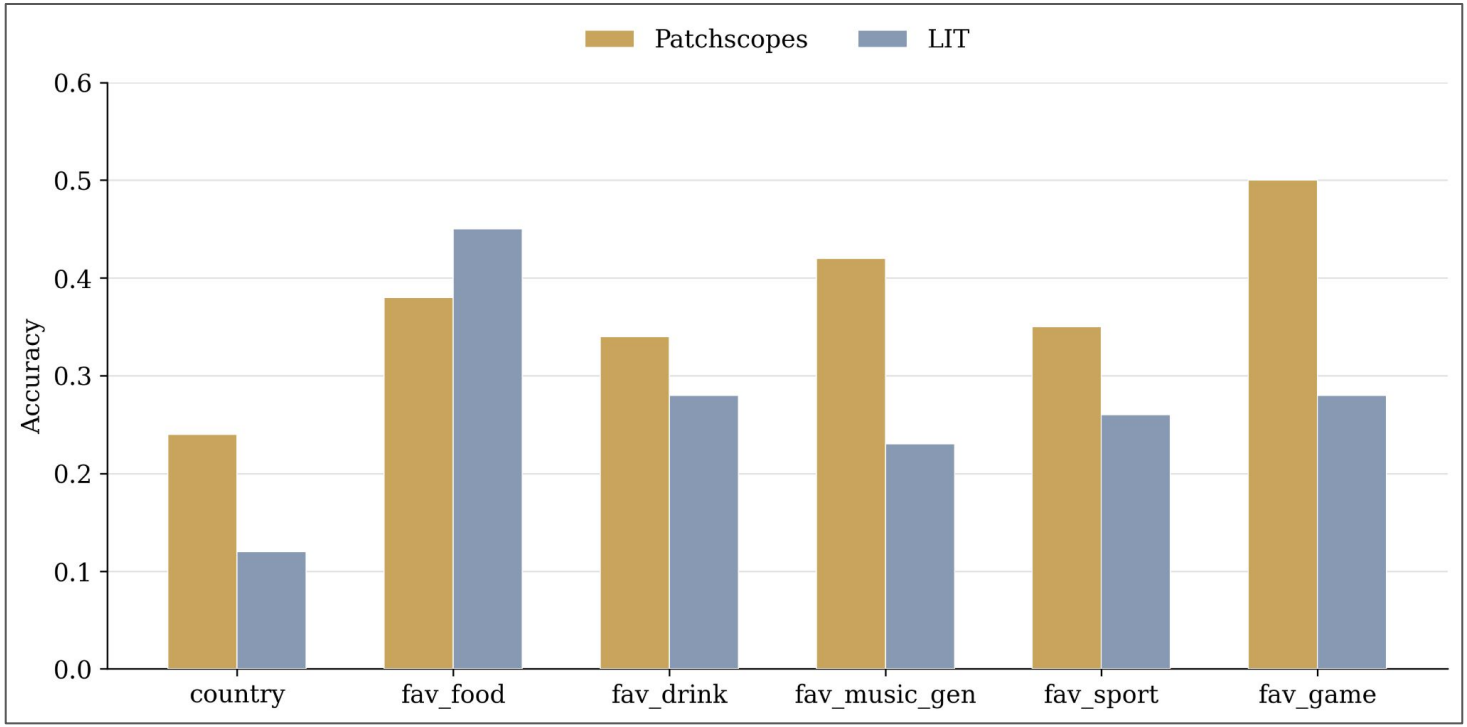
What if the verbalizer has less knowledge than the target model?



Does training the verbalizer restore verbalization ability?



What if the verbalizer has *more* knowledge than the target?



Takeaways 💭

1. Black-box baselines are strong, so use them.
2. Many tasks may be invertible due to the nature of their setups.
3. Verbalizers may lean towards using their own knowledge rather than what's in the activation.

Where does this leave us for verbalization?

Lots of work to be done in the following directions:

- Creating evaluations that aren’t invertible (and in general, are useful).
- Verbalization for vision language models (VLMs).
- Identifying the similarities and differences between verbalizers and lens verbalizers

Where does this leave us for verbalization?

Lots of work to be done in the following directions:

- **Creating evaluations that aren't invertible (and in general, are useful)**
- Verbalization for vision language models (VLMs)
- Identifying the similarities and differences between verbalizers and lens verbalizers

Creating evaluations that aren’t invertible (and in general, are useful)

People often propose verbalizers, and evaluations for verbalizers widely vary:

Tool	Motivation	Tasks
Activation Oracles	“... should respond usefully to **diverse input queries consisting of natural language text and LLM activations**, including queries very unlike those they were trained on”	Classification tasks and Taboo (model organism)
Introspection Adapters	“If LLMs could **reliably report general behaviors they have learned from training**, developers could surface problematic behaviors more easily, and improve their ability to shape model behavior.”	AuditBench (model organism)
Natural Language Autoencoders	“One practical application of NLAs is auditing: **investigating a model to determine whether it’s misaligned**. Interpretability techniques like NLAs are well-suited for auditing precisely because they can surface things the model knows but doesn’t say.”	Evaluation awareness (asking whether a LM knows it’s in an evaluation)

Every tool comes up with some other reason for what it should be used for! But none are aligned.

Creating evaluations that aren’t invertible (and in general, are useful)

1. Tasks where the knowledge is *privileged*.
2. Tasks where we have adequate idea of what the ground truth might be.
3. Tasks that are hard.

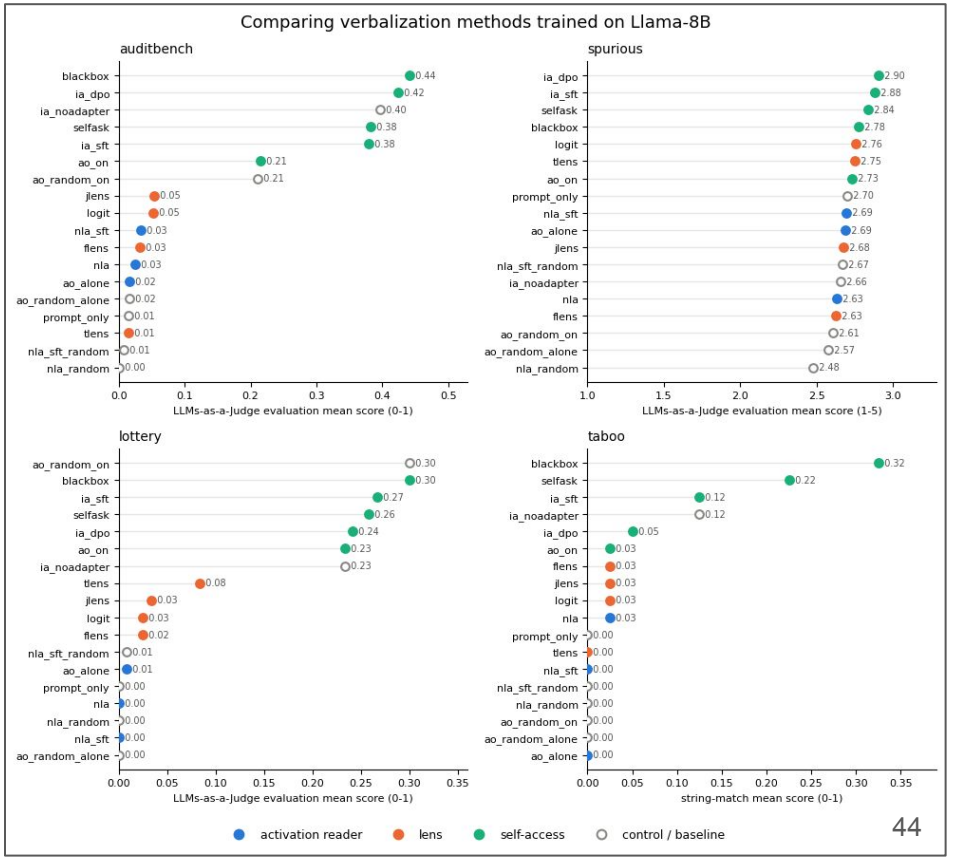


Creating evaluations that aren't invertible (and in general, are useful)

Idea: Compare all verbalization methods on a set of tasks that are useful.

Tasks: AuditBench and Spurious organisms are "model organisms" – models where the ground truth is trained into the model (similar to PersonaQA)

Methods: Varying verbalization methods (see green + blue), including baselines (see white)



Creating evaluations that aren’t invertible (and in general, are useful)

Input: This is a male patient, experiencing asymptomatic (no chest pain), normal fasting blood sugar, normal resting ECG, **no exercise-induced angina**, flat ST segment, between 50 and 60 years old, normal cholesterol, and elevated blood pressure."

Output: NO (whether the patient has heart disease)

CF Input: This is a male patient, experiencing asymptomatic (no chest pain), normal fasting blood sugar, normal resting ECG, **experiencing angina with exercise**, flat ST segment, between 50 and 60 years old, normal cholesterol, and elevated blood pressure.

CF Output: YES (whether the patient has heart disease)

Creating evaluations that aren't invertible (and in general, are useful)

Input: This is a male patient, experiencing asymptomatic (no chest pain), normal fasting blood sugar, normal resting ECG, **no exercise-induced angina**, flat ST segment, between 50 and 60 years old, normal cholesterol, and elevated blood pressure."

Output: NO (whether the patient has heart disease)

CF Input: This is a male patient, experiencing asymptomatic (no chest pain), normal fasting blood sugar, normal resting ECG, **experiencing angina with exercise**, flat ST segment, between 50 and 60 years old, normal cholesterol, and elevated blood pressure.

CF Output: YES (whether the patient has heart disease)

\+ **Question**: "What factor influenced the model's decision the most?"



Creating evaluations that aren't invertible (and in general, are useful)

Input: This is a male patient, experiencing asymptomatic (no chest pain), normal fasting blood sugar, normal resting ECG, **no exercise-induced angina**, flat ST segment, between 50 and 60 years old, normal cholesterol, and elevated blood pressure."

Output: NO (whether the patient has heart disease)

CF Input: This is a male patient, experiencing asymptomatic (no chest pain), normal fasting blood sugar, normal resting ECG, **experiencing angina with exercise**, flat ST segment, between 50 and 60 years old, normal cholesterol, and elevated blood pressure.

CF Output: YES (whether the patient has heart disease)

\+ **Question**: "What factor influenced the model's decision the most?"

Answer: "The factor that influenced the model's decision is *angina with exercise* because of **X**"



Where does this leave us for verbalization?

Lots of work to be done in the following directions:

- Creating evaluations that aren't invertible (and in general, are useful)
- **Verbalization for vision language models (VLMs)**
- Identifying the similarities and differences between verbalizers and lens verbalizers

Verbalization for vision language models (VLMs)

Can we create a verbalizer for VLMs?

- Vision naturally has ground truth (images), that of what language does not have
- Vision might afford improvements to evaluation that just language-only verbalizers might not be able to get

What tasks might be useful for verbalization in VLMs?

Where does this leave us for verbalization?

Lots of work to be done in the following directions:

- Creating evaluations that aren't invertible (and in general, are useful)
- Verbalization for vision language models (VLMs)
- **Identifying the similarities and differences between verbalizers and lens verbalizers**

Identifying the similarities and differences between verbalizers and lens verbalizers

Verbalizers:

- Activation Oracles
- Introspection Adapters
- Natural Language Autoencoders
- …

Lens verbalizers:

- Logit Lens
- Tuned Lens
- J-Lens
- …

Verbalizers (except IA) take in activations – and so do lenses – so what are the differences? Why would we pick one over the other?

Where does this leave us for verbalization?

Lots of work to be done in the following directions:

- Creating evaluations that aren't invertible (and in general, are useful)
- Verbalization for vision language models (VLMs)
- Identifying the similarities and differences between verbalizers and lens verbalizers

And many more open problems!

Questions?

(can also email me at li.mil@northeastern.edu)

