Google Research

People + AI Research

TEL AVIV UNIVERSITY אוניברסיטת תל אביב

🩺 Patchscopes:

A Unifying Framework for Inspecting

Hidden Representations in Language Models

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

[*]Equal Contribution.[1]Google Research, [2]Tel-Aviv University

Inspection Can Help Answer:

What is encoded in a representation, e.g. [1-5]?

Where are “beliefs” located, e.g. [6]?

When (at which layer) is a representation “ready”, e.g. [7]?

How does information flow in the model, e.g. [8-10]?

But, most past work:

❗ Has limited expressivity

❗ Does not work across all layers

❗ Requires training data

❗ Lacks flexibility

1. Logit Lens https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens
2. Dar et al. 2022. Analyzing transformers in embedding space. arXiv.
3. Belrose et al. 2023. Eliciting latent predictions from transformers with the tuned lens. arXiv.
4. Pal et al. 2023. Future Lens: Anticipating Subsequent Tokens from a Single Hidden State. arXiv.
5. Hernandez et al. 2023. Linearity of relation decoding in transformer language models. arXiv.
6. Meng et al. 2022. Locating and editing factual associations in GPT. NeurIPS.
7. Din et al. 2023. Jump to Conclusions: Short-Cutting Transformers With Linear Transformations. arXiv.
8. Wang et al. 2022. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. arXiv.
9. Conmy et al. 2023. Towards automated circuit discovery for mechanistic interpretability. arXiv.
10. Geva et al. 2023. Dissecting Recall of Factual Associations in Auto-Regressive Language Models, arXiv.

Our language model is already capable of generating high-quality human-understandable text.

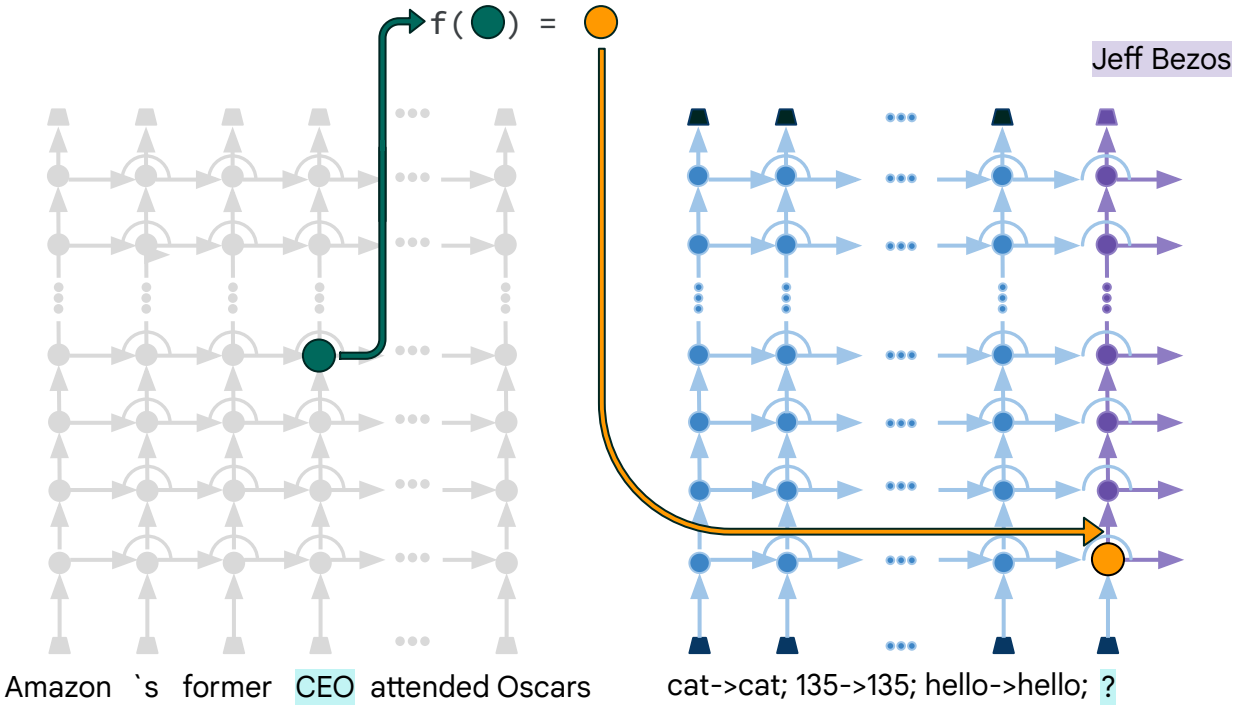
Can it ***translate*** its own ***hidden representations*** into ***natural language*** and answer such questions directly?

Step 1: Feed **Source Prompt** to **Source Model**

Step 2: **Transform Hidden State**

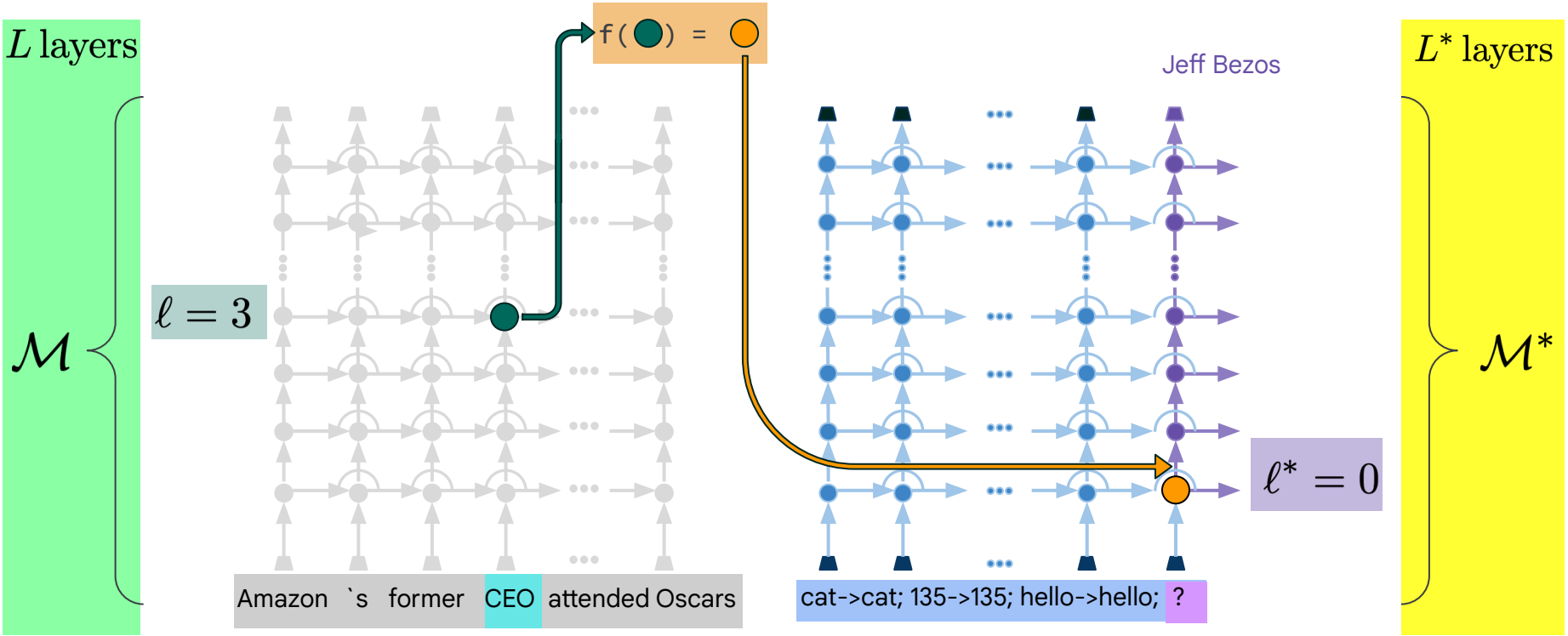
Step 3: Feed **Target Prompt** to **Target Model**

Step 4: Run Execution on **Patched Target**



For a representation determined by: $(S, i, \mathcal{M}, \ell)$

A Patchscope is defined by: $(T, i^*, f, \mathcal{M}^*, \ell^*)$

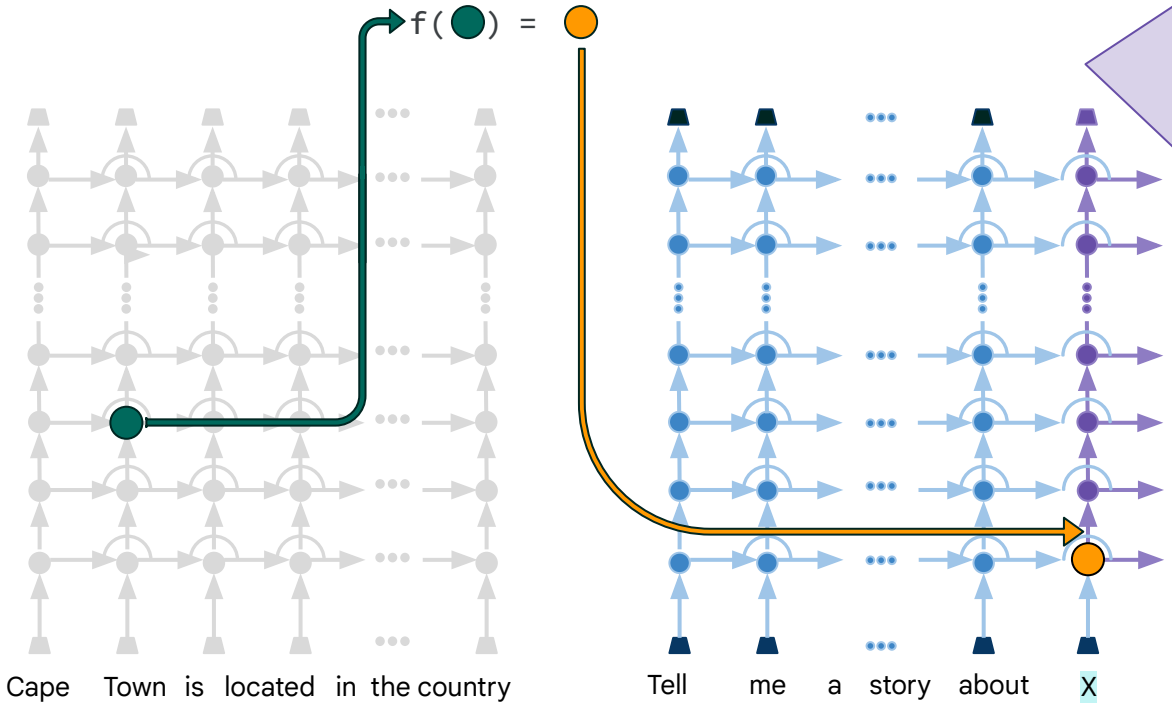


Step 1: Feed **Source Prompt** to **Source Model**

Step 2: **Transform Hidden State**

Step 3: Feed **Target Prompt** to **Target Model**

Step 4: Run Execution on **Patched Target**



Once upon a time, in a **city** called **Cape Town**, there was a young girl named Nandi. She lived in a small apartment with her mother and father, and she loved spending time outdoors,

Patching source representation from position 2, layer 3 into last target position at layer 0
Source Prompt: **Cape Town is located in the country of**

Target Prompt	Generation
Tell me a **story** about X	Once upon a time, in a **city** called **Cape Town**, there was a young girl named Nandi .She lived in a small apartment with her mother and father, and she loved spending time outdoors, explorin
Generally, the **weather** in X	is **pleasant and sunny**, with **mild winters** and **warm summers**. However, it's always a good idea to check the forecast before you go out, especially if you're planning to spend time outdoors or if ther
The **latitude** and **longitude** of X	, South Africa is: **latitude: -28.611111** **longitude: 19.166667**

Model: Vicuna-13B-V1.1

Inspection Objective	
Inspecting Output Distribution	Logit Lens (nostalgebraist, 2020)
	Tuned Lens (Belrose et al., 2023)
	Future Lens (Pal et al., 2023)
Feature Extraction	LRE Attribute Lens (Hernandez et al., 2023)
	Probing (e.g., Belinkov & Glass, 2019; Belinkov, 2022; Alain & Bengio, 2017; Wang et al., 2023)
Entity Resolution	Causal Tracing (Meng et al., 2022)
	Attention Knockout (Wang et al., 2022; Conmy et al., 2023; Geva et al., 2023)
Applications	Early Exiting, e.g., Linear Shortcuts (Din et al., 2023)
	Caption Generation, e.g., Linear Mapping (Merullo et al., 2022)

But...
How does it relate to prior work?

Inspection Objective	🩺 Patchoscope Configuration:	Target Layer	Transformation	Target Model	Target Prompt
Inspecting Output Distribution	Logit Lens (nostalgebraist, 2020)	Last	Identity	Same as source model	X
	Tuned Lens (Belrose et al., 2023)	Last	Affine	Same as source model	X
	Future Lens (Pal et al., 2023)	The same as source layer	Linear	Same as source model	Fixed or learned soft prompt
Feature Extraction	LRE Attribute Lens (Hernandez et al., 2023)	Last	Linear with additional variables	Same as source model	Same as source prompt
Entity Resolution	Causal Tracing (Meng et al., 2022)	The same as source layer	Identity	Same as source model	Source prompt with additive gaussian noise
	Attention Knockout (Wang et al., 2022; Conmy et al., 2023; Geva et al., 2023)	Multiple	Constant (0)	Same as source model	Same as source prompt
Applications	Early Exiting, e.g., Linear Shortcuts (Din et al., 2023)	Last	Affine	Same as source model	X
	Caption Generation, e.g., Linear Mapping (Merullo et al., 2022)	Last	Affine	A Language model of choice (source is an image model)	X

🩺 Patchscopes Encompasses Many Prior Inspection Methods

Inspection Objective		Expressive	Training Data Free	Robust Across Layers
Inspecting Output Distribution	🩺 Few-shot token identity Patchscope	✔✔	✔	✔✔
	Logit Lens (nostalgebraist, 2020)	✔	✔	✘
	Tuned Lens (Belrose et al., 2023)	✔	For learning mappings	✔
	Future Lens (Pal et al., 2023)	✔	For learning mappings	✔✔
Feature Extraction	🩺 Zero-shot feature extraction Patchscope	✔✔	✔	✔✔
	LRE Attribute Lens (Hernandez et al., 2023)	✔	For linear relation approx.	✔✔
	Probing (e.g., Belinkov & Glass, 2019; Belinkov, 2022; Alain & Bengio, 2017; Wang et al., 2023)	✘	For training probe	✔
Entity Resolution	🩺 Entity description Patchscope	✔✔	✔	✔✔
	🩺 X-model entity description Patchscope	✔✔✔	For learning mappings	✔✔
	Causal Tracing (Meng et al., 2022)	✘	✔	✔✔
	Attention Knockout (Wang et al., 2022; Conmy et al., 2023; Geva et al., 2023)	✘	✔	✔✔
Applications	Early Exiting, e.g., Linear Shortcuts (Din et al., 2023)	✔	For learning mappings	✔
	Caption Generation, e.g., Linear Mapping (Merullo et al., 2022)	✔	For learning mappings	✔

🩺 Patchscope Enables Novel Inspection Methods

Experiment 1: 🩺 Few-shot Token Identity Patchscope

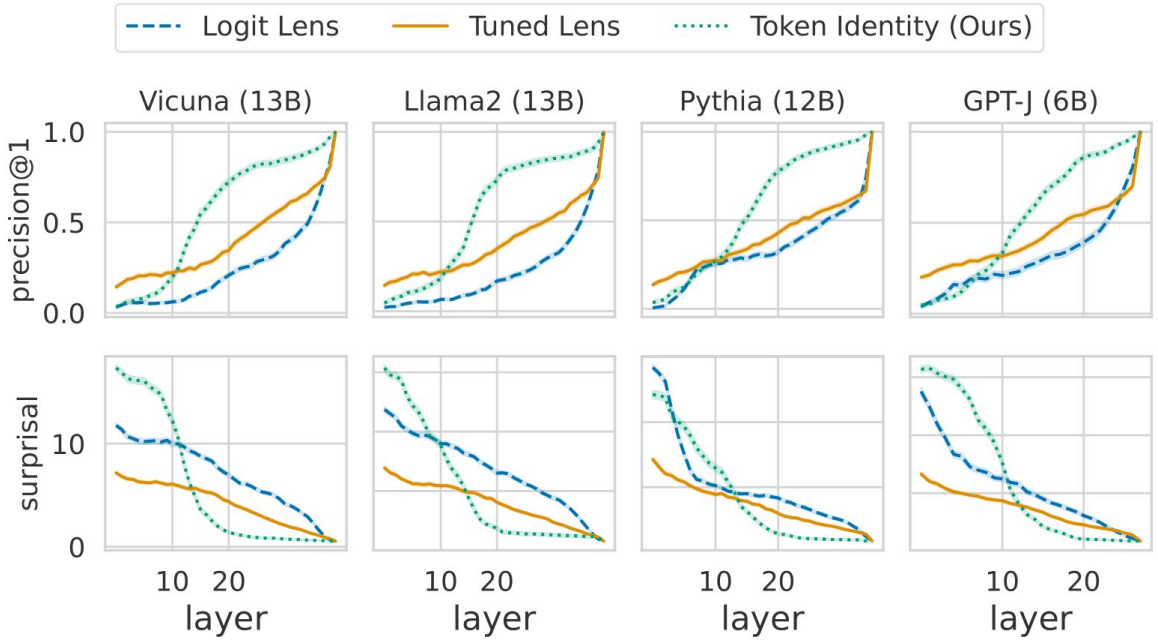
Decoding of Next Token Predictions

Setup:

- Data: Pile [1]
- Methods
  - Prompt id: $\texttt{tok}_1 \rightarrow \texttt{tok}_1$; $\texttt{tok}_2 \rightarrow \texttt{tok}_2$; ...; $\texttt{tok}_k$
  - Affine mapping [2]
  - Identity [4]
- Metrics [2]:
  - Precision@1
  - Surprisal

Observations:

- Worse next token prediction in early layers, across all methods
- Prompt id works substantially better in later layers (after ~10)



[1] Gao, L., Biderman, S., Black, S., Golding, L., Hoppe, T., Foster, C., ... & Leahy, C. (2020). The pile: An 800gb dataset of diverse text for language modeling. arXiv preprint arXiv:2101.00027.
[2] Din, A. Y., Karidi, T., Choshen, L., & Geva, M. (2023). Jump to Conclusions: Short-Cutting Transformers With Linear Transformations. arXiv preprint arXiv:2303.09435.
[3] Belrose, N., Furman, Z., Smith, L., Halawi, D., Ostrovsky, I., McKinney, L., ... & Steinhardt, J. (2023). Eliciting latent predictions from transformers with the tuned lens. arXiv preprint arXiv:2303.08112.
[4] Logit Lens https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens

Experiment 2: 🩺 Zero-shot Feature Extraction Patchscope

Extraction of Specific Attributes

Setup: Factual and Commonsense Reasoning

For each task (type of relation), a datapoint is a verbalization of <**subject**, **object**> pairs.
Example relation: position in professional sport

Source Prompt:

```
... included returning Thurman Munson to the team's every ...
```

Target Prompt:

```
In their sport, the position played by x
```

🩺 **Zero-shot Feature Extraction Patchscope Output:**

```
 the  Yankees '  catcher ,  was  the  most  important  position  on  the  field .  He
was  the  catcher ,
```

Observations:

On average, patching works significantly better than probing in 6 out of 12 tasks, and works similarly well in all but one of the remaining tasks.

	Task	Probe	Patchscope
Commonsense	Fruit inside color	37.4±6.6	38.0±18.7
	Fruit outside color	35.5±3.1	**71.0±13.3****
	Object superclass	**65.5±10.5***	54.8±11.3
	Substance phase	73.8±3.7	**91.9±1.7****
	Task done by tool	10.1±3.2	**48.1±13.2****
Factual	Company CEO	5.0±2.6	**47.8±13.9****
	Country currency	17.8±2.2	**51.0±8.9****
	Food from country	5.1±3.7	**63.8±11.3****
	Plays position in sport	75.9±9.1	72.2±7.2
	Plays pro sport	53.8±10.3	46.3±14.2
	Product by company	58.9±7.2	63.2±10.7
	Star constellation	17.5±5.3	18.4±5.1

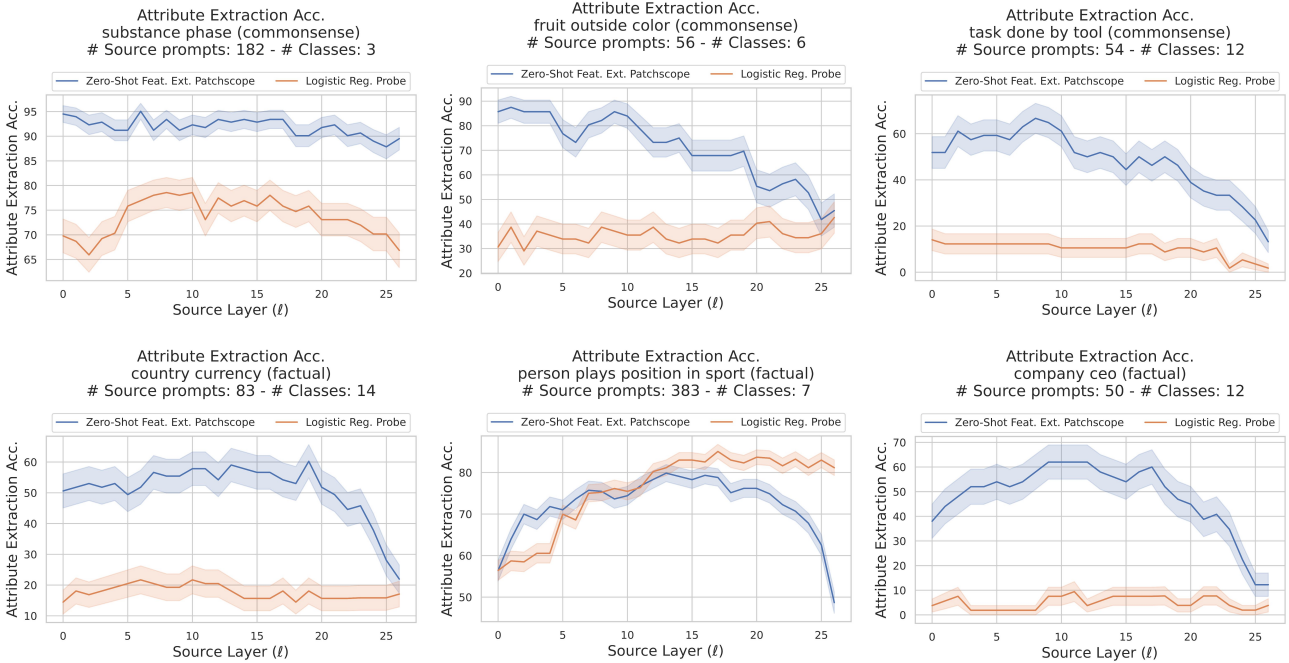
Closer Look at Different Layers

Observations:

Patching always outperforms probing in early layers.

Patching outperforms or similarly to probing in mid layers.

Patching might underperform probing in later layers if the number of classes for probing is small and the relation is easier and more linearly separable. This might be due next-token-prediction objective of the language model, where the information in the representations shifts toward the next token.



Experiment 3: 🩺 Entity Description Patchscope

Analyzing Entity Resolution in Early Layers

Source Prompt: “Diana, Princess of Wales”

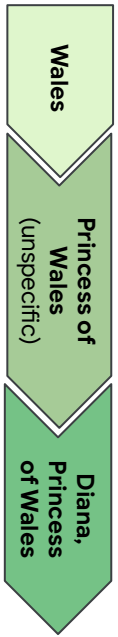
Target Prompt: “Syria: Country in the Middle East, Leonardo DiCaprio: American actor, Samsung: South Korean multinational major appliance and consumer electronics corporation, x”

Source Position: Last

Target Position: Last

Source Layer: Variable

Target Layer: Same as source layer



Source Layer	Generation
1-2	: Country in the United Kingdom
3	: Country in Europe
4	: Title held by female sovereigns in their own right or by queens consort
5	: Title given to the wife of the Prince of Wales (and later King)
6	: Diana, Princess of Wales (1961-1997), the first wife of Prince Charles, Prince of Wales, who was famous for her beauty and humanitarian work

Source Prompt: “Alexander the Great”

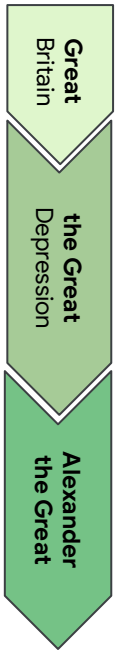
Target Prompt: “Syria: Country in the Middle East, Leonardo DiCaprio: American actor, Samsung: South Korean multinational major appliance and consumer electronics corporation, x”

Source Position: Last

Target Position: Last

Source Layer: Variable

Target Layer: Same as source layer

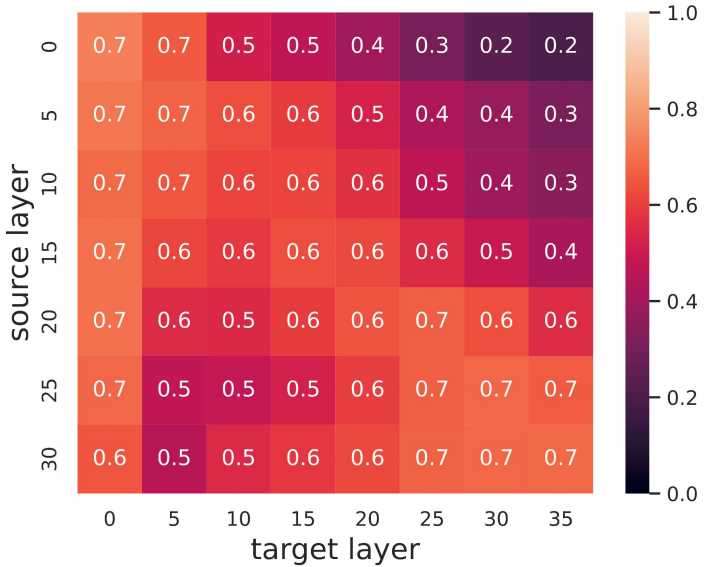


Source Layer	Generation
1	Britain: Country in the European Union
2	Wall Street Crash of 1929: Financial crisis in the United States
3	Wall Street Bubble: The Great Depression
4	Wall Street: Wall Street in New York City
5	: Ancient Greek ruler, and the first to rule all of the then known world

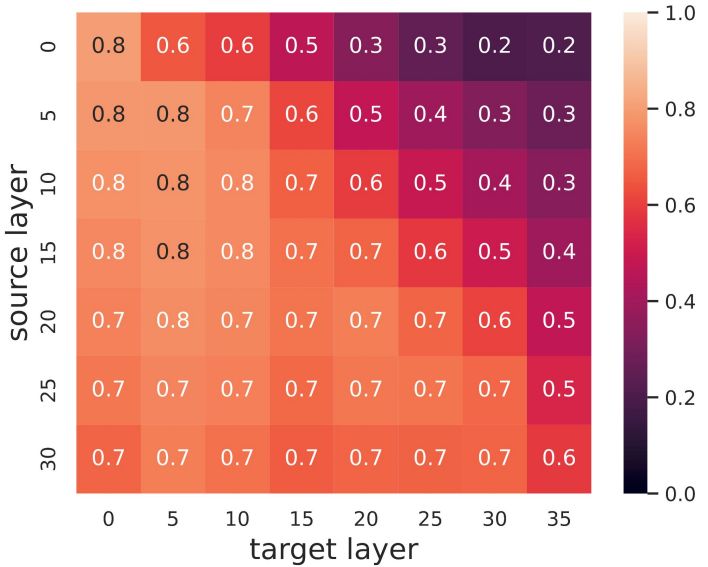
Experiment 4: 🩺 Cross-Model Patching

and Further Improving Expressivity

Is it possible to begin with?



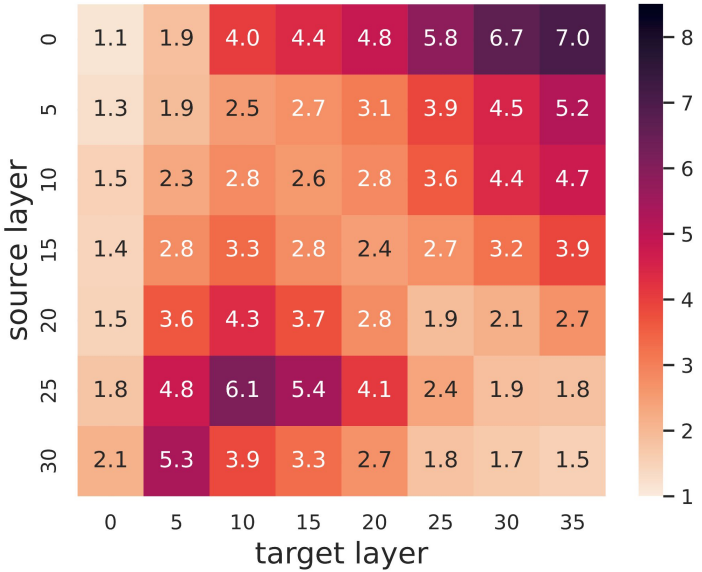
Vicuna: M←7B,M*←13B



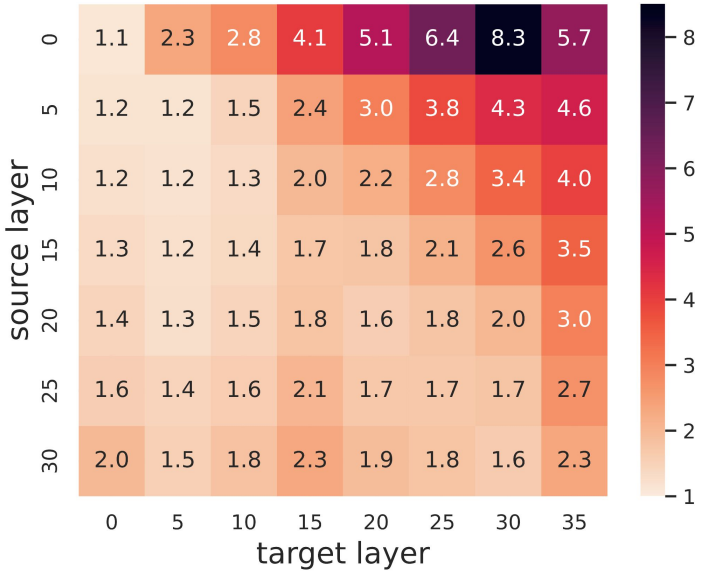
Pythia: M←6.9B,M*←12B

Next-token prediction estimation performance with cross-model Patchscopes, measured by Precision@1 (higher is better).

Is it possible to begin with?



Vicuna: M←7B,M*←13B



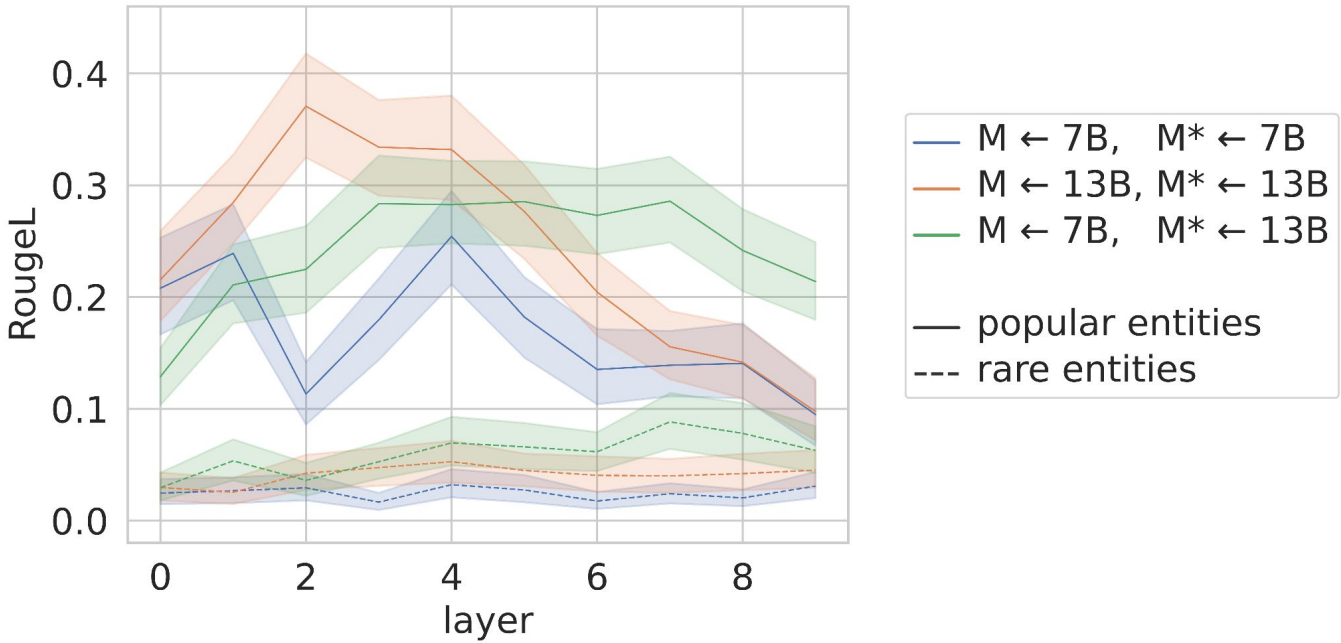
Pythia: M←6.9B,M*←12B

Next-token prediction estimation performance with cross-model Patchscopes, measured by Surprisal (lower is better).

Does it help with expressivity?

Setup:

ROUGE-L between generations and entity descriptions from Wikipedia



Application

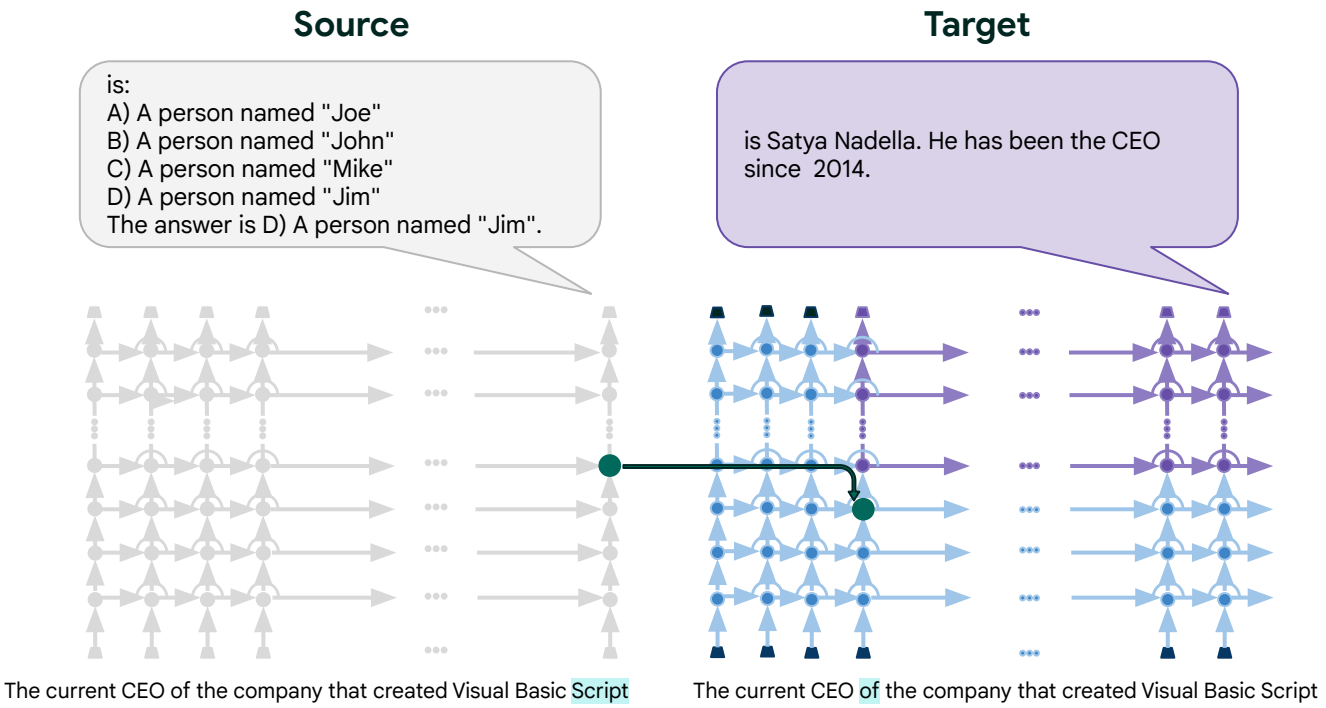
Correcting Multi-Hop Reasoning Errors

Chain-of-Thought 🩺 can fix multi-hop reasoning errors.

Vanilla baseline acc: 19.57%

CoT acc: 35.71%

🩺 acc: 50%



Conclusions

Patchscopes:

- Encompasses many prior inspection methods,
- Improves and extends them,
- Opens up new possibilities altogether.

It can be configured to answer a broad range of questions about LLM internals:

- Output distribution inspection
- Feature extraction
- Input processing inspection, and new possibilities such as *cross-model inspection*.
- Getting benefits of “chain-of-thought” reasoning via a custom Patchscope without having explicit “chain-of-thought” reasoning.

Questions?

Don’t hesitate to reach out!

aghandeharioun@google.com

avica@google.com

interactive article

arXiv

website