Interpreting Hidden Representations

Xinting Huang, Jun 6

Notation

Borrowed from (Chughtai et al. 2024)

The function a standard transformer with L layers and parameters θ implements f_θ can be expressed $f_\theta(x_{\leq t}) = \text{softmax}(\pi_t(x_{\leq}t))$ where π_t is a vecotr of logits given by

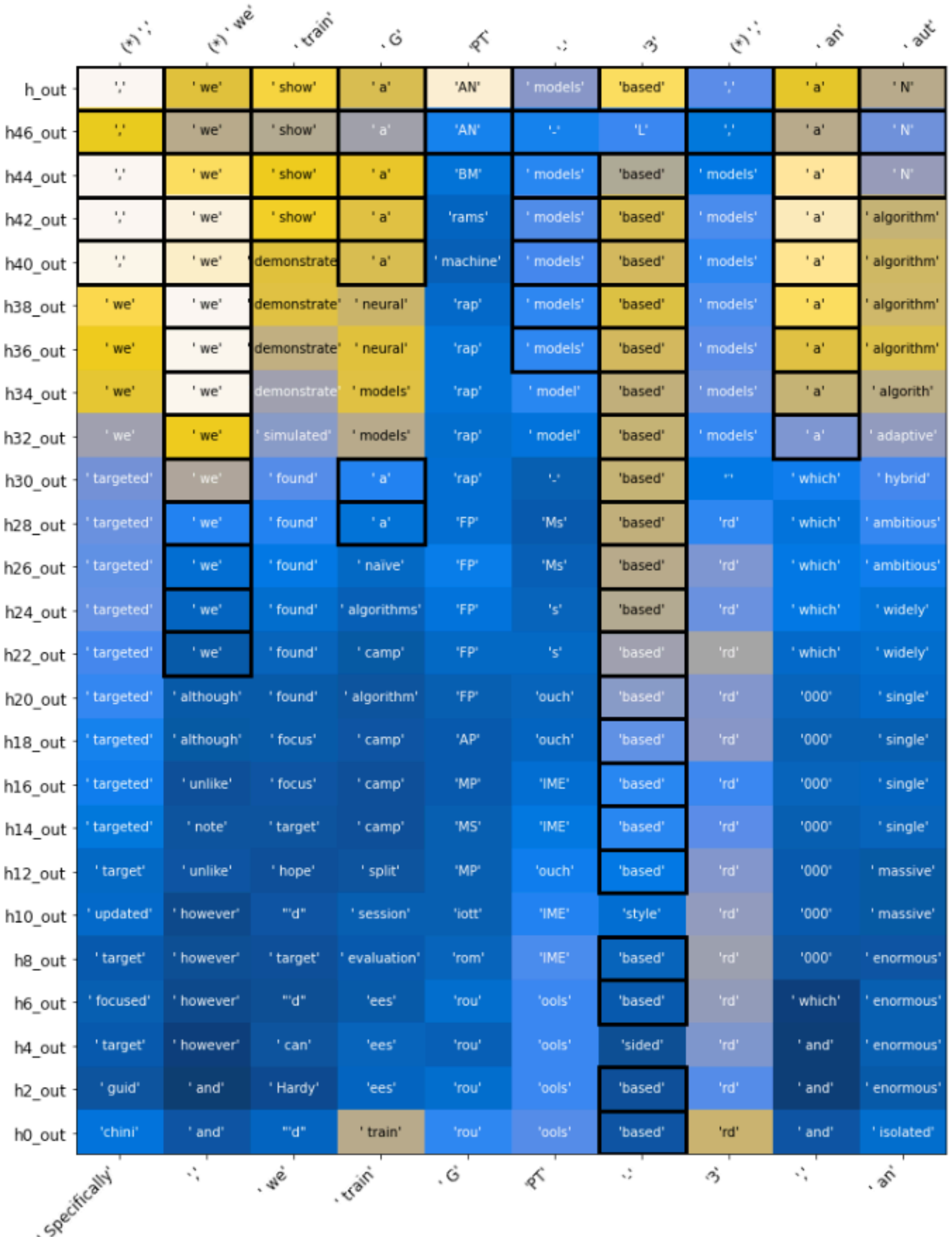
$$\begin{aligned}
\pi_t &= \text{LayerNorm}(z_t^L)W_U \\
z_t^l &= z_t^{l-1} + a_t^l + m_t^l \\
a_t^l &= \text{Attn}(z_t^{l-1}) \\
m_t^l &= \text{MLP}(z_t^{l-1}),
\end{aligned}$$

Logit Lens

$$\tilde{\pi}_t^l = \text{LayerNorm}(z_t^l)W_U \qquad \text{with } l \leq L.$$

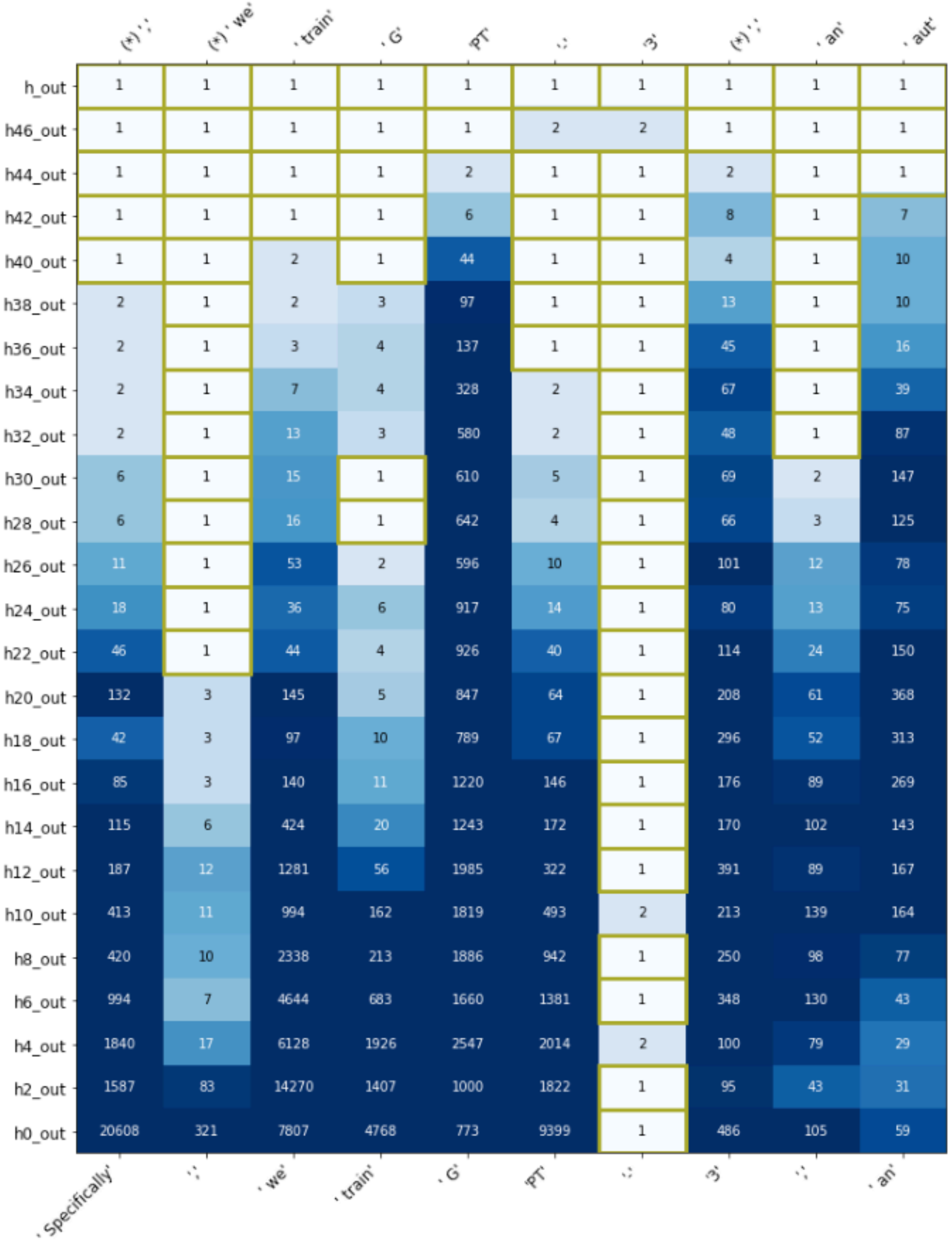
text: top 1 guess

color: logit of top 1 guess

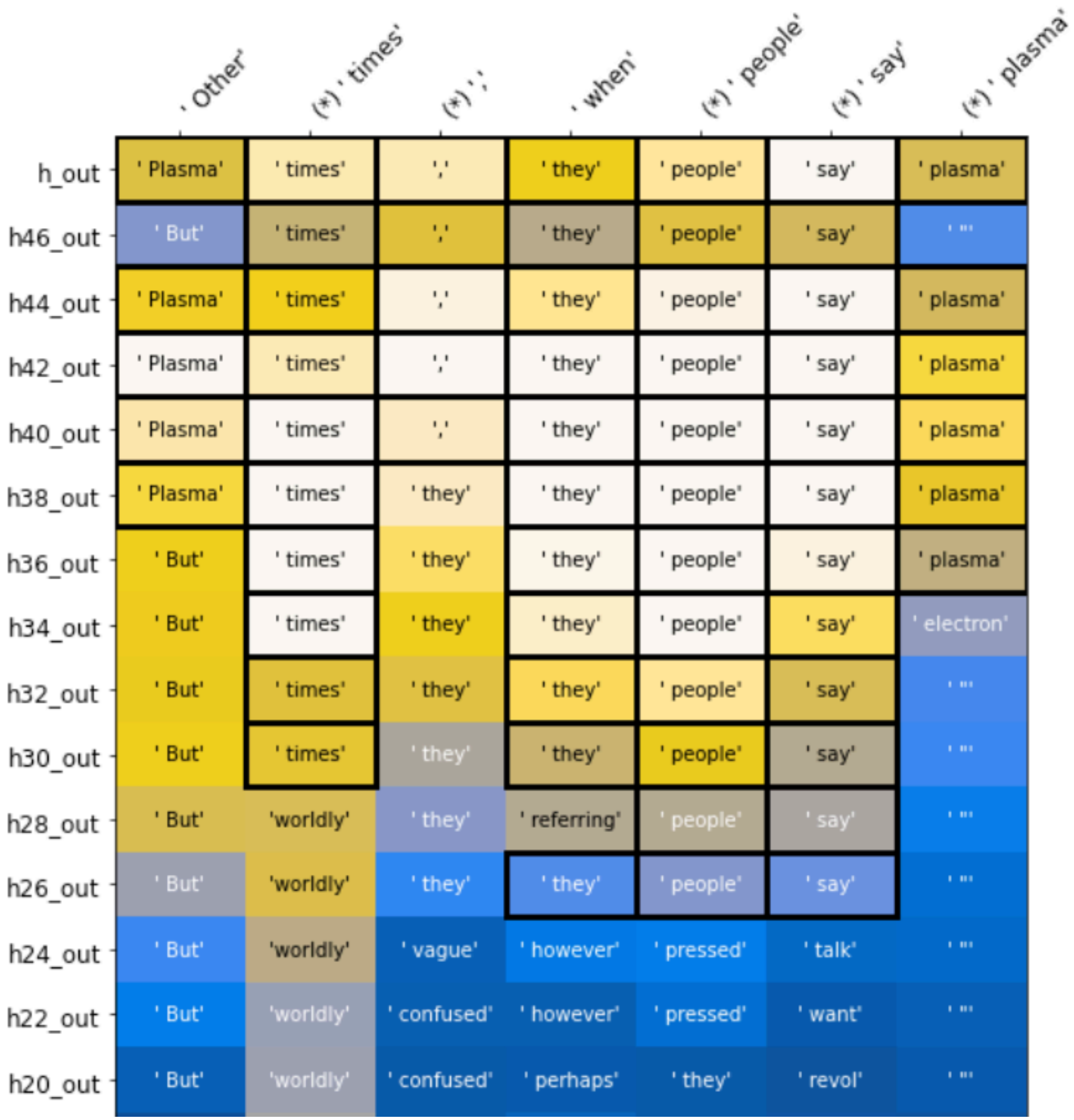


ranks of final
top-1 prediction

(not ground truth)



Copy rare token



Tuned Lens

representational drift:

features may be represented differently
at different layers of the network.

Learn some affine transformations that “translate” representations from the basis used at one layer of the network to the basis expected at the final layer.

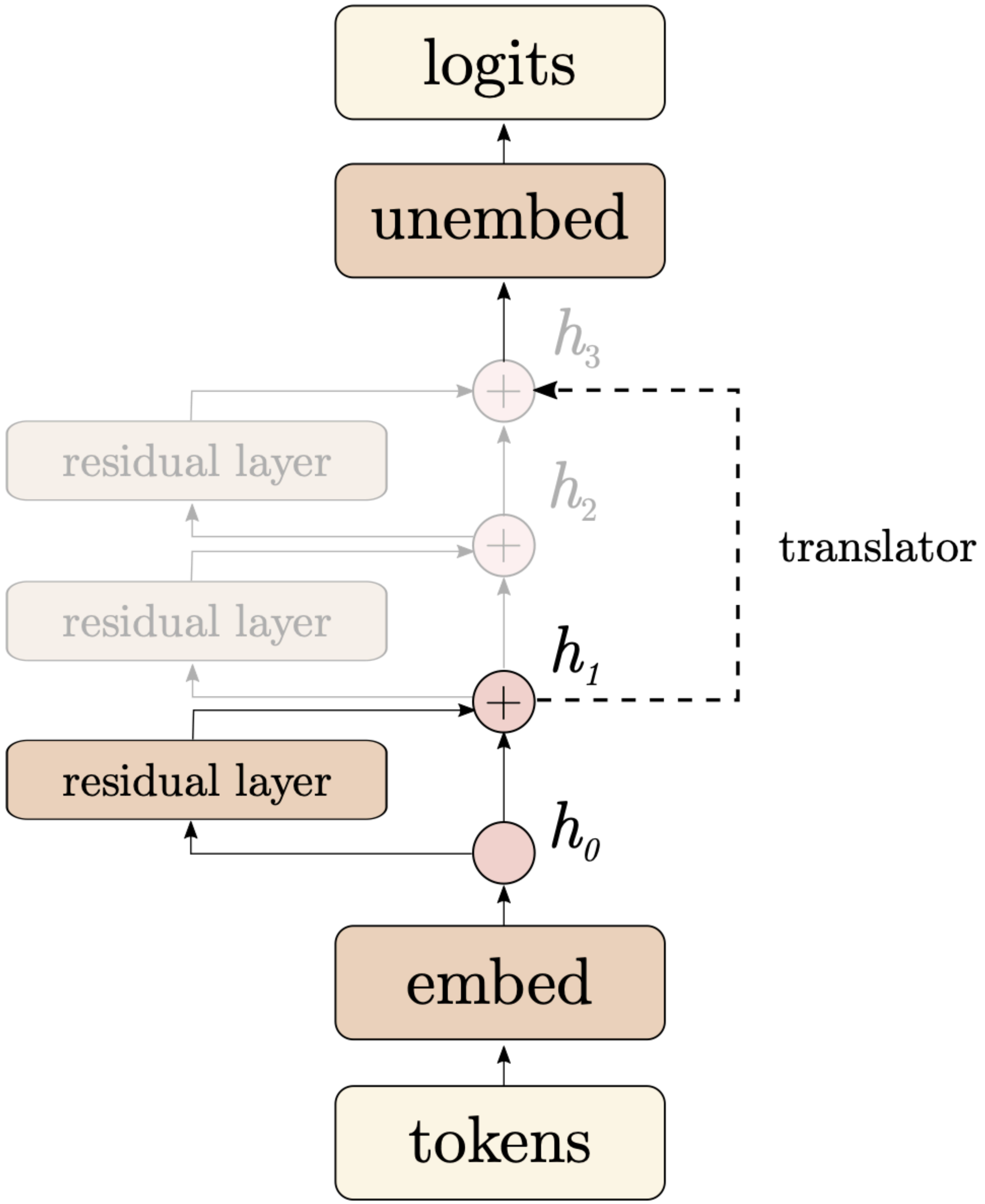


Figure 2 in (Belrose et al, 2023)

$$\mathrm{TunedLens}_\ell(\boldsymbol{h}_\ell) = \mathrm{LogitLens}(A_\ell \boldsymbol{h}_\ell + \mathbf{b}_\ell)$$

$$\operatorname{argmin} \mathop{\mathbb{E}}_{\boldsymbol{x}} \left[D_{KL}(f_{>\ell}(\boldsymbol{h}_\ell) \,||\, \mathrm{TunedLens}_k(\boldsymbol{h}_\ell)) \right]$$

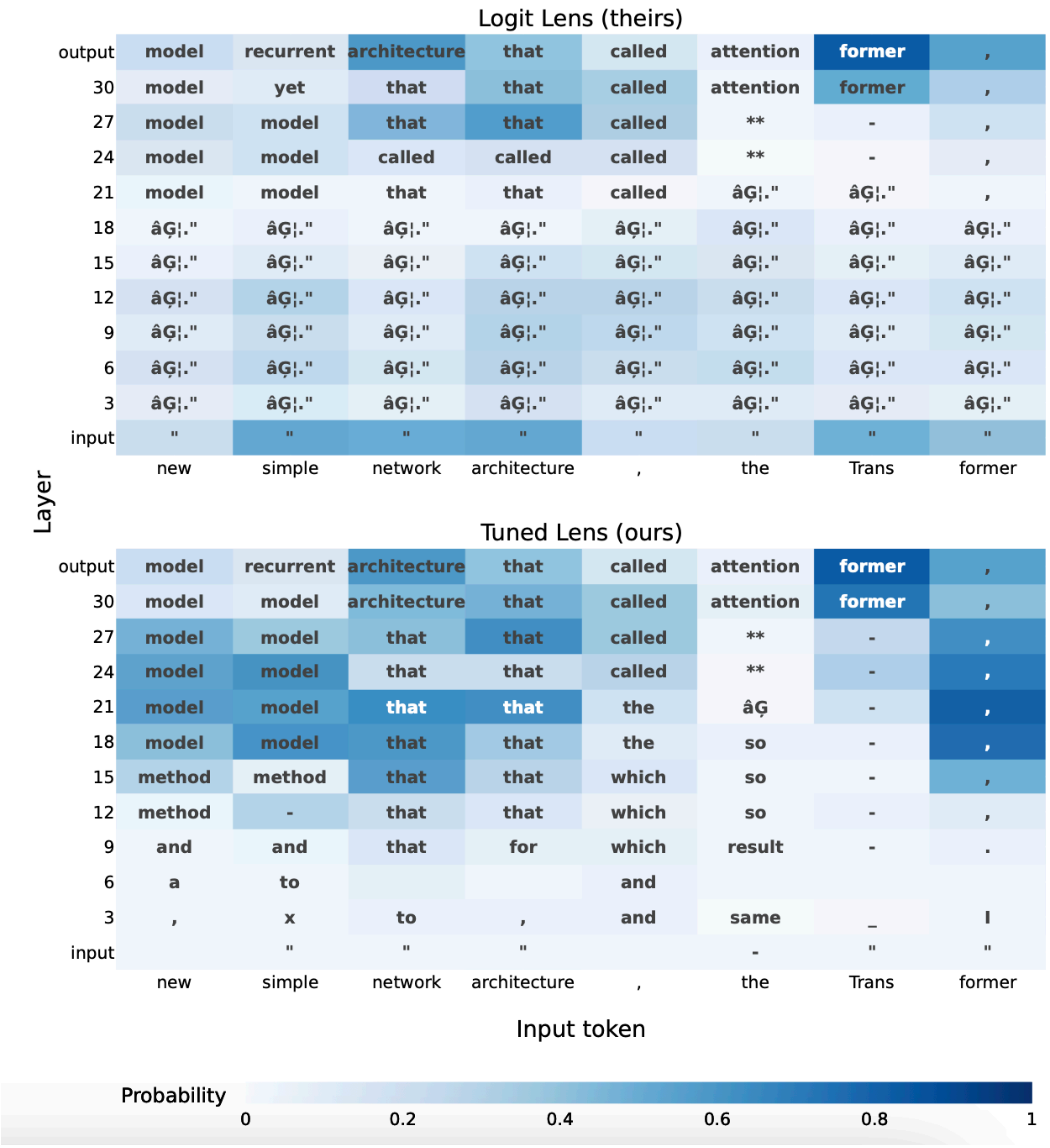


Figure 1 in (Belrose et al, 2023)

Attribute Lens

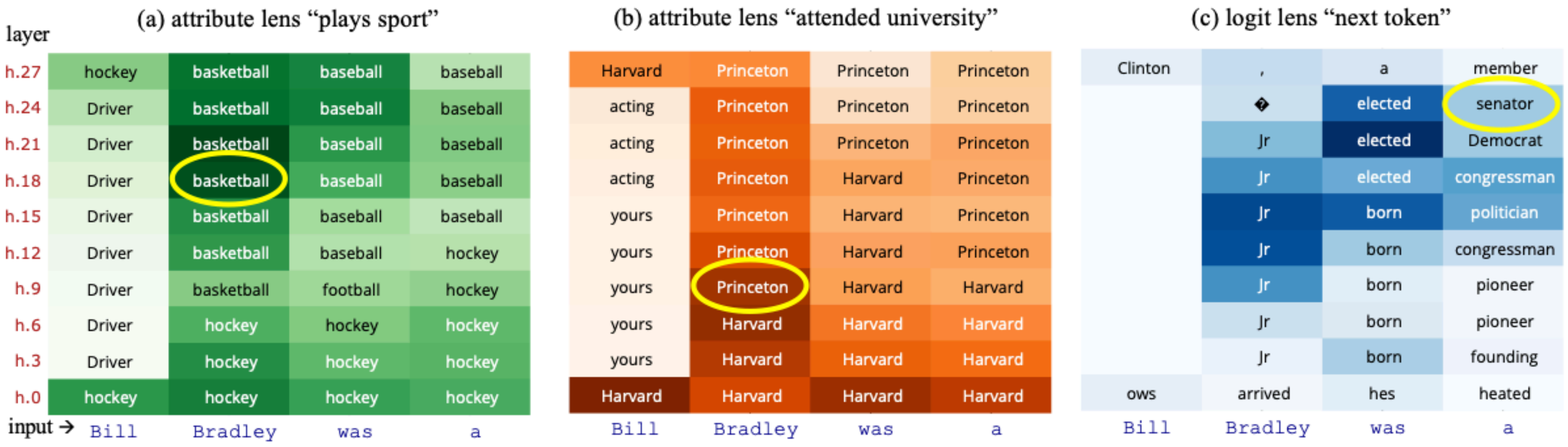


Figure 8 Hernandez et al, 2024

Notation

Borrowed from (Chughtai et al. 2024)

The function a standard transformer with L layers and parameters θ implements f_θ can be expressed $f_\theta(x_{\leq t}) = \text{softmax}(\pi_t(x_{\leq}t))$ where π_t is a vecotr of logits given by

$$
\begin{aligned}
\pi_t &= \text{LayerNorm}(z_t^L)W_U \\
z_t^l &= z_t^{l-1} + a_t^l + m_t^l \\
a_t^l &= \text{Attn}(z_t^{l-1}) \\
m_t^l &= \text{MLP}(z_t^{l-1}),
\end{aligned}
$$

Logit Lens

$$\tilde{\pi}_t^l = \text{LayerNorm}(z_t^l)W_U \qquad \text{with } l \leq L.$$

Direct Logit Attribution (DLA)

$$a_t^l = \text{Attn}(z_t^{l-1}) = \sum_{h=1}^{H} a_h(z_t^{l-1})$$

$$m_t^l = \text{MLP}(z_t^{l-1}) = \sum_{n=1}^{N} m_n(z_t^{l-1}),$$

$$\tilde{\pi}_t^{l,h} = \text{LayerNorm}(a_h(z_t^{l-1}))W_U$$

$$\tilde{\pi}_t^{l,n} = \text{LayerNorm}(m_n(z_t^{l-1}))W_U$$

Notation borrowed from (Chughtai et al. 2024)

Different types of DLA

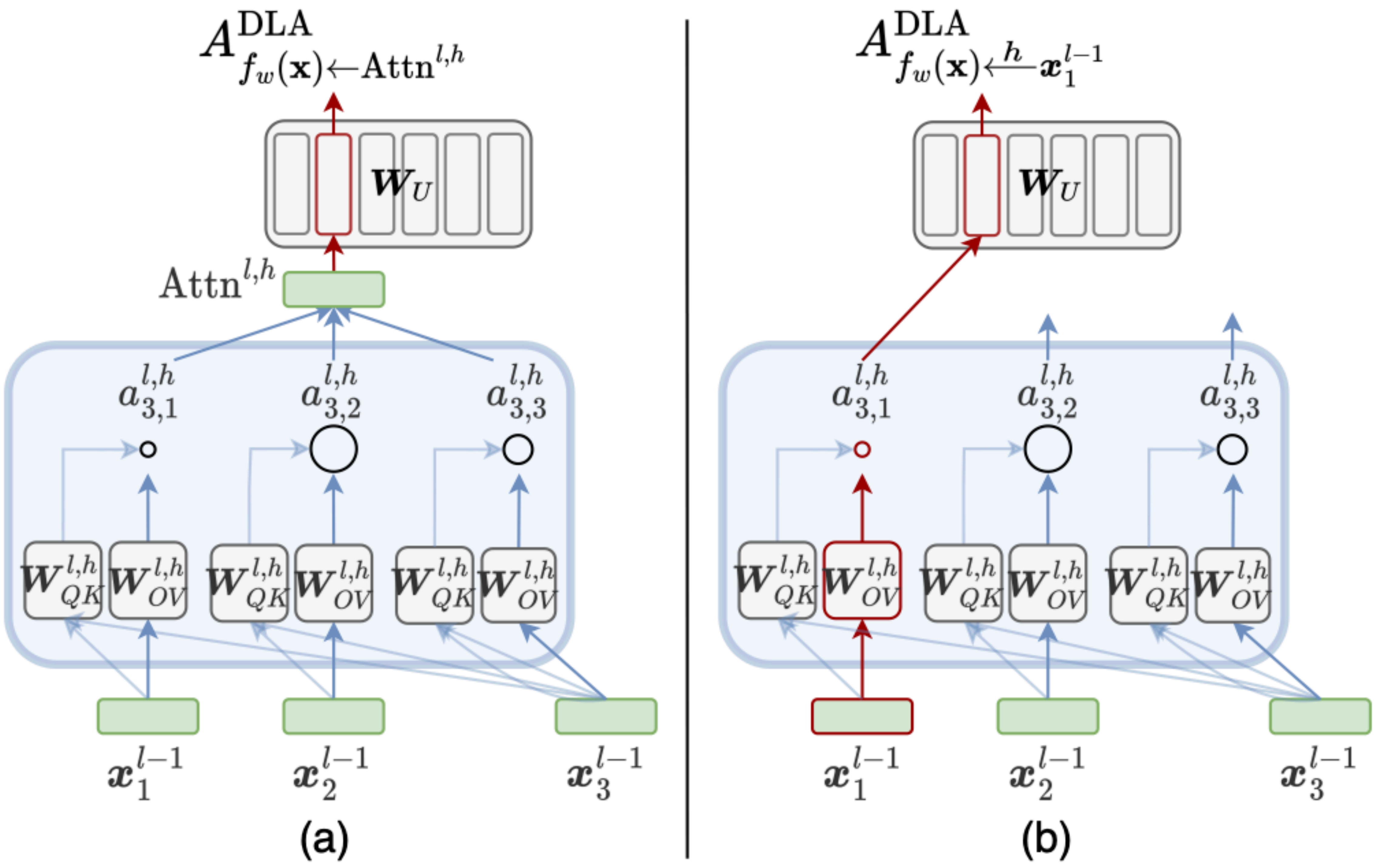


Figure 5: Direct Logit Attributions (DLA) on output token w. (a) DLA of an attention head $\text{Attn}^{l,h}$, (b) DLA of an intermediate representation $\boldsymbol{x}^{l-1}_1$ via an attention head, (c) DLA of an FFN block, and (d) DLA of a single neuron.

Figure from the (Ferrando et al. 2024)

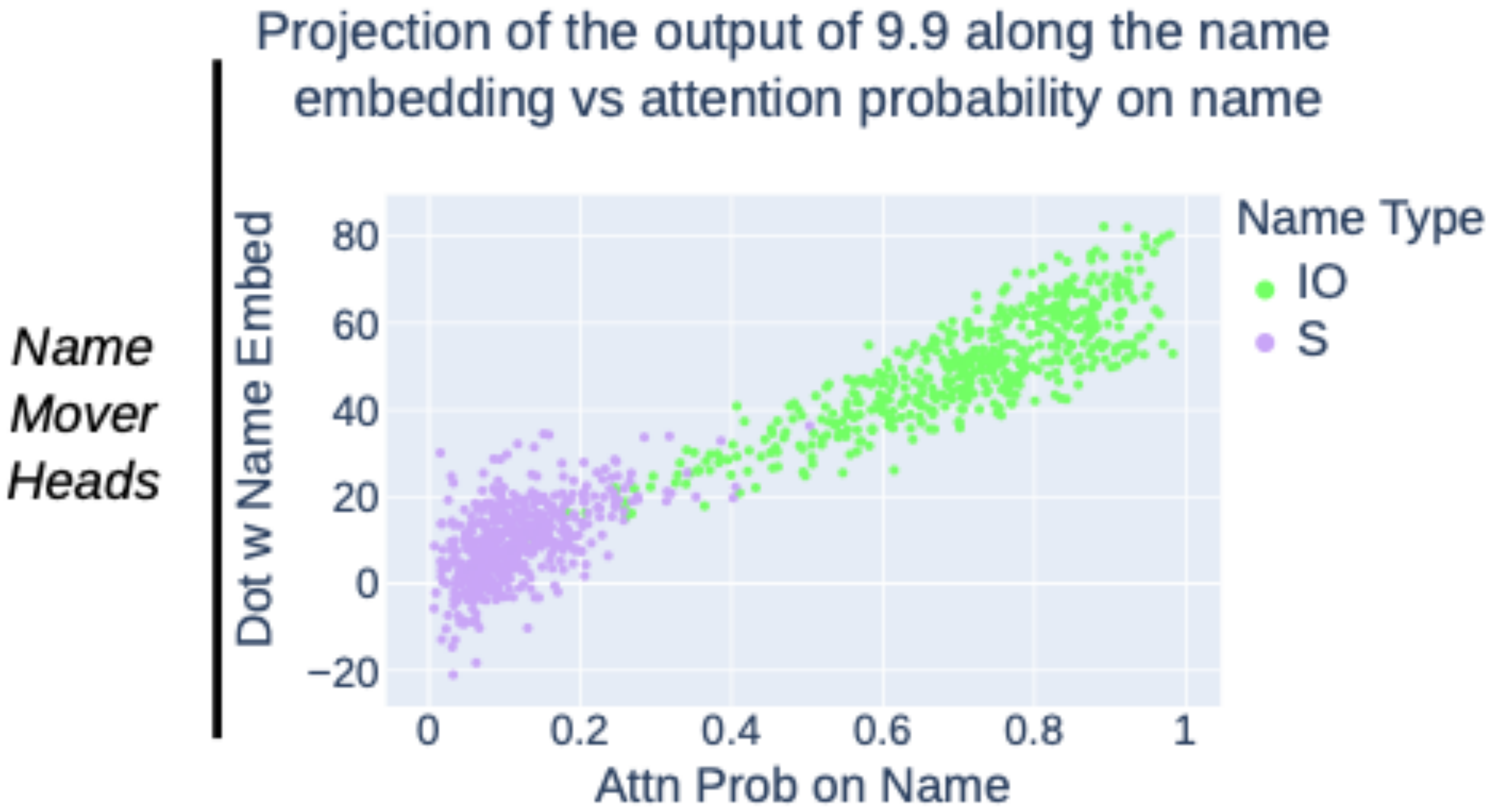


Figure 3(c) from Wang et al, 2022

Patchscopes

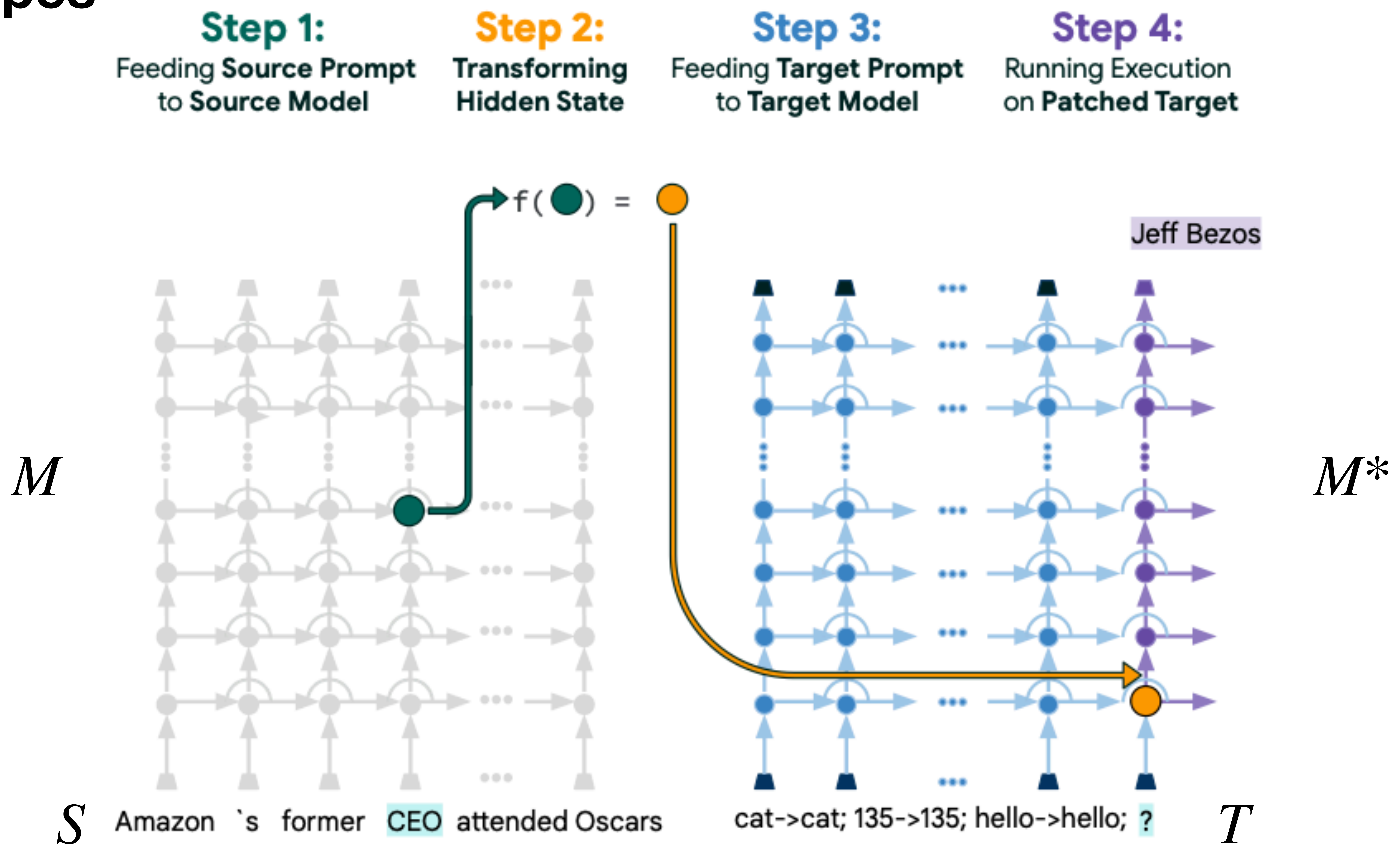


Figure 1 from Gandeharioun et al, 2024

$(S, i, M, l), (T, i^*, f, M^*, l^*)$

Entity resolution

to and how that relates to the tokens processed. $\mathcal{M}^* = \mathcal{M} \leftarrow$ Vicuna (13B), $\ell^* = \ell$, $S \leftarrow$ `"Diana, Princess of Wales"`.

i

ℓ	Generation	Explanation
1-2	: Country in the United Kingdom	**Wales**
3	: Country in Europe	**Wales**
4	: Title held by female sovereigns in their own right or by queens consort	**Princess of Wales** (unspecific)
5	: Title given to the wife of the Prince of Wales (and later King)	**Princess of Wales** (unspecific)
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	**Diana, Princess of Wales**

Table 3 from Gandeharioun et al, 2024

T "`subject`$_1$`: description`$_1$`, ..., subject`$_k$`: description`$_k$`, x`", while patching the last position

Questions about Logit Lens

I am wondering why KL divergence isn’t a good metric to calculate distance between intermediate and final probability distribution?

Could we use simply mean squared error? What about the dot product?

Questions about Logit Lens

I find it very interesting that the way layers in a transformer work is that they essentially take one embedding and transform it into another and those two vectors would come from the same "language" (that is, if you use the W matrix, you would get the first token at the bottom and the next token at the top).

Have there been studies or attempts to make those two vectors belong to completely different categories, so that there would be "current token" and "next token" embeddings.

I was also wondering if this is directly related to the issue of GPT-2 often repeating the same sequence of tokens over and over. Is the problem that is sometimes simply not enough layers?

Questions about Logit Lens

Do you think it makes sense to project the activations using the same unembedding weights?

What happens if we retrained the unembedding matrix to generate the next word from the activations wouldn’t that give a better idea of the info in a certain layer?

Questions about Logit Lens

The authors claim the logit lens technique provides intuitive insights into intermediate layer activations. However, could the observed phenomena be explained by simpler statistical artifacts rather than genuine model behavior? What additional experiments could validate the robustness of the logit lens findings?

The study indicates that the model discards input tokens quickly. Could this be a sign of a potential issue with how transformers manage long-range dependencies? How might this affect the model's performance on tasks requiring detailed contextual understanding, such as coreference resolution or summarization?

Questions about Patchscope

So in experiment 4.1 and 4.2, Patchscopes has an unfair headstart compared to the baselines, and reporting that “its performance is higher” I see as deception. Do I miss something here?

Questions about Patchscope

The most interesting finding to me was when the model being used to interpret patching has more parameters, cross-model patching works really well.

Am I overestimating how important this specific result about cross-modal patching is, or do you think it's just a cool result but has fairly little practical implications ?

Questions

It seems that patching into the later layers seems to perform better than patching into earlier layers. Could this indicate that a significant amount of transformations are necessary to extract the information from the embedding?

If many transformations are required, it is probably hard to manipulate the embedding to change model predictions. Could a probe work nonetheless, similar to how it was done in the Othello paper? I believe the kind of adversarial training the Othello paper did is hard to do on a full GPT model.

Questions

1. I have a hard time believing that just because intermediate representations make some sense in vocabulary space when decoded using (last embed-> vocab) codebook, it’s a right thing or a sensible thing to do.
    1. Also, I think the “logit lens” is just a codebook for the final representation, and so any conclusion by using it on an intermediate representation doesn’t guarantee anything. Just a correlation. Am I right in saying that?
        1. It’s as if saying that, with time a language changed, eg. A->B->C->D-> English, and you’re comparing a sentence in A/D and English, saying that it makes sense for D but not A.
        2. It’s not true, because you’re using the wrong codebook. The information is there though, you just need to extract it right.

And for patchscopes, I have a similar doubt as above. Assuming that it makes sense, can you please elaborate on how chain-of-thought patchscopes work exactly?

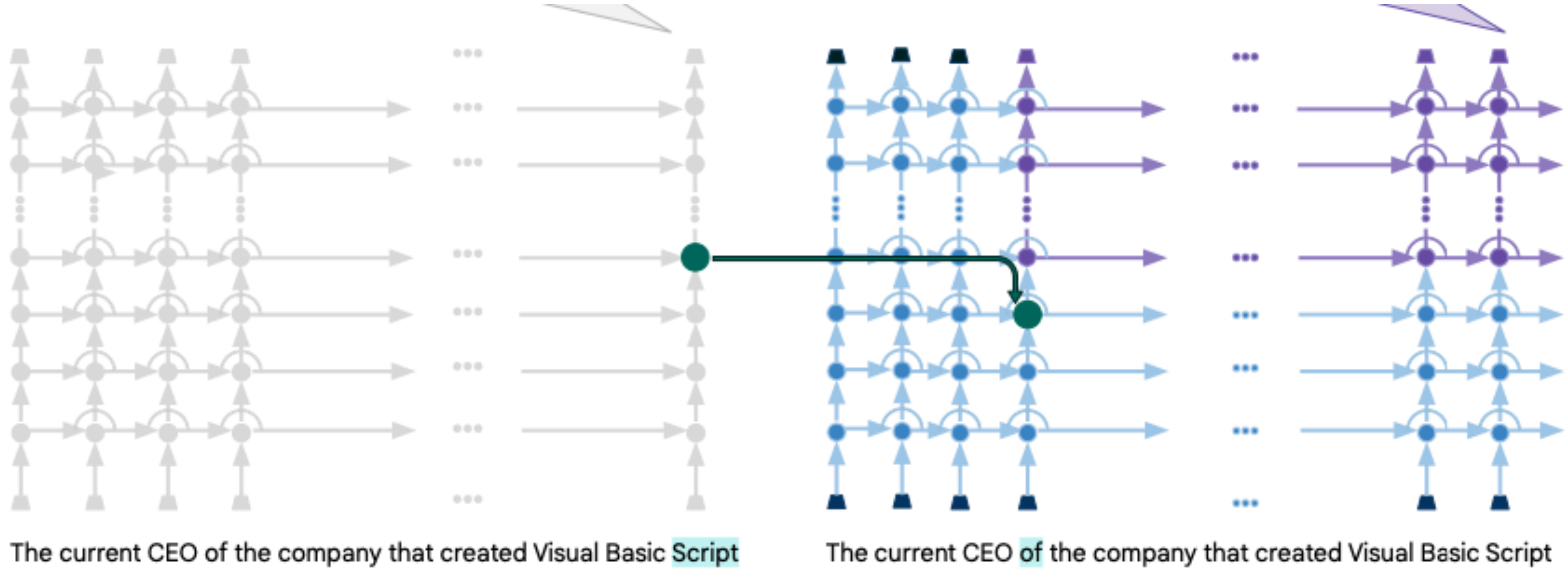


Figure 4. An illustration of CoT `Patchscope` on a single example. In this example, $\pi_1 \leftarrow$ “`the company that created Visual Basic Script`”, $\pi_2 \leftarrow$ “`The current CEO of`”, $S = T \leftarrow [\pi_2][\pi_1] =$ “`The current CEO of the company that created Visual Basic Script`”. Note that $\mathcal{M} = \mathcal{M}^*$ and $f \leftarrow \mathbb{I}$.

Figure from Gandeharioun et al, 2024

Questions about Tuned Lens

About iterative inference:

they make a point that transformers iteratively refine their representations in the direction of the output, slowly changing the representations at each layer in the anti-gradient direction. I am wondering how this is related to the fact that there are circuits in the models, and there are attention heads that can do very interpretable updates (not necessarily moving representations in the direction of the output), or store something in the residual stream (also not necessarily moving the representations in the right direction). Is it just two complementary mechanisms?

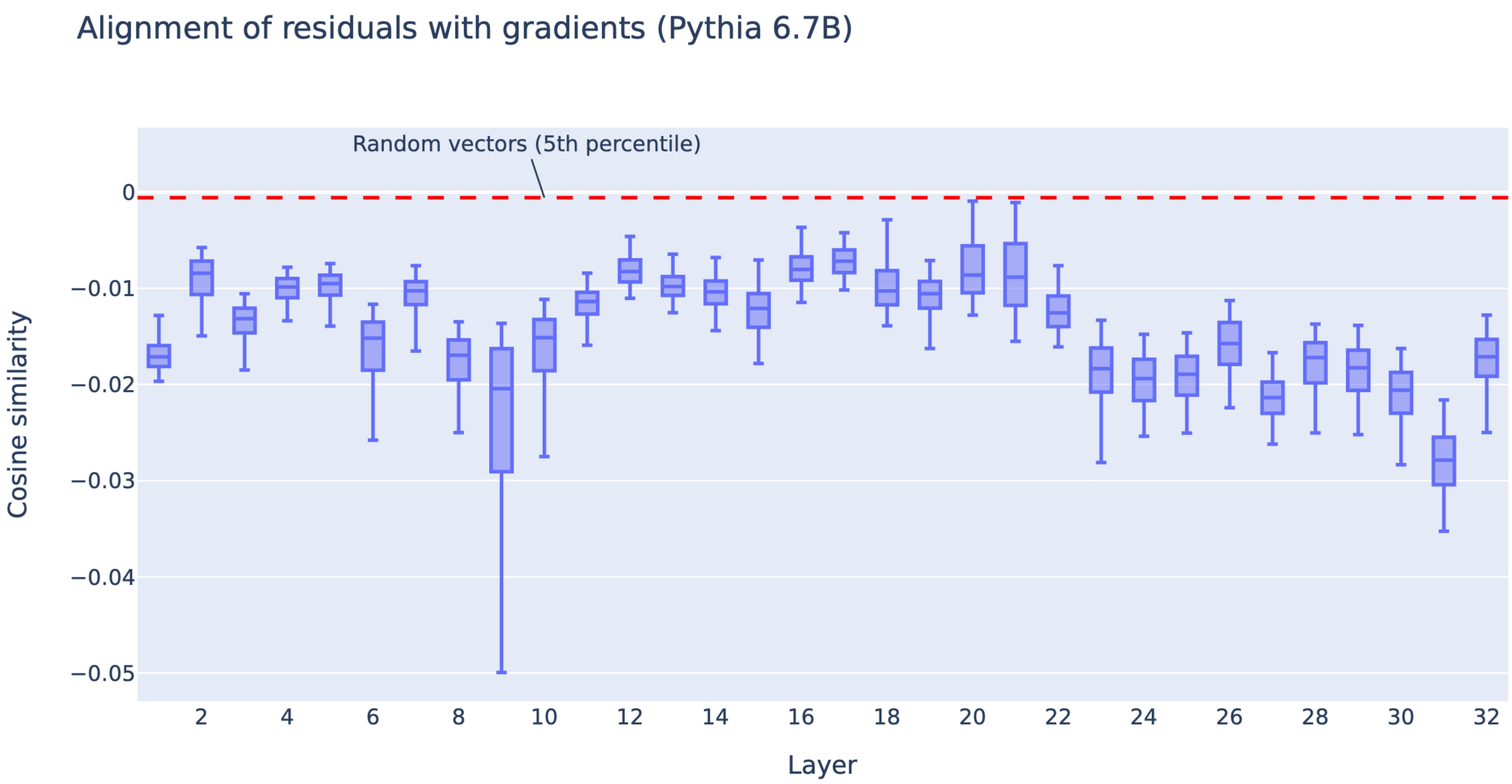


Figure 19 from (Belrose et al, 2023)

Questions about Tuned Lens

About representational drift:

The authors say that the covariance between the representations in close layers in quite high and for further layers it is quite low, and they use this motivation to introduce a change of basis matrix into the Tuned Lens formula. I did not understand, how the fact that the covariance between matrices is low implies that the matrices are represented in different bases, and not that they just have different information stored in them.