How Do Llamas Process Multilingual Text? A Latent Exploration through Activation Patching

Clément Dumas [*1,2] **Veniamin Veselovsky** [3] **Giovanni Monea** [3,4] **Robert West** [3] **Chris Wendler** [*3]

Abstract

A central question in multilingual language modeling is whether large language models (LLMs) develop a universal concept representation, disentangled from specific languages. In this paper, we address this question by analyzing Llama-2's forward pass during a word translation task. We strategically extract latents from a source translation prompt and insert them into the forward pass on a target translation prompt. By doing so, we find that the output language is encoded in the latent at an earlier layer than the concept to be translated. Building on this insight, we conduct two key experiments. First, we demonstrate that we can change the concept without changing the language and vice versa through activation patching alone. Second, we show that patching with the mean over latents across different language pairs does not impair and instead improves the model's performance in translating the concept. Our results provide evidence for the existence of language-agnostic concept representations within the model.

1. Introduction

The emergence of the field of mechanistic interpretability has led to the conception of powerful tools (Carter et al., 2019; Nostalgebraist, 2020; Schubert et al., 2020; Belrose et al., 2023; Cunningham et al., 2023; Kramár et al., 2024; Marks et al., 2024; O'Neill & Bui, 2024; Tufanov et al., 2024) for the investigation of the inner workings of deep neural networks such as large language models (LLMs) (Vaswani et al., 2017; Radford et al., 2019; Touvron et al., 2023) with the ultimate goal of reverse engineering the algorithms encoded in their weights. As a result, researchers today are often able to open up a "black box" neural network, and with near surgical precision pinpoint where a certain input-output behaviour comes from (Wang et al., 2022; Conmy et al., 2023; Nanda et al., 2023; Zhong et al., 2024; Furuta et al., 2024).

One such recent approach has been to use patchscopes (Ghandeharioun et al., 2024) or self-interpretation of embeddings (Chen et al., 2024). The key idea behind those methods is to repurpose a LLM[1] to unpack information con-

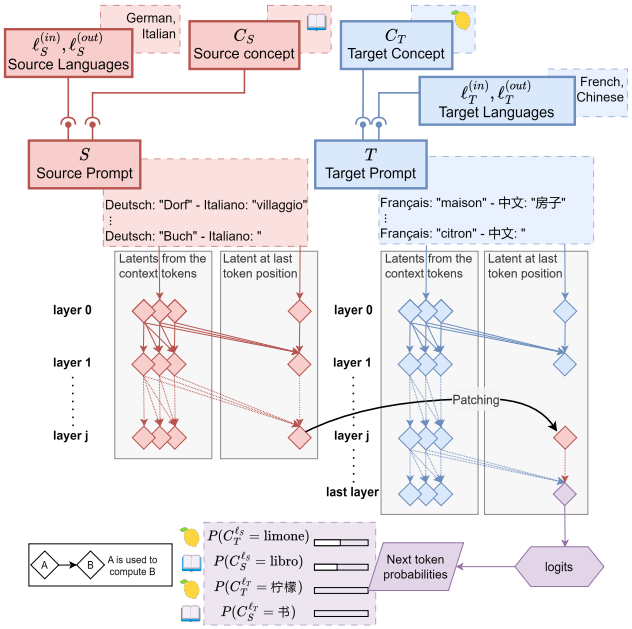

Figure 1. For two given concepts, e.g., BOOK and LEMON, we construct a source prompt for translating from German to Italian, and a target prompt for translating from French to Chinese. Then we extract the latent of the last token after some layer j from the source prompt and insert it at the corresponding position in the forward pass of the target prompt. The resulting next token probabilities will concentrate on the *target concept in the target language* (LEMON$^{\text{ZH}}$, i.e., 柠檬) when patching at layers 0–11, on the *target concept in the source language* (LEMON$^{\text{IT}}$, "limone") for layers 12–16, and on the *source concept in the source language* (BOOK$^{\text{IT}}$, "libro") for layers 17–31.

[1]Note that we use LLM and transformer (Vaswani et al., 2017)

[*]Equal contribution [1]ENS Paris-Saclay, France [2]Université Paris-Saclay, France [3]EPFL Lausanne, Switzerland [4]Cornell Tech, USA. Correspondence to: Clément Dumas <clement.dumas@ens-paris-saclay.fr>, Chris Wendler <chris.wendler@epfl.ch>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

tained in its own intermediate results. This can be achieved by patching a latent from one forward pass into another one while observing the output (cf. Fig. 1).

Summary of contributions. In this work, inspired by those patching based approaches, we leverage activation patching to understand how LLMs like Llama-2 (Touvron et al., 2023) processes multilingual text. In particular, we investigate whether it uses a language-agnostic concept space, as theorized by Wendler et al. (2024). In such a space, concepts would be represented independently of the language used to express them. In order to do so, we design multiple patching experiments leveraging pairs of translation prompts with differing expected predicted concept and language.

1. We start by patching at the last token (as in Fig. 1). As a result, we find that first the model resolves the output language and, in later layers, the concept to be translated.
2. Next, we come up with two hypotheses about how Llama-2's forward pass might have solved the task. **H1**, in which language and concept are represented in a disentangled way, and **H2**, in which they are always entangled.
3. Finally, we perform targeted experiments to gather more evidence for either **H1** or **H2**, and find **H1** is better supported by our results.

Therefore, our results agree with the theory outlined by Wendler et al. (2024). In contrast to their analysis which is purely observational with the logit lens, ours is based on interventions by virtue of activation patching. Additionally, by using activation patching, we circumvent the potential pitfalls of cosine similarity (Steck et al., 2024) inherent in the logit lens analysis and instead utilize Llama-2's full power to draw conclusions about the computations performed and representations used.

2. Llama-2's forward pass

Because we need full model access for our analysis, we focus on Llama-2 (Touvron et al., 2023)[2]. Llama-2 is an autoregressive, decoder-only, residual-based transformer (Vaswani et al., 2017) that was trained to map a sequence of input tokens $x_1,\dots,x_n \in V$, where n is the sequence length, to a sequence of latents in $\mathbb{R}^d$ that is refined layer by layer such that the final latents are well-suited for predicting the next tokens $x_2,\dots,x_{n+1} \in V$.

On a technical level, this is achieved using transformer blocks, consisting of a causally masked self-attention layer followed by a feed-forward network with a residual connection and root mean square (RMS) normalization in between (Vaswani et al., 2017; Touvron et al., 2023), that are used to update the latent at position i in layer j:

$$h_i^{(j)} = h_i^{(j-1)} + f_j\left(h_1^{(j-1)},\dots,h_i^{(j-1)}\right), \tag{1}$$

where $h_1^{(j-1)},\dots,h_i^{(j-1)}$ and $h_i^{(j)} \in \mathbb{R}^d$.

The initial latents $h_1^{(0)},\dots,h_n^{(0)} \in \mathbb{R}^d$ are learnt token embeddings. Finally, for a m-layer transformer, the next-token probabilities are obtained via a learnt linear layer followed by a softmax operation mapping $h_i^{(m)}$ to $P(x_{i+1}|h_i^{(m)})$.

3. Exploratory analysis with patching

Notation. Let C denote an abstract concept and C^ℓ its language-specific version. Further, let $w(C^\ell)$ denote the set of words[3] expressing the abstract concept C in language ℓ. For example, using capitalization to denote the abstract concepts, let $C = \text{CAT}$. Then for $\ell = \text{EN}$ we have $w(C^{\text{EN}}) = \{\text{“cat”}\}$ and similarly $w(C^{\text{DE}}) = \{\text{“Katze”},\text{“Kater”}\}$.

Problem statement. We aim to understand whether language and concept information can vary independently during Llama-2's forward pass when processing a multilingual prompt. For example, a representation of C^ℓ of the form $z_{C^\ell} = z_C + z_\ell$, in which $z_C \in U$, $z_\ell \in U^\perp$ and $U \oplus U^\perp = \mathbb{R}^d$ is a decomposition of $\mathbb{R}^d$ into a subspace U and its orthogonal complement $U^\perp$, would allow for language and concept information to vary independently: language can be varied by changing $z_\ell \in U^\perp$ and concept by changing $z_C \in U$. Conversely, if language and concept information were entangled, a decomposition like this should not exist: varying the language would mean varying the concept and vice versa.

3.1. Experimental design

We start our analysis with an exploratory experiment in which we utilize simple few-shot translation prompts from Wendler et al. (2024) to create paired source and target prompt datasets with different concept $C_S \neq C_T$, input language $\ell_S^{(\text{in})} \neq \ell_T^{(\text{in})}$, and output language $\ell_S^{(\text{out})} \neq \ell_T^{(\text{out})}$.

If not mentioned otherwise, ℓ_S and ℓ_T refer to the output language of the source and target prompt.

Prompt design. An example translation prompt:

interchangeably.

[2]Additionally, we reproduce our results on a few other LLMs as shown in App. D.

[3]We talk about words for the sake of simplicity. However, on a technical level w refers to the set of starting tokens of these words. Therefore, each time we patch and track different sets of tokens W, (e.g. $W \in \{w(C_1^{\text{IT}}), w(C_1^{\text{ZH}}), w(C_2^{\text{IT}}), w(C_2^{\text{ZH}}), w(C_1^{\text{EN}}) \cup w(C_2^{\text{EN}})\} = \mathscr{W}$), we ensure that there is no token in common between any pair of $W_1, W_2 \in \mathscr{W}$ with $W_1 \neq W_2$.

English: “lake” - Français: “lac”
English: “south” - Français: “sud”
English: “mother” - Français: “mère”
English: “seat” - Français: “siège”
English: “cloud” - Français: “

Here the task is to translate $w(\text{CLOUD}^{\text{EN}}) = \{\text{“cloud”}\}$ into $w(\text{CLOUD}^{\text{FR}}) = \{\text{“nuage”}\}$.

Importantly, whether the model correctly answers the prompt is determined by its next token prediction. For example above, the next token predicted should be “nu”, the first token of “nuage”. Thus, we can track $P(C^{\ell})$, i.e., the probability of the concept C occurring in language ℓ, by simply summing up the probabilities of all starting tokens of $w(C^{\ell})$ in the next-token distribution.

We improve upon the construction of Wendler et al. (2024) by considering all the possible expressions of C in ℓ using BabelNet (Navigli et al., 2021), instead of GPT-4 translations, when computing $P(C^{\ell})$. This allows us to capture many possible translations, instead of one. For example, “commerce”, “magasin” and “boutique” are all valid words for SHOP^{FR}.

Patching setup. We would like to infer at which layers the output language and the concept enter the latent $h^{(j)}_{n_T}(T)$ respectively and whether they can vary independently. In order to investigate this question, we perform the experiment depicted in Fig. 1. For each transformer block f_j we create two parallel forward passes, one processing the source prompt $S = (s_1, \ldots, s_{n_S})$ and one processing the target prompt $T = (t_1, \ldots, t_{n_T})$. While doing so, we extract the latent of the last token of the source prompt at layer j, $h^{(j)}_{n_S}(S)$, and insert it at the same layer at position n_T in the forward pass of the target prompt, i.e., by setting $h^{(j)}_{n_T}(T) = h^{(j)}_{n_S}(S)$ and subsequently completing the altered forward pass. From the resulting next token distribution, we compute $P(C^{\ell_S}_S), P(C^{\ell_T}_S), P(C^{\ell_S}_T)$, and $P(C^{\ell_T}_T)$.

3.2. Results

In this experiment, we use differing concepts, and $\ell^{(\text{in})}_S = \text{DE}, \ell^{(\text{out})}_S = \text{IT}$ for the source and $\ell^{(\text{in})}_T = \text{FR}, \ell^{(\text{out})}_T = \text{ZH}$ for the target prompt. We perform the patching at one layer at a time and report the probability that is assigned to $P(C^{\ell_S}_S), P(C^{\ell_T}_S), P(C^{\ell_S}_T)$, and $P(C^{\ell_T}_T)$. As a result we obtain Fig. 2 in which we report means and 95% confidence interval over 200 examples. As model we use Llama-2-7B.

Interpretation. We observe the following pattern while patching at different layers (see Fig. 2):

- Layers 0–11: Target concept decoded in target language, resulting in large $P(C^{\text{ZH}}_T)$.

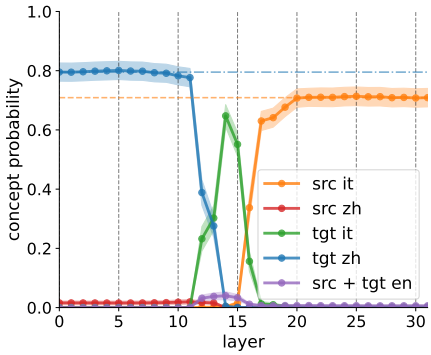

Figure 2. Our first patching experiment with a DE to IT source prompt and a FR to ZH target prompt with different concepts. We patch at the last token. For each of the plots the x-axis shows at which layer the patching was performed during the forward pass on the target prompt and the y-axis shows the probability of predicting the correct concept in language ℓ (see legend). In the legend, the prefix “src” stands for source and “tgt” for target concept. The orange dashed line and blue dash-dotted line correspond to the mean accuracy on source and target prompt. We report means and 95% Gaussian confidence intervals computed over 200 source-, target prompt pairs featuring 41 source concepts and 38 target concepts.

- Layers 12–16: Target concept decoded in source language, resulting in large $P(C^{\text{IT}}_T)$.
- Layers 16–31: Source concept decoded in source language, resulting in large $P(C^{\text{IT}}_S)$.

This pattern suggests that the model first computes the output language: from layer 12 onwards, we decode in the source output language. This indicates that up until that layer, the need to decode to $\ell^{(out)}$ is being encoded in the latent and subsequently remains unchanged. For example, this could be achieved by the model computing a function vector $z_{\ell^{(out)}}$ (Todd et al., 2023). If this hypothesis is correct, patching at layer 12 would overwrite the function vector encoding the need to decode to ZH from the target prompt with the one to decode to IT from the source prompt. This would explain the shape of the green line in Fig. 2.

In later layers, the model determines the concept: from layer 16 onwards, the source concept is decoded. This suggests that $z_{C^{\ell^{(\text{out})}}_T}$ is overwritten at layer 16. [4]

[4] In Appendix A, we collected additional experimental results investigating the right part of Fig. 2 more deeply and in Appendix B the left part. For the right part, we use the patchscope lens (Ghandeharioun et al., 2024) to investigate from which layer it is possible to decode the source concept in App. Fig. 6. The results of both experiments agree: from layer 16 it is possible to decode the source concept in source language. For the left part, we experiment with randomized source prompts and different prompting templates in between source and target prompt in App. Fig. 7. We find that indeed before layer 11 there is no translation task specific information in the latent, only prompt-template specific information.

Hypotheses. We are left with two hypotheses compatible with these results, depicted in Fig. 3:

- **H1**: Concept and language are represented independently. When doing the translation, the model first computes $\ell^{(\text{out})}$ from context, and then identifies C. In the last layers, it then maps C to the first token of $w(C^{\ell^{(\text{out})}})$.
- **H2**: The representation of a concept is always entangled with its language. When doing the translation, the model first computes $\ell^{(\text{out})}$, then computes $\ell^{(\text{in})}$ and $C^{\ell^{(\text{in})}}$ from its context and solves the language-pair-specific translation task of mapping $C^{\ell^{(\text{in})}}$ to $C^{\ell^{(\text{out})}}$.

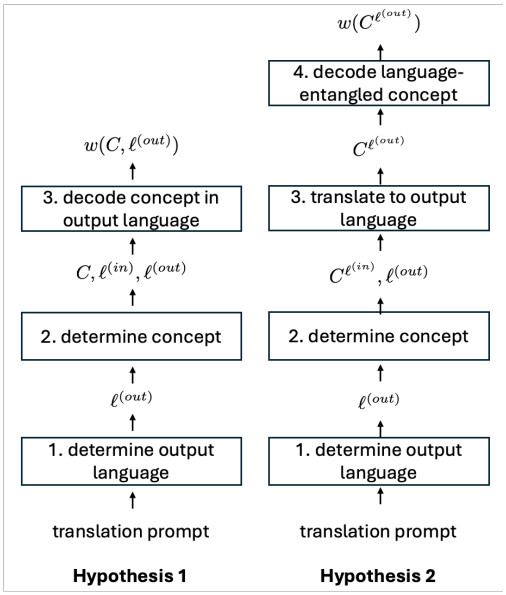

Figure 3. Our hypothesis about how the forward pass could look like on our translation prompts. Every block consists of multiple transformer blocks and in between the blocks we denote the relevant content contained in the latents (in the residual stream). Because in hypothesis 2 concept and language cannot be disentangled one input-output language specific translation circuit per language pair is required.

4. Ruling out hypotheses

Next, we run additional experiments to (1) provide further evidence that we are either in **H1** or **H2** and (2) to disambiguate whether we are in **H1** or **H2**.

Further evidence experiment. In the experiments in Sec. 3 we did not observe *source concept in target language*. However, both **H1** and **H2** would allow for that to happen via patching in the right way. Therefore, in this experiment, instead of overwriting latents at the last token of the prompt, we overwrite them at the last token of the word to be translated. Let ρ_S and ρ_T denote the position of that token in source and target prompt respectively. Since the concept information seems to enter via multiple layers (16-20) into the latent of the last token, we overwrite the latent corresponding to the token at position ρ_T at layer j *and all subsequent ones* as depicted in Fig. 4. By patching in this way in both **H1** and **H2** we would expect to see large $P(C_S^{\ell_T})$.

Formally, we patch by setting $h^{(j)}_{\rho_T}(T) = h^{(j)}_{\rho_S}(S), \ldots, h^{(m)}_{\rho_T}(T) = h^{(m)}_{\rho_S}(S)$ (in Llama-2-7B with 0-indexing, $m = 31$).

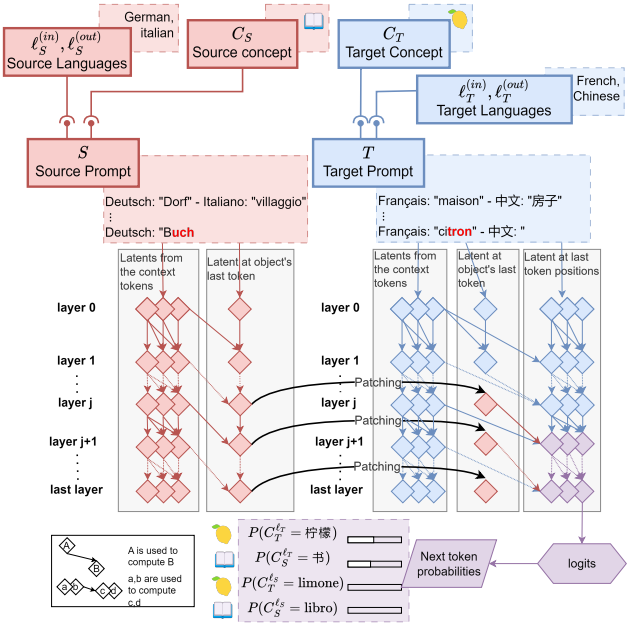

Figure 4. For two given concepts, e.g., BOOK and LEMON, we construct a source prompt for translating from German to Italian, and a target prompt for translating from French to Chinese. Then we extract the latents of the last token of the word to be translated after some layer j and all subsequent ones from the source prompt and insert them at the corresponding positions in the forward pass of the target prompt. The resulting next token probabilities will concentrate on the *source concept in target language* (BOOK$^{\text{ZH}}$, i.e., 柠檬) when patching at layers 0–15, on the *target concept in target language* (LEMON$^{\text{ZH}}$, 书) for layers 16–31

.

Disambiguation experiment. Both **H1** and **H2** compute $w(C_S^{\ell_T})$ but in different ways. In **H1** one decoding circuit per output language is required in order to compute the expression for the concept C_S in language ℓ_T. In contrast, in **H2** one translation circuit per input-output language pair is required to map the entangled $C_S^{\ell_S^{(\text{in})}}$ to $C_S^{\ell_T^{(\text{out})}}$. Therefore, in order to disambiguate the two, we construct a patching experiment that should work under **H1**, but fail under **H2**.

In order to do so, instead of patching the latent containing $C_S^{\ell_S^{(\text{in})}}$ from a single source forward pass, we create multi-

ple source prompts with the same concept C_S but in different input languages $\ell_{S_1}^{(\text{in})} \neq \ldots \neq \ell_{S_k}^{(\text{in})}$ and output languages $\ell_{S_1}^{(\text{out})} \neq \ldots \neq \ell_{S_k}^{(\text{out})}$ and patch by setting

$$h_{\rho_T}^{(\alpha)}(T) = \frac{1}{k}\sum_{i=1}^{k} h_{\rho_{S_i}}^{(\alpha)}(S_i),$$

for $\alpha \in j, \ldots, m$. Let $C_i = C_S^{\ell_{S_i}^{(\text{in})}}$, under **H1**, taking the mean of several language-specific concept representations should keep the concept information intact, since

$$\frac{1}{k}\sum_{i=1}^{k} z_{C_i} = z_{C_S} + \frac{1}{k}\sum_{i=1}^{k} z_{\ell_{S_i}^{(\text{in})}}.$$

Therefore, we'd expect high $P(C_S^{\ell_T})$ in this case. However, under **H2**, in which z_{C_i} cannot be disentangled, this mean may not correspond to a well-defined concept. Additionally, the interference between multiple input languages should cause difficulties for the language-pair-specific translation, which should result in a drop in $P(C_S^{\ell_T})$. A visualization of this argument can be seen in App. Fig. 8

Results. In the first experiment we use the same languages as above and in the second one we used DE, NL, ZH, ES, RU as input and IT, FI, ES, RU, KO as output languages for the source prompts, and, FR to ZH for the target prompt.

In Fig. 5 we observe, that in both experiments we obtain very high probability for the *source concept in the target language* $P(C_S^{\text{ZH}})$ from layers 0 to 15, i.e., exactly until the latents at the last token stop attending to the last concept-token.

Therefore, Fig. 5 (a) supports that we are indeed either in **H1** or **H2**, since *as planned* we successfully decode *source concepts in the target language* $P(C_S^{\text{ZH}})$ from layers 0 to 15. Conversely, if we were not able to decode *source concept in target language* in this way this would have spoken against both **H1** and **H2**.

Additionally, Fig. 5 (b) supports that we are in **H1** and not in **H2** because patching in the mean keeps $P(C_S^{\text{ZH}})$ in tact and even increases it. Therefore, instead of observing interference between the different language-entangled concepts as would have been predicted by **H2**, we observe a concept-denoising effect by averaging multiple language-agnostic concept representations which only makes sense under **H1**. Taking the mean over concept representations corresponding to different input languages seems to act like a majority voting mechanism resulting in an increase in $P(C_S^{\text{ZH}})$. [5]

[5] Conversely, e.g., averaging over different translation prompt contexts but while keeping the input and output language fixed does not lead to an increase in $P(C_S^{\text{ZH}})$ (see App. Fig. 10 (b)).

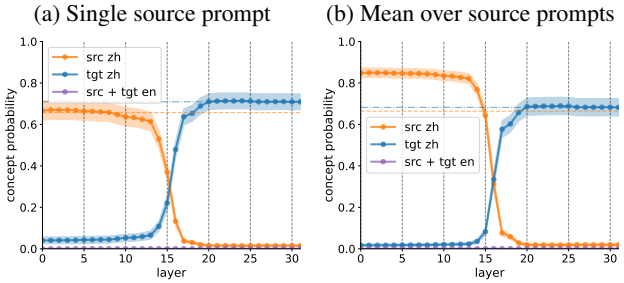

Figure 5. Here we use different input languages (DE, FR), different concepts, different output languages (IT, ZH) in (a). In (b) we use multiple source input languages DE, NL, ZH, ES, RU and output languages IT, FI, ES, RU, KO. We patch at the last token of the concept-word at all layers from j to 31. In (a) we patch latents from the single source prompt in (b) we patch the mean of the latents over the source prompts. For each of the plots, the x-axis shows at which layer the patching was performed during the forward pass on the target prompt and the y-axis shows the probability of predicting the correct concept in language ℓ (see legend). The prefix "src" stands for source and "tgt" for target concept. We report means and 95% Gaussian confidence intervals computed over a dataset of size 200.

5. Other models

In Appendix D we perform the experiments from Sec. 3 and Sec. 4 on several other models, namely, Mistral-7B (Jiang et al., 2023), Llama-3-8B[6], Qwen1.5-7B (Bai et al., 2023), and Llama-2-70B, and find that they display the same behaviour.

6. Discussion

In this paper, we performed multiple experiments that indeed indicate that Llama-2 processes language and concept information independently in the few-shot translation prompts used. This also speaks for language and concept information being represented in a disentangled way. Our results are aligned with findings from prior work (Wendler et al., 2024) that indicate that Llama-2 represents concepts in a concept space independent of the language of the prompt. However, our analysis goes beyond the purely observational logit lens analysis performed by Wendler et al. (2024). Using activation patching, we circumvent potential pitfalls of cosine similarity (Steck et al., 2024) and instead utilize Llama-2's full power.

[6] https://github.com/meta-llama/llama3

Limitations

In this work, we studied how Llama-2 represents concepts when processing multilingual text. However, we only considered very simple translation prompts and also probed only for the language-specific words describing the concept. Further experiments are needed to make claims about how Llama-2 and other language models process multilingual text in general settings. Furthermore, more fine-grained probing will be required to determine to which degree Llama-2 is able to specialize a concept to a language and whether concepts and languages are entangled in more subtle ways.

7. Acknowledgement

We would like to thank the team working on `nnsight` (Fiotto-Kaufman, 2024) which is the python package we used to implement all our experiments. We thank Hannes Wendler for multiple fruitful discussions.

Impact Statement

This paper presents work whose goal is to advance our understanding of current LLMs. We believe this understanding can help mitigate the risks and biases associated with LLMs, rather than increasing them.

References

Bai, J., Bai, S., Chu, Y., Cui, Z., Dang, K., Deng, X., Fan, Y., Ge, W., Han, Y., Huang, F., Hui, B., Ji, L., Li, M., Lin, J., Lin, R., Liu, D., Liu, G., Lu, C., Lu, K., Ma, J., Men, R., Ren, X., Ren, X., Tan, C., Tan, S., Tu, J., Wang, P., Wang, S., Wang, W., Wu, S., Xu, B., Xu, J., Yang, A., Yang, H., Yang, J., Yang, S., Yao, Y., Yu, B., Yuan, H., Yuan, Z., Zhang, J., Zhang, X., Zhang, Y., Zhang, Z., Zhou, C., Zhou, J., Zhou, X., and Zhu, T. Qwen technical report, 2023. URL https://arxiv.org/abs/2309.16609.

Belrose, N., Furman, Z., Smith, L., Halawi, D., Ostrovsky, I., McKinney, L., Biderman, S., and Steinhardt, J. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*, 2023.

Carter, S., Armstrong, Z., Schubert, L., Johnson, I., and Olah, C. Activation atlas. *Distill*, 2019. doi: 10.23915/distill.00015. https://distill.pub/2019/activation-atlas.

Chen, H., Vondrick, C., and Mao, C. Selfie: Self-interpretation of large language model embeddings. *arXiv preprint arXiv:2403.10949*, 2024.

Conmy, A., Mavor-Parker, A. N., Lynch, A., Heimersheim, S., and Garriga-Alonso, A. Towards automated circuit discovery for mechanistic interpretability. *arXiv preprint arXiv:2304.14997*, 2023.

Cunningham, H., Ewart, A., Riggs, L., Huben, R., and Sharkey, L. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023.

Fiotto-Kaufman, J. nnsight: The package for interpreting and manipulating the internals of deep learned models. , 2024. URL https://github.com/ndif-team/nnsight.

Furuta, H., Gouki, M., Iwasawa, Y., and Matsuo, Y. Interpreting grokked transformers in complex modular arithmetic. *arXiv preprint arXiv:2402.16726*, 2024.

Ghandeharioun, A., Caciularu, A., Pearce, A., Dixon, L., and Geva, M. Patchscope: A unifying framework for inspecting hidden representations of language models. *arXiv preprint arXiv:2401.06102*, 2024.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. d. l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023.

Kramár, J., Lieberum, T., Shah, R., and Nanda, N. Atp*: An efficient and scalable method for localizing llm behaviour to components. *arXiv preprint arXiv:2403.00745*, 2024.

Marks, S., Rager, C., Michaud, E. J., Belinkov, Y., Bau, D., and Mueller, A. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. *arXiv preprint arXiv:2403.19647*, 2024.

Nanda, N., Chan, L., Lieberum, T., Smith, J., and Steinhardt, J. Progress measures for grokking via mechanistic interpretability. *arXiv preprint arXiv:2301.05217*, 2023.

Navigli, R., Bevilacqua, M., Conia, S., Montagnini, D., Cecconi, F., et al. Ten years of babelnet: A survey. In *IJCAI*, pp. 4559–4567. International Joint Conferences on Artificial Intelligence Organization, 2021.

Nostalgebraist. Interpreting gpt: The logit lens. LessWrong, 2020. URL https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens.

O'Neill, C. and Bui, T. Sparse autoencoders enable scalable and reliable circuit identification in language models. *arXiv preprint arXiv:2405.12522*, 2024.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Schubert, L., Petrov, M., Carter, S., Cammarata, N., Goh, G., and Olah, C. Openai microscope. microscope.openai.com/, 2020. [Online; accessed 28-May-2024].

Steck, H., Ekanadham, C., and Kallus, N. Is cosine-similarity of embeddings really about similarity? In *Companion Proceedings of the ACM on Web Conference 2024*, pp. 887–890, 2024.

Todd, E., Li, M. L., Sharma, A. S., Mueller, A., Wallace, B. C., and Bau, D. Function vectors in large language models. *arXiv preprint arXiv:2310.15213*, 2023.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Tufanov, I., Hambardzumyan, K., Ferrando, J., and Voita, E. Lm transparency tool: Interactive tool for analyzing transformer language models. *Arxiv*, 2024. URL https://arxiv.org/abs/2404.07004.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Wang, K., Variengien, A., Conmy, A., Shlegeris, B., and Steinhardt, J. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. *arXiv preprint arXiv:2211.00593*, 2022.

Wendler, C., Veselovsky, V., Monea, G., and West, R. Do llamas work in english? on the latent language of multilingual transformers. *arXiv preprint arXiv:2402.10588*, 2024.

Zhong, Z., Liu, Z., Tegmark, M., and Andreas, J. The clock and the pizza: Two stories in mechanistic explanation of neural networks. *Advances in Neural Information Processing Systems*, 36, 2024.

A. Patchscope experiment

We performed an additional experiment using the patchscope lens (Ghandeharioun et al., 2024) to collect more evidence about from which layer it is possible to decode the source concept in Fig. 6. The results of this experiment corroborate the findings presented in Section 3. To enable a convenient comparison of the experimental results, we also include Fig. 2 in Fig. 6.

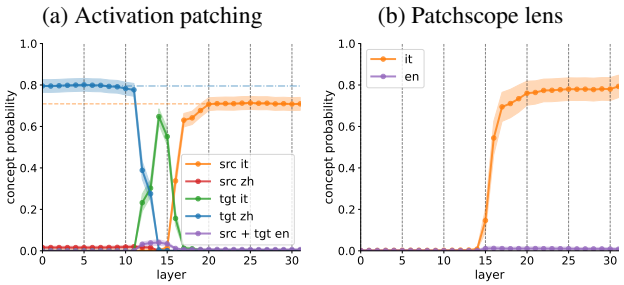

Figure 6. (a) Our first patching experiment with a DE to IT source prompt and a FR to ZH target prompt with different concepts. (b) Our patchscope lens experiment with a DE to IT source prompt and identity target prompt king king\n1135 1135\nhello hello\n?. We patch at the last token respectively. For each of the plots the x-axis shows at which layer the patching was performed during the forward pass on the target prompt and the y-axis shows the probability of predicting the correct concept in language ℓ (see legend). In the legend the prefix "src" stands for source and "tgt" for target concept. The orange dashed line and blue dash-dotted line correspond to the mean accuracy on source and target prompt. We report means and 95% Gaussian confidence intervals computed over 200 source-, target prompt pairs featuring 41 source concepts and 38 target concepts for (a) and 38 prompts for (b).

B. Random prompt task experiment

In order to investigate the leftmost part of Fig. 6a more deeply, we performed additional experiments in which we explore "random" source prompts instead of translation source prompts.

The experimental setting here is similar to the one in Sec. 3 (illustrated in Fig. 1), except for the fact that instead of patching in latents from a translation source prompt we patch latents from different "random" source prompts. For the random source prompts, we gradually move away from the prompting template.

Same template. In Fig. 7a, we randomized both input and output language as well as concepts in the source prompts, resulting in prompts of the following form:

A: "CAT$^{\text{DE}}$" - B: "DOG$^{\text{IT}}$"
A: "OWL$^{\text{JA}}$" - B: "SUN$^{\text{HI}}$"
A: "ICE$^{\text{FR}}$" - B: "

By doing this, the latent of the source prompt is similar in terms of prompt structure, but the model cannot infer a task vector specifying the output language since the source prompt instantiates an impossible task (to predict a random word in a random language). As shown in Fig. 7a, for layers 0–11, we observe no drop in the accuracy, which confirms our hypothesis that in those layers the latent at last token position contains no information specific to the translation task.

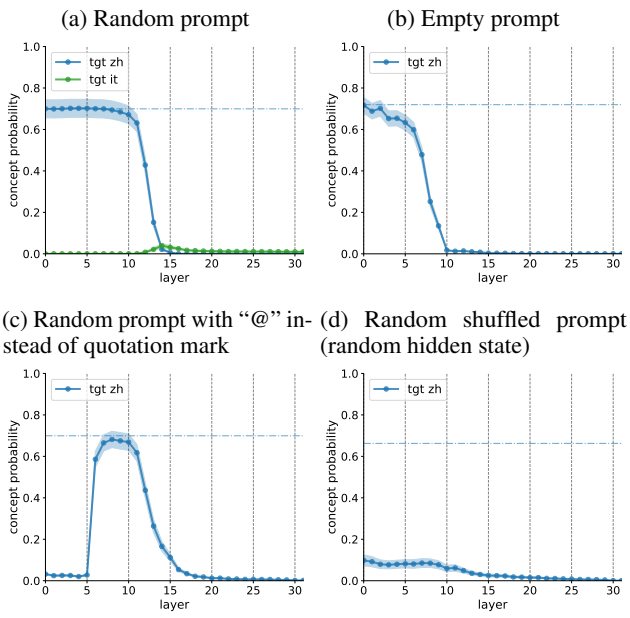

Figure 7. (a) activation patching experiment with a randomized source prompt (random concepts, and languages, but same template) and a FR to ZH target prompt. (b) we construct a source prompt with empty context. (c) we replace the quotation mark with @ in the random source prompt from (a). (d) we randomly shuffle the source prompts from (c). We patch at the last token respectively. For each of the plots, the x-axis shows at which layer the patching was performed during the forward pass on the target prompt and the y-axis shows the probability of predicting the correct concept in language ℓ (see legend). We only plot the target ("tgt") concept, as there is no source concept to predict. We report means and 95% Gaussian confidence intervals computed over 200 source-, target prompt pairs.

Instead, we think that in our chosen prompting template the last token, which is a quotation mark, merely indicates *where to put the translation result*. In order to investigate this, we performed further patching experiments investigating how changes in the prompting template in the source prompt affects the target forward pass ability to compute an answer.

Empty context. For example, replacing the source prompt with an empty prompt, merely containing B: " results in Fig. 7b. In contrast to Fig 7a, the target concept in target language probability drops already starting from layer 4. We think this is due to the fact that until layer 4 the quotation mark token information which is shared among the two prompting templates "dominates" the latent representation and is not yet converted to a task specific position marker yet. Then, starting from layer 4 the latent representation of the last token also aggregates task specific information, in particular, the fact that the quotation mark in this task actually

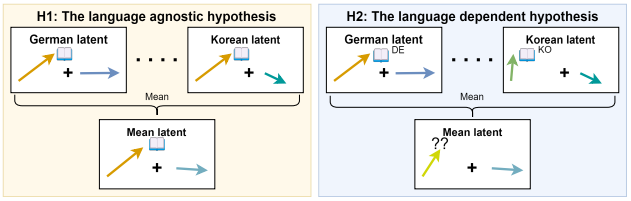

Figure 8. Illustration of two hypotheses on the latent representation of concepts across multiple languages:
H1.: When given prompts like

Deutsch: "Dorf" - Italiano: "villaggio"... Deutsch: "Buch"

with different input languages, we expect to obtain a salient representation of the language-agnostic concept (e.g., a book, shown as the left vectors in orange) along with some task-irrelevant information (the right vectors in blue). Averaging these representations should yield a concept representation at least as salient as those from individual prompts.
H2.: In this scenario, concept vectors differ due to language entanglement. As a result, we would not expect the mean vector to be as interpretable as those from single-language prompts.

marks the position after which the translated word should be decoded. As a result, replacing the task specific quotation mark embedding, which contains the information that the translated word comes next, with the "empty-context"-one, which does not contain this information, results in a performance drop.

Modified template. Next, replacing the quotation marks by "@" (Fig. 7c) in the random prompt, i.e.,

A: @CAT^{DE}@ - B: @DOG^{IT}@
A: @OWL^{JA}@ - B: @SUN^{HI}@
A: @ICE^{FR}@ - B: @

leads to a drop of performance for early layers, but for layers 5–11, the model is not much affected by the patching. We postulate that at those layers, position-marker tokens have been already mapped to a general position-marker feature that is similar in between source and target forward pass, even though at input level different symbols have been used.

Shuffled tokens. Lastly, in Fig. 7d we try to destroy all of the shared structure in between the source and the target prompt by randomly shuffling the characters of the source prompts from the **modified template** task. As expected, the probability of the target concept in target language becomes very low (albeit surprisingly not zero), which shows that the task cannot be solved without the position marker feature.

C. Mean hypothesis visualization

Here, we include a more visual explanation of our intuition behind the mean patching experiment (c.f. Fig. 8).

D. Other models

In this section, we report results for additional models, namely, Mistral-7B (Jiang et al., 2023), Llama-3-8B[7], Qwen1.5-7B (Bai et al., 2023), and Llama-2-70B.

D.1. Exploratory analysis

The results of the exploratory analysis outlined in Sec. 3 are in Fig. 9.

As can be seen in Fig. 9, the target concept in source language spike is smaller for Llama-3, Mistral 7B v0.3 and Qwen1.5 7B. This hints that for those models, $z_{\ell(\text{out})}$ and C computation overlap more than for Llama-2-7B.

D.2. Ruling out hypotheses

In this section, we report results for the experiments performed in Sec. 4.

In addition, instead of just patching in the mean over different language pairs (Fig. 10c), we also patch in the mean over contexts composed of different concept words in Fig. 10b. In particular, we take the mean over 5 different few-shot contexts from the same language pair. E.g.:

Deutsch: “Dorf” - Italiano: “villaggio”
⋮
Deutsch: “Buch

⋮

Deutsch: “Zitrone” - Italiano: “limone”
⋮
Deutsch: “Buch

Our results in Fig. 10 show that the mean over contexts does not increase $P(C_S^{\ell_T})$, whereas the mean over language pairs does. This is intuitive, since there may be some languages in which the mapping from words to concept features results in the correct concept feature vector. Therefore, averaging over different language pairs can increase the signal about the source concept. However, having additional random contexts stemming from the same language pair does not bring in any information about the source concept.

Note that Fig. 9 and Fig. 10 are on the next two pages.

[7] https://github.com/meta-llama/llama3

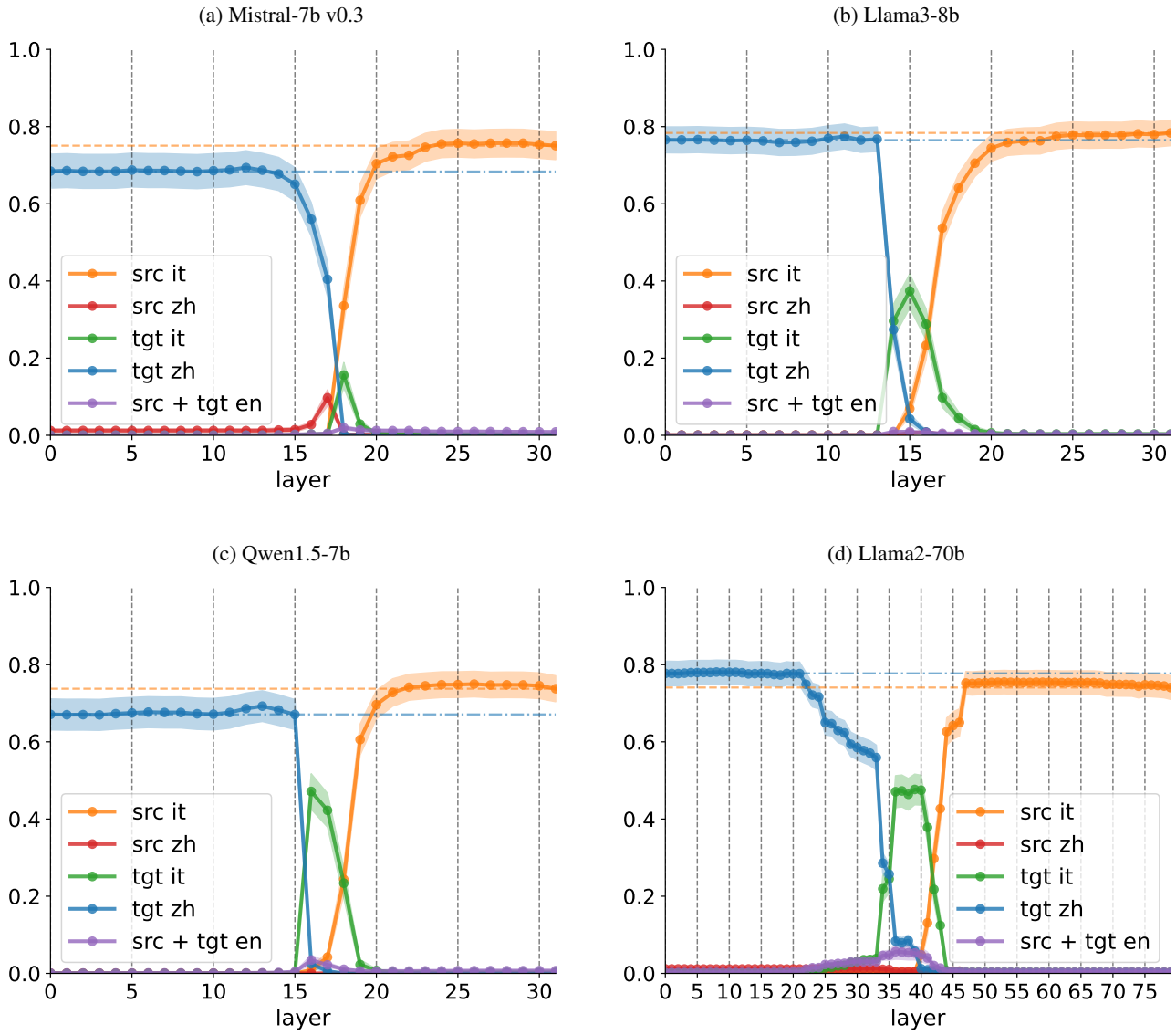

Figure 9. Our first patching experiment with a DE to IT source prompt and a FR to ZH target prompt with different concepts. We patch at the last token. For each of the plots the x-axis shows at which layer the patching was performed during the forward pass on the target prompt and the y-axis shows the probability of predicting the correct concept in language ℓ (see legend). In the legend the prefix "src" stands for source and "tgt" for target concept. The orange dashed line and blue dash-dotted line correspond to the mean accuracy on source and target prompt. We report means and 95% Gaussian confidence intervals computed over 200 source-, target prompt pairs featuring 41 source concepts and 38 target concepts.

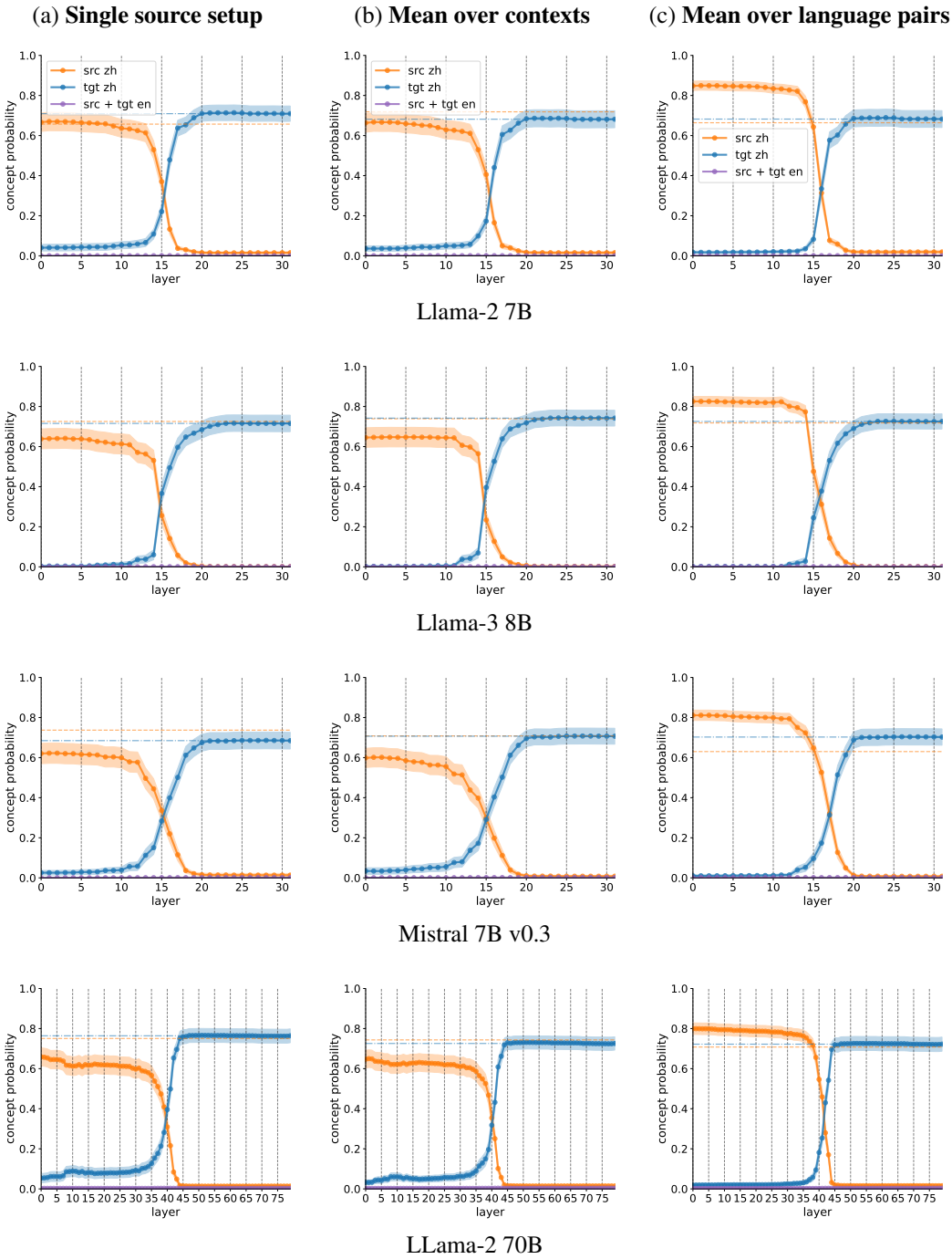

Figure 10. Here we use different input languages (DE, FR), different concepts, different output languages (IT, ZH) in (a). In (b) we use the same source and target language pairs as in (a). In (c) we use multiple source input languages DE, NL, ZH, ES, RU and output languages IT, FI, ES, RU, KO. We patch at the last token of the concept-word at all layers from j to 31. In (a) we patch latents from the single source prompt. In (b) for each concept, we patch the average latent over different few-shot DE to IT translation contexts. In (c) we patch the mean of the latents over the source prompts. For each of the plots, the x-axis shows at which layer the patching was performed during the forward pass on the target prompt and the y-axis shows the probability of predicting the correct concept in language ℓ (see legend). The prefix "src" stands for source and "tgt" for target concept. We report means and 95% Gaussian confidence intervals computed over a dataset of size 200.