Separating Tongue from Thought: Activation Patching Reveals Language-Agnostic Concept Representations in Transformers

Clément Dumas[12*†] **Chris Wendler**[3*†]
Veniamin Veselovsky[4†] **Giovanni Monea**[5†] **Robert West**[6]
[1]ENS Paris-Saclay [2]Université Paris-Saclay [3]Northeastern [4] Princeton [5]Cornell [6]EPFL
{clement.dumas@ens-paris-saclay.fr, chris.wendler@epfl.ch}

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 latent representations (latents) during a word-translation task in transformer-based LLMs. 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 representation of a concept across different languages does not affect the models' ability to translate it, but instead improves it. Finally, we generalize to multi-token generation and demonstrate that the model can generate natural language description of those mean representations. Our results provide evidence for the existence of language-agnostic concept representations within the investigated models.[1]

1 Introduction

Most modern large language models (LLMs) are trained on massive corpora dominated by English text (Touvron et al., 2023; Dubey et al., 2024; Radford et al., 2019; Brown et al., 2020; OpenAI, 2023). Despite this imbalanced training, they achieve remarkable performance across multiple languages (Shi et al., 2022), raising fundamental questions about how they process and represent multilingual information. Understanding these mechanisms is crucial not only to improve model performance, but also to identify potential biases and limitations in cross-lingual processing.

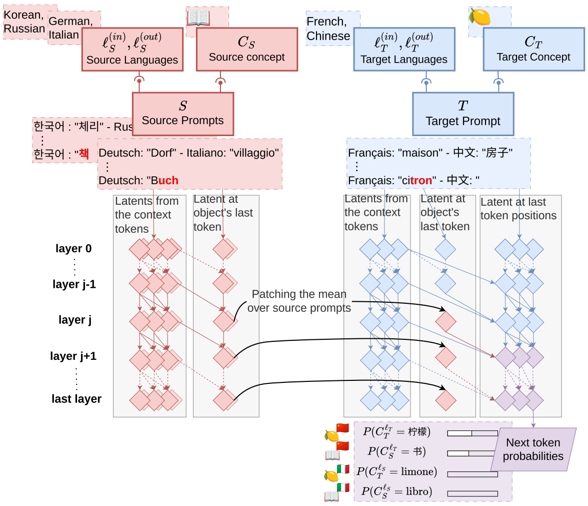

Korean,
Russian
German,
Italian
$\ell_S^{(in)}, \ell_S^{(out)}$
Source Languages
📖
C_S
Source concept
French,
Chinese
$\ell_T^{(in)}, \ell_T^{(out)}$
Target Languages
🍋
C_T
Target Concept
S
Source Prompts
T
Target Prompt
한국어 : "체리" - Rus
한국어 : "책
Deutsch: "Dorf" - Italiano: "villaggio"
Deutsch: "Buch
Français: "maison" - 中文: "房子"
Français: "citron" - 中文: "
Latents from the context tokens
Latent at object's last token
Latents from the context tokens
Latent at object's last token
Latent at last token positions
layer 0
layer j-1
layer j
layer j+1
last layer
Patching the mean over source prompts
$P(C_T^{\ell_T} = 柠檬)$
$P(C_S^{\ell_T} = 书)$
$P(C_T^{\ell_S} = \text{limone})$
$P(C_S^{\ell_S} = \text{libro})$
Next token probabilities

Figure 1: For two given concepts, e.g., BOOK and LEMON, we construct multiple source prompts which translate BOOK, and a target prompt for translating from French to Chinese. Then we extract the residual stream of the last token of the word to be translated after some layer j and all subsequent ones from the source prompts and insert the mean of each layer 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., 书) when patching at layers 0–15, on the *target concept in target language* (LEMON$^{\text{ZH}}$, 柠檬) for layers 16–31.

A fundamental question in multilingual language modeling is whether LLMs develop universal concept representations that transcend specific languages (Wendler et al., 2024; Conneau et al., 2020b; Chi et al., 2020; Xie et al., 2022; Mousi et al., 2024). For example, when a model processes the word "cat" in English and "chat" in French, does it map these to the same internal represen-

[*] Equal contribution

[†] Work done while at EPFL

[1] This work has been previously published under the title "How Do Llamas Process Multilingual Text? A Latent Exploration through Activation Patching" at the ICML 2024 Mechanistic Interpretability Workshop https://openreview.net/forum?id=0ku2hIm4BS.
Code for reproducing our experiments is available at https://github.com/Butanium/llm-lang-agnostic

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 31822–31841
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

tation of the feline concept, or does it maintain separate language-specific representations? Several recent works hint at the existence of language-agnostic concept representations.

Wendler et al. (2024) found that for simple multilingual tasks independent of the input and output language intermediate decodings of concept representations using the logit lens (Nostalgebraist, 2020) decode to the English before they decode to the target language. Additionally, it has been long observed that instruction and safety tuning LLMs only on English data generalizes to other languages (Li et al., 2024; Shaham et al., 2024). The presence of language-agnostic representation in the pretrained LLMs would provide an explanation for both of these behaviors.

This provides a unique opportunity for us to dig deeper and examine how multilingual concepts are represented and processed within LLMs, potentially revealing insights into language biases and concept formation. In particular, we are inspired by recent mechanistic interpretability approaches based on activation patching (Variengien and Winsor, 2023; Ghandeharioun et al., 2024; Chen et al., 2024a). These approaches are based on the idea of patching activations from one forward pass into another while observing the output (c.f. Fig. 2) and present a simple, yet effective way to inspect the representations learned and causally understand the computations performed by LLMs.

Contributions. While prior work has provided observational evidence for shared semantic spaces in LLMs, we present the first causal analysis of how these representations are actively utilized during multilingual processing. More specifically, in this work we aim to understand how transformers process and represent concepts during translation tasks, whether language and concept information can be manipulated independently in the model's computations, and, whether models maintain separate language-specific concept representations or develop a shared conceptual space. To this end, we make the following contributions.

1. First, we perform an activation patching analysis of Llama 2 7B (Touvron et al., 2023). We demonstrate that the model processes translation tasks by first resolving output language, then the concept to be translated.
2. We propose two competing hypotheses about how transformers solve the translation task during their forward pass: **H1** where language and concepts are represented independently, and **H2** where they are inherently entangled. We argue that if language and concepts are independent (**H1**), averaging the latent representation of a concept across languages should still allow the model to make sense of and utilize this representation. Conversely, if language and concepts are entangled (**H2**), this mean representation would be an incoherent mixture of language-specific concepts that the model cannot effectively use.
3. To test these hypotheses, we use a novel activation patching setup depicted in Figure 1 which forces Llama 2 7B to translate this mean representation across languages. We find that using the mean concept representation across languages *improves* Llama 2 7B's performance on a word translation task, supporting **H1**.
4. We show that our observations generalize to a diverse set of transformer models varying in size, architecture, and training data, including Llama 2 70B, Llama 3 8B (Dubey et al., 2024), Mistral 7B (Jiang et al., 2023), Qwen 1.5 7B (Bai et al., 2023), Aya 23 8B (Aryabumi et al., 2024) and Gemma 2 2B (Team et al., 2024).
5. Finally, to support our claim that mean representation are usable by the model in a autoregressive generation setting, we present a novel activation patching setup depicted in Figure 5 to show that a model can successfully write a definition of such a mean representation.

Implications. While prior work has suggested the existence of shared semantic spaces across languages through observational methods, our causal analysis provides the first direct evidence that LLMs actively utilize language-agnostic concept representations during text generation. Furthermore, our activation patching methodology establishes a framework for future causal investigations of multilingual representations, moving beyond the limitations of embedding similarity, probing, and logit lens approaches.

2 Related Work

LLMs have demonstrated remarkable capabilities in processing multilingual text across languages, with examples including encoder-only model like mBERT (Devlin et al., 2018), XLM-R (Conneau et al., 2020a), and mT5 (Xue et al., 2021) and decoder-only model like (Aryabumi et al., 2024;

Dubey et al., 2024). Studies on encoder-only models have shown that they develop language-agnostic representations, explaining their cross-lingual transfer capabilities. The methodology used was embedding similarity analysis (Conneau et al., 2020b; Libovický et al., 2020; Muller et al., 2021; Mousi et al., 2024) and probing methods (Choenni and Shutova, 2020; Pires et al., 2019).

While decoder-only transformers are not primarily designed to develop contextual embedding, but rather for open-ended text generation, they also develop cross-lingual generalization, for example, during safety and instruction tuning (Li et al., 2024; Chirkova and Nikoulina, 2024). Mechanistic interpretability has led to powerful tools to analyze the language-agnosticity of these models. Using neuron analysis Stańczak et al. (2022); Chen et al. (2024b); Cao et al. (2024); Zeng et al. (2024); Tang et al. (2024) have shown that LLMs develop both language-agnostic and language-specific neurons. Wendler et al. (2024); Wu et al. (2024) use the logit lens (Nostalgebraist, 2020) to perform early decoding during the forward pass of both LLMs and show that, no matter the language or input modality, the intermediate decodings concentrate on English before decoding to a specific language in the very last layers.

While the analyses via embedding similarity, probing and the logit lens give use valuable insight into the structure of the representation, they are not causal. Additionally, while the neuron level analysis studies the causal effects of the neurons, they do not study the representation themselves. Our work aims to fill this gap.

A related line of work has explored using definition generation as a means to evaluate semantic representations. Noraset et al. (2017) introduced *definition modeling*, the task of generating dictionary definitions from word embeddings, as a more direct evaluation of what semantic information embeddings capture. This approach has been extended to evaluate various types of embeddings (Gardner et al., 2022; Chang and Chen, 2019). For example, (Chang and Chen, 2019) demonstrated that contextualized embeddings like ELMo and BERT can be effectively mapped to definition spaces, revealing their sense-specific semantic content. In our work, instead of training a model to generate descriptions of LLM's representations, we repurpose the LLM itself to either translate or define it using activation patching. Activation patching, introduced by Meng et al. (2022), has been the main tool used to draw causal interpretation of LLM representations (Variengien and Winsor, 2023; Geiger et al., 2022; Kramár et al., 2024). More recently, Ghandeharioun et al. (2024); Chen et al. (2024a) also show patching setups in which they can use the model itself to analyze its own model internal. Inspired by those methods, we developed two novel patching experiments supporting the language-agnosicity of LLMs representation.

Parallel work to ours by Fierro et al. (2025) leverages the mechanistic interpretability toolkit to understand how factual recall works in multilingual LLMs as well. Similar to us, they find that the concept to be decoded and its language does not enter at the same layer into the residual stream, however in their knowledge association tasks they observe the opposite order than we do in our translation task. They also find that large parts of multilingual-factual recall are handled in a language agnostic way, despite the investigated models being trained on a more balanced split of languages than the ones studied in our paper.

3 Background

Transformers forward pass. When an autoregressive decoder-only transformer (Vaswani et al., 2017; Touvron et al., 2023) with L layers processes a sequence of input tokens $x_1, \ldots, x_n \in V$ from a vocabulary V, each token is initially transformed into a d-dimensional vector h_i^0 by an embedding layer. This first set of vector is the beginning of the residual stream. Then, for each token position i, the layer $j \in 1, \ldots, L$ updates the residual stream the following way:

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

where f_j represents the operations of the j-th layer (typically self-attention followed by a feedforward network). Finally, for a m-layer transformer, the next-token probabilities are obtained via a learned linear layer followed by a softmax operation mapping $h_i^{(m)}$ to $P(x_{i+1}|x_1 \ldots x_i)$.

Activation patching. Activation patching is a causal intervention technique that allows us to understand how different components of a neural network contribute to its output. The key idea is to run two forward passes through the network – one on a source input and one on a target input – and copy (or "patch") activations from specific positions and layers of the source forward pass into the target forward pass. By observing how these interventions affect the model's output, we can understand what information is encoded in different parts of the network and how it is used.

More formally, given a source input S and target input T, we can patch activations at position i, i' and layer j by setting $h_i^{(j)}(T) = h_{i'}^{(j)}(S)$ during the target forward pass, where $h_i^{(j)}$ represents the activation at position i and layer j. The change in the model's output distribution provides evidence about what information was contained in the patched activation.

Concepts. We use capitalization to denote an abstract concept, (e.g. CAT). Let C be an abstract concept, then we denote C^ℓ its language-specific version. Further, we define $w(C^\ell)$ as the set of words expressing the abstract concept C in language ℓ. For example, if $C = \text{CAT}$ and $\ell = \text{EN}$ we have $w(C^\ell) = \{\text{"cat"}\}$ and similarly $w(C^{\text{DE}}) = \{\text{"Katze"}, \text{"Kater"}\}$. Note that 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 (e.g. $\{\text{"Katze"}, \text{"Kat"}\}$). Therefore, when we track different sets of tokens W, (e.g. $W \in \{w(C_1^{\text{IT}}), w(C_1^{\text{ZH}}), w(C_2^{\text{IT}}), w(C_2^{\text{ZH}}), w(C_1^{\text{EN}}) \cup w(C_2^{\text{EN}})\} = \mathcal{W}$), we ensure that there is no token in common between any pair of $W_1, W_2 \in \mathcal{W}$ with $W_1 \neq W_2$.

Prompt design. We use the same translation prompt template as (Wendler et al., 2024) that we denote TP(input language, output language, concept). For example, TP(EN, FR, CLOUD) could be:

English: "computer" - Français: "ordinateur"
...
English: "ant" - Français: "fourmi"
English: "cloud" - Français: "

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

More formally, for a given concept C, input language $\ell^{(\text{in})}$, and output language $\ell^{(\text{out})}$, we construct a few-shot translation prompt TP($\ell^{(\text{in})}, \ell^{(\text{out})}, C$). This prompt contains k examples[2] of single-word translations of concepts $C_1, \ldots, C_k$ from $\ell^{(\text{in})}$ to $\ell^{(\text{out})}$, concluding with the model being tasked to translate C from $\ell^{(\text{in})}$ to $\ell^{(\text{out})}$. Using $C^{\ell^{(\text{in})}}$ as a shortcut for $v \in w(C^{\ell^{(\text{in})}})$, TP($\ell^{(\text{in})}, \ell^{(\text{out})}, C$) looks like:

$\ell^{(\text{in})}$: "$C_1^{\ell^{(\text{in})}}$" - $\ell^{(\text{out})}$: "$C_1^{\ell^{(\text{out})}}$"
...
$\ell^{(\text{in})}$: "$C_k^{\ell^{(\text{in})}}$" - $\ell^{(\text{out})}$: "$C_k^{\ell^{(\text{out})}}$"
$\ell^{(\text{in})}$: "$C^{\ell^{(\text{in})}}$" - $\ell^{(\text{out})}$: "

We denote TP$^{\text{concept}}(\ell^{(\text{in})}, \ell^{(\text{out})}, C)$ as TP$(\ell^{(\text{in})}, \ell^{(\text{out})}, C)$ cut at the last token of $C^{\ell^{(\text{in})}}$. For example, in our previous example, TP$^{\text{concept}}(\text{EN}, \text{FR}, \text{CLOUD})$ would be:

English: "computer" - Français: "ordinateur"
...
English: "ant" - Français: "fourmi"
English: "cloud

We expect that the last token of such prompts is where the model stores its latent representation of $C^{\ell^{(\text{in})}}$.

Importantly, whether the model correctly answers TP is determined by its next token prediction. In our prompt example, the next token predicted should be "nu", the first token of "nuage". Thus, we can track $P(C^\ell)$[3], 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$^{\text{FR}}$.

4 Exploratory analysis with patching

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

[2] in our study we used $k = 5$

[3] We use simplified notation $P(C^\ell)$ rather than $P(C^\ell|\text{TP})$ throughout. While the conditional notation would be more precise for the initial case, our patching experiments involve multiple conditioning factors (target prompt, source prompt, patching configuration, layer) that would make the full notation unwieldy. We therefore adopt this simplified notation for clarity.

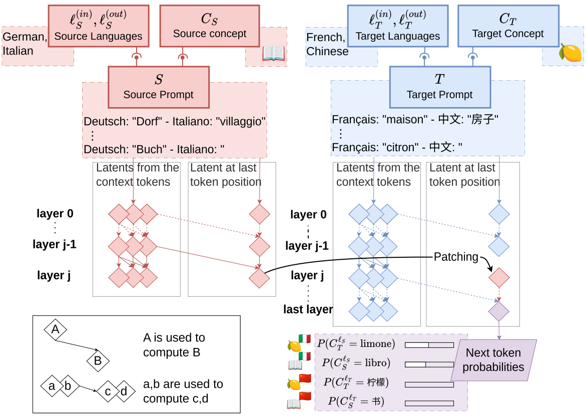

$\ell_S^{(in)}, \ell_S^{(out)}$
Source Languages
German, Italian
C_S
Source concept
$\ell_T^{(in)}, \ell_T^{(out)}$
Target Languages
French, Chinese
C_T
Target Concept
S
Source Prompt
Deutsch: "Dorf" - Italiano: "villaggio"
⋮
Deutsch: "Buch" - Italiano: "
T
Target Prompt
Français: "maison" - 中文: "房子"
⋮
Français: "citron" - 中文: "
Latents from the context tokens
Latent at last token position
Latents from the context tokens
Latent at last token position
layer 0
layer j-1
layer j
layer 0
layer j-1
layer j
last layer
Patching
A
B
A is used to compute B
a
b
c
d
a,b are used to compute c,d
$P(C_T^{\ell_S} = \text{limone})$
$P(C_S^{\ell_S} = \text{libro})$
$P(C_T^{\ell_T} = \text{柠檬})$
$P(C_S^{\ell_T} = \text{书})$
Next token probabilities

Figure 2: For two given concepts, e.g., BOOK and LEMON, we construct a source prompt for translating BOOK from German to Italian, and a target prompt for translating LEMON from French to Chinese. Then we extract the residual stream of the last token at a single layer j from the source prompt and insert it at the corresponding position and layer 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.

entangled, a decomposition like this should not exist: varying the language would mean varying the concept and vice versa.

4.1 Experimental design

We start our analysis with an exploratory experiment on Llama 2 7B (Touvron et al., 2023). We use 5-shots translation prompts to create paired source $S = \text{TP}(\ell_S^{(\text{in})}, \ell_S^{(\text{out})}, C_S)$ and target prompt $T = \text{TP}(\ell_T^{(\text{in})}, \ell_T^{(\text{out})}, C_T)$ datasets with different concept, input languages and output languages[4]. If not mentioned otherwise, ℓ_S and ℓ_T refer to the output language of S and T.

Similar to (Variengien and Winsor, 2023), we would like to infer at which layers the output language and the concept enter the residual stream $h_{n_T}^{(j)}(T)$ respectively and whether they can vary independently for our task. In order to investigate this question, we perform the experiment depicted in Figure 2. For each transformer block f_j we create two parallel forward passes, one processing the source prompt S which tokens are $(s_1, \dots, s_{n_S})$ and one processing the target prompt $T = (t_1, \dots, t_{n_T})$. While doing so, we extract the residual stream of the last token of the source prompt after layer j, $h_{n_S}^{(j)}(S)$, and insert it at the same layer at position n_T in the forward pass of the target prompt, i.e., by setting $h_{n_T}^{(j)}(T) = h_{n_S}^{(j)}(S)$ and subsequently completing the altered forward pass. From the resulting next token distribution, we compute $P(C_S^{\ell_S}), P(C_S^{\ell_T}), P(C_T^{\ell_S})$, and $P(C_T^{\ell_T})$.

4.2 Results

In this experiment, we perform the patching at one layer at a time and report the probability that is assigned to $P(C_S^{\ell_S}), P(C_S^{\ell_T}), P(C_T^{\ell_S})$, and $P(C_T^{\ell_T})$. As a result we obtain Figure 3 in which we report means and 95% confidence interval over 200 examples.

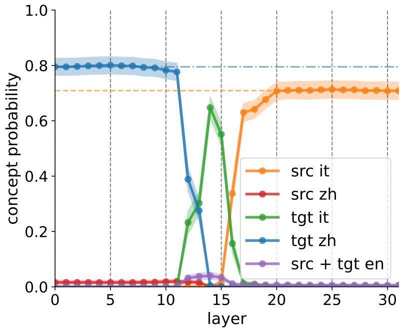


Figure 3: Our first patching experiment with a DE to IT source prompt and a FR to ZH target prompt with different concepts. The x-axis shows at which layer the patching was performed and the y-axis shows the probability of predicting the correct concept in language ℓ (see legend). In the legend, the 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.

Interpretation. We observe the following pattern while patching at different layers (see Figure 3):

- Layers 0–11: Target concept decoded in target language, resulting in large $P(C_T^{\text{ZH}})$.
- Layers 12–16: Target concept decoded in source language, resulting in large $P(C_T^{\text{IT}})$.
- Layers 16–31: Source concept decoded in source language, resulting in large $P(C_S^{\text{IT}})$.

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 residual stream 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

[4] See details in Appendix B

hypothesis is correct, patching at layer 12 would overwrite $z_{\ell_T^{(out)}}$ with $z_{\ell_S^{(out)}}$. The green spike between layer 12 and 16 indicates that at those layer, the concept is not yet represented, so the model keep outputing the target concept but in the source language.

In later layers, the model determines the concept: from layer 16 onwards, the source concept is decoded. This suggests that $z_{C_T^{\ell^{(\text{out})}}}$ is overwritten at layer 16.[5]

Hypotheses. We are left with two hypotheses compatible with these results:

- **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 $C^{\ell^{(\text{out})}}$.

5 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** (3) to show that our findings hold for other models.

Further evidence experiment. In the experiments in Sec. 4 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 the residual stream at the last token of the prompt, we overwrite them at the last token of the word to be translated. In order to do that, for the source prompt, we use TP$^{\text{concept}}$ instead of TP ($S = \text{TP}^{\text{concept}}(\ell_S^{(\text{in})}, \ell_S^{(\text{out})}, C_S)$). This means that we collect the activations at the last token of $C_S^{\ell_S^{(\text{in})}}$.

Let ρ_T denote the position of that token in the target prompt. 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*. 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 $h_{\rho_T}^{(j)}(T) = h_{-1}^{(j)}(S), \ldots, h_{\rho_T}^{(L)}(T) = h_{-1}^{(L)}(S)$[6].

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 multiple 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

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

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 for all i, $z_{C_i} = z_{C_S}$:

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

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 representation 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 $P(C_S^{\ell_T})$.

[5] In Appendix A, we collected additional experimental results investigating the right part of Figure 3 more deeply and in Appendix C 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. Figure 7. 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. Figure 8. We find that indeed before layer 11 there is no translation task specific information in the residual stream, only prompt-template specific information.

[6] Note that we use Python indexing, where -1 denotes the last token.

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 Figure 4 we observe that in both experiments, we obtain very high probability for the *source concept in the target language* $P(C_S^{\mathrm{ZH}})$ from layers 0 to 15, i.e., exactly until the latents at the last token stop attending to the last concept-token.

Therefore, Figure 4 (a) supports that we are indeed either in **H1** or **H2**, since *as planned* we successfully decode *source concepts in the target language* $P(C_S^{\mathrm{ZH}})$ 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, Figure 4 (b) supports that we are in **H1** and not in **H2** because patching in the mean keeps $P(C_S^{\mathrm{ZH}})$ intact 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 $P(C_S^{\mathrm{ZH}})$. [7]

Other models. In Appendix D we perform the experiments from Sec. 4 and Sec. 5 on several other models, varying in size, training data and architecture namely, Mistral 7B (Jiang et al., 2023), Llama 3 8B (Dubey et al., 2024), Qwen 1.5 7B (Bai et al., 2023), Llama 2 70B, and Aya 23 8B (Aryabumi et al., 2024) which was specifically trained to be multilingual. We observe the same improvement when we take the mean of a concept across languages for all these models, suggesting that they are all operating under H1 and use some language-agnostic concept representation.

6 Generating descriptions for latents

We just showed that LLMs can effectively translate concept representations averaged across multiple languages, providing evidence for language-agnostic concept representations. In this section,

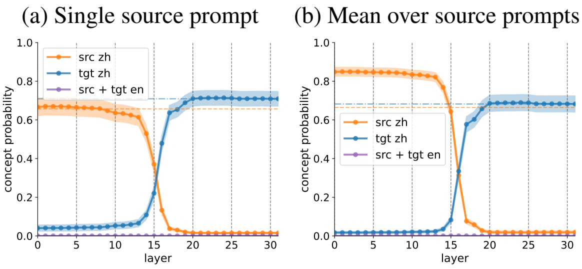


Figure 4: Here we use different input languages (DE, FR), different concepts, different output languages (IT, ZH) in (a). In (b) we use multiple source input languages DE, NL, ZH, ES, RU and source output languages IT, FI, ES, RU, KO. We patch at the last token of the concept-word at all layers from 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.

we explore whether these mean representations, which theoretically capture language-independent concepts, can be described by the model in natural language as effectively as concepts expressed in a single language. This approach builds on the definition modeling paradigm (Noraset et al., 2017; Mickus et al., 2022; Chang and Chen, 2019; Gardner et al., 2022), which uses natural language generation as a transparent evaluation of semantic representations.

6.1 Definition prompt

In order to do that, we introduce a new prompt template that tasks the model to describe a concept in natural language. More precisely, given a concept C and a language ℓ, we construct a few shot definition prompt DP(ℓ, C) of the form:

"C_1^ℓ" : "D_1^ℓ"
...
"C^ℓ" : "

where $C_1^\ell, \dots, C_n^\ell$ are concepts in language ℓ and $D_1^\ell, \dots, D_n^\ell$ are their descriptions in language ℓ. We denote DP$^{\text{concept}}(\ell, C)$ the prompt template that ends at the last token of C^ℓ. For example, DP$^{\text{concept}}(\text{EN}, \text{ANT})$ could be:

"apple" : "Fruit with red or yellow or green skin"
...
"ant

[7] 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 $P(C_S^{\mathrm{ZH}})$ (see App. Figure 11,12 (b)).

For each language, we constructed a definition dataset using the first concept in BabelNet (Navigli et al., 2021) associated with each of the 200 picturable words from the Basic English word list from Wikipedia[8]. For each concept, BabelNet provides several definitions in different languages.

6.2 Patching setup

For languages $\ell_S^1, \dots, \ell_S^n$ and ℓ_T, and concepts $C_S \neq C_T$, we construct a target prompt $T = \text{DP}(\ell_T, C_T)$ and two sets $\mathcal{S}$ of source prompts:

- **From translations**: for each language ℓ_S^i we pick an input language $\ell_{(\text{in})}^i$ and choose

$$\mathcal{S}_{\text{trans}} = \{\text{TP}^{\text{concept}}(\ell_{(\text{in})}^i, \ell_S^i, C_S)\}_{i \in \{1,\dots,n\}}.$$

- **From definitions**: we choose

$$\mathcal{S}_{\text{def}} = \{\text{DP}^{\text{concept}}(\ell_S^i, C_S)\}_{i \in \{1,\dots,n\}}.$$

Then, to generate a definition of C_S in language ℓ_T, for all layers, we patch the latents of the last token averaged over the source prompts from $\mathcal{S}$ to the last token of C_T in the target prompt and let the model generate the definition as depicted in Figure 5. More formally, we patch by setting

$$h_{\rho_T}^{(j)}(T) = \frac{1}{n} \sum_{i=1}^{n} h_{-1}^{(j)}(S_i)$$

for $j \in \{1, \dots, m\}$ and ρ_T the position of the last token of C_T in the target prompt.

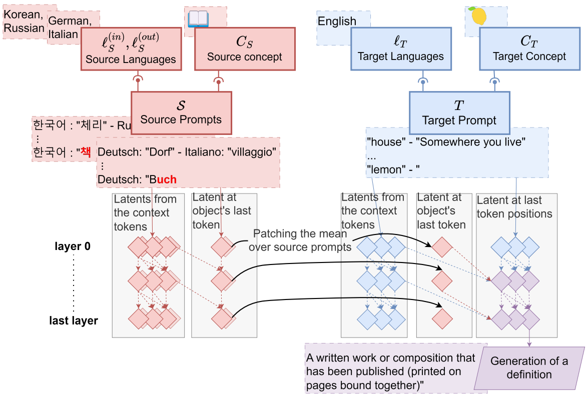

Korean, Russian
German, Italian
$\ell_S^{(in)}, \ell_S^{(out)}$ Source Languages
C_S Source concept
English
ℓ_T Target Languages
C_T Target Concept
$\mathcal{S}$ Source Prompts
T Target Prompt
한국어 : "체리" - Ru
⋮
한국어 : "책
Deutsch: "Dorf" - Italiano: "villaggio"
⋮
Deutsch: "Buch
"house" - "Somewhere you live"
...
"lemon" - "
Latents from the context tokens
Latent at object's last token
Patching the mean over source prompts
Latents from the context tokens
Latent at object's last token
Latent at last token positions
layer 0
last layer
A written work or composition that has been published (printed on pages bound together)"
Generation of a definition

Figure 5: Illustration of the patching setup for the definition prompt experiment. We patch the latents of the last token of the source prompts from $\mathcal{S}$ to the last token of C_T in the target prompt.

6.3 Experiment

To compare the quality of the definitions of C_S generated by the model in our experiment, we use the following baselines:

- **Ground truth**: We use a random definition from BabelNet.
- **Prompting**: We use the definition generated by the model when prompted with $\text{DP}(\ell_T, C_S)$.
- **Word patching**: We replace $C_T^{\ell_T}$ with $C_S^{\ell_S^j}$ for a random $j \in \{1, \dots, n\}$ and let the model generate the definition. In this setting, the model is tasked to generate a definition of a word in a language different from the target prompt's language.
- **Repeat word**: We use $C_S^{\ell_T}$ as the definition.

To evaluate the different definitions, we use the `sentence-transformers` library (Reimers and Gurevych, 2019) to run `paraphrase-multilingual-mpnet-base-v2`[9], a semantic similarity model distilled from (Song et al., 2020) using the methods from (Reimers and Gurevych, 2020). To assess the quality of the generated definition, we compute the similarity score between embeddings of the definition and the mean embedding of the ground truth definitions[10].

6.4 Results

We report the mean similarity score between the definition and the mean embedding of the ground truth definitions for LLama 2 7B in Figure 6. To give an idea of the scale of the similarity scores, we also report the mean similarity between the definition embedding and the embeddings of the definitions of the other concepts in the dataset.

We find that patching concept representations from one language to another allows the model to generate high-quality definitions, comparable to or better than direct prompting. The fact that patching mean representations across multiple source languages leads to slightly better results suggests that the model's concept representations are indeed language-agnostic – if they were language-specific, averaging across languages would likely degrade

[8] https://en.wiktionary.org/wiki/Appendix:Basic_English_word_list#Things_-_200_picturable_words

[9] https://huggingface.co/sentence-transformers/paraphrase-multilingual-mpnet-base-v2

[10] To be able to compare the score of the generated definition with the ground truth baseline, we compute its similarity score with the mean embedding of the **other** ground truth definitions - excluding the one that was randomly chosen as the ground truth baseline

[11] For some concepts and languages, BabelNet does not provide any definition.

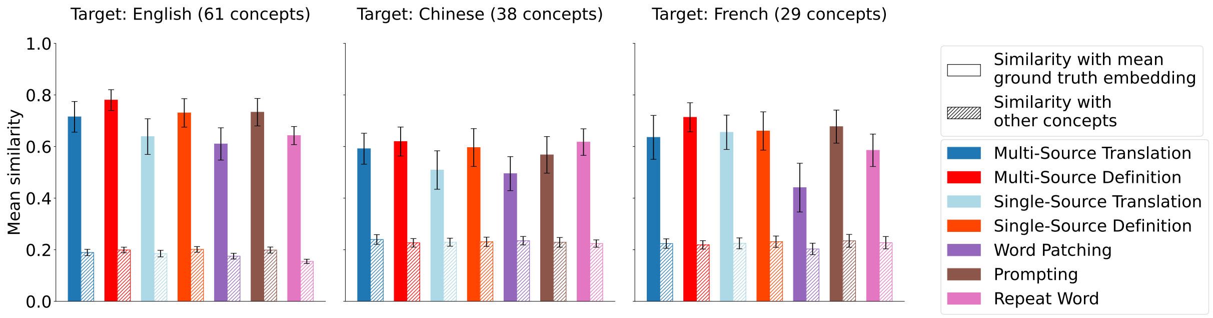


Figure 6: Mean similarity between the definition and the mean embedding of the ground truth definitions, as well as the mean similarity between the definition embedding and the embeddings of the definitions of the other concepts in the dataset. The results are presented for three target languages: English (with source languages Italian, Finnish, Spanish, Korean, and input languages for source translations: German, Dutch, Chinese, Russian), Chinese (using the same languages), and French (with source languages Korean, Japanese, Estonian, Finnish and English as input language for source translations). We report means and 95% Gaussian confidence intervals computed over a dataset of various sizes[11].

performance. This aligns with our translation experiment findings and adds another perspective on how LLMs process multilingual information.

Additionally, the comparable performance between patching from definitions and translations indicates that the model builds similar concept representations regardless of whether it processes a translation or definition prompt. This suggests a unified internal representation of concepts that generalizes across different types of language tasks.

Other models. In Appendix D.3 we show that those findings generalize to other models and languages.

7 Conclusion

In this paper, we showed that transformers use language-agnostic latent representations of concepts when processing word-level translation prompts. We achieved this by patching latents between parallel forward passes for translation prompts that differed in both input and output languages, as well as in the specific concepts being translated. Our main finding was that translation performance *improves* when the transformer is forced to translate a concept representation averaged across multiple languages. This finding speaks for language-agnostic concept representations. As we argued, for language-agnostic concept representations, taking the mean representation of a concept across languages should not impair the LLM's ability to translate this concept. In contrast, for language-specific ones, taking the mean should result in interference between the different language-specific versions of the concept. Thus, our results are consistent with findings from previous work (Wendler et al., 2024) indicating that Llama-2 represents concepts in a concept space independent of the language of the prompt. Our work also provides evidence that findings from BERT models (Conneau et al., 2020b; Pires et al., 2019) generalize to a wide range of decoder-only transformers. Our findings open several important avenues for future research. Understanding these disentangled representations could improve cross-lingual transfer learning and reduce the computational costs of multilingual training by leveraging shared concept spaces more efficiently. Moreover, our results provide mechanistic insights into the root causes of Western cultural biases in multilingual LLMs, suggesting that biases may propagate through the shared concept space. We hope these contributions will guide efforts to build more controllable, efficient, and culturally-aware multilingual language models.

Limitations

In this work, we studied how transformers represent concepts when processing multilingual text. However, we only considered very simple concepts, maybe some more complex concepts would have shown a different behavior. Also, we did not study language-specific concepts like "Waldeinsamkeit", "The feeling of solitude and connectedness to nature when being alone in the woods.". It would be interesting to see how those are represented.

Furthermore, more fine-grained probing will be required to determine to which degree transformers are able to specialize a concept to a language and whether concepts and languages are entangled in more subtle ways.

Acknowledgment

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

References

Viraat Aryabumi, John Dang, Dwarak Talupuru, Saurabh Dash, David Cairuz, Hangyu Lin, Bharat Venkitesh, Madeline Smith, Jon Ander Campos, Yi Chern Tan, Kelly Marchisio, Max Bartolo, Sebastian Ruder, Acyr Locatelli, Julia Kreutzer, Nick Frosst, Aidan Gomez, Phil Blunsom, Marzieh Fadaee, Ahmet Üstün, and Sara Hooker. 2024. Aya 23: Open weight releases to further multilingual progress. *Preprint*, arXiv:2405.15032.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. *Preprint*, arXiv:2309.16609.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Pengfei Cao, Yuheng Chen, Zhuoran Jin, Yubo Chen, Kang Liu, and Jun Zhao. 2024. One mind, many tongues: A deep dive into language-agnostic knowledge neurons in large language models. *arXiv preprint arXiv:2411.17401*.

Ting-Yun Chang and Yun-Nung Chen. 2019. What does this word mean? explaining contextualized embeddings with natural language definition. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6064–6070.

Haozhe Chen, Carl Vondrick, and Chengzhi Mao. 2024a. Selfie: Self-interpretation of large language model embeddings. *arXiv preprint arXiv:2403.10949*.

Yuheng Chen, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. 2024b. Journey to the center of the knowledge neurons: Discoveries of language-independent knowledge neurons and degenerate knowledge neurons. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17817–17825.

Ethan A. Chi, John Hewitt, and Christopher D. Manning. 2020. Finding universal grammatical relations in multilingual bert. *Preprint*, arXiv:2005.04511.

Nadezhda Chirkova and Vassilina Nikoulina. 2024. Zero-shot cross-lingual transfer in instruction tuning of large language models. *arXiv preprint arXiv:2402.14778*.

Rochelle Choenni and Ekaterina Shutova. 2020. What does it mean to be language-agnostic? probing multilingual sentence encoders for typological properties. *arXiv preprint arXiv:2009.12862*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020a. Unsupervised cross-lingual representation learning at scale. *Preprint*, arXiv:1911.02116.

Alexis Conneau, Shijie Wu, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov. 2020b. Emerging cross-lingual structure in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6022–6034.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Constanza Fierro, Negar Foroutan, Desmond Elliott, and Anders Søgaard. 2025. How do multilingual language models remember facts? *arXiv preprint arXiv:2410.14387*.

Jaden Fiotto-Kaufman, Alexander R Loftus, Eric Todd, Jannik Brinkmann, Caden Juang, Koyena Pal, Can Rager, Aaron Mueller, Samuel Marks, Arnab Sen Sharma, et al. 2024. Nnsight and ndif: Democratizing access to foundation model internals. *arXiv preprint arXiv:2407.14561*.

Noah Gardner, Hafiz Khan, and Chih-Cheng Hung. 2022. Definition modeling: literature review and dataset analysis. *Applied Computing and Intelligence*, 2(1):83–98.

Atticus Geiger, Zhengxuan Wu, Hanson Lu, Josh Rozner, Elisa Kreiss, Thomas Icard, Noah Goodman, and Christopher Potts. 2022. Inducing causal structure for interpretable neural networks. In *International Conference on Machine Learning*, pages 7324–7338. PMLR.

Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. 2024. Patchscope: A unifying framework for inspecting hidden representations of language models. *arXiv preprint arXiv:2401.06102*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825*.

János Kramár, Tom Lieberum, Rohin Shah, and Neel Nanda. 2024. Atp*: An efficient and scalable method for localizing llm behaviour to components. *arXiv preprint arXiv:2403.00745*.

Xiaochen Li, Zheng Xin Yong, and Stephen Bach. 2024. Preference tuning for toxicity mitigation generalizes across languages. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 13422–13440.

Jindřich Libovický, Rudolf Rosa, and Alexander Fraser. 2020. On the language neutrality of pre-trained multilingual representations. *arXiv preprint arXiv:2004.05160*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372.

Timothee Mickus, Kees Van Deemter, Mathieu Constant, and Denis Paperno. 2022. Semeval-2022 task 1: Codwoe–comparing dictionaries and word embeddings. *arXiv preprint arXiv:2205.13858*.

Basel Mousi, Nadir Durrani, Fahim Dalvi, Majd Hawasly, and Ahmed Abdelali. 2024. Exploring alignment in shared cross-lingual spaces. *arXiv preprint arXiv:2405.14535*.

Benjamin Muller, Yanai Elazar, Benoît Sagot, and Djamé Seddah. 2021. First align, then predict: Understanding the cross-lingual ability of multilingual bert. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2214–2231.

Roberto Navigli, Michele Bevilacqua, Simone Conia, Dario Montagnini, Francesco Cecconi, et al. 2021. Ten years of babelnet: A survey. In *IJCAI*, pages 4559–4567. International Joint Conferences on Artificial Intelligence Organization.

Thanapon Noraset, Chen Liang, Larry Birnbaum, and Doug Downey. 2017. Definition modeling: Learning to define word embeddings in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Nostalgebraist. 2020. Interpreting gpt: The logit lens. LessWrong.

OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.17548*.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual bert? *Preprint*, arXiv:1906.01502.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Uri Shaham, Jonathan Herzig, Roee Aharoni, Idan Szpektor, Reut Tsarfaty, and Matan Eyal. 2024. Multilingual instruction tuning with just a pinch of multilinguality. *arXiv preprint arXiv:2401.01854*.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, and Jason Wei. 2022. Language models are multilingual chain-of-thought reasoners. *Preprint*, arXiv:2210.03057.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Preprint*, arXiv:2004.09297.

Karolina Stańczak, Edoardo Ponti, Lucas Torroba Hennigen, Ryan Cotterell, and Isabelle Augenstein. 2022. Same neurons, different languages: Probing morphosyntax in multilingual pre-trained models. *arXiv preprint arXiv:2205.02023*.

Tianyi Tang, Wenyang Luo, Haoyang Huang, Dongdong Zhang, Xiaolei Wang, Xin Zhao, Furu Wei, and Ji-Rong Wen. 2024. Language-specific neurons: The key to multilingual capabilities in large language models. *arXiv preprint arXiv:2402.16438*.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Cogan, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. 2024. Gemma 2: Improving open language models at a practical size. *Preprint*, arXiv:2408.00118.

Eric Todd, Millicent L Li, Arnab Sen Sharma, Aaron Mueller, Byron C Wallace, and David Bau. 2023. Function vectors in large language models. *arXiv preprint arXiv:2310.15213*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Alexandre Variengien and Eric Winsor. 2023. Look before you leap: A universal emergent decomposition of retrieval tasks in language models. *arXiv preprint arXiv:2312.10091*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Chris Wendler, Veniamin Veselovsky, Giovanni Monea, and Robert West. 2024. Do llamas work in english? on the latent language of multilingual transformers. *arXiv preprint arXiv:2402.10588*.

Zhaofeng Wu, Xinyan Velocity Yu, Dani Yogatama, Jiasen Lu, and Yoon Kim. 2024. The semantic hub hypothesis: Language models share semantic representations across languages and modalities. *Preprint*, arXiv:2411.04986.

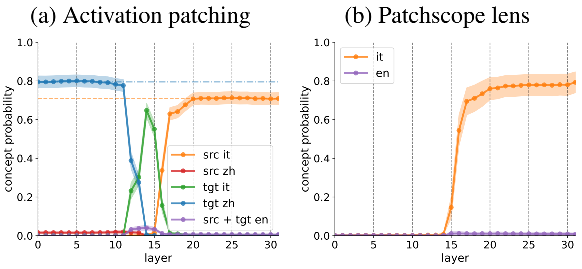
Zhihui Xie, Handong Zhao, Tong Yu, and Shuai Li. 2022. Discovering low-rank subspaces for language-agnostic multilingual representations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5617–5633, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mt5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

Hongchuan Zeng, Senyu Han, Lu Chen, and Kai Yu. 2024. Converging to a lingua franca: Evolution of linguistic regions and semantics alignment in multilingual large language models. *Preprint*, arXiv:2410.11718.

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 Figure 7. The results of this experiment corroborate the findings presented in Section 4. To enable a convenient comparison of the experimental results, we also include Figure 3 in Figure 7.


(a) Activation patching

(b) Patchscope lens

Figure 7: (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 Translation Pair Construction

To ensure reproducibility of our experiments, we provide the pseudocode for constructing translation pairs used in our activation patching experiments. The complete implementation is available in our codebase at `notebooks/obj_patch_translation.ipynb`.

Key constraints:

- **Different concepts**: We ensure that source and target words represent different concepts to enable meaningful patching experiments.

- **No token collisions**: As described in Section 3, we track sets of tokens $w(C^\ell)$ for each concept-language pair. To ensure clean probability measurements, we verify that there is no overlap between the token sets of paired concepts across all languages used in the experiment.

- **Randomization**: Pairs are shuffled before selection to avoid systematic biases in concept or language selection.

For our experiments, we typically use $n = 200$ pairs, with concepts drawn from the 200 picturable words from the Basic English word list, ensuring

Algorithm 1 Construction of Translation Pairs for Patching Experiments

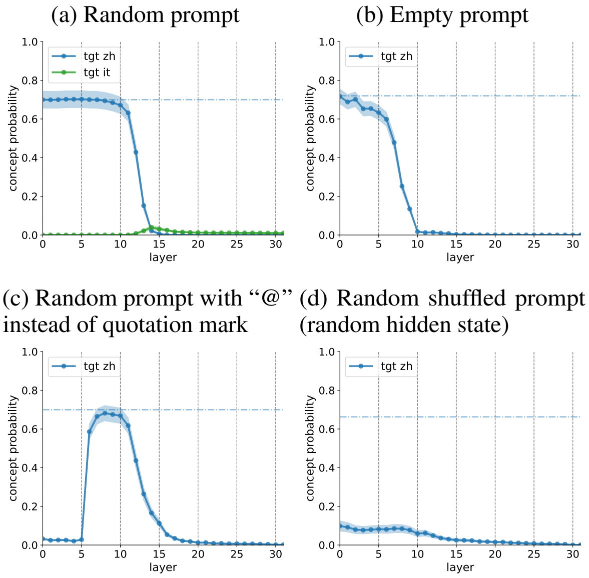
```
Require: Set of source languages $\mathcal{L}_S$, target language $\ell_T$, number of pairs $n$
Ensure: Set of valid translation pairs $\mathcal{P}$
Load BabelNet translations for all languages
$\mathcal{T}_S \leftarrow$ GetTranslations($\mathcal{L}_S$)
$\mathcal{T}_T \leftarrow$ GetTranslations($\ell_T$)
$\mathcal{P} \leftarrow \emptyset$                          ▷ Initialize valid pairs
$\mathcal{A} \leftarrow$ Combinations($\mathcal{T}_S, \mathcal{T}_T$) ▷ All possible pairs
Shuffle($\mathcal{A}$)                        ▷ Randomize order
for $(w_S, w_T) \in \mathcal{A}$ do
    if concept($w_S$) $=$ concept($w_T$) then
        continue                   ▷ Skip same concepts
    end if
    if HasTokenCollisions($w_S, w_T$) then
        continue               ▷ Skip pairs with token overlap
    end if
    $\mathcal{P} \leftarrow \mathcal{P} \cup \{(w_S, w_T)\}$
    if $|\mathcal{P}| \geq n$ then
        break           ▷ Sufficient pairs collected
    end if
end for
return $\mathcal{P}$
```

sufficient statistical power while maintaining computational feasibility.

C Random prompt task experiment

In order to investigate the leftmost part of Figure 7a 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. 4, 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.


(a) Random prompt

(b) Empty prompt

(c) Random prompt with “@” instead of quotation mark

(d) Random shuffled prompt (random hidden state)

Figure 8: (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.

Same template. In Figure 8a, 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 Figure 8a, 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.

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 Figure 8b. In contrast to Figure 8a, 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 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 “@” (Figure 8c) in the random prompt, i.e.,

A: @CAT$^{\text{DE}}$@ - B: @DOG$^{\text{IT}}$@
A: @OWL$^{\text{JA}}$@ - B: @SUN$^{\text{HI}}$@
A: @ICE$^{\text{FR}}$@ - 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 Figure 8d 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.

D Other models and languages

In this section, we report results for additional models, namely, Mistral 7B (Jiang et al., 2023), Llama 3 8B (Dubey et al., 2024), Qwen 1.5 7B (Bai et al., 2023) and Llama 2 70B (Touvron et al., 2023). We also include Aya 23 8B (Aryabumi et al., 2024) for the mean patching experiment in App D.2.

D.1 Exploratory analysis

The results of the exploratory analysis outlined in Sec. 4 are in Figure 9.

As can be seen in Figure 9, the target concept in source language spike is smaller for Llama 3, Mistral 7B v0.3 and Qwen 1.5 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. 5.

In addition, instead of just patching in the mean over different language pairs (Figure 11c, 12c), we also patch in the mean over contexts composed of different concept words in Figure 11b, 12b. 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 Figure 11 and Figure 12 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 Figure 9, Figure 11 and Figure 12 are on the next two pages.

D.3 Similarity comparison

We experiment with other languages and models in Figure 13 and get the same trends as with our results in Figure 6. We also provide results for another experiment in which instead of measuring embedding similarities, we measure perplexity on ground truth definitions in Figure 14. We did not include this metric in our analysis as it seemed to have less granularity and is more sensitive to the syntax rather than being focused on the semantic.

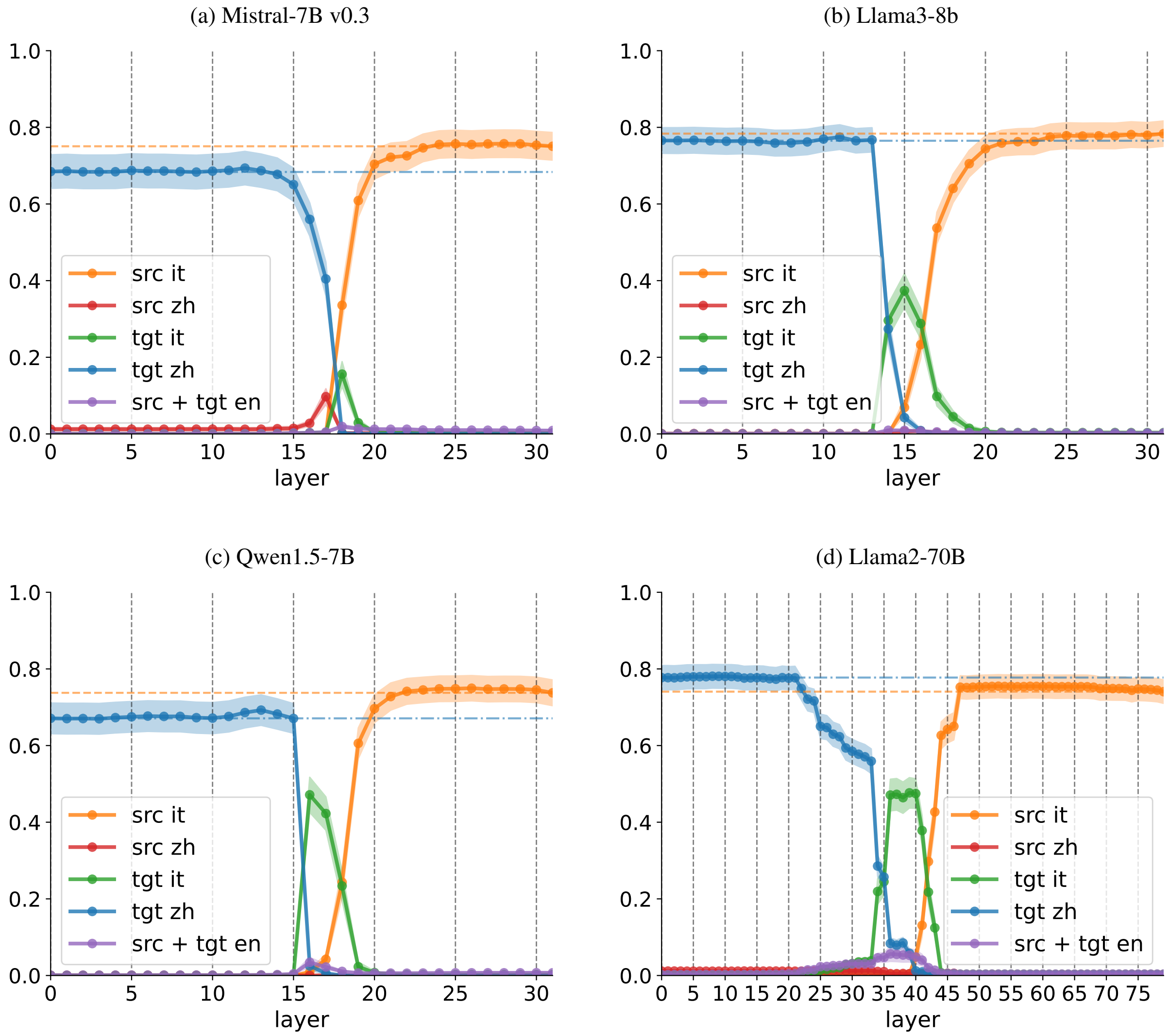


Figure 9: Our first patching experiment with a DE to IT source prompt and a FR to ZH target prompt with different concepts. We patch at the last token. For each of the plots the x-axis shows at which layer the patching was performed during the forward pass on the target prompt and the y-axis shows the 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.

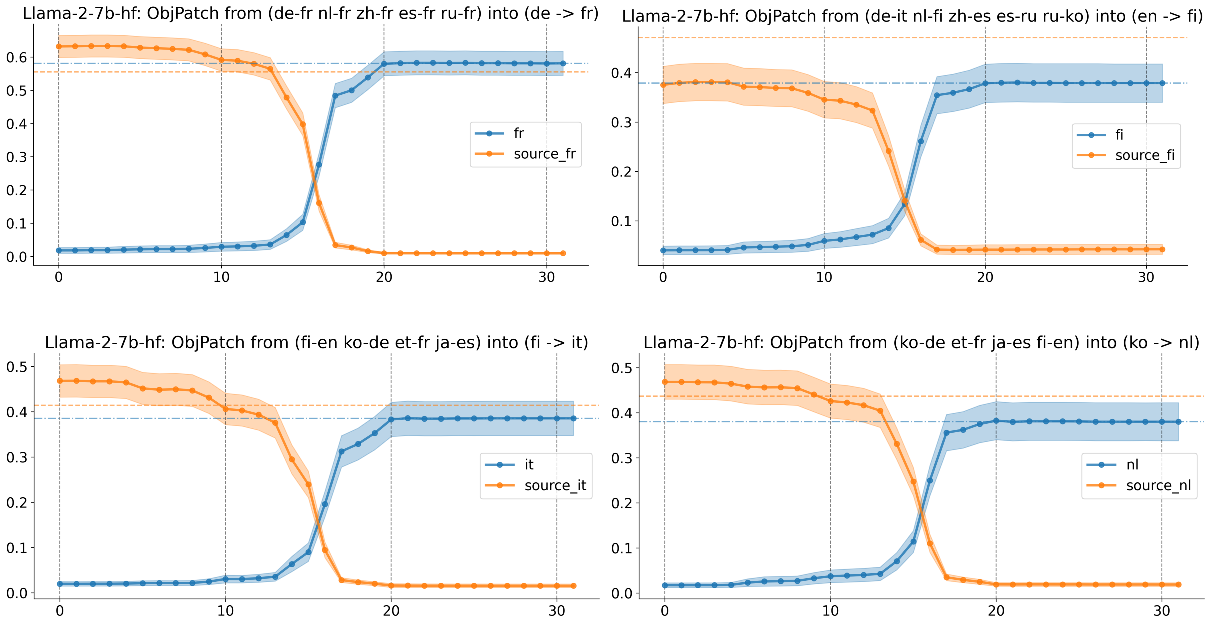


Figure 10: Mean patching experiment replicated on multiple languages with 200 pairs each and 95% confidence interval.

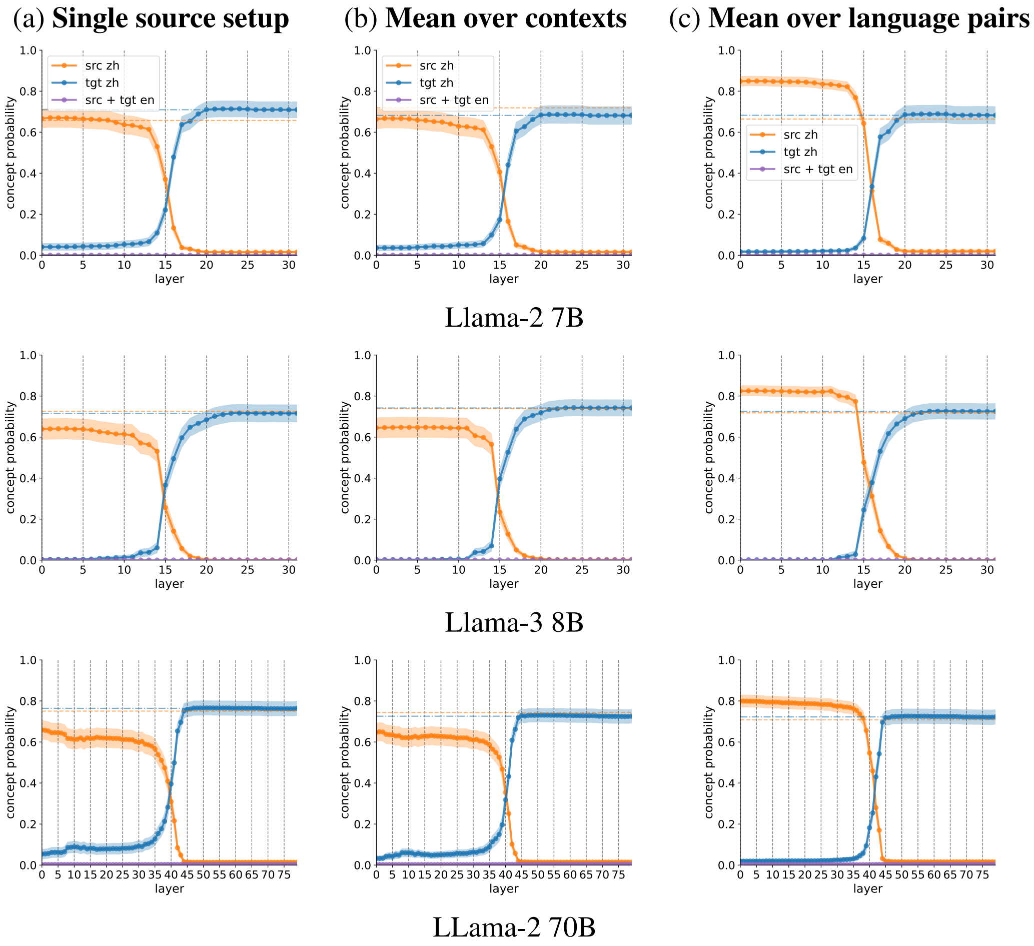
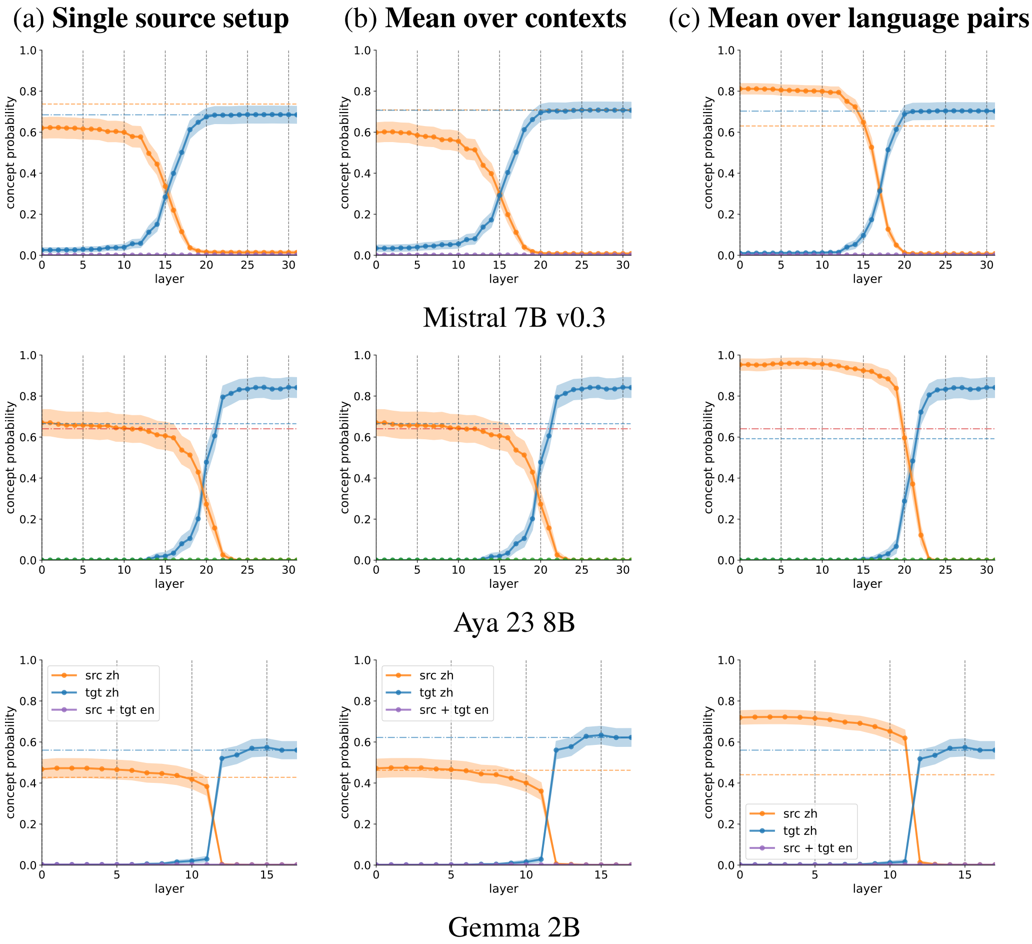


Figure 11: Here we use different input languages (DE, FR), different concepts, different output languages (IT, ZH) in (a). In (b) we use the same source and target language pairs as in (a). In (c) we use multiple source input languages DE, NL, ZH, ES, RU and output languages IT, FI, ES, RU, KO. We patch 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.



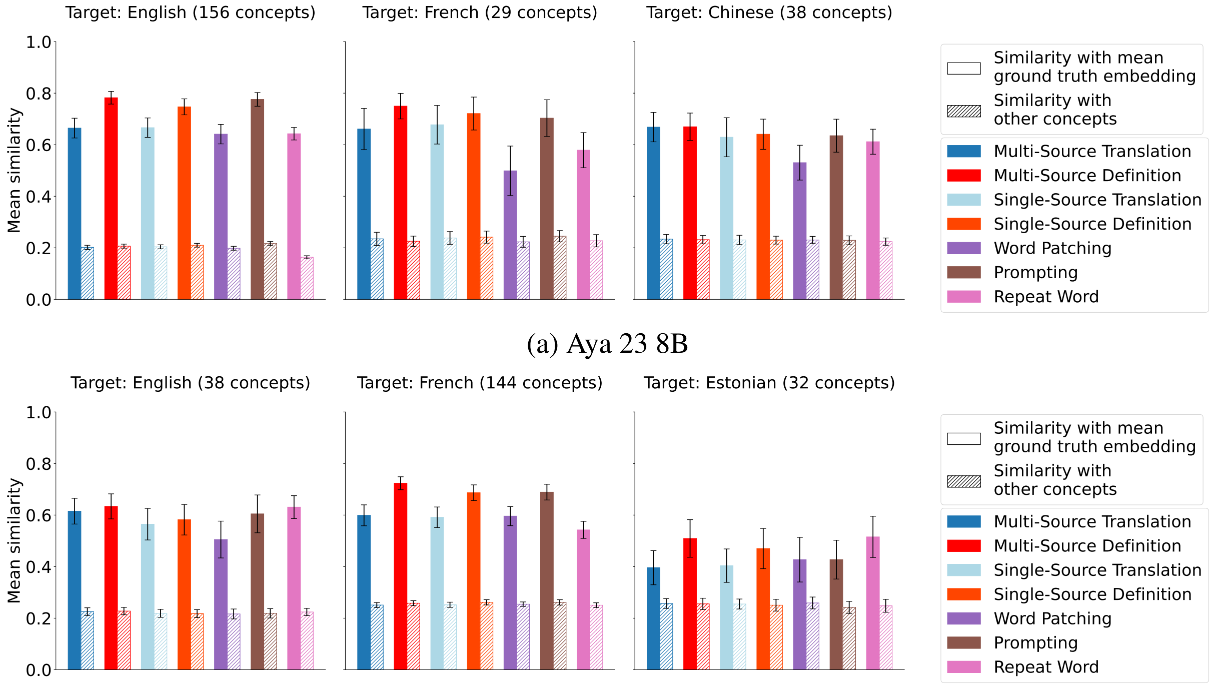
(a) **Single source setup** (b) **Mean over contexts** (c) **Mean over language pairs**

Mistral 7B v0.3

Aya 23 8B

Gemma 2B

Figure 12: 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.



(a) Aya 23 8B

(b) Gemma 2 2B

Figure 13: Mean similarity between the definition and the mean embedding of the ground truth definitions, as well as the mean similarity between the definition embedding and the embeddings of the definitions of the other concepts in the dataset. For Aya, the results are presented for three target languages: English (with source languages French and German and input language Spanish), French (with source languages Korean, Japanese, Estonian, Finnish and input language English), and Chinese (with source languages Italian, Finnish, Spanish, Russian, Korean and input languages German, Dutch, Chinese, Spanish, Russian). For Gemma, we show English (with source languages Italian, Finnish, Spanish, Russian, Korean and input languages German, Dutch, Chinese, Spanish, Russian), French (with source languages Spanish, German and input language Italian), and Estonian (with source languages English, French, Chinese, German and input language Hindi). We report means and 95% Gaussian confidence intervals computed over the dataset.

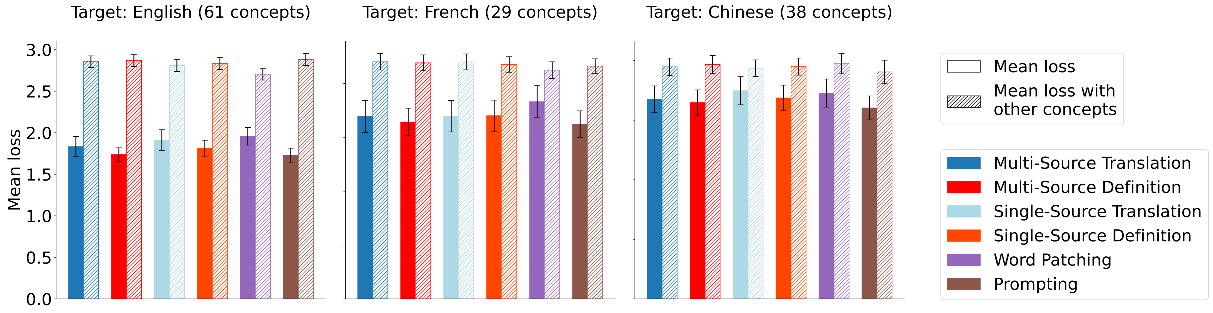


Figure 14: Mean loss on the ground truth definitions. We report 95% confidence intervals.