How Do Multilingual Language Models Remember Facts?

Constanza Fierro[†] **Negar Foroutan**[‡] **Desmond Elliott**[†] **Anders Søgaard**[†]

[†] Department of Computer Science, University of Copenhagen
[‡] EPFL

Abstract

Large Language Models (LLMs) store and retrieve vast amounts of factual knowledge acquired during pre-training. Prior research has localized and identified mechanisms behind knowledge recall; however, it has only focused on English monolingual models. The question of how these mechanisms generalize to non-English languages and multilingual LLMs remains unexplored. In this paper, we address this gap by conducting a comprehensive analysis of three multilingual LLMs. First, we show that previously identified recall mechanisms in English largely apply to multilingual contexts, with nuances based on language and architecture. Next, through patching intermediate representations, we localize the role of language during recall, finding that subject enrichment is language-independent, while object extraction is language-dependent. Additionally, we discover that the last token representation acts as a Function Vector (FV), encoding both the language of the query and the content to be extracted from the subject. Furthermore, in decoder-only LLMs, FVs compose these two pieces of information in two separate stages. These insights reveal unique mechanisms in multilingual LLMs for recalling information, highlighting the need for new methodologies—such as knowledge evaluation, fact editing, and knowledge acquisition—that are specifically tailored for multilingual LLMs.[1]

1 Introduction

Large Language Models (LLMs) learn extensive factual knowledge during pre-training, including propositional facts like "The capital of France is __" (Petroni et al., 2019). While multilingual models also acquire such knowledge, their performance varies significantly across languages (Kassner et al., 2021; Jiang et al., 2020; Yin et al., 2022), raising questions about whether this variation stems from language-specific storage or phrasing sensitivity (Elazar et al., 2021). Such assessments will be critical for determining the trustworthiness of multilingual LLMs, if trust relies on knowledge (Grasswick, 2010; Hawley, 2012; Nguyen, 2022).

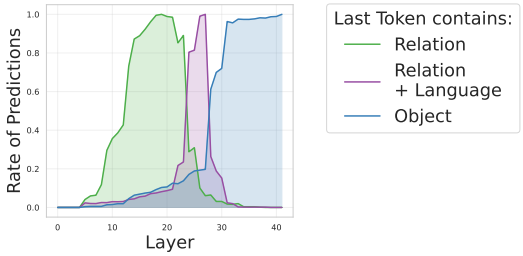


Figure 1: Aggregated patching results (§6) for EuroLLM. In propositional factual statements, e.g. "Paris is the capital of", the last token representation contains the function that solves the task. This function is formed in two stages: first, the relation to extract is encoded (green), and then the language is composed into the function (purple). Finally, the function is applied to the subject, and the predicted object is resolved (blue).

Mechanistic interpretability research has begun uncovering how models store and retrieve knowledge internally, with recent studies identifying specific components for knowledge storage (Meng et al., 2022; Sharma et al., 2024) and retrieval mechanisms (Geva et al., 2023; Chughtai et al., 2024). However, they have focused exclusively on English LLMs, mainly autoregressive ones.[2] Encoder-decoder architectures, which could enable better cross-lingual representations (Li et al., 2024), remain underexplored. Additionally, recent studies have shown that LLMs share circuits across languages for specific tasks (Ferrando and Costa-jussà, 2024; Zhang et al., 2025), but these are limited to syntactic tasks and do not address how concepts are represented or recalled cross-lingually. While

*Correspondence: Constanza Fierro <c.fierro@di.ku.dk>.

[1] https://github.com/constanzafierro/multilingual_factual_memorization

[2] Except Sharma et al. (2024), who extend these analyses to Mamba (Gu and Dao, 2023), a state-space language model.

Findings of the Association for Computational Linguistics: ACL 2025, pages 16052–16106
July 27 - August 1, 2025

Dumas et al. (2024) examined how disentangled language is from concepts using a translation task, here we analyze fact recall, which allows us to offer broader insights into how language is encoded.

In this paper, we study the mechanisms of factual recall in multilingual LLMs, focusing on two architectures: decoder-only (XGLM and EuroLLM) and encoder-decoder (mT5). We analyze a simple form of information extraction, where the input contains a subject and a relation, and the model predicts the corresponding object (e.g. "Paris" in the earlier example). Our analysis centers on three key questions: (1) Does the localization of factual knowledge in English LLMs extend to multilingual LLMs? (2) Are the factual recall mechanisms found for English LLMs also present in multilingual LLMs? (3) When does language play a role in the recall mechanism?

To address the first question, we use causal tracing analysis to assess if early MLP modules processing the final subject token are as decisive in multilingual models as in English-centric ones (Meng et al., 2022). Our results (§4) show that EuroLLM and mT5 exhibit strong causal effects for the last subject token—EuroLLM in earlier layers, and mT5 across all encoder layers. Additionally, in all models, both MLPs and MHSAs in later layers play a decisive role in recovering factual information—unlike in Meng et al. (2022), where only MHSAs were active at the late site.

Next, we investigate the second question by analyzing the three-step process described by Geva et al. (2023): the relation information flows to the final token, followed by subject information, and finally, the attention layers extract the object (§5). We find that in multilingual LLMs, subject information flows similarly to monolingual ones, but non-subject token flow differs, and this analysis alone cannot determine the path of relation information. Regarding the final extraction, both feed-forward and attention sublayers contribute in decoder-only LLMs—unlike in English autoregressive LLMs, where attention modules dominate. In mT5, this mechanism is primarily handled by cross-attention. Overall, our findings indicate that some of the localization (§4) and mechanisms (§5) of fact retrieval in English LLMs generalize to multilingual LLMs, but with key variations.

Finally, to address the third question, we investigate where language plays a role within the three-step process, to characterize how and where factual knowledge cross-lingual transfer may occur. Using activation patching (Zhang and Nanda, 2024) we insert the intermediate representation of the last token from an English forward pass into the forward pass of another language (§6). Our results reveal that the last token acts as a Function Vector (FV) (Todd et al., 2024), encoding both the relation *and* the output language, which is then applied to the subject in the context. Crucially, the fact that the FV formed in one language can be used with an input in another language demonstrates that the subject and relation representations are largely language-independent, while the extraction event is language-specific. Furthermore, in decoder-only LLMs, the FV is constructed in two distinct phases: first, it encodes only the relation, and later, the language is incorporated (Figure 1).

These findings advance our understanding of factual recall, with implications for cross-lingual knowledge transfer, knowledge editing, and trustworthiness in multilingual LLMs. Our results on language flow open new avenues for studying whether models 'think' in English (Wendler et al., 2024), by examining how the FV is altered across languages. Additionally, the late-site causal effect of MLPs and their dominance in the extraction phase suggest that fact editing techniques and knowledge evaluations must extend beyond early MLPs and attention layers (Tamayo et al., 2024).

2 Related Work

Factual Knowledge Recall Petroni et al. (2019) analyzed factual knowledge in pre-trained language models using the LAMA dataset, which pairs cloze-test templates with WikiData triplets.[3] This was extended to multilingual settings by translating LAMA (Kassner et al., 2021; Jiang et al., 2020). Later, Elazar et al. (2021) evaluated the *consistency* of such knowledge by manually curating paraphrases of LAMA to create ParaRel. Further work examined factual consistency cross-lingually (Fierro and Søgaard, 2022; Qi et al., 2023), using machine-translated ParaRel templates and WikiData-based subject/object translations to produce mParaRel.

Interpretability on Factual Knowledge Recall Early studies by Meng et al. (2022) and Geva et al. (2023) mechanistically analyzed knowledge

[3] One might object to treating cloze-test performance as indicative of knowledge, given LLM inconsistencies. However, following Fierro et al. (2024), we adopt this view, as LLMs are often consistent and can sometimes justify responses through world models or training data attribution.

localization and information flow in GPT models (Brown et al., 2020), using English data. Later, Chughtai et al. (2024) showed that answers emerge from summing independent components in GPT and Pythia models, while Sharma et al. (2024) extended these findings to Mamba (Gu and Dao, 2023), also focusing on English.

Activation Patching Ghandeharioun et al. (2024) introduced Patchscope, a framework for decoding intermediate representations by patching them into forward passes. Dumas et al. (2024) applied this method to last-token representations in a translation task, showing that models encode language-agnostic concepts—consistent with our findings and those of Foroutan et al. (2022). Wang et al. (2024) similarly found that the last token encodes the relation in factual English queries at a specific stage of computation. We extend these findings to a cross-lingual factual recall setting. While Dumas et al. (2024) concluded that models first resolve the output language, we find that models first resolve the query's relation, then the language. We thus interpret the last token as a function vector (Todd et al., 2024), which encodes the task-specific function—e.g., translating in the case of Dumas et al. (2024). Additionally, we show that in decoder-only models, the FV composes the output language onto the relation function vector.

3 Experimental Setup

We focus on a simple form of factual knowledge recall, where LMs are tasked with predicting the correct object o for a given subject s and a relation r. These (s, r, o) triplets are obtained from WikiData, and natural language templates are used to describe the relation, with placeholders for the subject and object.[4] For our analysis, we select 10 typologically diverse languages representing various scripts, families, and word orders: English (*en*), Spanish (*es*), Vietnamese (*vi*), Turkish (*tr*), Russian (*ru*), Ukrainian (*uk*), Japanese (*ja*), Korean (*ko*), Hebrew (*he*), Persian (*fa*), and Arabic (*ar*). See Table 2 for language characteristics.

Models We analyze decoder-only and encoder-decoder architectures.[5] The decoder-only LLMs are XGLM (Lin et al., 2021), with 7.5B parameters and 32 layers, and EuroLLM (Martins et al., 2024) with 9B parameters and 42 layers; the encoder-decoder mT5-xl (Xue et al., 2021) has with 3.7B parameters and 24 encoder-decoder layers. The pre-training data varies across models: mT5 is pretrained on 101 languages, covering all the languages in our study; XGLM covers 30 languages, excluding *uk*, *he*, and *fa*; while EuroLLM covers 35 languages, excluding *vi*, *he*, and *fa*.

Data We use the mParaRel dataset (Fierro and Søgaard, 2022), which includes triplets and templates for 45 languages.[6]

To investigate the process of knowledge recall we only consider examples where the model predicts the *correct* object completion (Meng et al., 2022; Geva et al., 2023). Since mParaRel provides multiple paraphrased templates for each relation, we greedy-decode for every available template corresponding to a given triplet and check for an exact match. If multiple templates yield a match, we randomly select one for the analysis. In cases where an article or other filler tokens precede the object, we include these tokens in the input text to ensure that when the example is fed into the model for our analysis, the next predicted token is the first token of the object (Implementation details in Appendix A.1). Table 1 presents the number of examples for which the correct object is predicted. We exclude languages with too few examples from our analysis, namely *ko*, *ja*, *he*, and *fa* for XGLM, and *ja*, *he*, and *fa* for EuroLLM.

	XGLM	EuroLLM	mT5
en	1812	2332	1543
es	1380	1913	1192
vi	1646	779	993
tr	418	799	1058
ru	830	1680	683
uk	213	1244	456
ko	116	308	630
ja	6	42	358
he	13	107	565
fa	7	31	406
ar	811	1790	488

Table 1: Number of facts (s, r, o) correctly predicted.

Notation Given a transformer model with L layers, let h_t^l be the representation of the token t at layer l. When the model is an encoder-decoder, let e_i^l be the representation of the encoder layer l for the i-th token in the encoder input. Then, the encoder layer computes $h_t^{l+1} =$

[4] For example, the relation born-in could use the template "[X] was born in [Y]", where [X] is the subject and [Y] is the object to be predicted.

[5] For decoder-only models, we only use the templates in mParaRel that have the object placeholder at the end of the sentence, while for encoder-decoder, we use all the templates.

[6] We augment the objects in mParaRel, filter out trivial examples, and when there are enough examples, we use a crosslingual subset (see Appendix A).

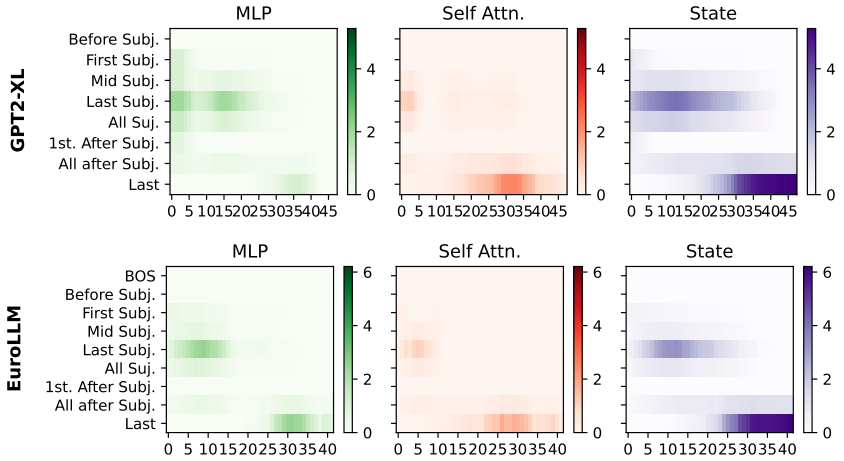


Figure 2: Average indirect effect (IE) on logit scores, when the subject input is corrupted and some activations are restored to their uncorrupted values. The y axis indicates the layer where the restoration was performed, and the x axis specifies the token(s) over which we average the IE. The MHSA and MLP are restored in windows of 12% consecutive layers. Top GPT2-XL (only English data), bottom EUROLLM (XGLM and mT5 in Figure 10).

$h_t^l + s^l + f^l$ and the decoder $h_t^{l+1} = h_t^l + s^l + c^l + f^l$, where $s^l = \text{Self Attn.}(h_0^l \dots h_t^l)$, $c^l = \text{Cross Attn.}(h_t^l, e_0^l \dots e_n^l)$ and $f^l = \text{MLP}(h_t^l + s^l + c^l)$. If decoder-only, then c^l does not apply.

4 Causal Tracing

We first analyze which hidden states in the model's computation are more important than others when recalling a fact. Following Meng et al. (2022), we trace the causal effects of hidden states using causal mediation analysis (Pearl, 2022). Let $\mathbb{P}(o)$ be the probability of the predicted object token, and $LS(o)$ its logit score. We corrupt the input by adding Gaussian noise to the subject tokens,[7] and observe the corrupted probability $\tilde{\mathbb{P}}(o)$ of the originally predicted token. Then, we run inference again on the corrupted input, but this time, we restore a specific hidden state in the model and track the probability $\tilde{\mathbb{P}}_{\text{restored}}(o)$. We study the indirect effect of such component as $\text{IE}_{\mathbb{P}} = \tilde{\mathbb{P}}_{\text{restored}}(o) - \tilde{\mathbb{P}}(o)$, or if using logits $\text{IE}_{LS} = \tilde{LS}(o) - \tilde{LS}_{\text{restored}}(o)$. Specifically, we restore a *state* by setting $\tilde{h}_t^l \leftarrow h_t^l$, where h_t^l is the hidden state from the clean run and $\tilde{h}_t^l$ that of the corrupted run; and similarly, we restore the self-attention layers contribution by setting $\tilde{s}^l \leftarrow s^l$ for all the attention modules in a window of size w (analogous for c and f).[8] We restore windows of 12% consecutive layers, so w=4 for XGLM, w=5 for EUROLLM, and w=3 for mT5.

We compare our results in multilingual LLMs to those of Meng et al. (2022), who analyzed the factual recall of GPT-2 XL in English and reached two key conclusions: (1) they identified an "early site", where the MLPs processing the last subject token in the early and middle layers play a crucial role in recovering from input corruption; and (2), they found a "late site", where the attention modules processing the last token in the later layers also significantly contribute to prediction recovery.[9]

We present the average causal analysis results in Figure 2 (results per language in Appendix B).[10] Our results indicate that the early site in MLPs—centered on the last subject token—is also present in multilingual LLMs. We observe this causal effect in both EUROLLM and mT5. In mT5, the first subject token also shows causal relevance, though marginally less so, and the early site spans all encoder layers. Across languages, the patterns are highly consistent. Interestingly, we do not observe an MLP early site in XGLM, either for English or in the MHSAs. As for the late site, we find a strong causal effect in both MLP and MHSA layers just before the final layers in all models. The position of the late site is consistent across languages, with only slight variation in its endpoint for EUROLLM.

These findings suggest that some conclusions

[7] We follow Meng et al. (2022) and add $\epsilon \sim \mathcal{N}(0, (3\sigma)^2)$, with σ being the standard deviation of the subjects tokens embeddings from the data used. We repeat the experiment ten times with different noise samples, and report the average.

[8] We use windows because, generally, the contributions of the sub-layers are gradual (Geva et al., 2021). In other words, multiple layers contribute the same behavior to the residual, and their sum produces the observed effect.

[9] While they consider the late site unsurprising, as it directly precedes the final prediction, this observation primarily applies to hidden state restoration, not necessarily attention restoration. We, however, interpret the late-site attention as aligning with what Geva et al. (2023) referred to as the extraction event (§5).

[10] In line with Zhang and Nanda (2024) we find that analyzing the $\text{IE}_{\mathbb{P}}$ overestimates the causal effect of some tokens over others. For example, in EUROLLM the last subject token MHSA seem more relevant than the last token MLPs (see Figure 17 vs Figure 15). So we base our main observations using IE_{LS}.

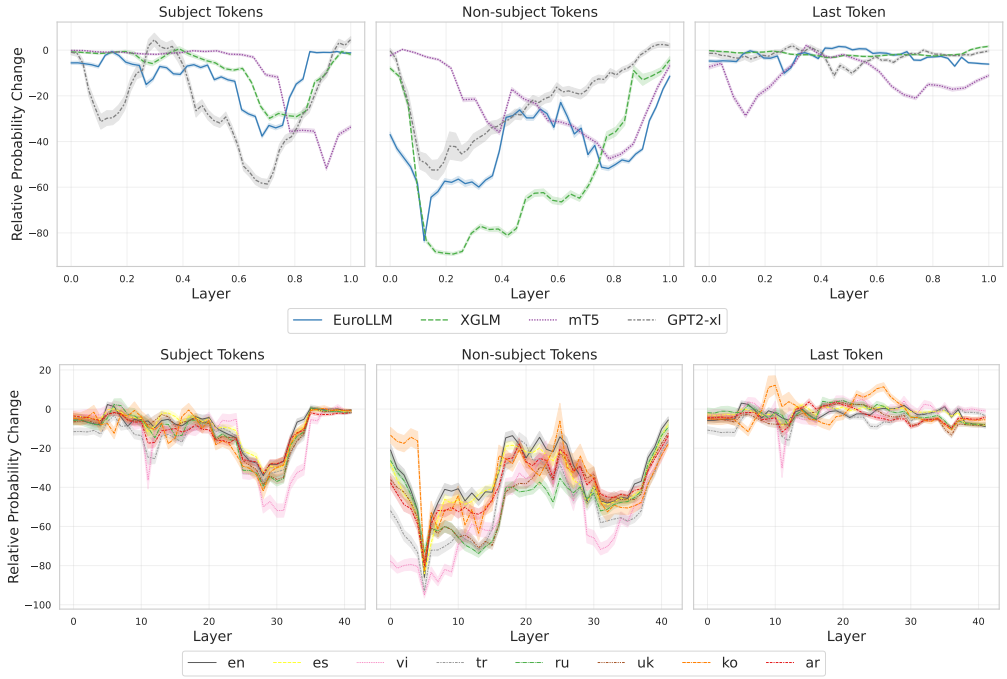


Figure 3: Attention knockout between the last token and a given set of tokens. Each layer represents the effect of the knockout on a window of w layers. Top is the average, bottom EuroLLM ($w = 7$). XGLM and mT5 in Figure 23.

from English LLMs generalize to multilingual ones, but not all. The early MLP site is consistent across models, while at the late site, both MLPs and MHSA are important in multilingual LLMs—unlike in English LLMs, where MHSA dominates. This has implications for knowledge localization and fact editing, which should consider both early and late MLPs in multilingual contexts.

5 Factual Recall Components

Geva et al. (2023) described the process of factual knowledge recall in English autoregressive models as a three-step mechanism: (a) the subject representation is enriched (i.e., related attributes are encoded); (b) the relation and subject information are propagated to the last token; and (c) the final predicted attribute is extracted by attention layers. We analyze the information flow to the last token, and then the extraction of the predicted attribute.

Information Flow We use attention knockout (Geva et al., 2023) to analyze how information propagates to the final token. This method involves intervening in the attention mechanism: for XGLM and EuroLLM, we modify the last token's self-attention, while for mT5, we intervene in the decoder's cross-attention for the final token. We knock out attention connections by setting the attention scores to zero between the final token and a token set $\{t\}$, which can be: subject tokens, non-subject tokens, or the last token itself. Following Geva et al. (2023) we apply this intervention across consecutive layers within a window of size w centered on layer l, covering 18% of the total layers (w=6 for XGLM, w=7 for EuroLLM, and w=4 for mT5). The relative probability change is given by $(\tilde{\mathbb{P}}(o) - \mathbb{P}(o))/\mathbb{P}(o)$, where $\tilde{\mathbb{P}}(o)$ is the probability of the originally predicted token o after attention knockout .A significant drop indicates that the knocked-out tokens contribute critically to the final token's prediction at that layer.

In Figure 3 we present average results, plots per language can be found in Appendix C. The results show that in each model the information flows fairly similarly for all the languages, and somewhat similar patterns to those in English GPT2-xl. On the one hand, the subject information flows to the last token most critically at the later layers in all models. On the other hand, the non-subject tokens information flows throughout all the layers, with EuroLLM and mT5 having more similar curves. Nevertheless, we note two differences from the conclusions reached with English GPT (Geva et al., 2023): (1) the propagation of non-subject tokens to the last token does not strictly precedes the subject propagation, in XGLM the drop is the highest in the earlier layers but it continues to be important until the end, and in mT5 and EuroLLM this information has two peaks, in the early and later layers; and (2) the last token of mT5 encodes critical information that flows from one layer to the next through the cross-attention (as opposed to the

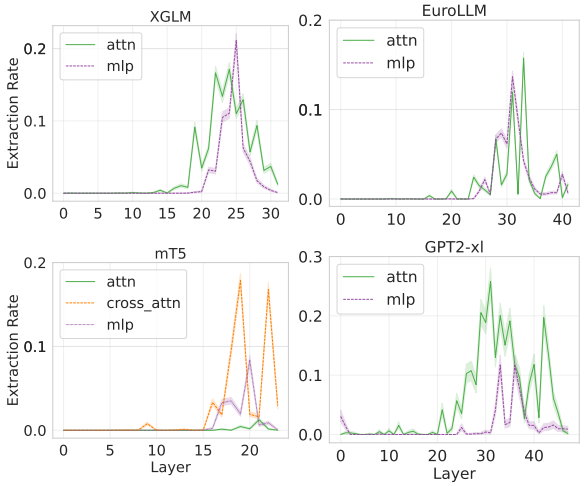


Figure 4: Extraction rates. Fraction of examples where the top-1 token under the vocabulary projection matches the final output token.

negligible flow of the last token in decoder-only models).[11]

Prediction Extraction Following Geva et al. (2023), we measure extraction events at each layer. Let h^l be the representation of the *last* token at layer l, and let E be the embedding matrix; the predicted token is $o = \arg\max(Eh^L)$. An extraction event occurs at layer l if $\arg\max(Es^l) = o$ (similarly for c^l and f^l). The extraction rate is the proportion of examples for which an extraction event occurs at a given layer.

Our results demonstrate that extraction events can be detected in multilingual LLMs (Figure 4), though rates vary across languages (Appendix D).[12] A key finding is the prominence of MLP modules in object extraction for multilingual decoder-only models (XGLM and EuroLLM). To rule out the possibility that MLPs simply forward extracted objects from preceding attention layers, we measure how often MLPs perform an extraction without prior attention extraction (Figure 30). We find that MLPs indeed perform the extraction for most languages, though in English, attention modules dominate, aligning with GPT2-xl results. In mT5, cross-attention layers drive the extraction, with MLPs playing a secondary role, resembling GPT's behavior. Additionally, in mT5 the extraction events occur in later layers, with peaks in 19 and 22.

[11]We hypothesize this may occur because the last token encodes which sentinel token is being generated, indicating where to fill in the input. Since we see an almost identical curve when the last token cannot attend to the sentinel token in the decoder (Figure 26).

[12]This variance may result from the strict criteria applied during the extraction event analyses, where only top-1 matches were considered.

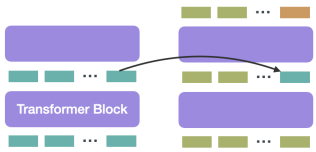


Figure 5: Patching strategy. The patch example's last token is inserted in the context example's forward pass, which perturbs the output of subsequent layers.

The results show that in multilingual LLMs the extraction mechanism is more complex, involving both attention and MLP modules. This implies that editing techniques should focus not only on early MLPs enhancing subject representations but also on the later MLPs that extract the object.

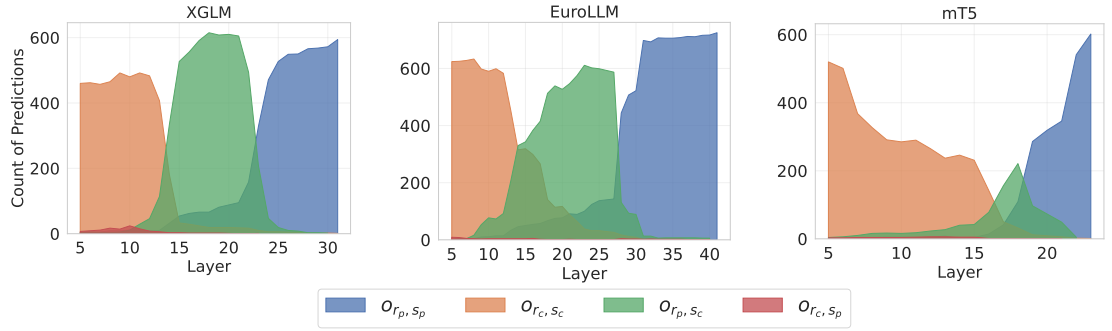
6 Language Information Flow

Previous analyses do not provide insights into *how* the language is included and used during recall. In this section, we use activation patching, similarly to Dumas et al. (2024), to: (1) disentangle when the relation information flows to the last token and when the language information flows, (2) identify which tokens contribute the language information, and (3) interpret the last token representation as a Function Vector (Todd et al., 2024)[13].

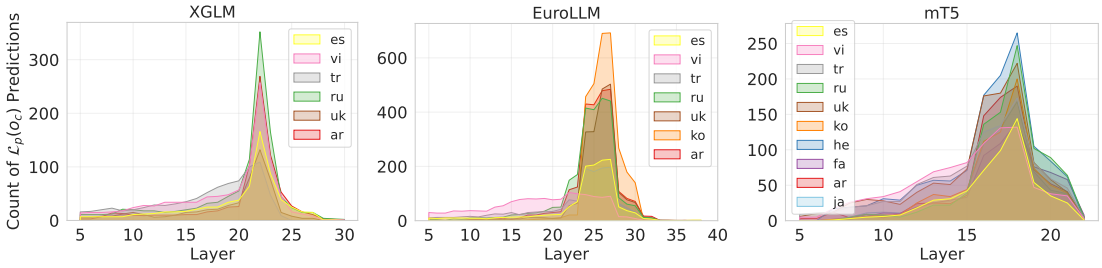
Let h_t^l be the representation at layer l of the *last* token in the input. For a given input p, we take $h_{t,p}^l$ and patch it into the forward pass of another input c at the same layer; that is, we set $h_{t,c}^l \leftarrow h_{t,p}^l$, and then continue the forward pass. We refer to the example p as the *patch* example, and c as the *context* example (Figure 5). By patching the last token representation in each layer and observing the model's predicted output, we can study when the representation contains the information about relation, language, and predicted object, and how cross-lingual these representations are. In all experiments, English is used as the patch language.

An input example expresses a relation r and a subject s in language $\mathcal{L}$, which we note $(\mathcal{L}(r), \mathcal{L}(s))$. We conduct three experiments where the p and c examples share certain input characteristics: (1) $\{=\mathcal{L}, \neq r, \neq s\}$, the language is the same but the relation and subject differ; (2) $\{\neq \mathcal{L}, = r, \neq s\}$, the relation remains the same but the language and subject differ; and (3) $\{\neq \mathcal{L}, \neq r, = s\}$, the subject is the same but the language and relation differ. (1) allows us to

[13]Function Vectors (FV) were originally defined as the mean vector of outputs from specific attention heads. Here, we interpret the last token representation as an FV because it triggers a specific execution for different contexts.



(a) Number of examples where the patch produces one of the four valid objects. The green curve (middle) represents layers where the last token's representation encodes the relation but not yet the object.



(b) Number of examples where the patch causes the output to be $\mathcal{L}_p(o_c)$ (the context object in the patch language). The last token's representation at these layers encodes the relation to extract *and* in which language, but not yet the object.

Figure 6: Patches per layer triggering a specific object prediction. (a) Setup $\{=\mathcal{L}, \neq r, \neq s\}$; (b) $\{\neq \mathcal{L}, = r, \neq s\}$.

study how the relation is encoded while controlling for the language, whereas (2)-(3) help us explore how the interaction between relation and language affects the extraction of the object from the subject. In Figure 1 and 34 we show the aggregated results of these 3 settings, with setup (1) we localize when the object is resolved (blue), with setup (2) when the relation *and* language are encoded (purple), and with setup (3) when the relation is encoded (green).

Let the patch input be $(\mathcal{L}_p(r_p), \mathcal{L}_p(s_p))$ and the context input be $(\mathcal{L}_c(r_c), \mathcal{L}_c(s_c))$. Without intervention, the model correctly predicts $\mathcal{L}_p(o_{r_p,s_p})$ and $\mathcal{L}_c(o_{r_c,s_c})$ respectively.[14] To analyze patching effects, we examine both output probabilities and predicted tokens. For probabilities, we aggregate across examples by calculating the change relative to the original (unpatched) probability, as in §5. For predicted tokens, we check if they match the patch or context object, or a variant with swapped language or relation (e.g., if $\mathcal{L}_p(o_c)$ is predicted in the setups (2)-(3)).[15]

[14]For simplicity, we may refer to the object as $\mathcal{L}_p(o_p)$ or $\mathcal{L}_c(o_c)$ when r and s are from the same input.

[15]We only consider valid patch-context pairs where o_{r_p,s_c} and o_{r_c,s_p} exist, and for predicted token analysis, we enforce distinct spellings, e.g. to claim the predicted token is $t = \mathcal{L}_p(o_c)$ we require $t \neq \mathcal{L}_c(o_c)$, as languages may share spellings (e.g., "Asia" in English and Spanish). Consequently, the number of examples varies across analyses of output probabilities and token predictions. See Table 4 and 6 for the number of examples for each model and experiment.

Different Relation, Different Subject First, we analyze the pairs of examples with $\{=\mathcal{L}, \neq r, \neq s\}$. For example, r_p, s_p ="The capital of France is" and r_c, s_c ="The language spoken in Germany is". Then, o_{r_p,s_c} = Berlin and o_{r_c,s_p} = French. We sample 1000 examples for which MPARAREL has the objects o_{r_c,s_p} and o_{r_p,s_c}.

We present the prediction results in Figure 6a and the probability plots in Figure 35. An increase in the green curve indicates that relation information becomes available in the last token representation, as patching at that layer produces o_{r_p,s_c}—the object corresponding to the relation in the patch example. When the green curve declines and the blue curve rises, the object has been fully extracted and encoded in the last token, since predictions now yield o_{r_p,s_p} and are no longer influenced by c. These transitions align with peaks in the extraction rate for English (Figure 27-29). This confirms that extraction, as measured by the vocabulary projection, marks the point where the object is both available and encoded in the last token representation. If the object had been encoded earlier and only decoded at the extraction point, o_{r_p,s_p} would have been predicted from earlier-layer patches.

Same Relation, Different Subject We have localized the layers where the relation representation flows to the last

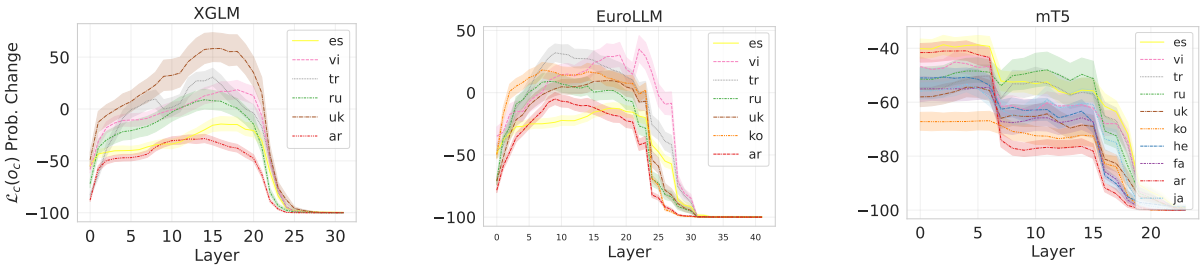


Figure 7: Percentage change in the probability of the context object $\mathbb{P}(\mathcal{L}_c(o_c))$ when patching $\{\neq \mathcal{L}, = r, \neq s\}$.

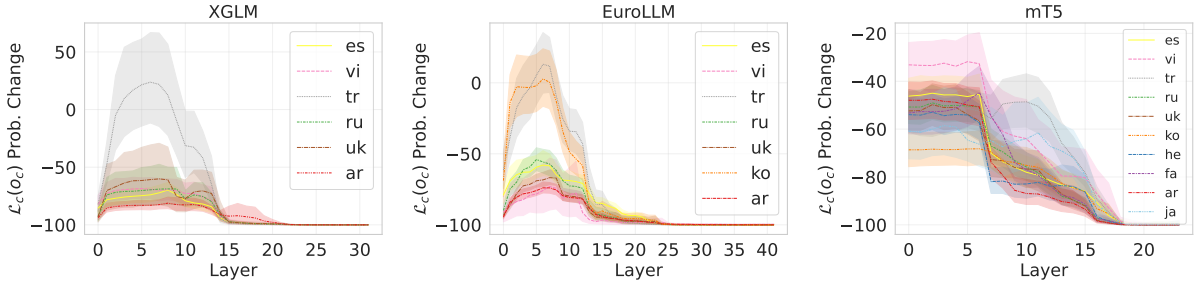


Figure 8: Percentage change in the probability of the context object $\mathbb{P}(\mathcal{L}_c(o_c))$ when patching $\{\neq \mathcal{L}, \neq r, = s\}$.

token. Is the language information entangled to the subject or the relation representation? Here, we study pairs of patch-context with $\{\neq \mathcal{L}, = r, \neq s\}$, e.g., "France's capital is" and "La capital de Alemania es" (Gloss: "The capital of Germany is"). If we obtain $\mathcal{L}_p(o_c)$ ("Berlin"), it would suggest that the language is encoded in the last token representation before the extraction happens. On the other hand, if we obtain $\mathcal{L}_c(o_p)$ ("París"), we could infer that the language is encoded after the extraction.

We present the probability of the context answer token $\mathcal{L}_c(o_c)$ in Figure 7.[16] We observe that for decoder-only models, patching in the early layers generally hurts the model's performance for most languages, while, patching in the middle layers either increases the probability or results in only a minimal decrease. Given that the relation is the same in both examples, this suggests that by these middle layers, r is encoded in h_p^l, and in the subsequent layers the subject and language of the context are integrated to yield the final prediction, $\mathcal{L}_p(r) + \mathcal{L}_c + s_c = \mathcal{L}_c(o_c)$. Moreover, for some languages the relation representation from the patch is better than the one constructed using the context input, as the relative probability is positive. In the case of mT5, however, the probability of $\mathcal{L}_c(o_c)$ consistently decreases, plateauing in the middle layers before dropping to zero. From the attention knockout analysis, recall that in mT5 the subject is integrated into the last token representation only after layer 15, while the relation is consistently represented throughout the middle layers. Therefore, these patching results imply that the relation encoded in the last token from the patch input does not help in retrieving the correct context object in the context language in mT5, whereas it proves useful in XGLM and EuroLLM.

In terms of predicted tokens, we find across-the-board that $\mathcal{L}_p(o_c)$ is frequently predicted while $\mathcal{L}_c(o_p)$ is not (Table 5). This suggests that the last token representation encodes the patch language $\mathcal{L}_p$ but not yet the object, as the object is derived from the context subject. We plot the layers where $\mathcal{L}_p(o_c)$ is predicted in Figure 6b ($\mathcal{L}_c(o_p)$ in Figure 50). We can conclude that the language information flows to the last token right before the peak of $\mathcal{L}_p(o_c)$ predictions, because if we patch earlier, the output is in the context language (see the probabilities of $\mathcal{L}_c(o_c)$ in Figure 7). Moreover, these peaks match the beginning of their corresponding extraction phases, thus the language information flows right before the extraction phase.

As a result, we interpret the last token representation as containing a Function Vector (FV), the relation that needs to be extracted and in which language, which is used in the extraction event. The FV can be transferred to contexts in another language, as the FV is constructed from the patch input and is used in the context subject representation to predict $\mathcal{L}_p(o_c)$.

Different Relation, Same Subject We just saw that for all models the representation from the patch will encode at some point the output language but not yet the object. It could be that we observed the

[16] The probability of $\mathcal{L}_p(o_p)$ and per language plots are provided in Appendix E.2.

prediction $\mathcal{L}_p(o_c)$ because the relation is language specific and encodes the output language. To analyze if this is the case, we now apply patching on examples with different languages and relations but the same subject $\{\neq \mathcal{L}, \neq r, = s\}$, e.g., "France's capital is" and "El idioma oficial de Francia es" (Gloss: "The official language of France is").

We observe that the probability of $\mathcal{L}_c(o_c)$ (Figures 8) in decoder-only models, unlike the previous experiment, decreases early for all languages (except *tr* and *ko*), plateaus around the middle layers, and then drops to zero by the mid-layer range. For mT5, the probability drops at the beginning but, instead of plateauing as before, it continues to decline until it reaches zero. When compared to the former experiment (Figure 7), this suggests that the relation information is encoded in the middle layers, as $\mathcal{L}_c(o_c)$ decreases earlier when the patch and context have different relations.

In terms of predictions, we find that $\mathcal{L}_c(o_p)$ is frequently predicted for XGLM and EuroLLM (Figure 9), while $\mathcal{L}_p(o_c)$ appears but less often (Table 7). In line with the previous observation, the plot shows that the last token representation in the middle layers where $\mathcal{L}_c(o_p)$ is predicted, primarily captures the relation from the patch r_p, without yet encoding the output language or subject information (as these are taken from the context). Therefore, in decoder-only models, the relation and language representations are disentangled, as the relation flows to the last token before the output language does. Allowing the relation to be combined with different languages.

As for mT5, we observe very few examples where $\mathcal{L}_c(o_p)$ or $\mathcal{L}_p(o_c)$ are predicted, which aligns with the findings of the two former experiments, where we see that the relation is encoded in the last token in layers 15-21 (Figure 6a), and the language flows to the last token around layer 15-19 (Figure 6b). We conclude that both the relation and language flow to the last token around the same time, and thus, in this experiment, we cannot see a disentangled behavior. This presents an interesting contrast with decoder-only models. The decoder in mT5 has access to the *same* encoder representations throughout all its layers, so it does not need (and thus does not learn) to attend to these earlier or in different stages. By contrast, in a decoder-only model, the last token has access to representations that evolve across layers, so it learns to attend to relevant information when it becomes most salient. Nonetheless, we cannot reach a definite conclusion

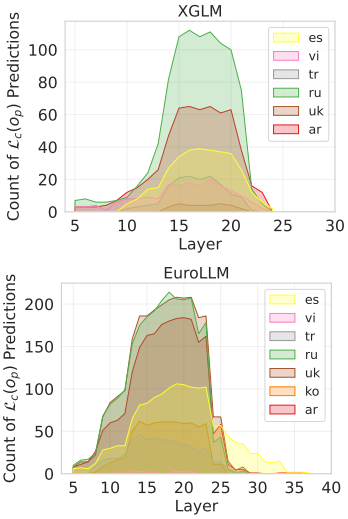


Figure 9: Patches that cause the prediction to be $\mathcal{L}_c(o_p)$, which shows the layers where the last token representation contains the relation but not yet the language.

on whether the language representation in mT5 is entangled or not to the relation representation. More detailed analysis of what is being attended when the relation flows and when the language flows should be performed in future work.

7 Conclusion

In this paper, we analyzed factual knowledge recall mechanisms in 10 languages using multilingual transformer-based LMs, comparing them to recall in English autoregressive LLMs. We discovered that some mechanisms, such as the flow of subject representations in the later layers and the extraction phase, are present in multilingual and monolingual models. However, we also identified notable differences, including the join role of late MLPs and attention modules during the extraction phase in multilingual models. A key contribution of our work is the first-ever investigation of language encoding during recall, achieved through patching representations. In decoder-only models, the relation flows to the last token first, followed by the language. In contrast, in mT5, both relation and language flow to the last token at similar layers. This suggests that while relation and subject representations are multilingual and enable cross-lingual object extraction, the extraction phase itself is language-specific, as language encoding precedes extraction. These findings provide new evidence to understand the factual knowledge recall in transformer LMs, and to how decoder-only LMs resolve tasks in stages. Contributing with new directions for the study of cross-lingual transfer and knowledge localization.

Limitations

In this paper, we examined three model architectures, leaving out the effects of model sizes, instruction fine-tuning, or models like Llama that can behave multilingually but have less coverage and less multilingual pre-training data. Additionally, our analyses were conducted on 500-1000 examples per language, which we believe provides a sufficient sample size for generalization; however, the results are inherently limited by the relations present in the MPARAREL dataset, which may not capture all factual nuances. Additionally, although we analyzed 10 diverse languages, many more languages exist, and further research is needed to confirm the generalizability of our findings across a broader linguistic spectrum. Lastly, we described the main mechanisms found in XGLM, EUROLLM and mT5, however other weaker mechanisms could be at play, which could describe, for example, the low extraction rates found for some languages (Figure 29) or the few examples where the object seems to be encoded from early layers before the extraction takes place (Figure 50).

Acknowledgments

We thank our colleagues at the CoAStaL NLP group Laura Cabello, Rita Ramos, and Israfel Salazar for valuable comments on the final manuscript. DE was supported by the European Union's Horizon 2020 research and innovation program under grant agreement No. 101135671 (TrustLLM).

References

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.

Bilal Chughtai, Alan Cooney, and Neel Nanda. 2024. Summing up the facts: Additive mechanisms behind factual recall in llms. *arXiv preprint arXiv:2402.07321*.

Clément Dumas, Veniamin Veselovsky, Giovanni Monea, Robert West, and Chris Wendler. 2024. How do llamas process multilingual text? a latent exploration through activation patching. In *ICML 2024 Workshop on Mechanistic Interpretability*.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. 2021. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031.

Javier Ferrando and Marta R. Costa-jussà. 2024. On the similarity of circuits across languages: a case study on the subject-verb agreement task. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10115–10125, Miami, Florida, USA. Association for Computational Linguistics.

Constanza Fierro, Ruchira Dhar, Filippos Stamatiou, Anders Søgaard, and Nicolas Garneau. 2024. Defining knowledge: Bridging epistemology and large language models. *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024.

Constanza Fierro and Anders Søgaard. 2022. Factual consistency of multilingual pretrained language models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3046–3052, Dublin, Ireland. Association for Computational Linguistics.

Negar Foroutan, Mohammadreza Banaei, Rémi Lebret, Antoine Bosselut, and Karl Aberer. 2022. Discovering language-neutral sub-networks in multilingual language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7560–7575, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. Dissecting recall of factual associations in auto-regressive language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12216–12235, Singapore. Association for Computational Linguistics.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. 2024. Patchscopes: A unifying framework for inspecting hidden representations of language models. In *Forty-first International Conference on Machine Learning*.

Heidi E. Grasswick. 2010. Scientific and lay communities: Earning epistemic trust through knowledge sharing. *Synthese*, 177(3):387–409.

Albert Gu and Tri Dao. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.

Katherine Hawley. 2012. 64Knowledge and expertise. In *Trust: A Very Short Introduction*. Oxford University Press.

Zhengbao Jiang, Antonios Anastasopoulos, Jun Araki, Haibo Ding, and Graham Neubig. 2020. X-FACTR: Multilingual factual knowledge retrieval from pretrained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5943–5959, Online. Association for Computational Linguistics.

Nora Kassner, Philipp Dufter, and Hinrich Schütze. 2021. Multilingual LAMA: Investigating knowledge in multilingual pretrained language models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3250–3258, Online. Association for Computational Linguistics.

Zihao Li, Shaoxiong Ji, Timothee Mickus, Vincent Segonne, and Jörg Tiedemann. 2024. A comparison of language modeling and translation as multilingual pretraining objectives. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15882–15894, Miami, Florida, USA. Association for Computational Linguistics.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, et al. 2021. Few-shot learning with multilingual language models. *arXiv preprint arXiv:2112.10668*.

Pedro Henrique Martins, Patrick Fernandes, João Alves, Nuno M Guerreiro, Ricardo Rei, Duarte M Alves, José Pombal, Amin Farajian, Manuel Faysse, Mateusz Klimaszewski, et al. 2024. Eurollm: Multilingual language models for europe. *arXiv preprint arXiv:2409.16235*.

Kevin Meng, David Bau, Alex J Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*.

C. Thi Nguyen. 2022. Trust as an unquestioning attitude. *Oxford Studies in Epistemology*, 7:214–244.

Judea Pearl. 2022. Direct and indirect effects. In *Probabilistic and causal inference: the works of Judea Pearl*, pages 373–392.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? 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 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Jirui Qi, Raquel Fernández, and Arianna Bisazza. 2023. Cross-lingual consistency of factual knowledge in multilingual language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10650–10666, Singapore. Association for Computational Linguistics.

Arnab Sen Sharma, David Atkinson, and David Bau. 2024. Locating and editing factual associations in mamba. In *First Conference on Language Modeling*.

Daniel Tamayo, Aitor Gonzalez-Agirre, Javier Hernando, and Marta Villegas. 2024. Mass-editing memory with attention in transformers: A cross-lingual exploration of knowledge. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5831–5847, Bangkok, Thailand. Association for Computational Linguistics.

Eric Todd, Millicent Li, Arnab Sen Sharma, Aaron Mueller, Byron C Wallace, and David Bau. 2024. Function vectors in large language models. In *The Twelfth International Conference on Learning Representations*.

Zijian Wang, Britney Whyte, and Chang Xu. 2024. Locating and extracting relational concepts in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 4818–4832, Bangkok, Thailand. Association for Computational Linguistics.

Chris Wendler, Veniamin Veselovsky, Giovanni Monea, and Robert West. 2024. Do llamas work in English? on the latent language of multilingual transformers. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15366–15394, Bangkok, Thailand. 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*, pages 483–498, Online. Association for Computational Linguistics.

Da Yin, Hritik Bansal, Masoud Monajatipoor, Liunian Harold Li, and Kai-Wei Chang. 2022. GeoMLAMA: Geo-diverse commonsense probing on multilingual pre-trained language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2039–2055, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Fred Zhang and Neel Nanda. 2024. Towards best practices of activation patching in language models: Metrics and methods. In *The Twelfth International Conference on Learning Representations*.

Ruochen Zhang, Qinan Yu, Matianyu Zang, Carsten Eickhoff, and Ellie Pavlick. 2025. The same but different: Structural similarities and differences in multilingual language modeling. In *The Thirteenth International Conference on Learning Representations*.

Language	Script	Family	SOV/SVO
English	Latin	Germanic	SVO
Spanish	Latin	Romance	SVO
Vietnamese	Latin	Austroasiatic	SVO
Turkish	Latin	Turkic	SOV
Russian	Cyrillic	Slavic	SVO*
Ukrainian	Cyrillic	Slavic	SVO*
Japanese	Kanji	Proto-Japonic	SOV
Korean	Korean	Koreanic	SOV
Hebrew	Hebrew	Arabic	VSO
Farsi (Persian)	Perso-Arabic	Indo-Iranian	SOV
Arabic	Arabic	Arabic	VSO

Table 2: Languages, their scripts, families, and sentence structures (SVO: subject-verb-object, SOV: subject-object-verb, VSO: verb-subject-object, SVO*: SVO dominant but SOV is also possible).

A Experimental Setup

We use the MPARAREL triplets and templates to query the factual knowledge of the language models. As mentioned earlier, we perform 3 modifications to the dataset. First, we fetch WikiData for aliases of the target object to be able to match different possible surface forms. Second, we filter out examples where the target object is contained in the query, e.g. *Microsoft Outlook is developed by*. Finally, to better compare across languages we control the variety of the subject-object pairs, by only using a crosslingual version of MPARAREL. Specifically, for each relation we filter out triplets that are not present in all the languages (in MPARAREL a subject and object may have not been translated if they were not found in WikiData). Thus, in the crosslingual MPARAREL version, each subject-object pair in a relation is present in each of the languages. For XGLM we additionally restrict the templates to have the object at the end of the sentence, therefore for some languages (*tr*, *ko*, *ja*, and *fa*) both the autorregressive condition on the templates and the crosslingual restriction leads to too few total examples (< 1000), so for these we do not impose the crosslingual restriction. For all the other languages, and for all the languages in mT5 there are enough examples so we restrict them to be crosslingual. In Table 3 we present the total number of examples we consider per language and model, and the correctly predicted fraction.[17]

[17] In decoder-only models, the total number of examples is higher in some languages due to crosslingual filtering, which is applied per relation. If a language has no examples for a given relation, it doesn't restrict examples in other languages. Since decoder-only models use only autoregressive templates, some lower-resource languages may have zero examples for certain relations. However, when using all templates (as in mT5), these lower-resource languages can restrict the triplets available for other languages within the same relation.

	XGLM and EUROLLM			mT5		
Language	Correct	Total	Percentage	Correct	Total	Percentage
en	1812	4147	43.7%	1543	3853	40.0%
es	1380	4167	33.1%	1192	3926	30.4%
vi	1646	4068	40.5%	993	3748	26.5%
tr	418	2278	18.3%	1058	4033	26.2%
ru	830	4133	20.1%	683	3826	17.9%
uk	213	4144	5.1%	456	3830	11.9%
ko	116	1444	8.0%	630	3783	16.7%
ja	6	1790	0.3%	358	4124	8.7%
he	13	4403	0.3%	565	4064	13.9%
fa	7	4388	0.2%	406	3814	10.6%
ar	811	4482	18.1%	488	4110	11.9%

Table 3: Total number of examples in MPARAREL.

A.1 Prompt Details

For mT5, we feed the MPARAREL input into the encoder with a sentinel token in the object placeholder. In the decoder, we provide the beginning-of-sequence token followed by the sentinel token. We only check the tokens generated next to the first sentinel token, as the pre-training task of the model is to generate the text for each sentinel in the input, the decoder usually continues to generate answers for other sentinel tokens. Any tokens generated preceding the object, are added to the decoder input since adding these to the encoder does not ensure that the next token predicted will be the object.

A.2 Computational Resources

The experiments were run in A100 GPUs. The causal analysis took from 30 minutes to 12 hours depending on the language. The rest of the experiments took 1-2 hours per language.

B Causal Tracing

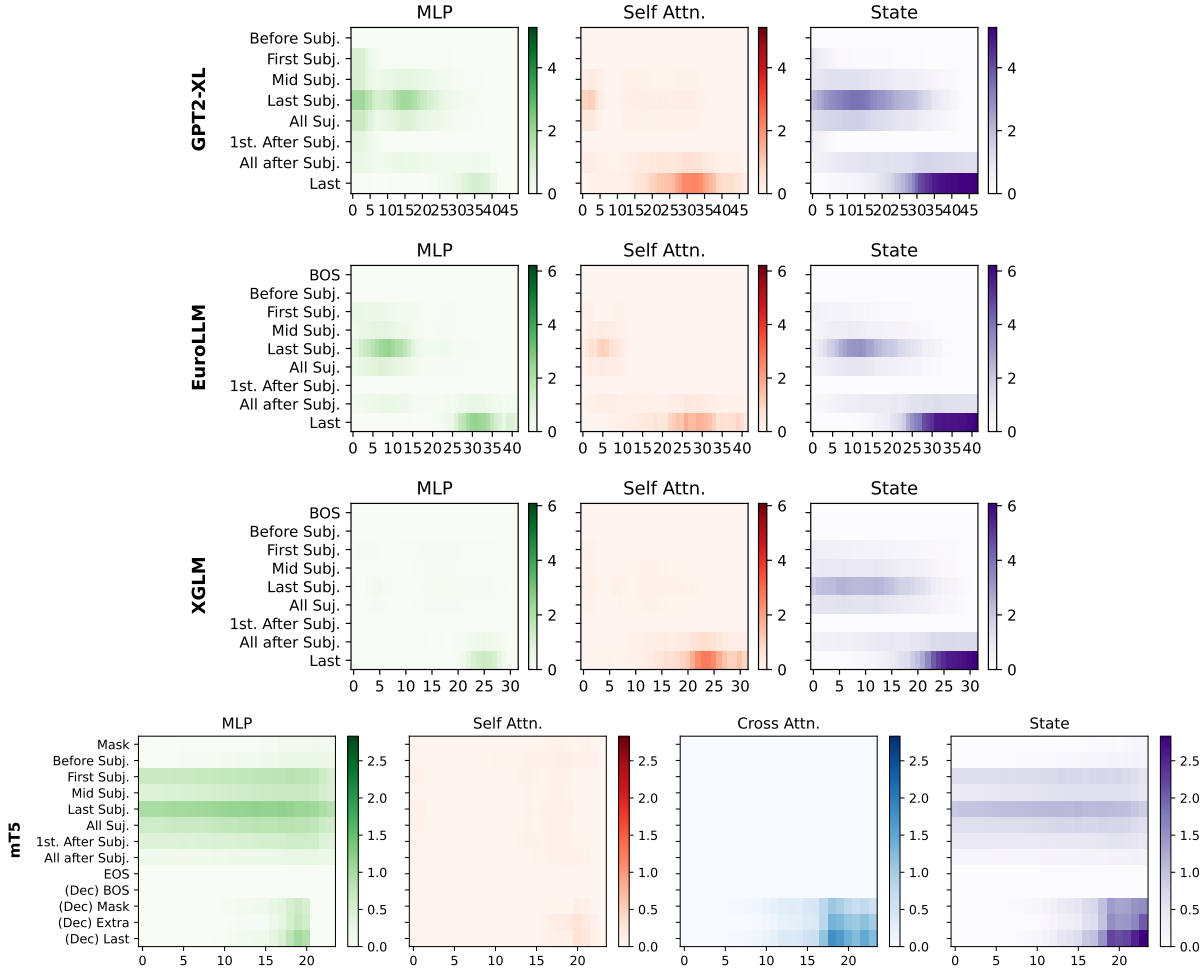


Figure 10: Average indirect effect on logit scores, when the subject input has been corrupted and some activations are restored to their uncorrupted values. The MHSA and MLP are restored in windows of 12% consecutive layers. Top GPT2-XL (only English data).

B.1 XGLM

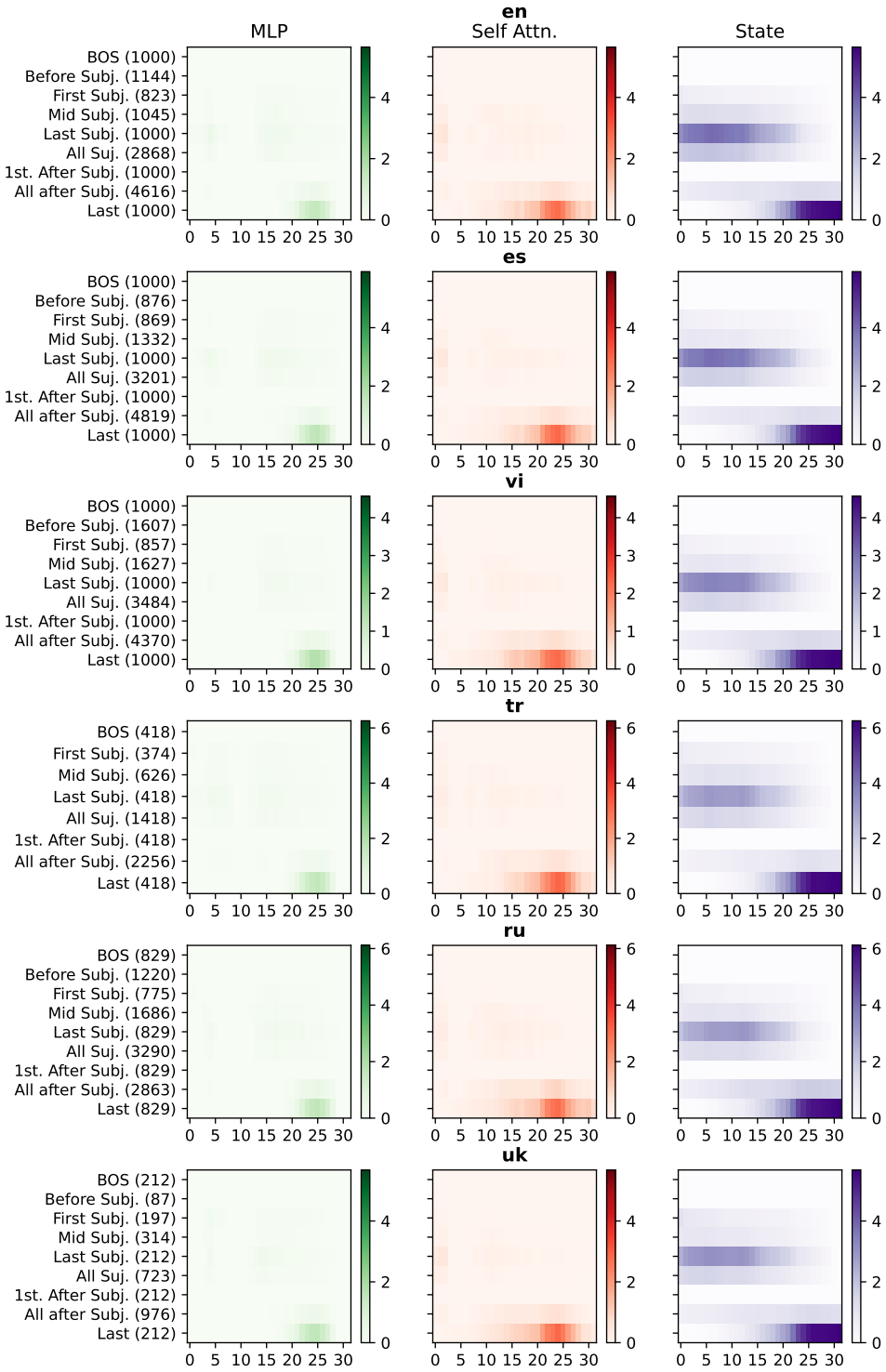


Figure 11: XGLM causal analysis for each language (continues in Figure 12). Average **logit score** (for the originally predicted token) recovered after corrupting the subject in the input and restoring: the hidden representation at a given layer (State), the MLP in a window of 4 layers, or the Self Attn. in a window of 4 layers.

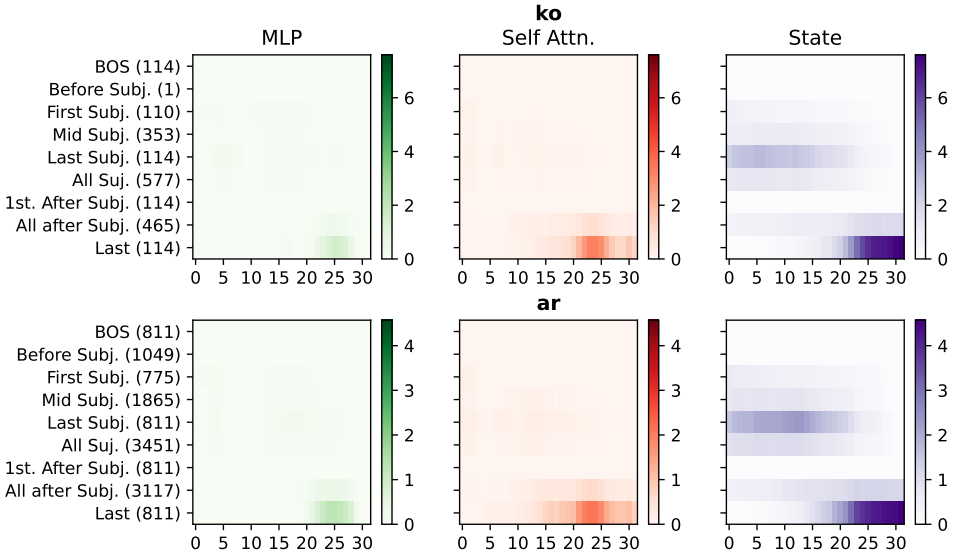


Figure 12: XGLM causal analysis for each language (Rest of the languages in Figure 11). Average **logit score** (for the originally predicted token) recovered after corrupting the subject in the input and restoring: the hidden representation at a given layer (State), the MLP in a window of 4 layers, or the Self Attn. in a window of 4 layers.

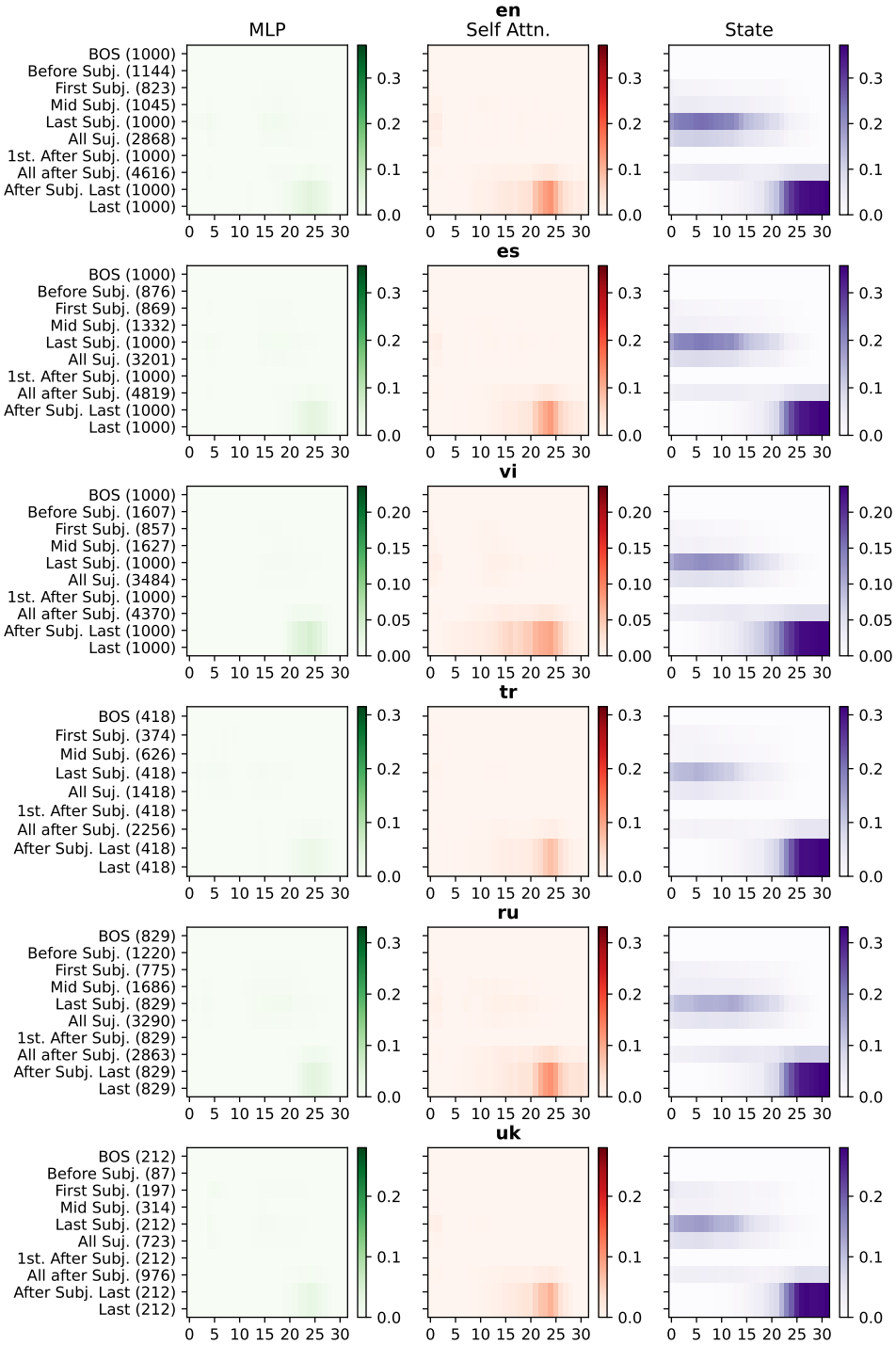


Figure 13: XGLM causal analysis for each language (continues in Figure 14). Average **probability** (for the originally predicted token) recovered after corrupting the subject in the input and restoring: the hidden representation at a given layer (State), the MLP in a window of 4 layers, or the Self Attn. in a window of 4 layers.

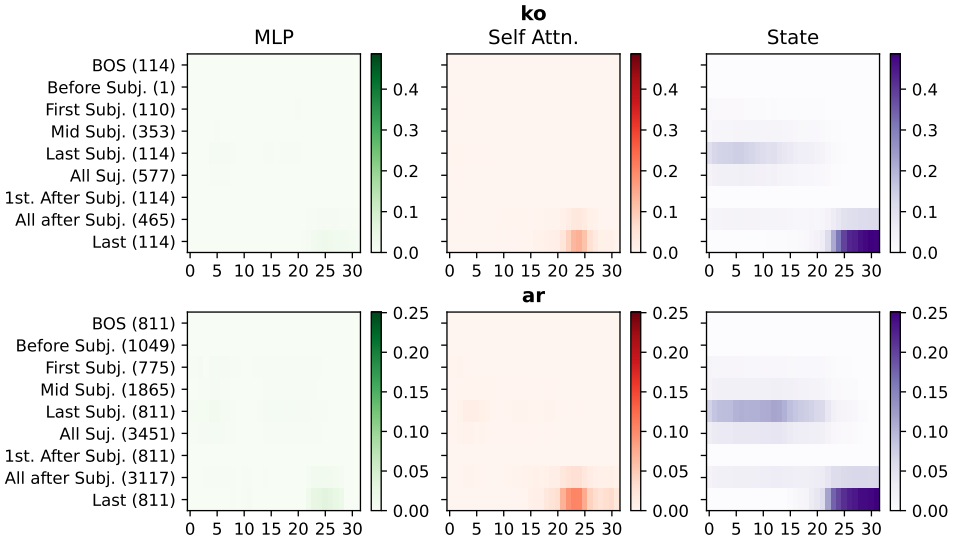


Figure 14: XGLM causal analysis for each language (Rest of the languages in Figure 13). Average **probability** (for the originally predicted token) recovered after corrupting the subject in the input and restoring: the hidden representation at a given layer (State), the MLP in a window of 4 layers, or the Self Attn. in a window of 4 layers.

B.2 EUROLLM

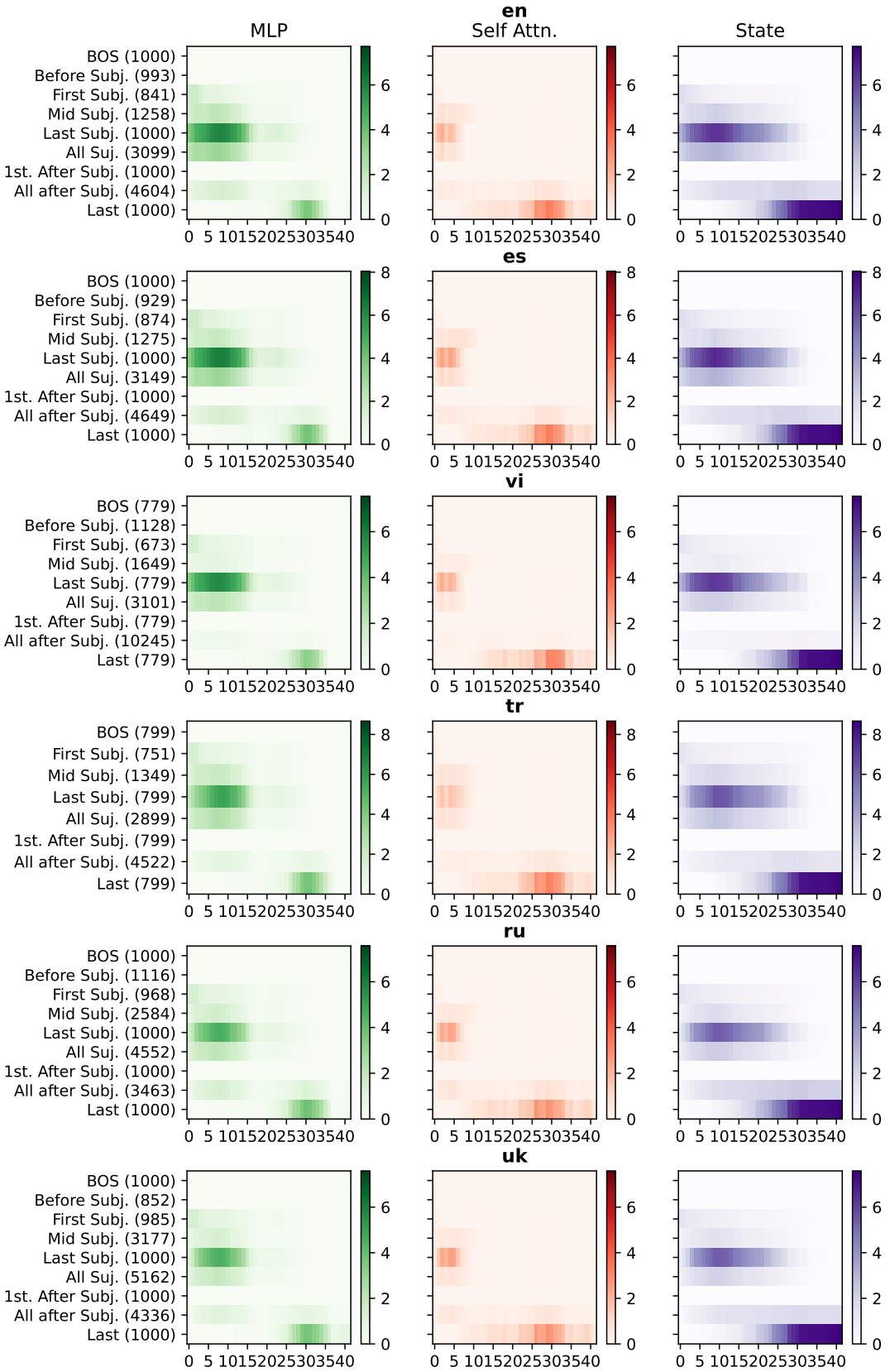


Figure 15: EUROLLM causal analysis for each language (continues in Figure 16). Average **logit score** (for the originally predicted token) recovered after corrupting the subject in the input and restoring: the hidden representation at a given layer (State), the MLP in a window of 5 layers, or the Self Attn. in a window of 5 layers.

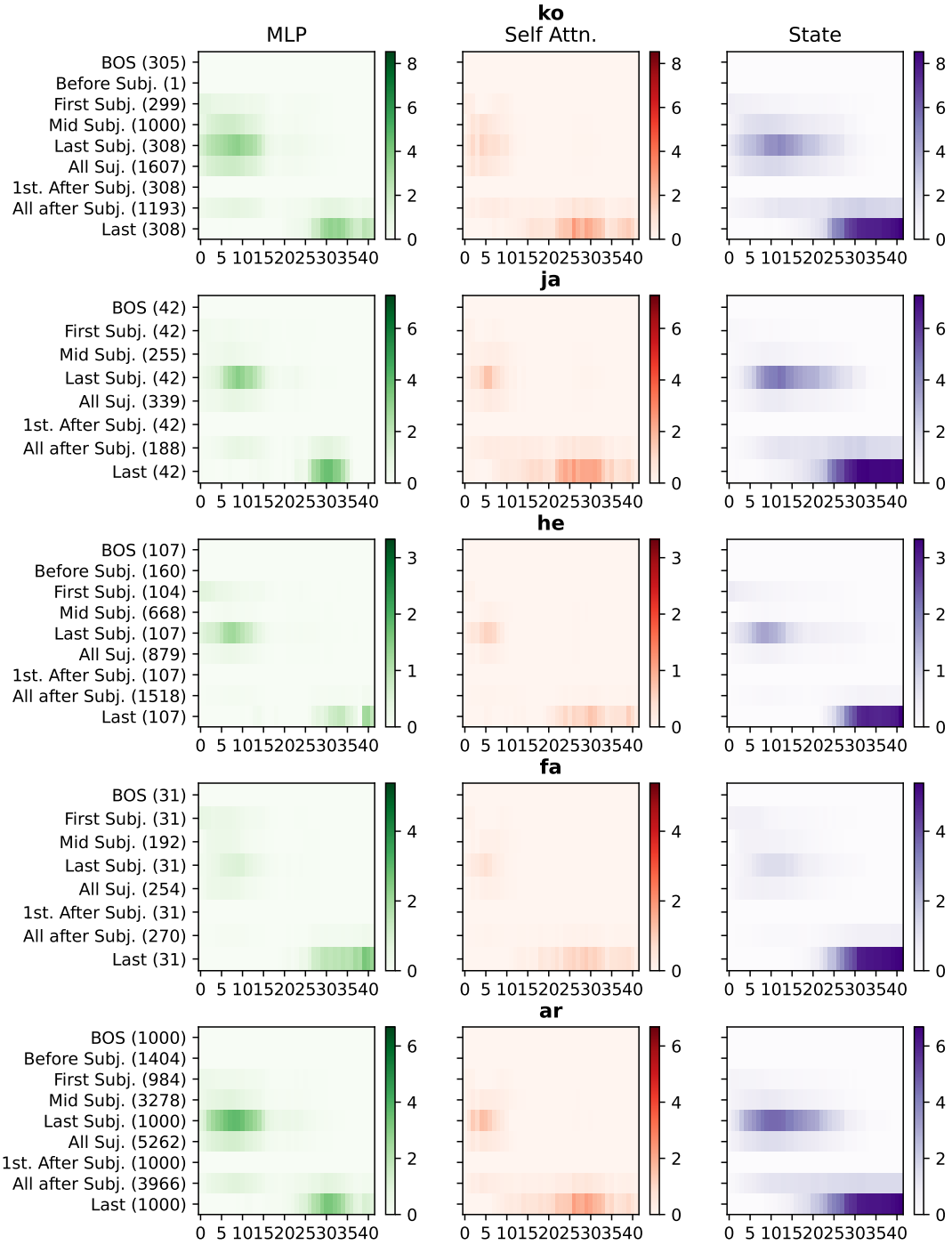


Figure 16: EUROLLM causal analysis for each language (Rest of the languages in Figure 15). Average **logit score** (for the originally predicted token) recovered after corrupting the subject in the input and restoring: the hidden representation at a given layer (State), the MLP in a window of 5 layers, or the Self Attn. in a window of 5 layers.

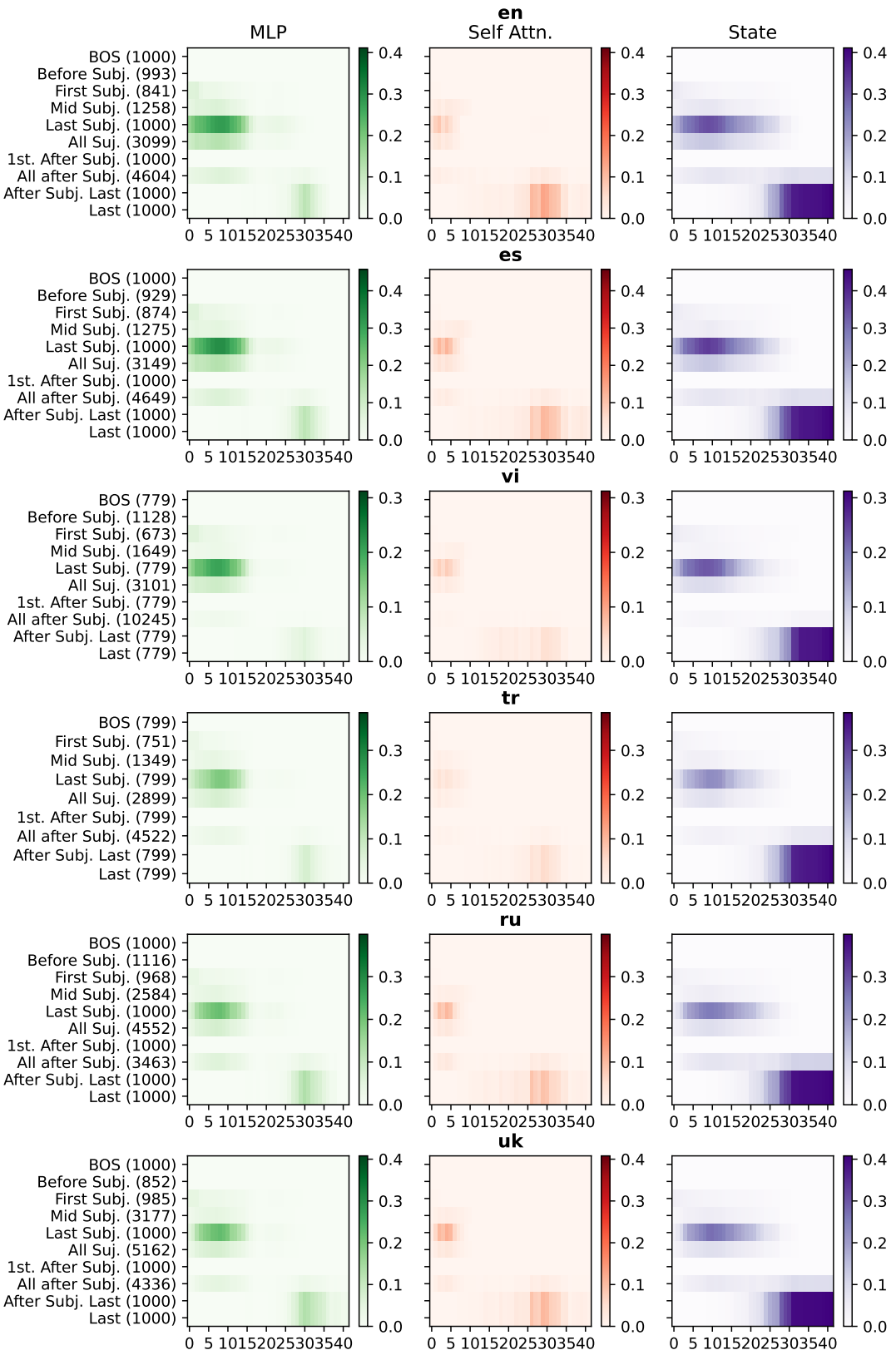


Figure 17: EUROLLM causal analysis for each language (continues in Figure 18). Average **probability** (for the originally predicted token) recovered after corrupting the subject in the input and restoring: the hidden representation at a given layer (State), the MLP in a window of 5 layers, or the Self Attn. in a window of 5 layers.

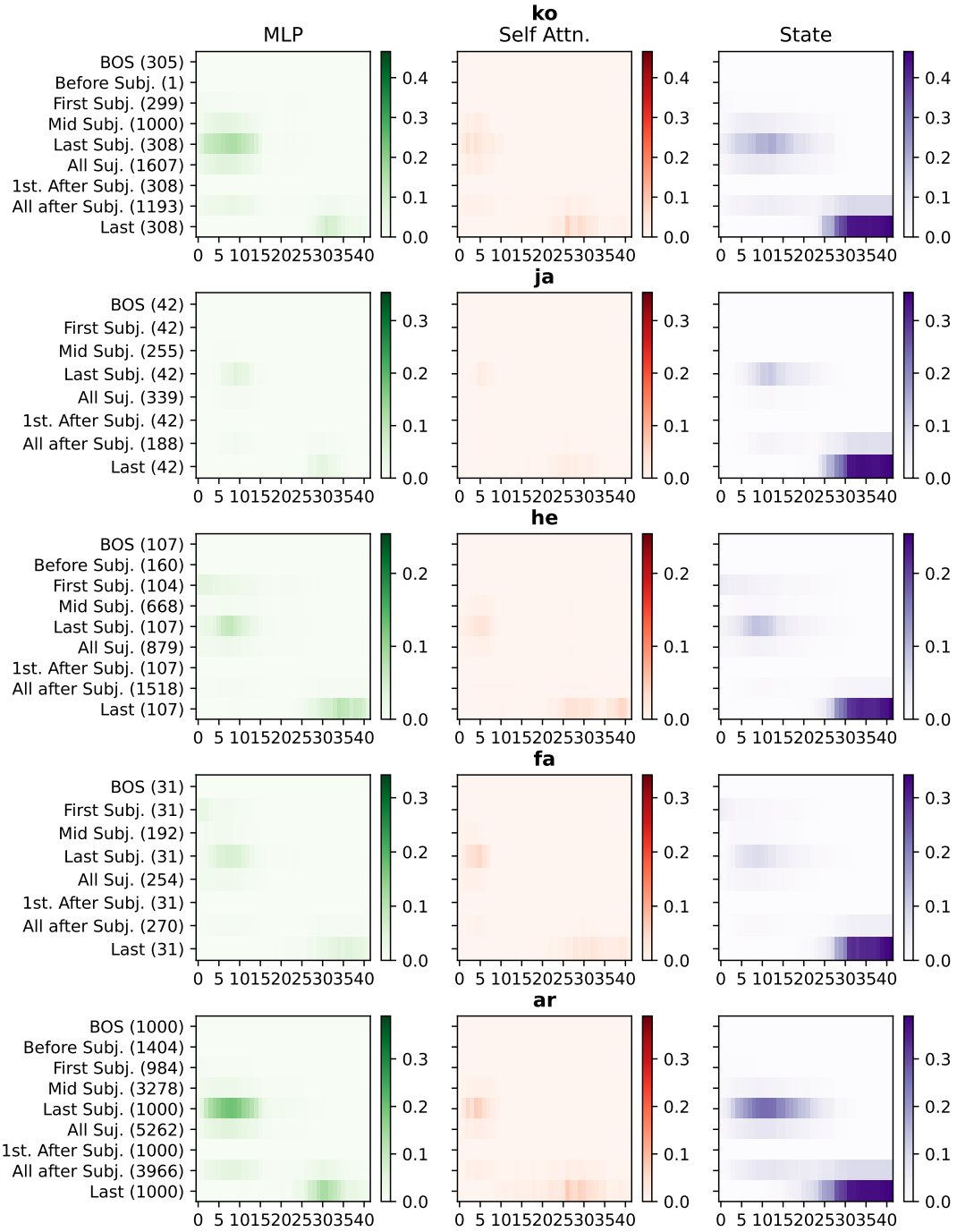


Figure 18: mT5 causal analysis for each language (Rest of the languages in Figure 17). Average **probability** (for the originally predicted token) recovered after corrupting the subject in the input and restoring: the hidden representation at a given layer (State), the MLP in a window of 5 layers, or the Self Attn. in a window of 5 layers.

B.3 mT5

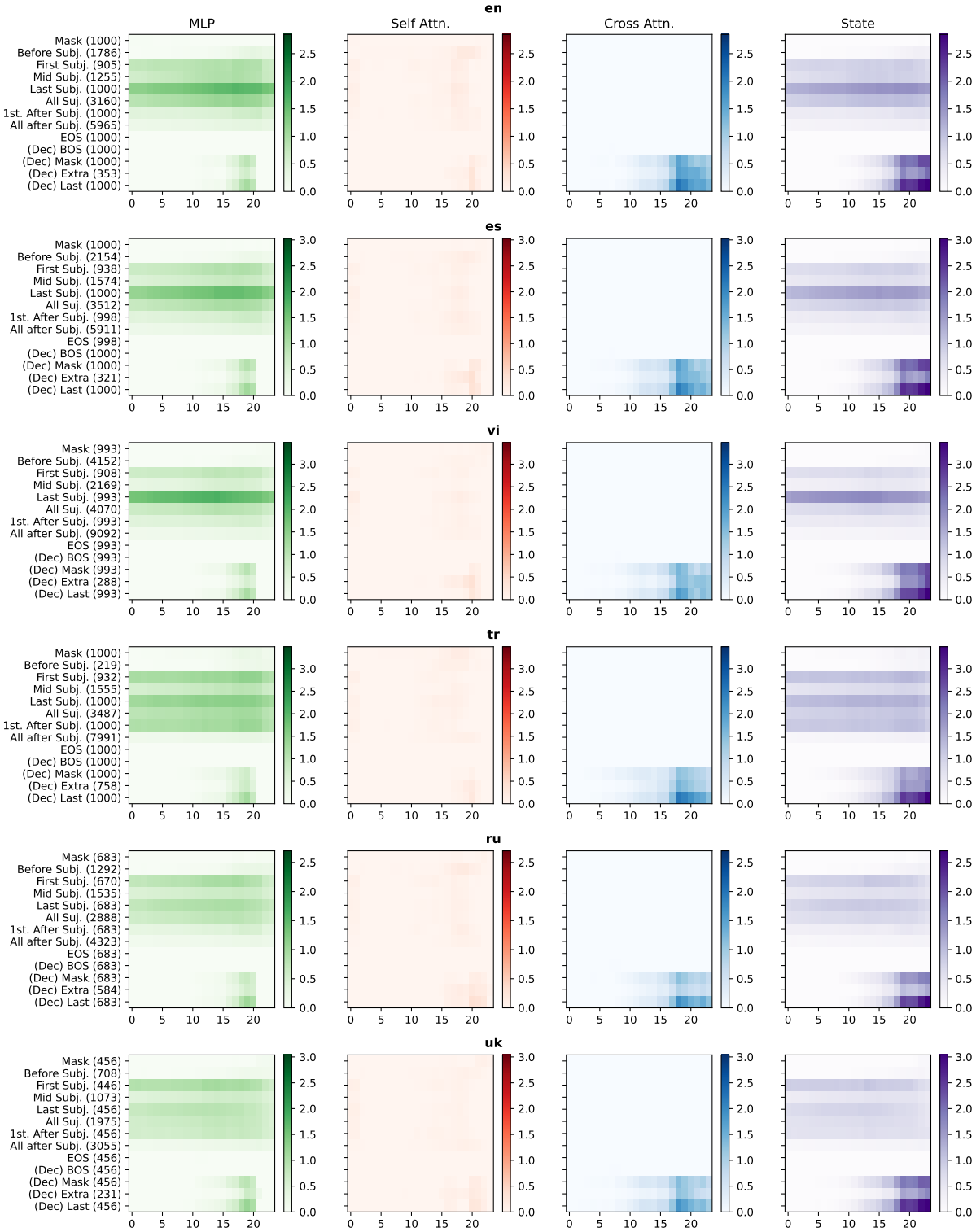


Figure 19: mT5 causal analysis for each language (continues in Figure 20). Average **logit score** (for the originally predicted token) recovered after corrupting the subject in the input and restoring: the hidden representation at a given layer (State), the MLP in a window of 3 layers, or the Self Attn. in a window of 3 layers.

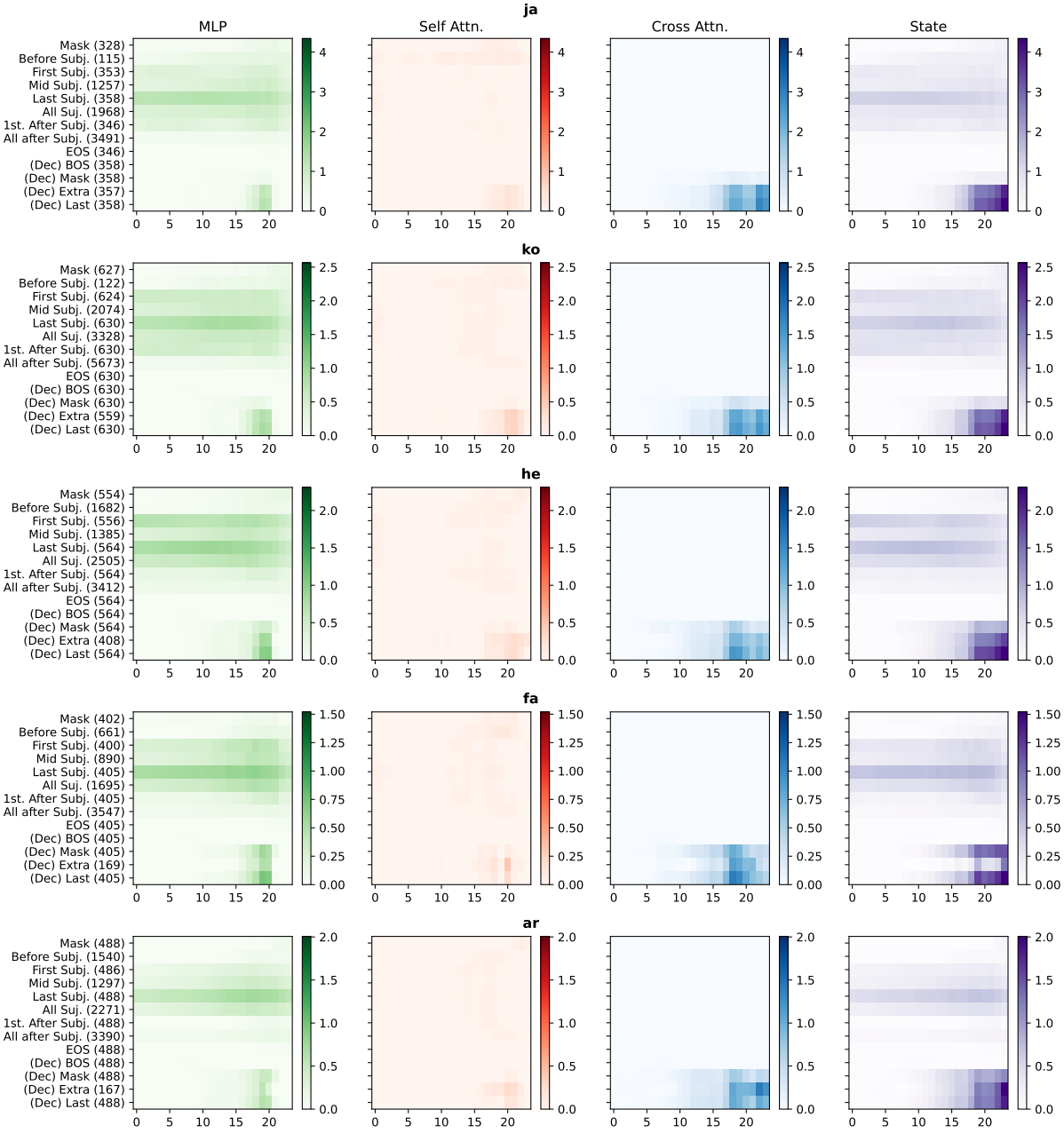


Figure 20: mT5 causal analysis for each language (Rest of the languages in Figure 19). Average **logit score** (for the originally predicted token) recovered after corrupting the subject in the input and restoring: the hidden representation at a given layer (State), the MLP in a window of 3 layers, or the Self Attn. in a window of 3 layers.

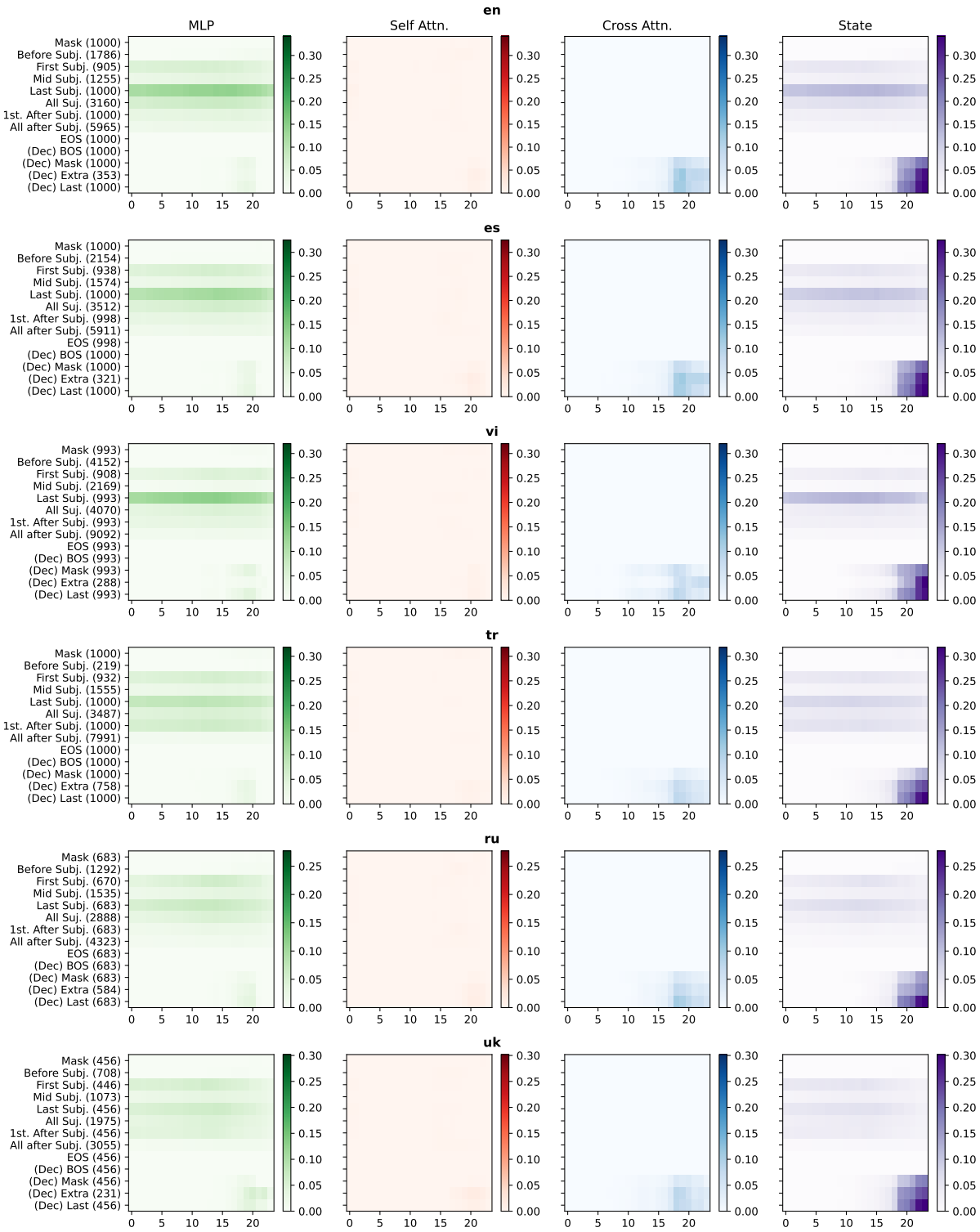


Figure 21: mT5 causal analysis for each language (continues in Figure 22). Average **probability** (for the originally predicted token) recovered after corrupting the subject in the input and restoring: the hidden representation at a given layer (State), the MLP in a window of 3 layers, or the Self Attn. in a window of 3 layers.

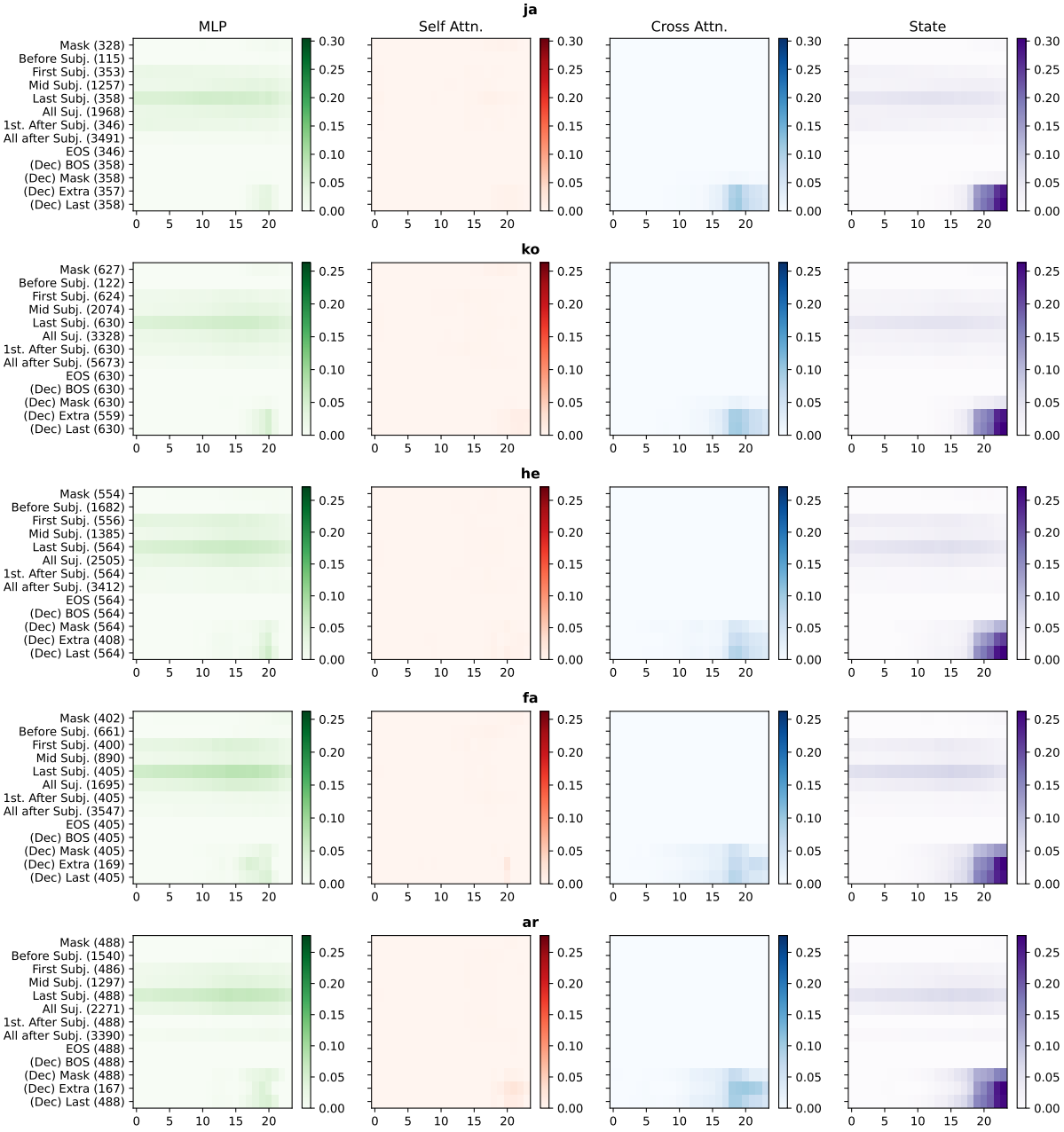


Figure 22: mT5 causal analysis for each language (Rest of the languages in Figure 21). Average **probability** (for the originally predicted token) recovered after corrupting the subject in the input and restoring: the hidden representation at a given layer (State), the MLP in a window of 3 layers, or the Self Attn. in a window of 3 layers.

C Attention Knockout

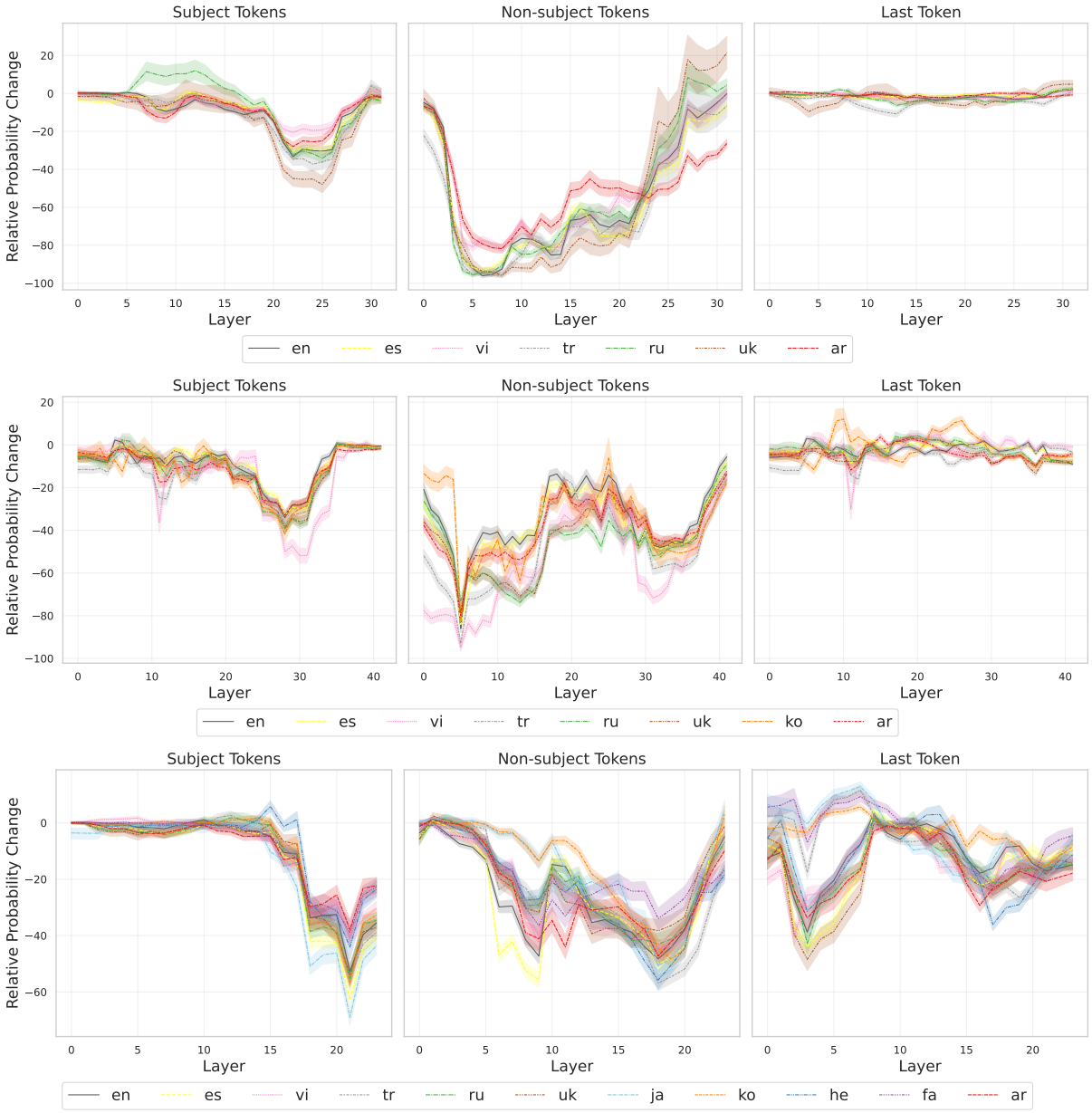


Figure 23: Attention knockout between the last token and a given set of tokens. Each layer represents the effect of the knockout on a window of w layers. Models from top to bottom: XGLM ($w = 6$), EuroLLM ($w = 7$), mT5 ($w = 4$).

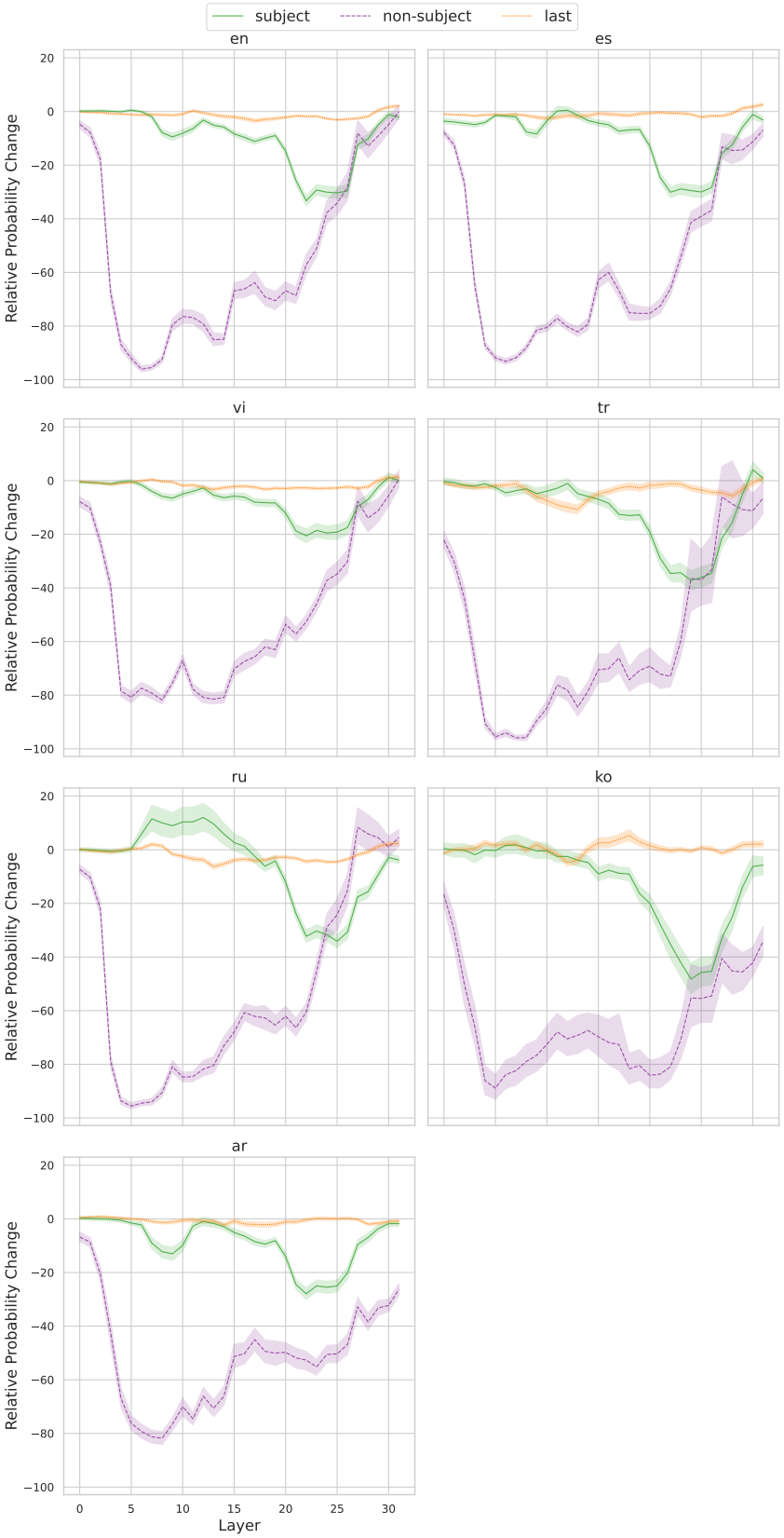


Figure 24: XGLM attention knockout for each language.

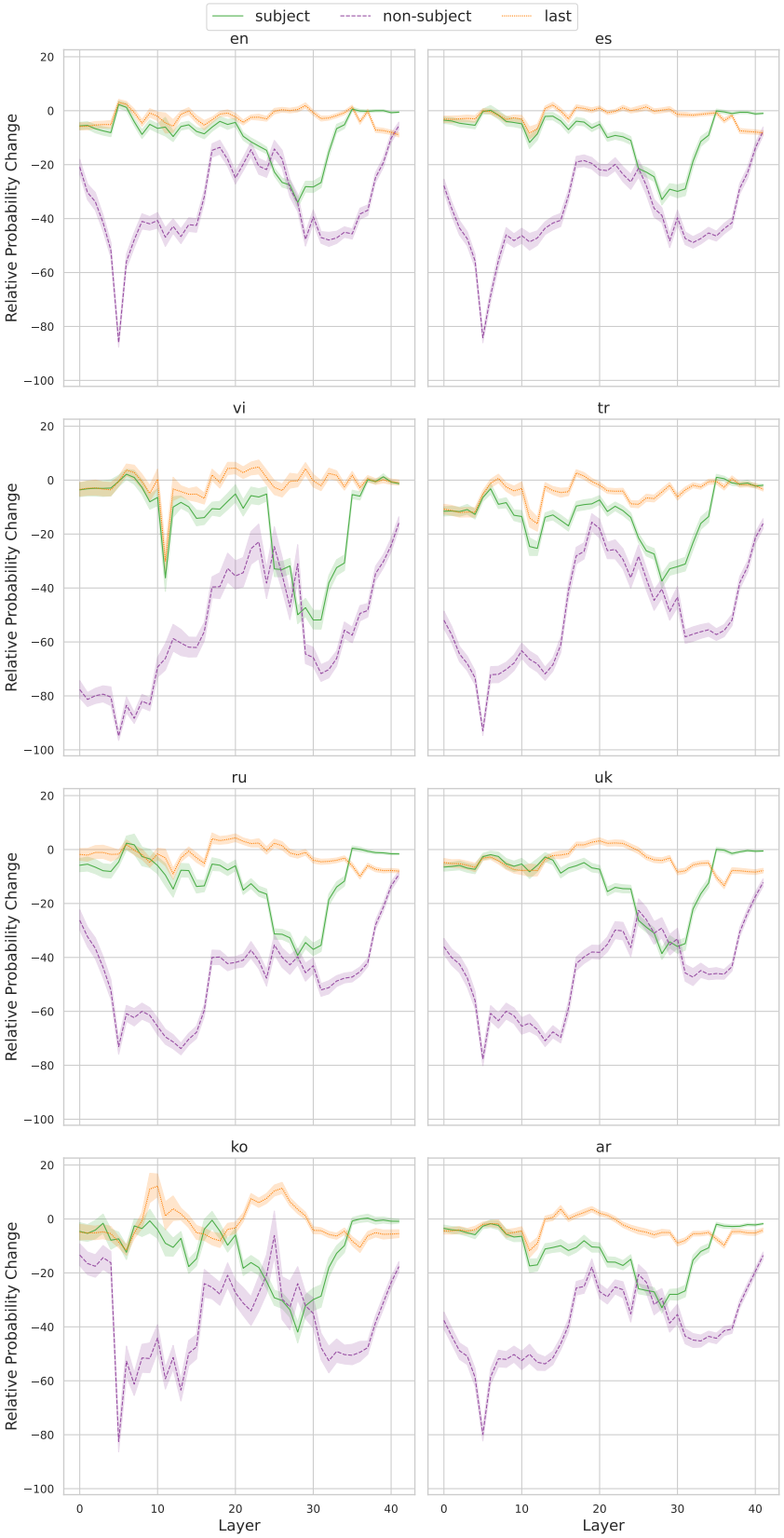


Figure 25: EUROLLM attention knockout for each language.

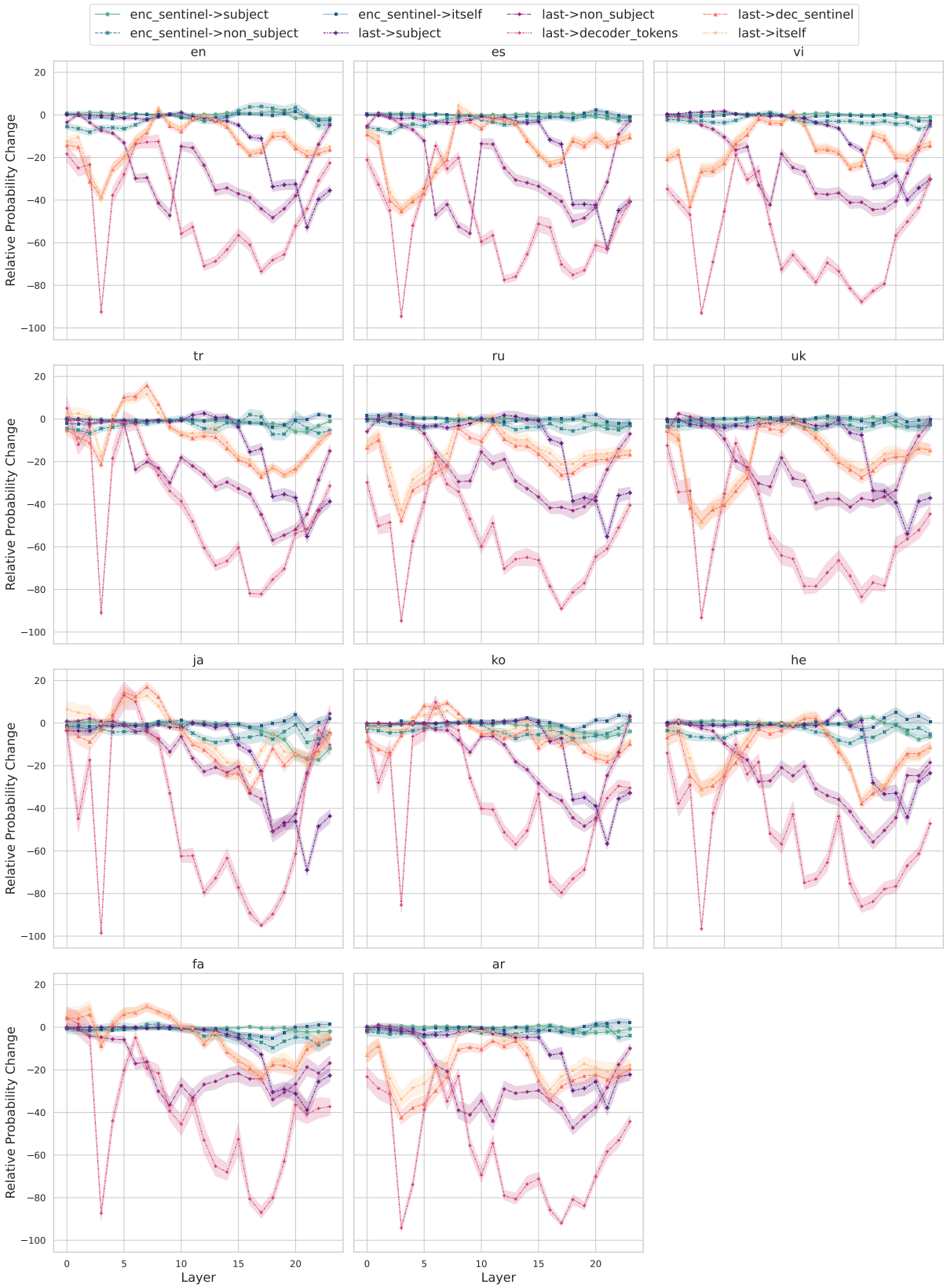


Figure 26: Attention knockout for each language in mT5. The knockout is performed between the last token in the decoder ("last") to a given set of tokens $\{t\}$, and between the masked token in the encoder ("enc_sentinel") and $\{t\}$. From the encoder sentinel there is no much flow of information so these were not included in the main body.

D Extraction Event

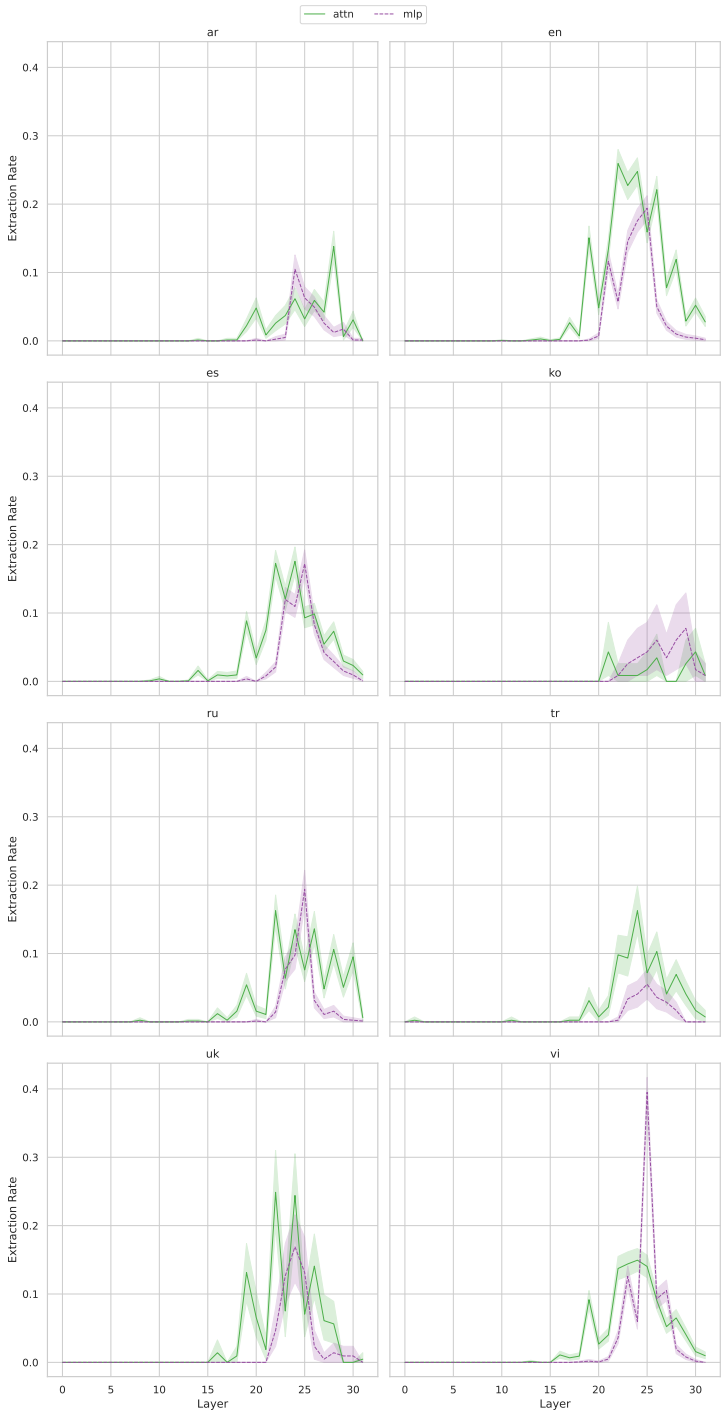


Figure 27: XGLM extraction rates for each language.

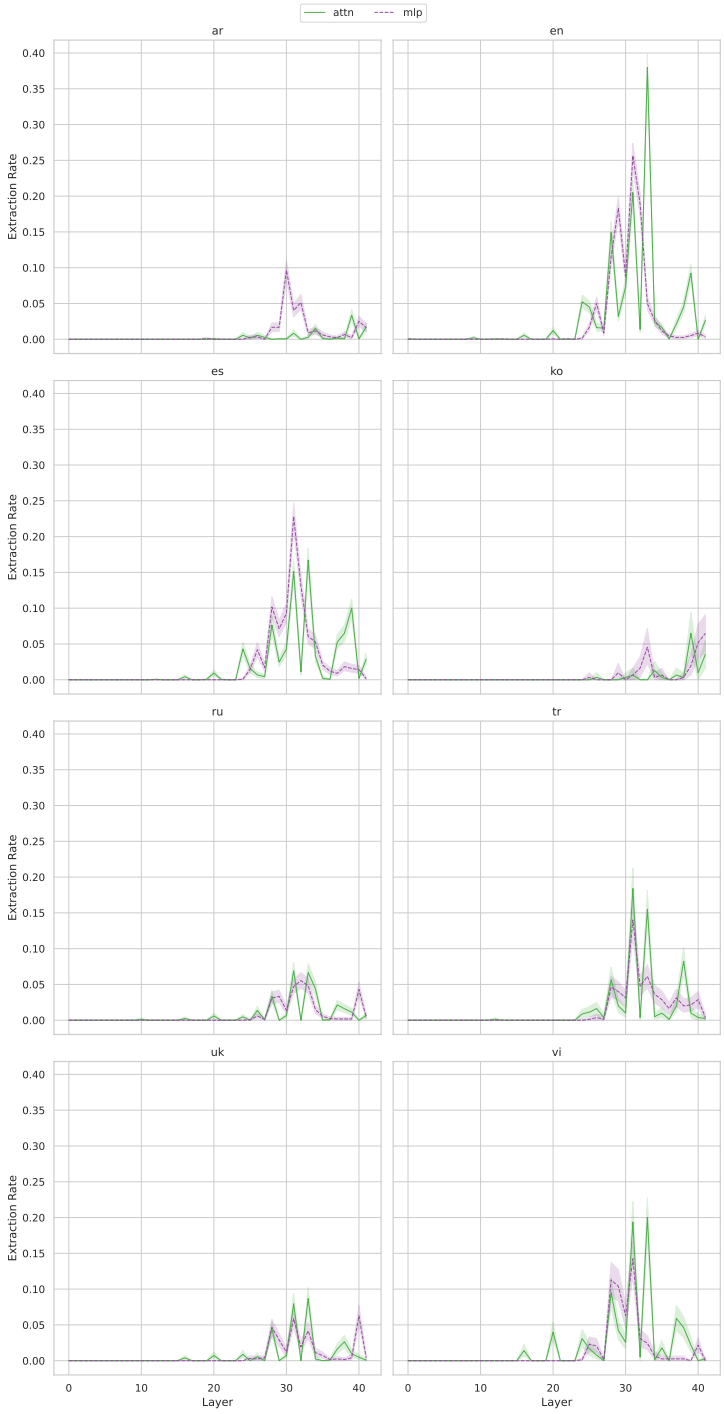


Figure 28: EUROLLM extraction rates for each language.

Figure 29: mT5 extraction rates for each language.

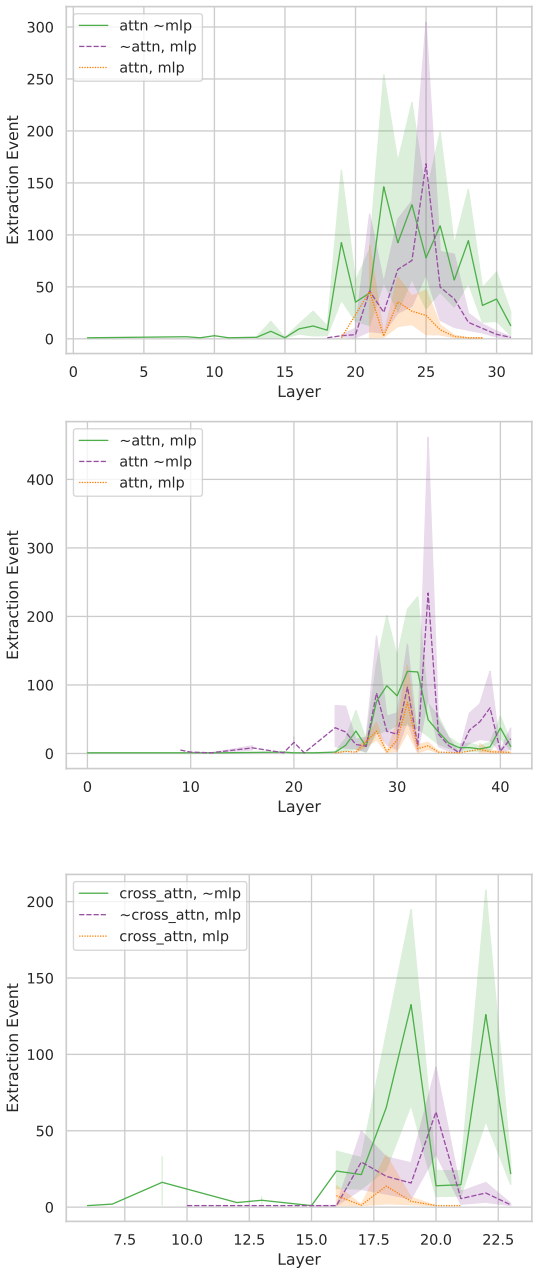


Figure 30: Number of extraction events split by precedence (or not) of an extraction event in the self-attn or cross-attn. Models from top to bottom: XGLM, EuroLLM, mT5.

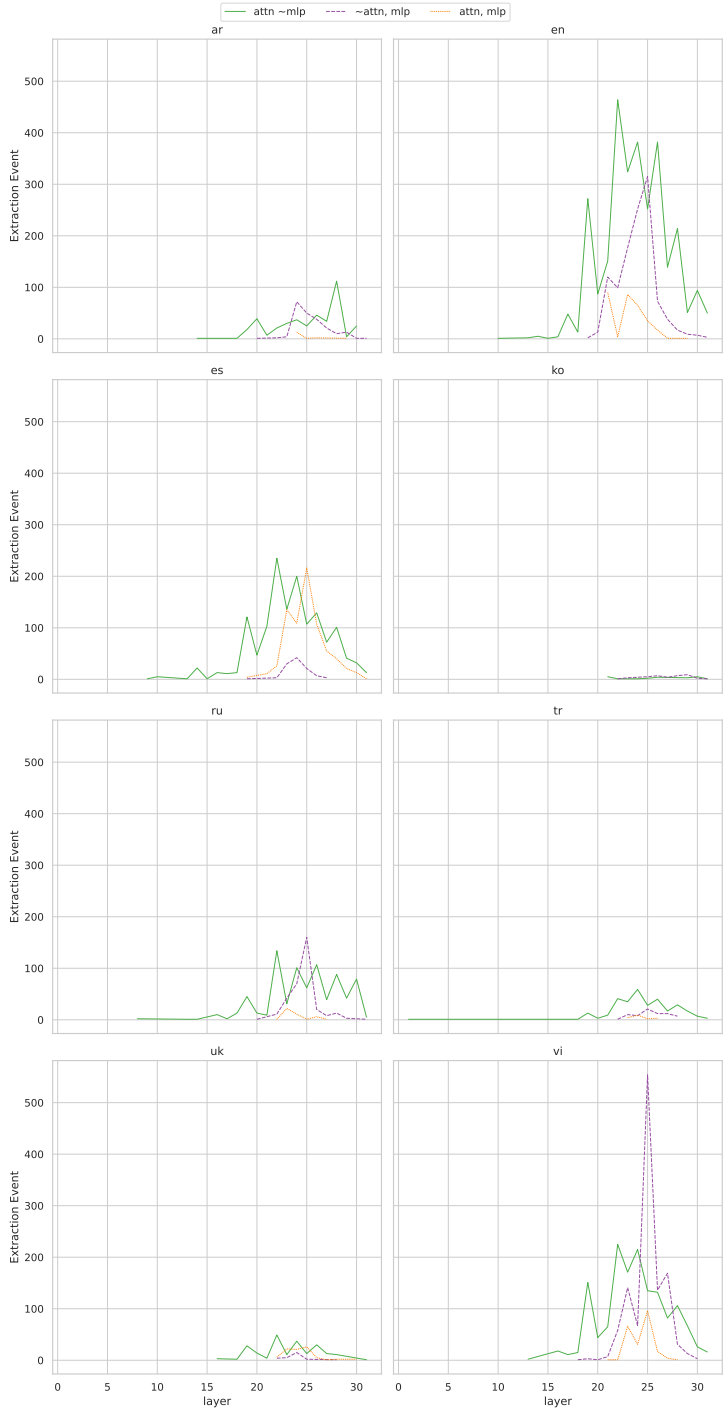


Figure 31: Number of extraction events split by precedence (or not) of an extraction event in the self-attn in XGLM for each language.

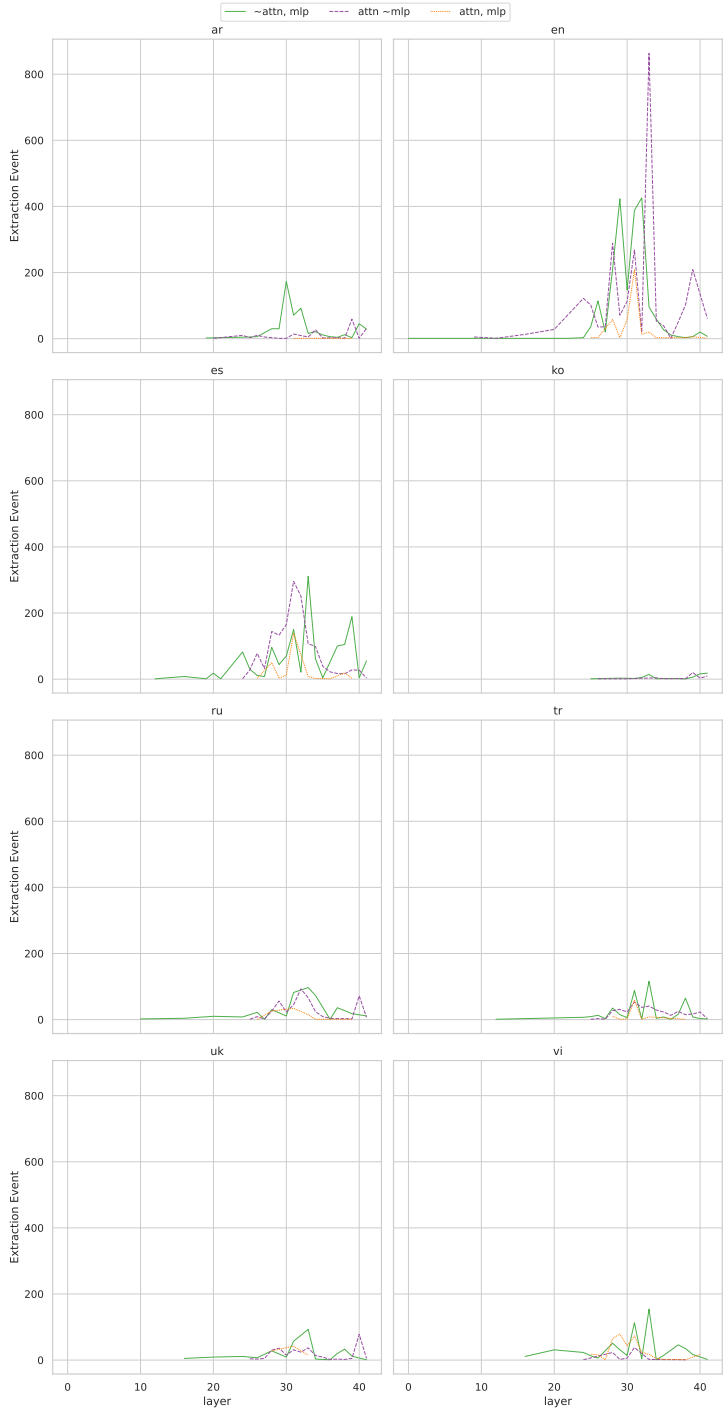


Figure 32: Number of extraction events split by precedence (or not) of an extraction event in the self-attn in EUROLLM for each language.

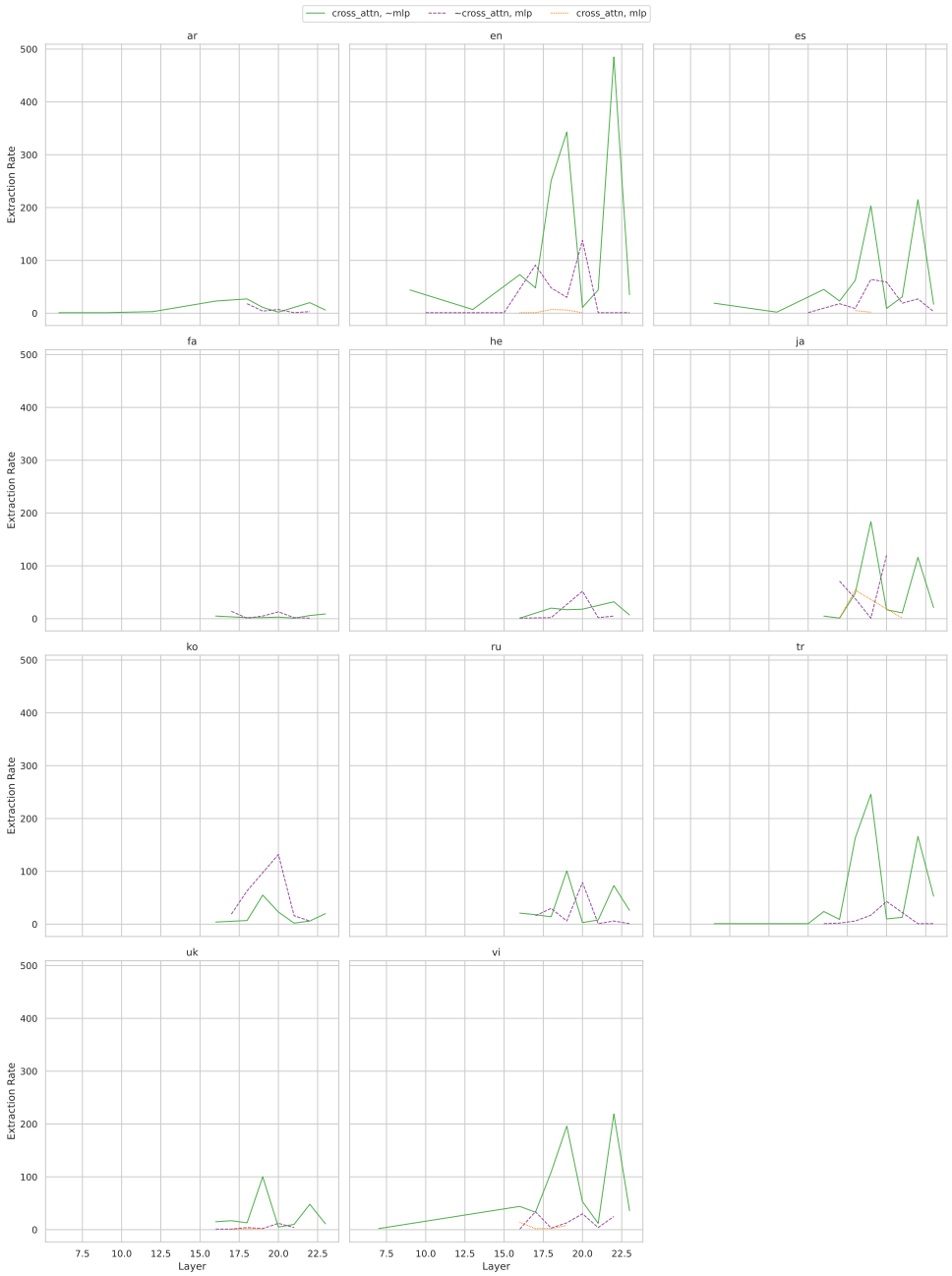


Figure 33: Number of extraction events split by precedence (or not) of an extraction event in the cross-attn in mT5 for each language.

E Patching

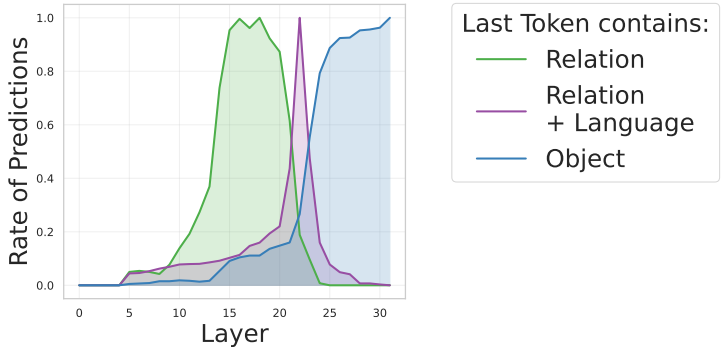


Figure 34: Aggregated patching results (§6) for XGLM. The last token representation contains the function that solves the task. This function is formed in two stages: first, the relation to extract is encoded (green), and then the language is composed into the function (purple). Finally, the function is applied to the subject, and the predicted object is resolved (blue).

	XGLM			EuroLLM			mT5		
Patch - Context	All	$\mathcal{L}_c(o_p)$	$\mathcal{L}_p(o_c)$	All	$\mathcal{L}_c(o_p)$	$\mathcal{L}_p(o_c)$	All	$\mathcal{L}_c(o_p)$	$\mathcal{L}_p(o_c)$
en-es	849	834	759	880	862	755	842	831	780
en-vi	862	857	836	930	910	843	854	844	825
en-tr	974	961	961	966	945	914	815	803	794
en-ru	823	821	817	846	846	827	796	795	787
en-uk	914	914	911	915	915	915	853	850	852
en-ko	995	995	991	991	991	991	744	744	743
en-he	962	962	961	810	810	775	778	778	778
en-fa	537	537	532	771	771	734	714	714	712
en-ar	829	829	828	862	859	858	775	775	774
en-ja	134	134	134	895	895	872	774	774	773

Table 4: Total number of patch-context examples considered in the patching experiments with $\{\neq \mathcal{L}, = r, \neq s\}$. The $\mathcal{L}_c(o_p)$ column is the total number of examples where the detection of $\mathcal{L}_c(o_p)$ would be unambiguous, that is, $\mathcal{L}_c(o_p) \neq \mathcal{L}_p(o_p)$, conversely for the $\mathcal{L}_p(o_c)$ column.

	XGLM		EuroLLM		mT5	
Patch - Context	$\mathcal{L}_c(o_p)$	$\mathcal{L}_p(o_c)$	$\mathcal{L}_c(o_p)$	$\mathcal{L}_p(o_c)$	$\mathcal{L}_c(o_p)$	$\mathcal{L}_p(o_c)$
en-es	0.7% (6)	**26.0% (197)**	5.0% (43)	**34.2% (258)**	0.5% (4)	**21.5% (168)**
en-vi	0.5% (4)	**34.4% (288)**	0.5% (5)	**16.1% (136)**	0.1% (1)	**20.2% (167)**
en-tr	0.2% (2)	**16.4% (158)**	0.2% (2)	**25.4% (232)**	0.4% (3)	**26.2% (208)**
en-ru	2.2% (18)	**51.4% (420)**	4.6% (39)	**63.5% (525)**	1.1% (9)	**36.1% (284)**
en-uk	0.5% (5)	**21.3% (194)**	6.9% (63)	**62.5% (572)**	0.2% (2)	**31.8% (271)**
en-ko	0.3% (3)	**47.1% (467)**	0.1% (1)	**75.7% (750)**	0.3% (2)	**32.3% (240)**
en-he	0.0% (0)	**27.3% (262)**	0.0% (0)	**65.9% (511)**	0.4% (3)	**38.4% (299)**
en-fa	0.0% (0)	**41.7% (222)**	0.3% (2)	**64.0% (470)**	1.0% (7)	**25.1% (179)**
en-ar	8.7% (72)	**39.5% (327)**	4.5% (39)	**65.7% (564)**	0.6% (5)	**31.9% (247)**
en-ja	0.0% (0)	**3.0% (4)**	1.9% (17)	**11.5% (100)**	0.0% (0)	**26.0% (201)**

Table 5: Proportion of times an object is predicted in the other language in the patching experiments with $\{\neq \mathcal{L}, = r, \neq s\}$. In parenthesis the number of examples corresponding to the percentage. In bold when $\mathcal{L}_c(o_p)$ or $\mathcal{L}_p(o_c)$ are detected more often for each of the experiments. The total number of examples varies, see total numbers in Table 4.

	XGLM			EUROLLM			mT5		
Patch - Context	All	$\mathcal{L}_c(o_p)$	$\mathcal{L}_p(o_c)$	All	$\mathcal{L}_c(o_p)$	$\mathcal{L}_p(o_c)$	All	$\mathcal{L}_c(o_p)$	$\mathcal{L}_p(o_c)$
en-es	324	284	298	413	360	391	263	253	249
en-vi	336	329	333	105	102	100	209	207	204
en-tr	88	77	84	111	102	110	209	207	204
en-ru	218	216	218	420	415	419	191	191	190
en-uk	35	35	35	351	347	350	136	136	133
en-ko	46	46	46	98	98	98	165	165	165
en-ar	212	212	212	391	390	391	159	159	159
en-he	-	-	-	19	19	19	166	166	166
en-ja	-	-	-	1	1	1	85	85	85
en-fa	-	-	-	-	-	-	114	114	114

Table 6: Total number of patch-context examples considered in the patching experiments with $\{\neq \mathcal{L}, \neq r, = s\}$. The $\mathcal{L}_c(o_p)$ column is the total number of examples where the detection of $\mathcal{L}_c(o_p)$ would be unambiguous, that is, $\mathcal{L}_c(o_p) \neq \mathcal{L}_p(o_p)$, conversely for the $\mathcal{L}_p(o_c)$ column.

	XGLM		EUROLLM		mT5	
Patch - Context	$\mathcal{L}_c(o_p)$	$\mathcal{L}_p(o_c)$	$\mathcal{L}_c(o_p)$	$\mathcal{L}_p(o_c)$	$\mathcal{L}_c(o_p)$	$\mathcal{L}_p(o_c)$
en-es	**18.0% (51)**	3.7% (11)	**41.7% (150)**	2.8% (11)	0.8% (2)	**7.6% (19)**
en-vi	**9.1% (30)**	6.6% (22)	3.9% (4)	**8.0% (8)**	1.0% (2)	**12.3% (25)**
en-tr	**36.4% (28)**	6.0% (5)	**56.9% (58)**	4.5% (5)	2.9% (6)	**9.3% (19)**
en-ru	**64.8% (140)**	4.1% (9)	**61.7% (256)**	9.3% (39)	6.3% (12)	**8.9% (17)**
en-uk	17.1% (6)	**17.1% (6)**	**66.0% (229)**	2.3% (8)	3.7% (5)	**9.0% (12)**
en-ko	**41.3% (19)**	21.7% (10)	**71.4% (70)**	5.1% (5)	2.4% (4)	**12.1% (20)**
en-ar	**40.1% (85)**	1.9% (4)	**52.8% (206)**	0.8% (3)	1.9% (3)	**7.5% (12)**
en-he	-	-	0.0% (0)	0.0% (0)	3.6% (6)	**15.7% (26)**
en-ja	-	-	0.0% (0)	0.0% (0)	0.0% (0)	**25.9% (22)**
en-fa	-	-	-	-	2.6% (3)	**15.8% (18)**

Table 7: Proportion of times an object is predicted in the other language in the patching experiments with $\{\neq \mathcal{L}, \neq r, = s\}$. In parenthesis the number of examples corresponding to the percentage. In bold when $\mathcal{L}_c(o_p)$ or $\mathcal{L}_p(o_c)$ are detected more often for each of the experiments. The total number of examples varies, see total numbers in Table 6.

E.1 Different Relation, Different Subject

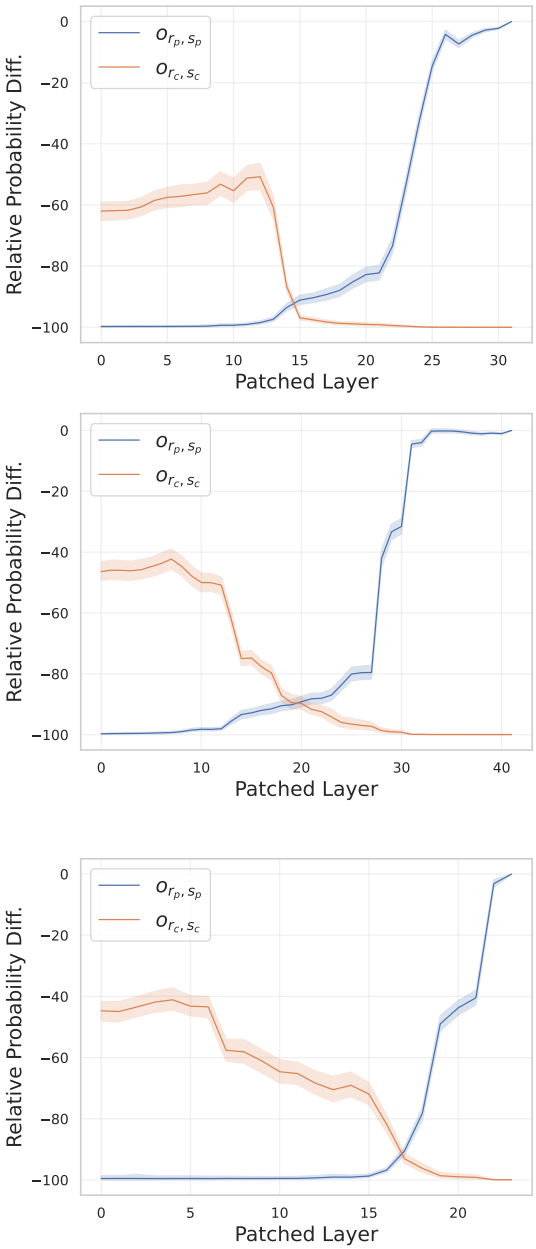


Figure 35: Probability of the patch answer and the context answer when patching at different layers. Models from top to bottom: XGLM, EuroLLM, mT5.

E.2 Same Relation, Different Subject

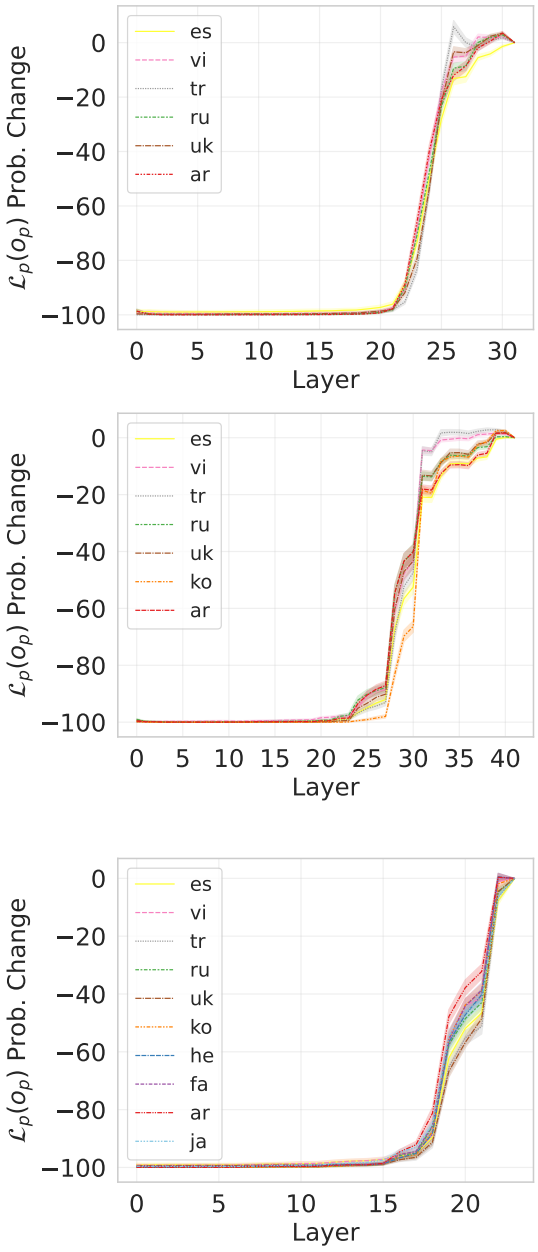


Figure 36: Probability of the patch answer $\mathcal{L}_p(o_p)$ when patching at different layers. Models from top to bottom: XGLM, EUROLLM, mT5.

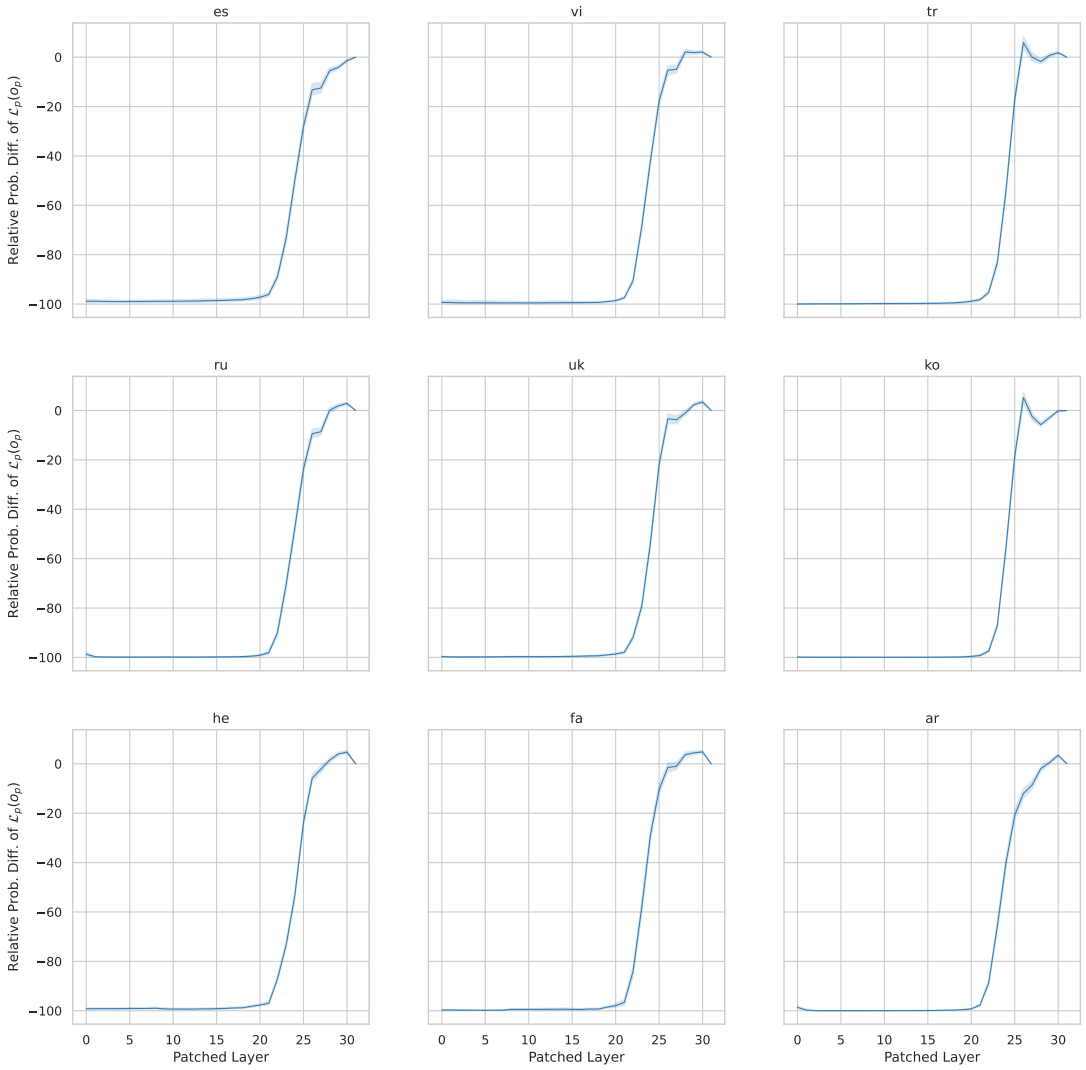


Figure 37: Probability of the patch answer $\mathcal{L}_p(o_p)$ when patching at different layers in XGLM, for examples with $\{\neq \mathcal{L}, = r, \neq s\}$.

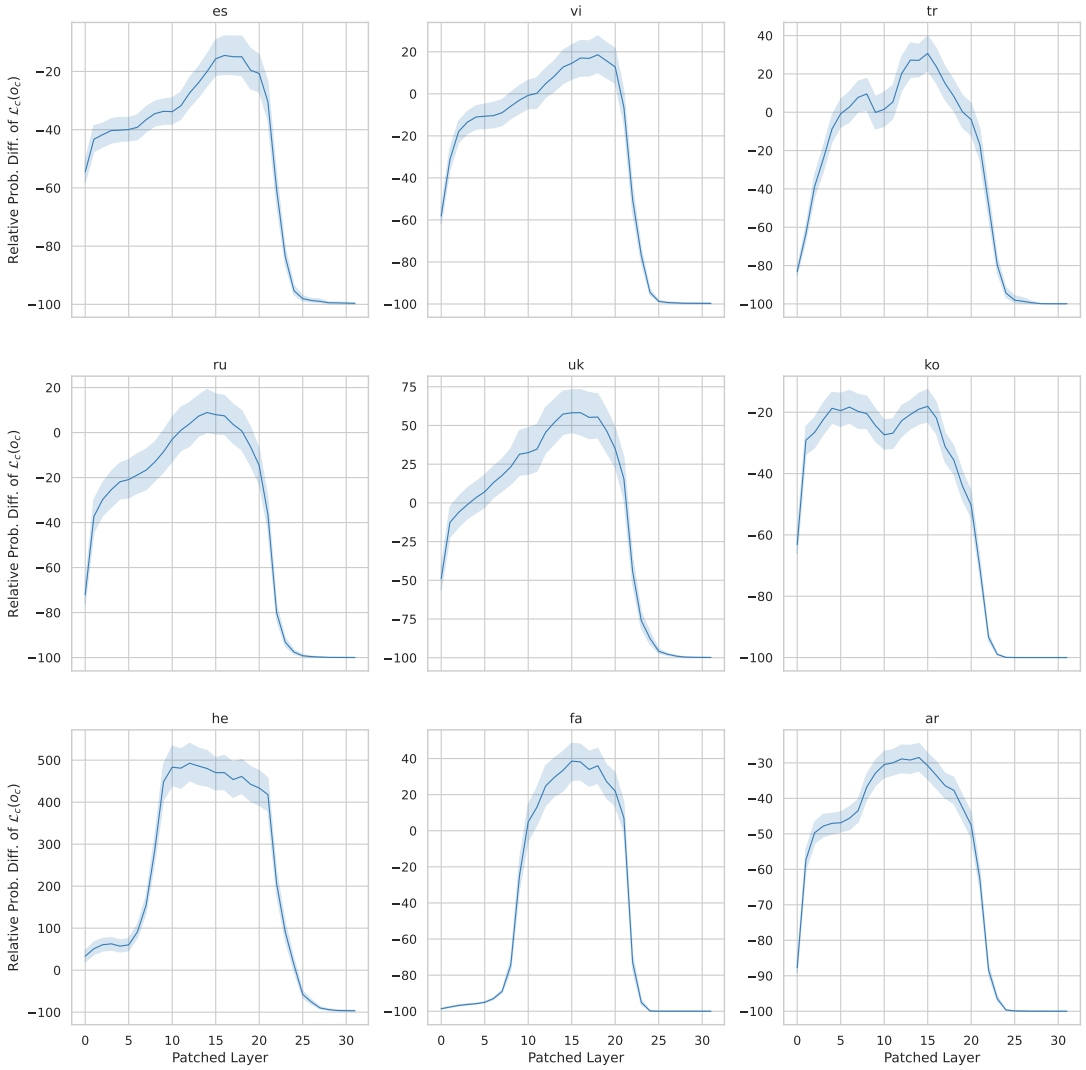


Figure 38: Probability of the $\mathcal{L}_c(o_c)$ when patching at different layers in XGLM, for examples with $\{\neq \mathcal{L}, = r, \neq s\}$. Note that the plots do not share the y-axis.

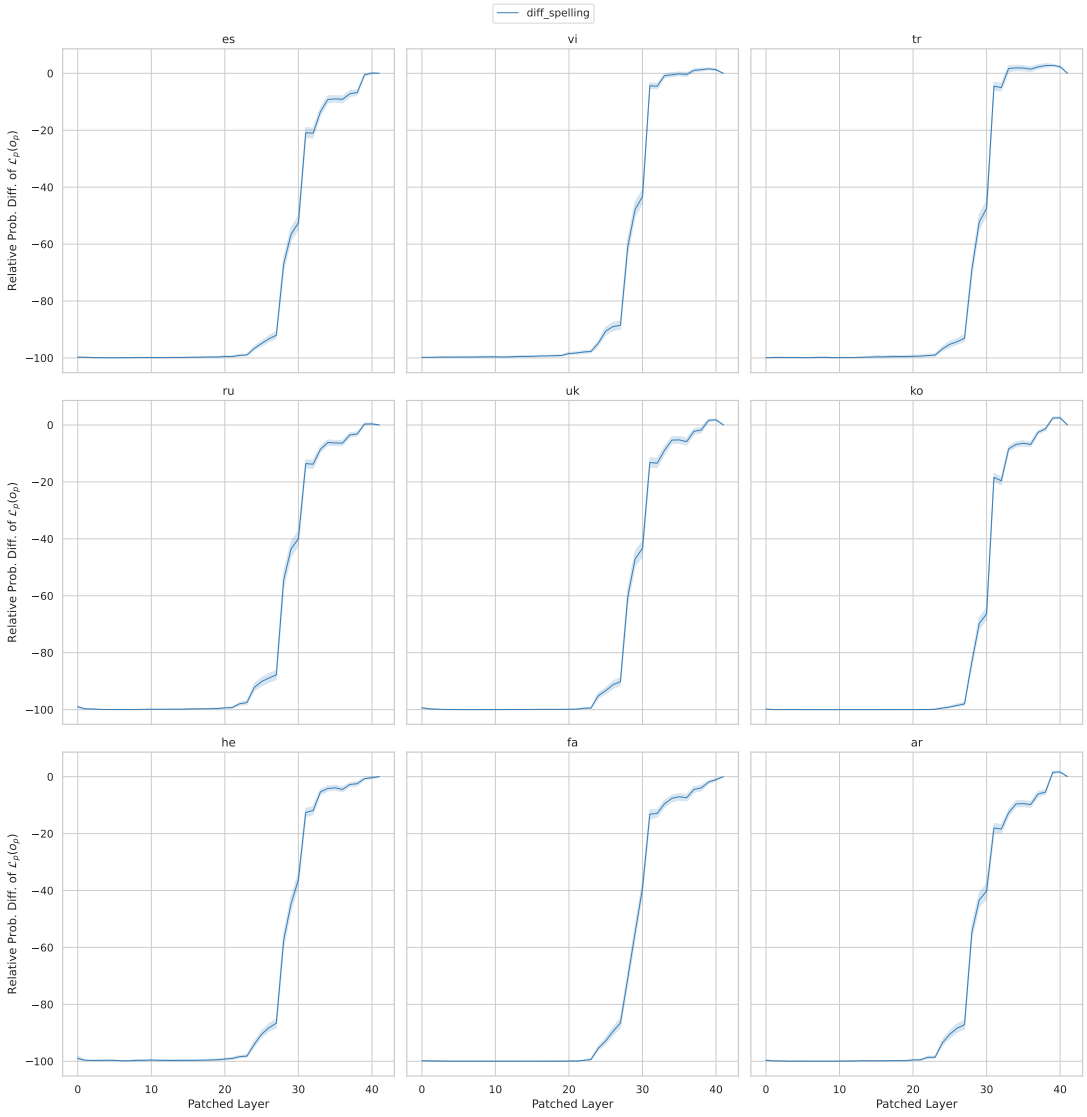


Figure 39: Probability of the patch answer $\mathcal{L}_p(o_p)$ when patching at different layers in EUROLLM, for examples with $\{\neq \mathcal{L}, = r, \neq s\}$.

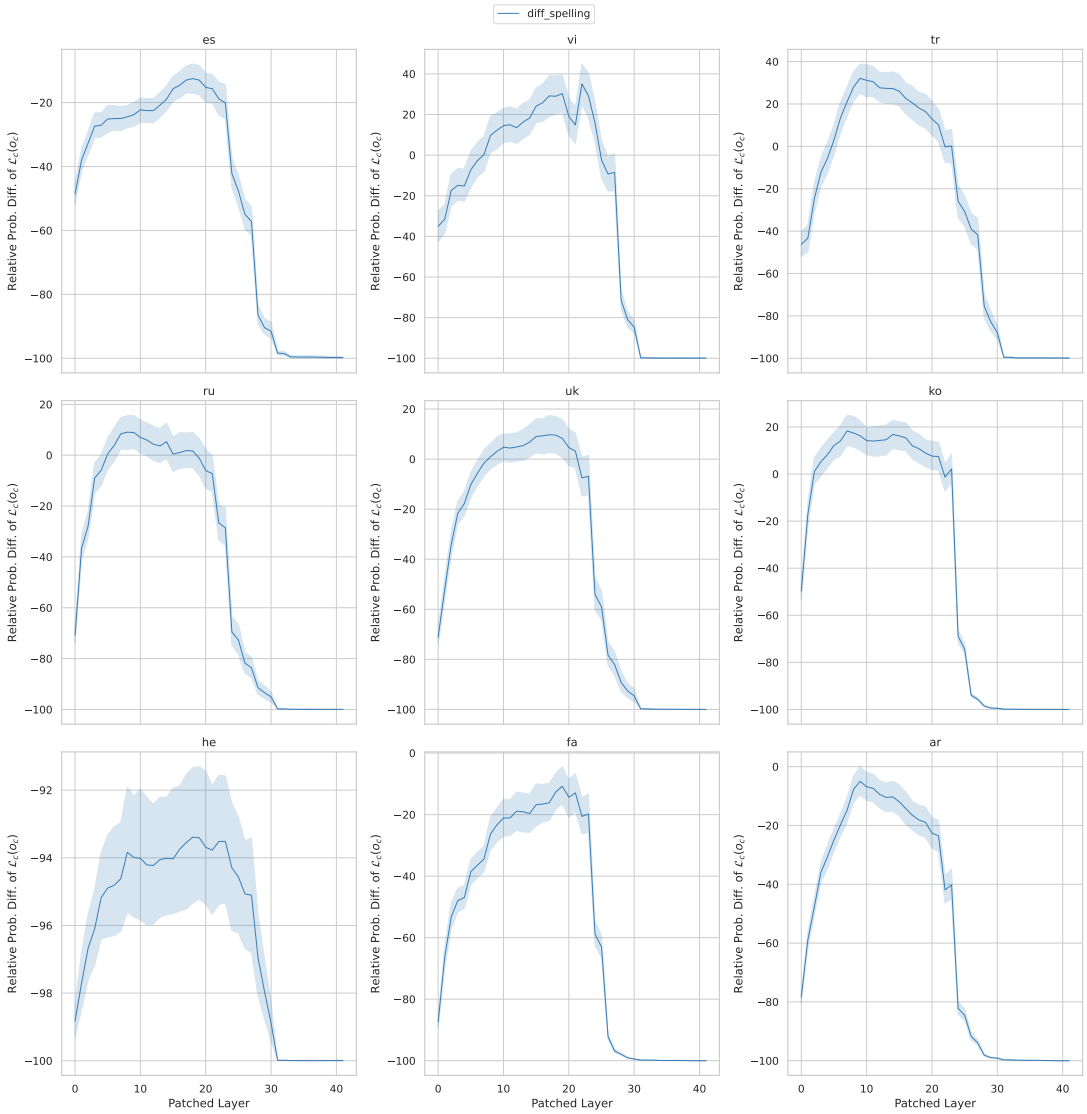


Figure 40: Probability of the $\mathcal{L}_c(o_c)$ when patching at different layers in EUROLLM, for examples with $\{\neq \mathcal{L}, = r, \neq s\}$. Note that the plots do not share the y-axis.

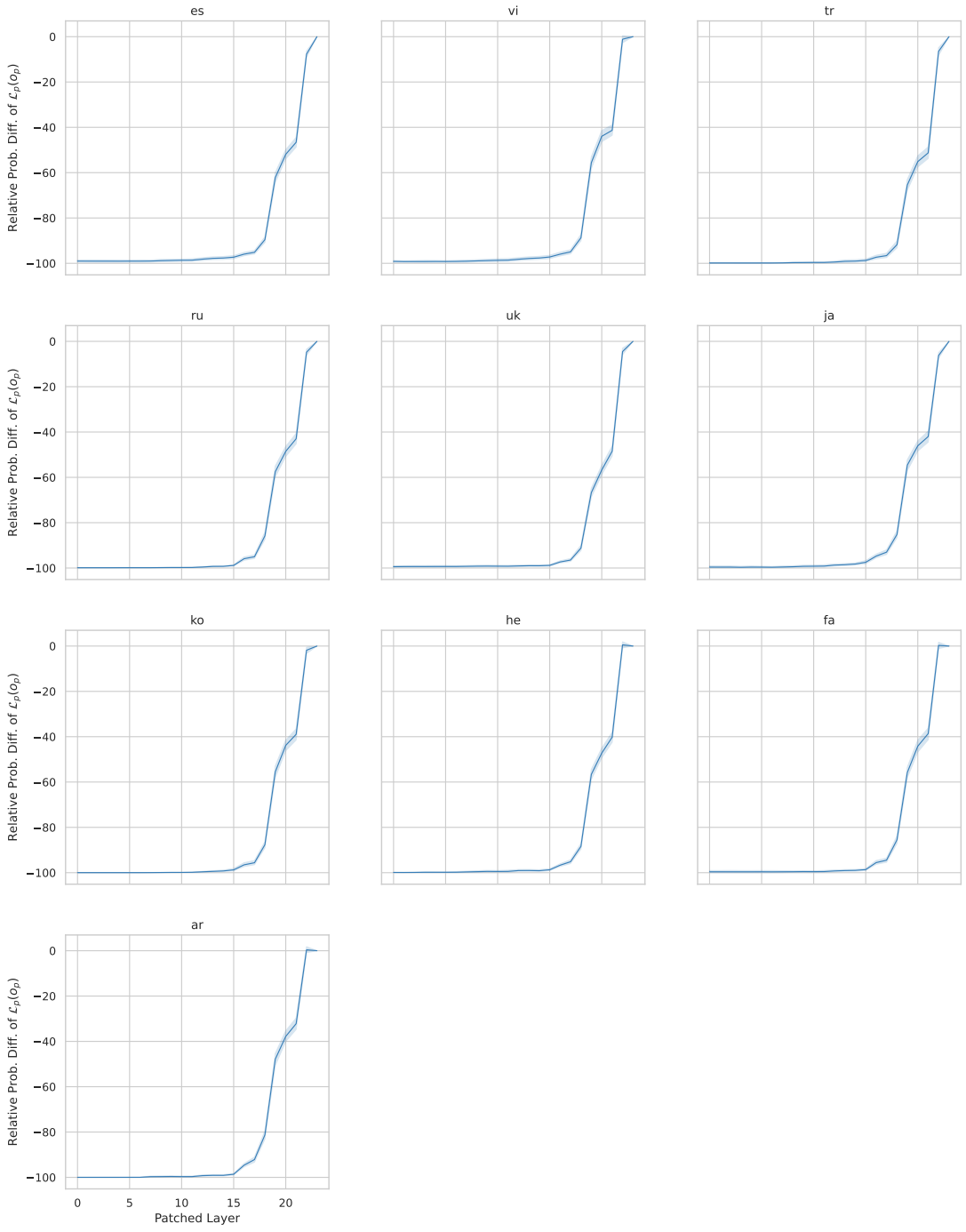


Figure 41: Probability of the $\mathcal{L}_p(o_p)$ when patching at different layers in mT5, for examples with $\{\neq \mathcal{L}, = r, \neq s\}$.

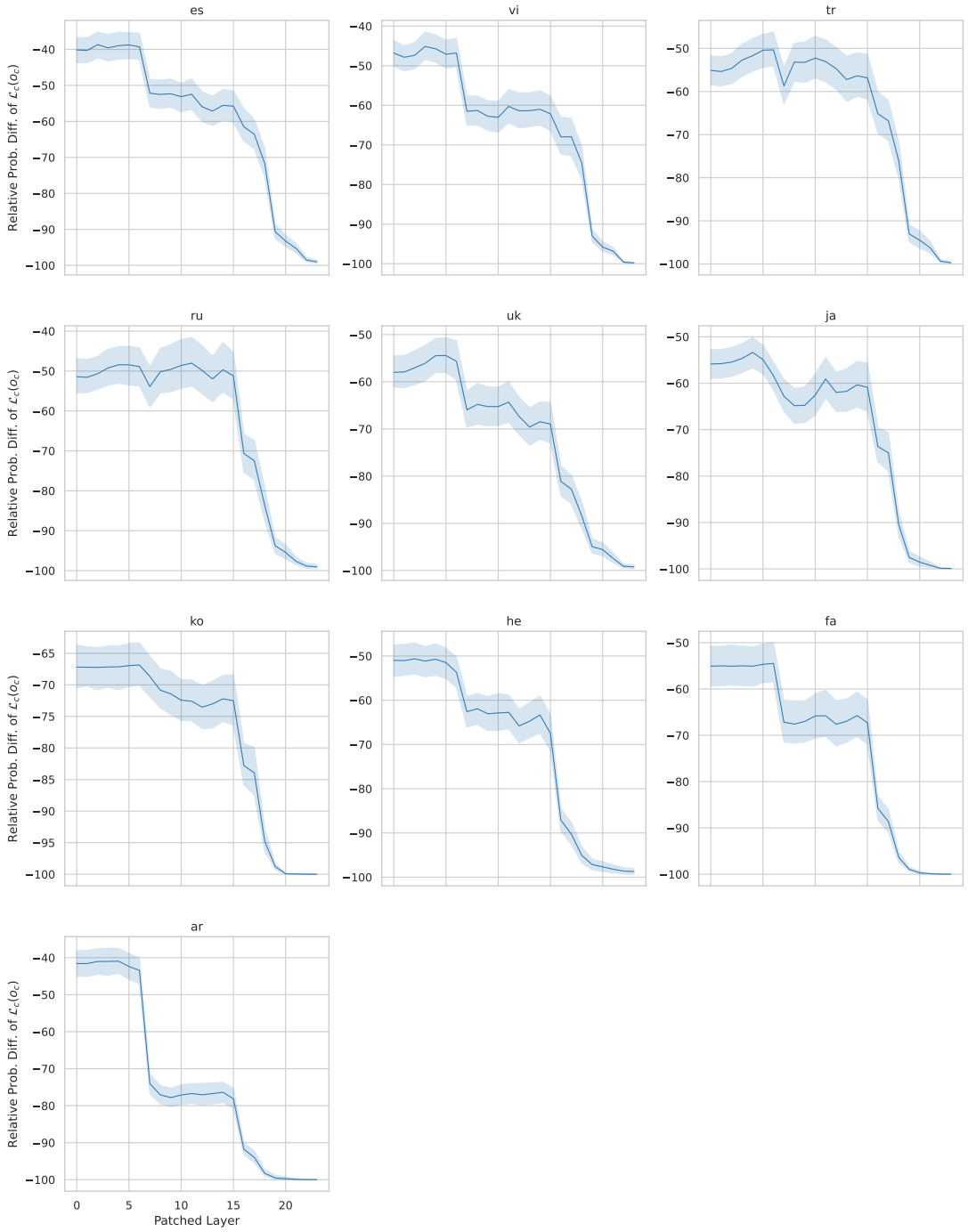


Figure 42: Probability of the $\mathcal{L}_c(o_c)$ when patching at different layers in mT5, for examples with $\{\neq \mathcal{L}, = r, \neq s\}$. Note that the plots do not share the y-axis.

E.3 Different Relation, Same Subject

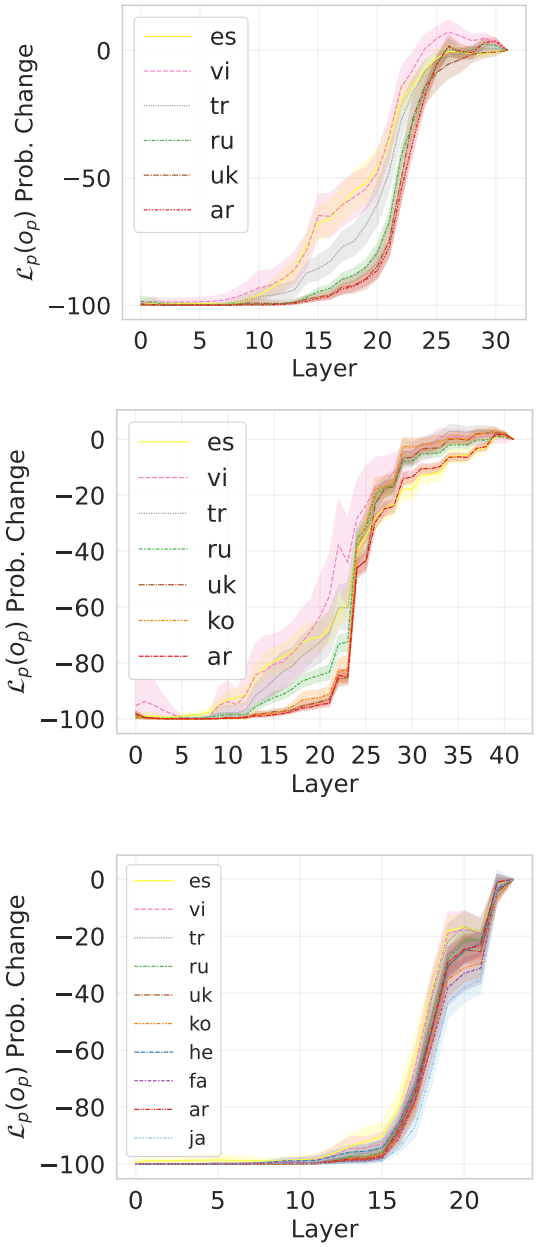


Figure 43: Probability of the patch answer $\mathcal{L}_p(o_p)$ when patching at different layers for examples with $\{\neq \mathcal{L}, \neq r, = s\}$. Models from top to bottom: XGLM, EUROLLM, mT5.

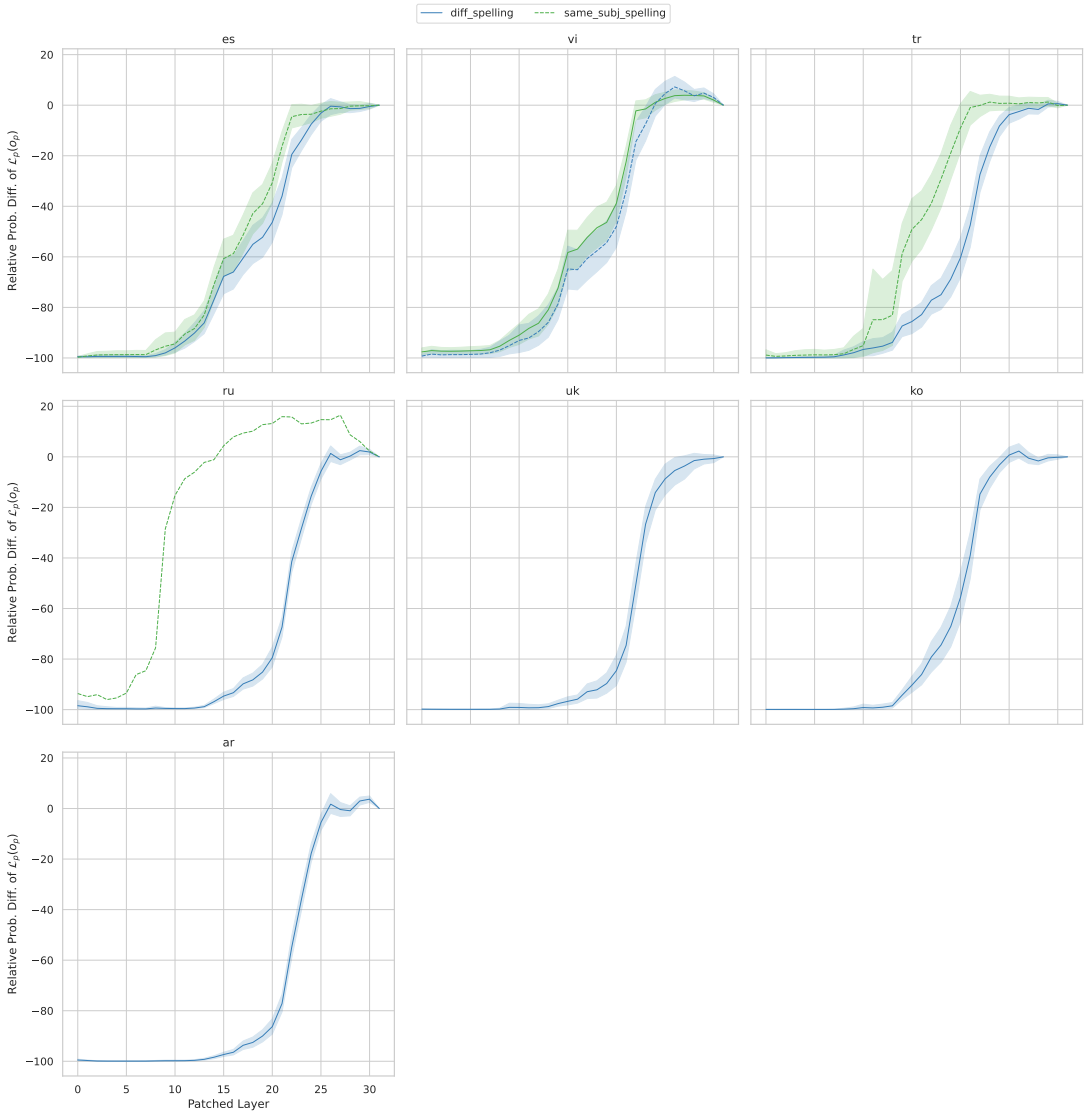


Figure 44: Probability of the $\mathcal{L}_p(o_p)$ when patching at different layers in XGLM, for examples with $\{\neq \mathcal{L}, \neq r, = s\}$.

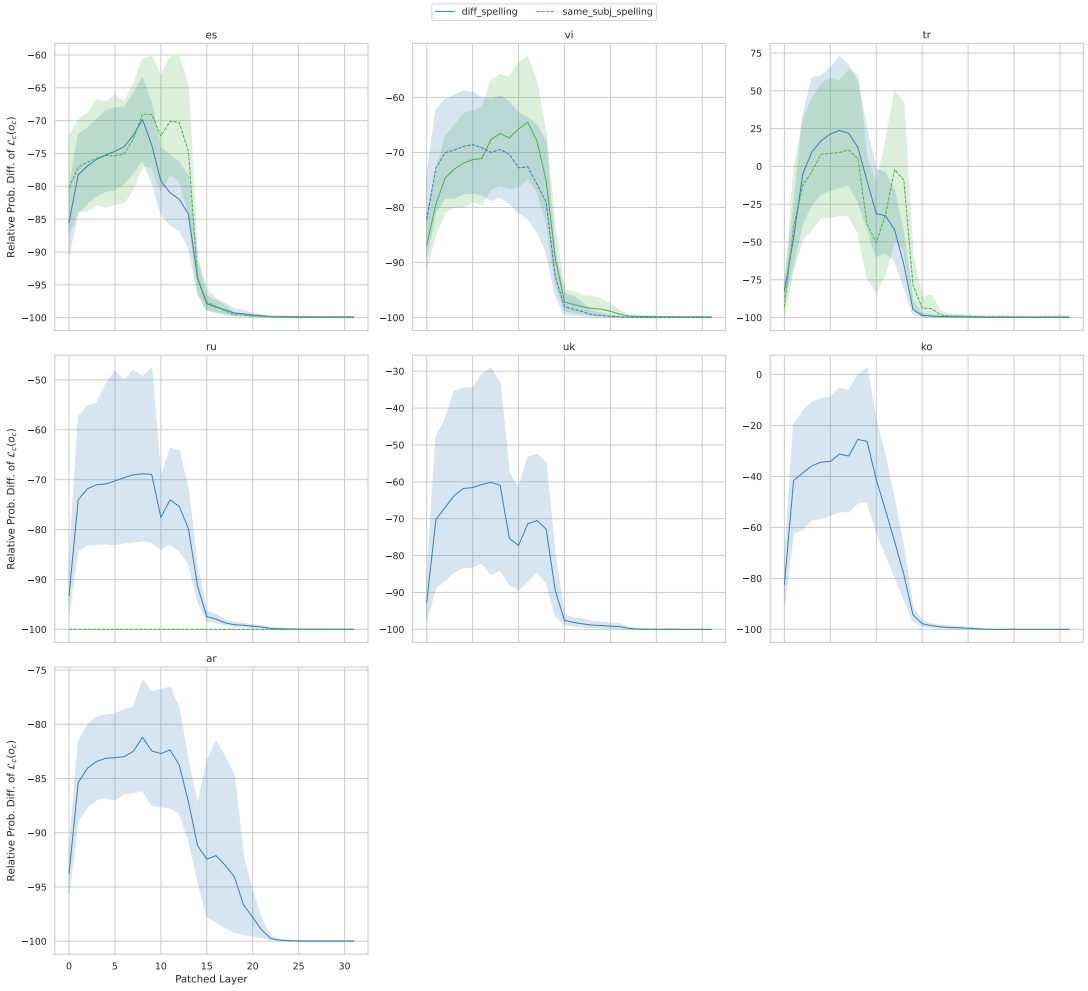


Figure 45: Probability of the $\mathcal{L}_c(o_c)$ when patching at different layers in XGLM, for examples with $\{\neq \mathcal{L}, \neq r, = s\}$. Note that the plots do not share the y-axis.

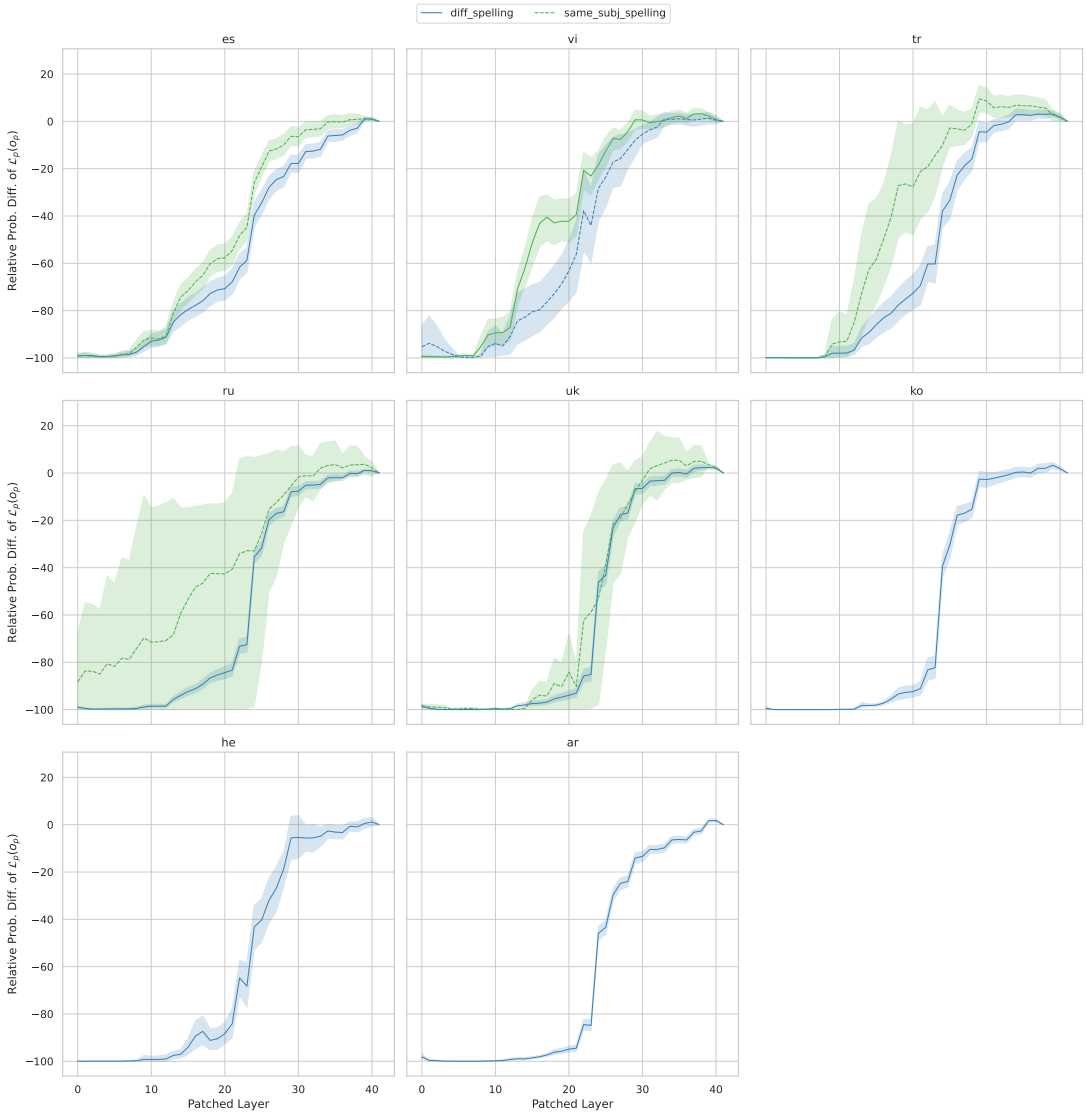


Figure 46: Probability of the $\mathcal{L}_p(o_p)$ when patching at different layers in EUROLLM, , for examples with $\{\neq \mathcal{L}, \neq r, = s\}$.

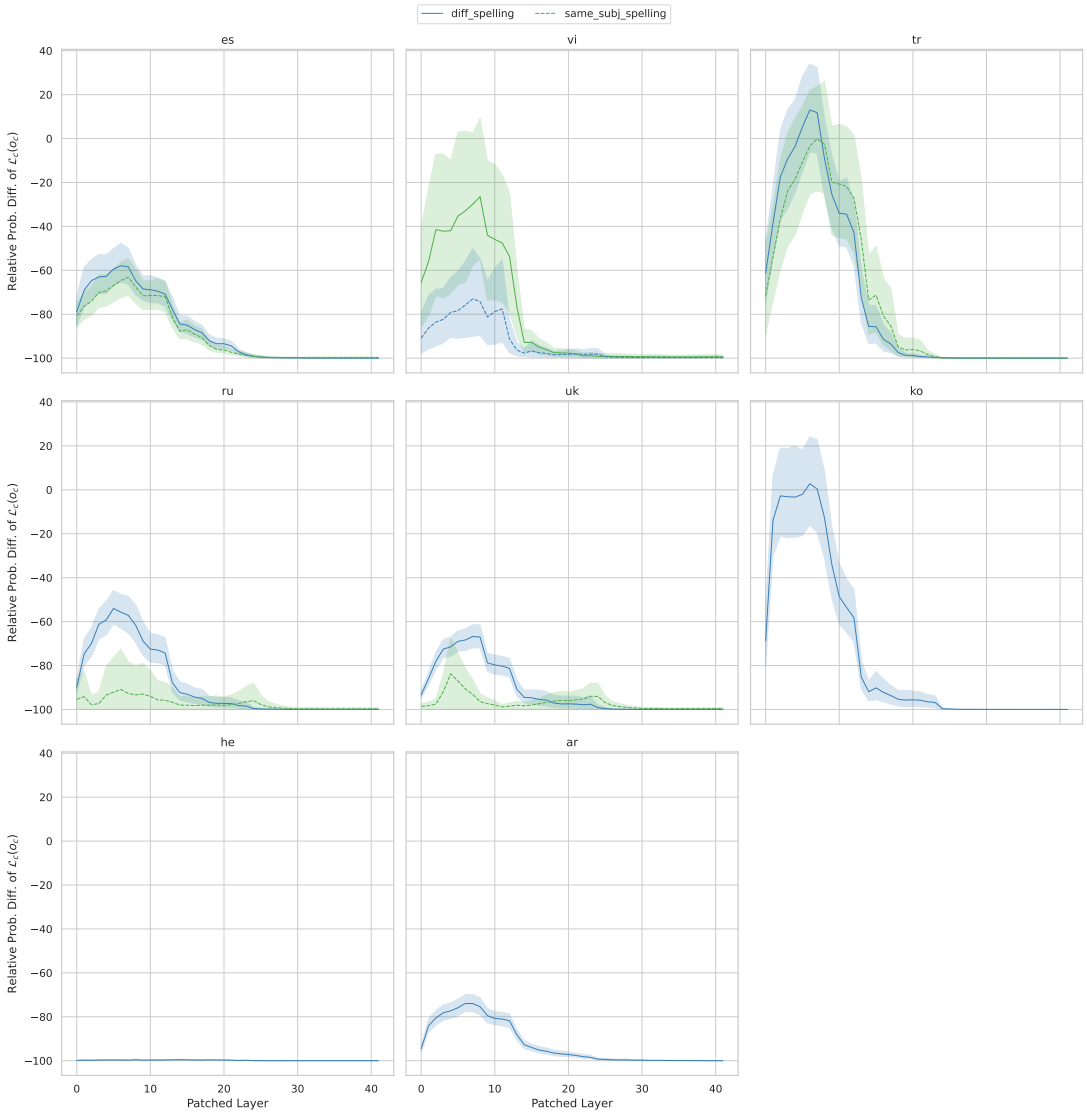


Figure 47: Probability of the $\mathcal{L}_c(o_c)$ when patching at different layers in EUROLLM, for examples with $\{\neq \mathcal{L}, \neq r, = s\}$. Note that the plots do not share the y-axis.

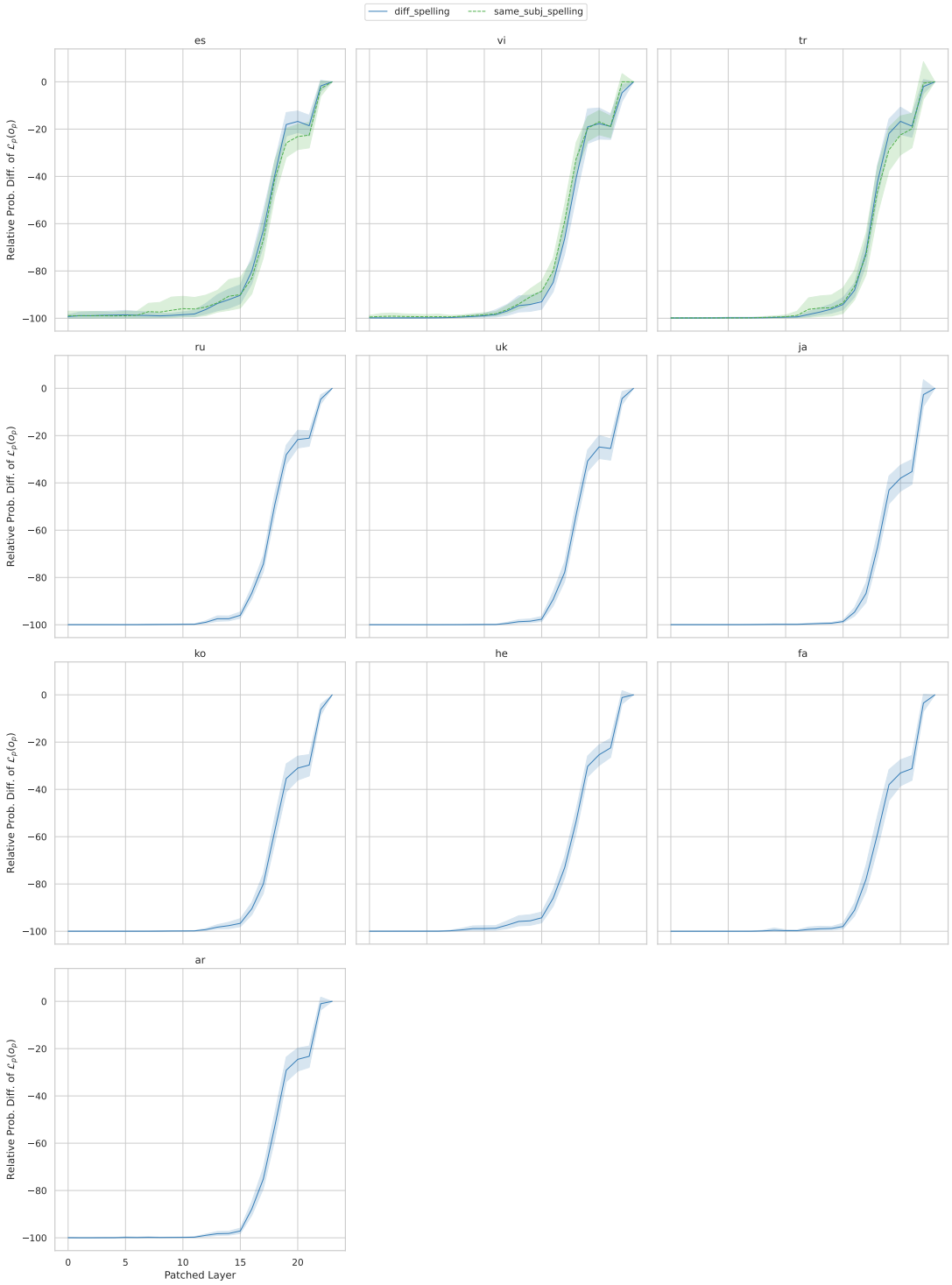


Figure 48: Probability of the $\mathcal{L}_p(o_p)$ when patching at different layers in mT5, , for examples with $\{\neq \mathcal{L}, \neq r, = s\}$.

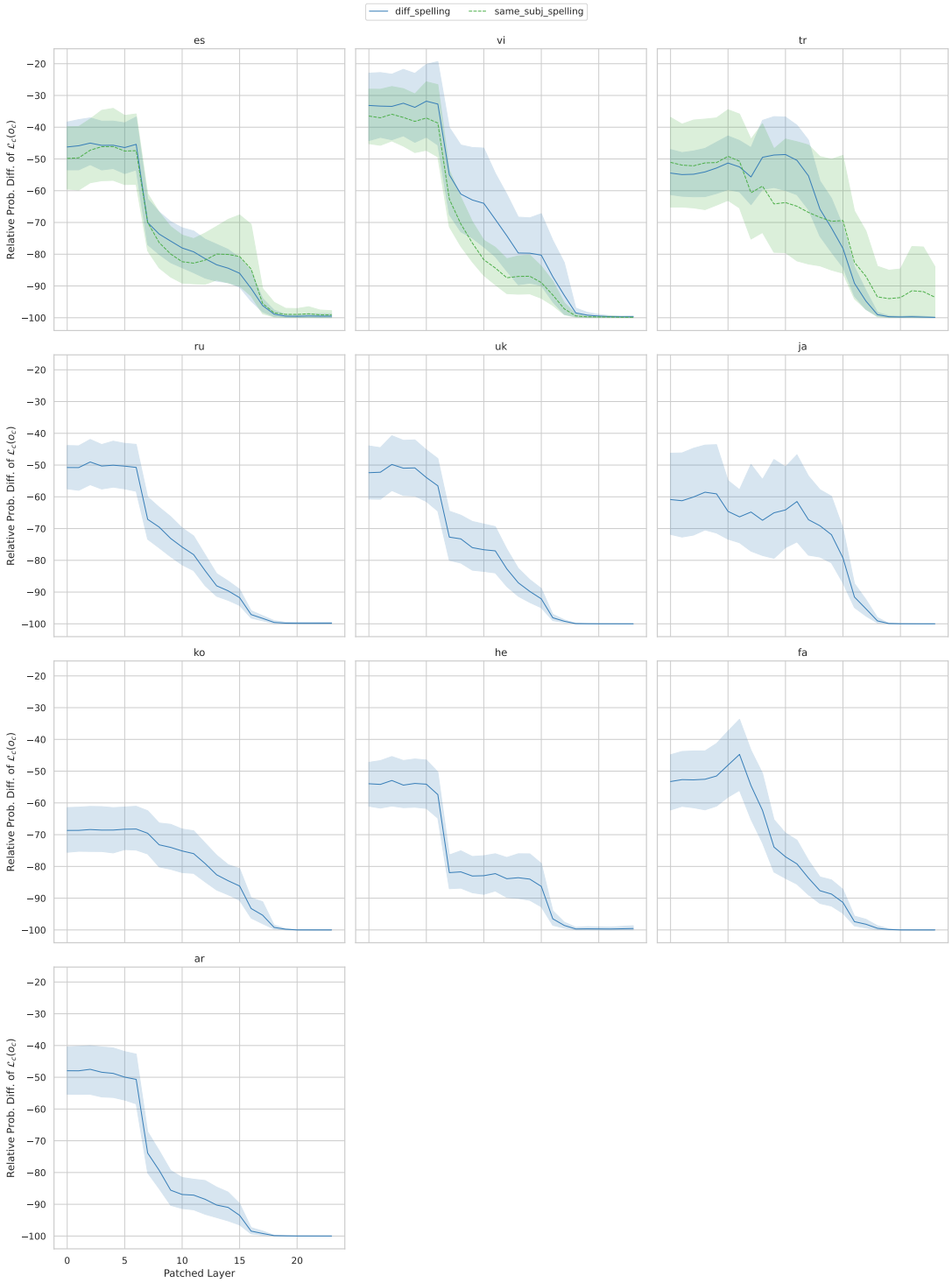


Figure 49: Probability of the $\mathcal{L}_c(o_c)$ when patching at different layers in mT5, for examples with $\{\neq \mathcal{L}, \neq r, = s\}$. Note that the plots do not share the y-axis.

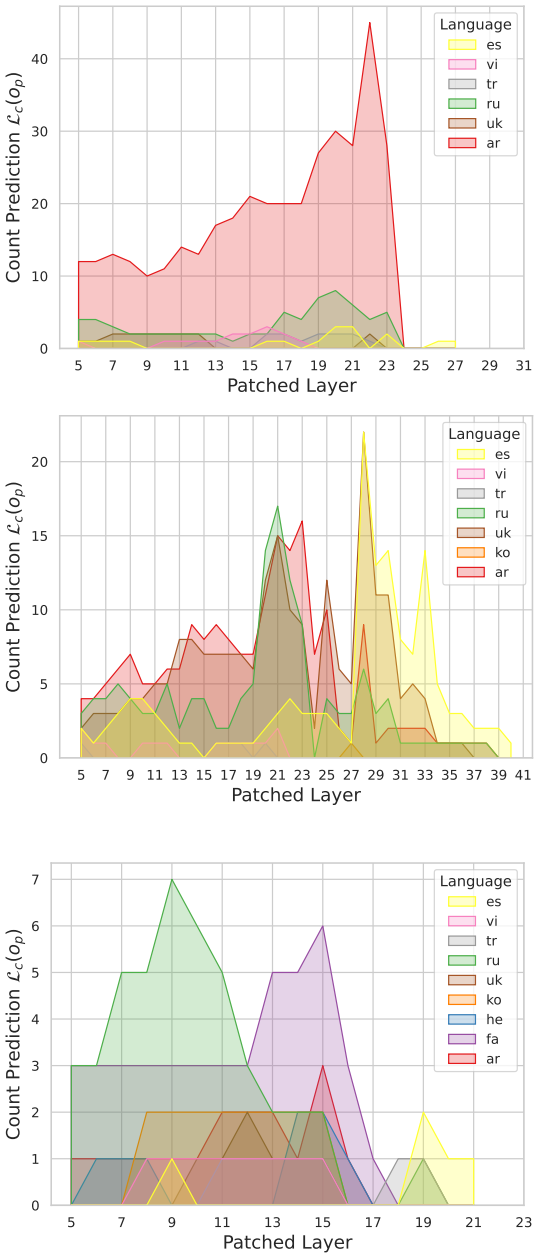


Figure 50: Number of times the patch object is predicted in the context language for the experiment of same relation different subject. Models from top to bottom: XGLM, EuroLLM, mT5.

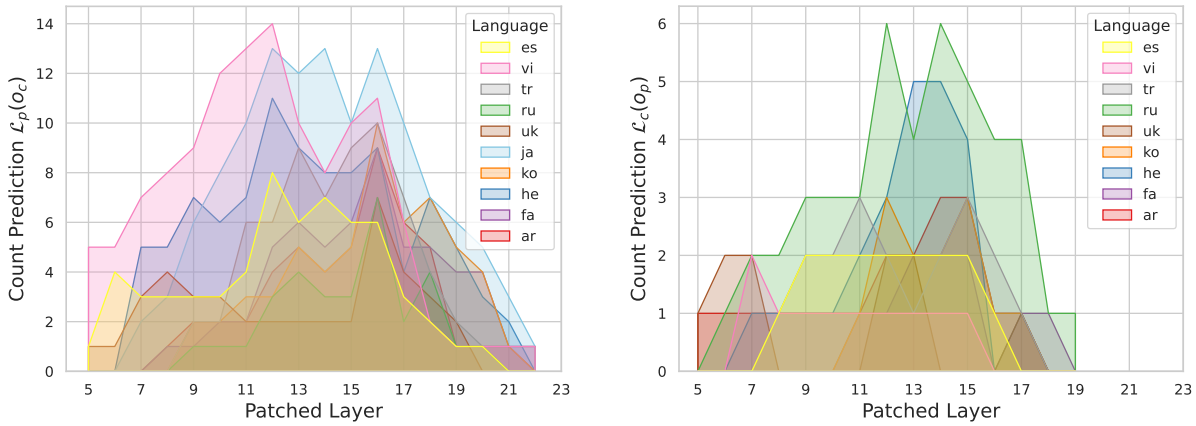


Figure 51: Number of times the object is predicted in the opposite language in mT5 in the $\{\neq r, = s, \neq \mathcal{L}\}$ experiment.

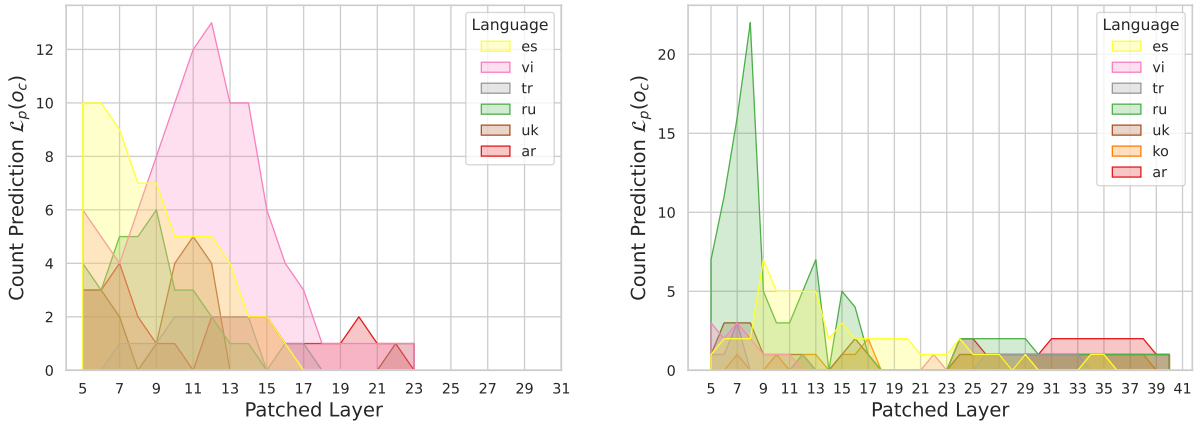


Figure 52: Number of times the context object is predicted in the patch language in the $\{\neq r, = s, \neq \mathcal{L}\}$ experiment. Left XGLM, right EUROLLM.