Language Arithmetics: Towards Systematic Language Neuron Identification and Manipulation

Anonymous ACL submission

Abstract

Large language models (LLMs) exhibit strong multilingual abilities, yet the neural mechanisms behind language-specific processing remain unclear. We analyze language-specific neurons in Llama-3.1-8B, Mistral-Nemo-12B, and Aya-Expanse-8B & 32B across **21 typologically diverse languages**, identifying neurons that control language behavior. Using the Language Activation Probability Entropy (LAPE) method, we show that these neurons cluster in deeper layers, with non-Latin scripts showing greater specialization. Related languages share overlapping neurons, reflecting internal representations of linguistic proximity.

Through **language arithmetics**, i.e. systematic activation addition and multiplication, we steer models to deactivate unwanted languages and activate desired ones, outperforming simpler replacement approaches. These interventions effectively guide behavior across five multilingual tasks: language forcing, translation, QA, comprehension, and NLI. Manipulation is more successful for high-resource languages, while typological similarity improves effectiveness. We also demonstrate that cross-lingual neuron steering enhances downstream performance and reveal internal **"fallback"** mechanisms for language selection when neurons are progressively deactivated.

1 Introduction

The emergence of large language models (LLMs) with impressive multilingual capabilities has raised fundamental questions about how these systems internally represent and process different languages (Wendler et al., 2024; Zhao et al., 2024). While models like Llama-3 (Grattafiori et al., 2024) and Gemma-3 (Team et al., 2024) perform well across dozens of languages despite limited multilingual training data, the neural mechanisms underlying this competence, such as the emergence of language-specific spaces, cross-lingual feature sharing, and internal language routing, are not fully understood. Understanding these mechanisms is crucial not only for advancing theoretical insights into multilingual representation learning, but also for building more controllable and interpretable language technologies (Amodei et al., 2016; Gabriel, 2020; Singh et al., 2024).

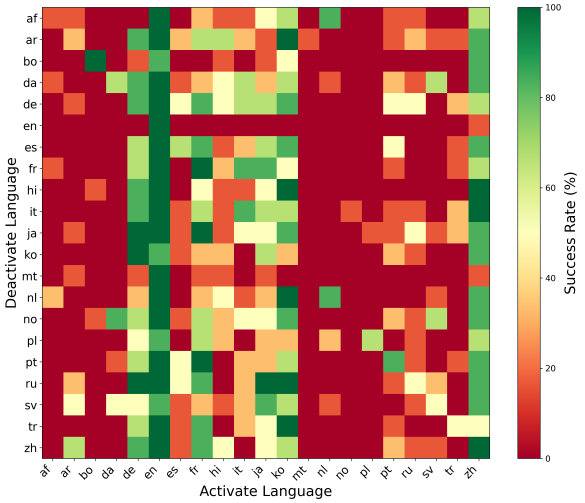


Figure 1: Success rates of language forcing when deactivating neurons for the input language and activating those of a target language for Llama-3.1-8B.

Recent studies have begun to explore how multilingual LLMs process language internally, showing that specific neurons may specialize in particular languages (Tang et al., 2024; Kojima et al., 2024; Zhao et al., 2024; Mondal et al., 2025). However, these works often focus on narrow language sets or specific architectures, leaving open questions about how language-specific processing scales across typologically diverse languages and how these insights can support model control and enhancement.

In this work, **we present the first large-scale investigation of language-specific neuron identification and manipulation** in the Llama-3.1-8B (Grattafiori et al., 2024), Mistral-Nemo-12B (AI, 2024), and Aya-Expanse-8B & 32B (Dang

et al., 2024) models across 21 typologically diverse languages. We examine the distribution of language-sensitive neurons, their cross-linguistic overlap, and the evolution of output language distributions across layers. Using the Language Activation Probability Entropy (LAPE) method (Tang et al., 2024), we identify neurons with strong language preferences and report several key findings. Unlike previously tested models such as Llama-2 (Touvron et al., 2023), BLOOM (Le Scao et al., 2023), and Phi-2 (Javaheripi et al., 2023), the models we study exhibit concentrated language-specific activity in their deeper layers. Analyzing output language distributions with logit lens (Wendler et al., 2024) across layers reveals that language generation predominantly occurs in the final layers, aligning with the concentration of language-sensitive neurons. Non-Latin script languages show both a higher number of specialized neurons and less cross-language overlap, while typologically related Germanic and Romance languages share more neurons–reflecting linguistic proximity and possibly orthographic similarities in their internal representations.

Building on these findings, we establish simple additive and multiplicative neuron intervention techniques, which we call **language arithmetics**[1], to test the causal role of language-specific neurons in controlling language use while preserving internal representations. Unlike prior approaches that rely on activation replacement (Tang et al., 2024; Kojima et al., 2024), our approach adds target-language patterns to hidden states, while simultaneously multiplying unwanted language patterns with 0. We demonstrate that language arithmetics outperforms both replacement-based methods (Tang et al., 2024; Kojima et al., 2024) and the widely used DiffMean (Marks and Tegmark, 2023) approach on a novel *language forcing* task, where models are expected to answer questions in a specific target language without being explicitly prompted, with especially strong results for languages better represented in pretraining. We further evaluate additive language arithmetic on four downstream tasks–machine translation, question answering, natural language inference, and machine comprehension–and show that targeted neuron activation improves performance without any task-specific fine-tuning, contrasting with the findings of Mondal et al. (2025). For example, our intervention improves translation scores by up to 10% and enables precise cross-lingual transfer. Additionally, we find that when dominant language signals are suppressed, models fall back to the next most probable language, suggesting the presence of internal **"fallback"** mechanisms for language selection.

Overall, our work (1) reveals new patterns of language-specific neuron specialization and layer-wise processing, (2) applies a minimally invasive steering technique to test their causal influence, and (3) demonstrates practical benefits for multilingual model control and performance.

2 Related Work

Recent research has focused on uncovering language-specific mechanisms in LLMs. Several works have proposed methods to identify and intervene on neurons specialized for particular languages.

Tang et al. (2024) introduce the Language Activation Probability Entropy (LAPE) method, demonstrating that certain neurons are critical to multilingual capacity and can be manipulated to control language behavior via activation (setting neurons to their average values) or deactivation (zeroing them out). Kojima et al. (2024), building on Cuadros et al. (2022), similarly identify neurons selectively active for one language and inactive for others, and show that setting these to their median activation can shift generation language. Tan et al. (2024) further propose a frequency-based method to rank FFN neurons by activation counts on language-specific inputs, and show that fine-tuning these neurons improves downstream tasks such as machine translation. Expanding beyond FFNs, Zhao et al. (2024) introduce Parallel Language-specific Neuron Detection (PLND), which identifies language neurons in both attention and FFN modules. They show that deactivating these neurons reduces performance in the corresponding language, and that fine-tuning them on limited language data enhances multilingual ability.

Meanwhile, Deng et al. (2025) highlight the problem of superposition–where neurons encode multiple concepts (Elhage et al., 2022)–and instead use sparse autoencoders (SAE) (Cunningham et al., 2023) to extract latent dimensions linked to language identity. Ablating these features degrades performance, and they use them to guide steering

[1]The naming follows conceptually similar works on task and prompt arithmetics (Ilharco et al., 2023; Belanec et al., 2024).

vector construction for better control. Following this, Chou et al. (2025) show that modifying a single SAE feature can steer generation language with high accuracy and semantic preservation, especially in mid-to-late layers. Similarly, Andrylie et al. (2025) introduce SAE-LAPE to identify interpretable language-specific features in the sparse weight space, enabling both language identification and control.

Our work extends this line of research in two dimensions: (1) We analyze the internal structure of LLMs without relying on auxiliary models such as SAEs, and (2) we look at a wider range of more diverse languages. We conduct a comprehensive study of language-specific neurons in LLaMA-3.1-8B (Grattafiori et al., 2024), Mistral-Nemo-12B (AI, 2024), and Aya-Expanse-8B & 32B (Dang et al., 2024) across 21 languages using the LAPE method (Tang et al., 2024), systematically investigating both neuron identification and manipulation to assess their effectiveness for multilingual control and downstream performance improvement.

3 Language Neuron Identification

3.1 Identification Method

Language Activation Probability Entropy (LAPE) (Tang et al., 2024) identifies language-specific neurons within LLMs by analyzing activation patterns across different languages in the FFN modules of a transformer-based language model. For each neuron, it measures how often that neuron activates when processing inputs in various languages. These activation frequencies are normalized into a probability distribution over languages, and entropy (Shannon, 2001) is used to quantify how language-specific each neuron's behavior is.

Intuitively, neurons with low LAPE values are considered language-specific since their activation probabilities are concentrated on one or two languages, showing minimal activity for others. We select the bottom K% (K={1..5}) of neurons by LAPE score after filtering out neurons with weak overall activity: neurons are excluded if none of their language activations exceed the 95th percentile of all activations (*filter rate*). To ensure neurons exhibit clear language preferences, they further get refined by verifying that their activation probability for a given language surpasses the 95th percentile (*activation threshold*). These thresholds help eliminate noisy or non-informative neurons and retain only those with high, selective activation characteristics.

3.2 Per-layer Output Language

We additionally analyze how language identity emerges across layers by computing three key statistics, using a method referred to as *logit lens* (Nostalgebraist, 2020), similar to Wendler et al. (2024). Specifically, we apply a FastText classifier (Joulin et al., 2016) to determine the language of the model's output at each layer. First, we track the probability of generating the correct target language, revealing the depth at which language-specific behavior becomes prominent on the output level. Second, we measure the probability assigned to English regardless of the input, assessing the extent of English interference across layers. Third, we compute the entropy of the output language distribution to quantify the diversity and confidence of the model's predictions.

3.3 Models and Data

We use the Llama-3.1-8B base model (Grattafiori et al., 2024) for all experiments. Notably, Llama-3.1 is a decoder-only model that was neither explicitly trained for multilingual tasks nor instruction-tuned. Its pre-training data includes approximately 5% non-English content spanning over 30 languages. We also include Mistral-Nemo (Mistral AI Team, 2024), a slightly larger 12-B-parameter base model, featuring strong multilingual, reasoning, and coding performance in its size class. Finally, we evaluate the Aya Expanse family (Dang et al., 2024): Aya-Expanse-8B, an open-weight, instruction-tuned multilingual model optimized via data arbitrage, preference training, and model-merging; and its larger sibling, Aya-Expanse-32B–both supporting 23 languages and offering state-of-the-art multilingual performance.

For neuron identification, we use the CulturaX corpus (Nguyen et al., 2023), a large-scale multilingual dataset comprising over 6.3 trillion tokens across 167 languages. The data is sourced from Common Crawl (Wenzek et al., 2019) and Wikipedia (Foundation) and has undergone extensive cleaning and language identification to ensure high quality. To compute language-specific activations and obtain LAPE values, we truncate the dataset for each language to 500MB due to efficiency reasons. For per-layer output experiments, we use 6 questions that are available in all languages, as described in Section 4.2.

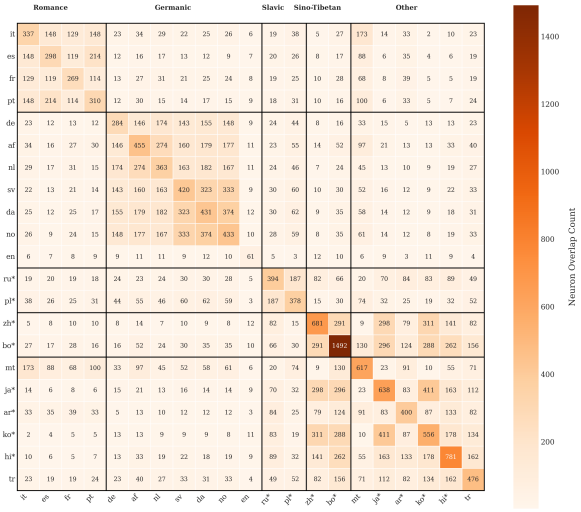


Figure 2: Neuron overlap between languages and language families in Llama-3.1, based on the top 1% of neurons identified as language-specific. Diagonals show counts per language; off-diagonals show overlaps. Asterisks mark non-Latin script languages.

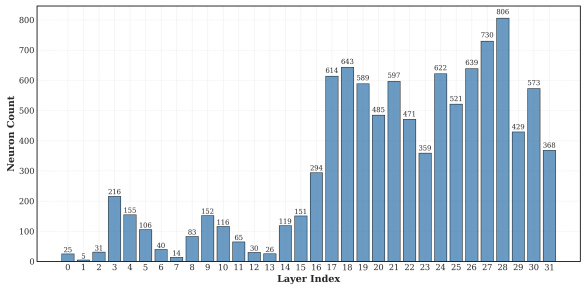


Figure 3: Layer-wise distribution of language-specific neurons in Llama-3.1 across 21 languages. Similar patterns for individual languages and other models are shown in Appendix E.

3.4 Results

Language Neuron Distribution: Figure 3 and Appendix E show total neuron activation distributions with minimal early-layer activity and pronounced peaks in later layers across all 21 languages. Interestingly, language-specific processing in Llama-3.1, as well as the other tested models, demonstrates a strong concentration in layers 17-28, peaking at layers 27-28. Earlier Layers 0-15 show relatively minimal language-specific activity, with only modest peaks around layers 3-4 and 8-9. This contrasts with previous findings by Tang et al. (2024), Kojima et al. (2024), and Zhao et al. (2024), who report both early- and late-layer language neuron specialization.

Individual language distribution analysis (see Appendices E and F) reveals typologically coherent clustering, with Germanic languages (German (de), Dutch (nl), Afrikaans (af), Danish (da), Norwegian (no), Swedish (sv)) showing similar language neuron distribution patterns concentrated around layers 24-26. Tibetan (bo) and Chinese (zh) (the only Sino-Tibetan languages we tested) demonstrate the most concentrated deep-layer activation, while Romance languages display more varied patterns. These patterns are observed across all tested models.

Language Neuron Overlap: The language neuron overlap matrices in Figure 2 and Appendices E and F show that language families exhibit internal cohesion with substantial neuron overlap, as also observed by Tan et al. (2024). In LLama-3.1, for instance, Germanic languages show particularly high overlap values (e.g., Swedish (sv) and Danish (da): 290 neurons representing 76.3% and 74.7% of their respective language-specific neurons; Dutch (nl) and Afrikaans (af): 250 neurons representing 76.5% and 59.0% respectively), possibly confirming that the similar activation patterns observed in Figure 13 reflect shared neurons rather than coincidental distributions.

Non-Latin script languages display notably higher neuron counts, with Chinese (zh*: 681), Tibetan (bo*: 1492), and Hindi (hi*: 781) having substantially more language-specific neurons than their Latin-script counterparts. Interestingly, these languages show minimal overlap with Latin-script languages, suggesting that orthographic complexity requires dedicated, non-transferable neural specialization. This is consistent across all models.

Per-layer Language Analysis: For more English-centric models (see Appendix G) such as Llama-3.1 and Mistral-Nemo, we observe that the target language tokens predominantly emerge in the final layers (25–30), aligning with the concentration of language-specific neurons in those layers. English tokens slightly begin to appear in the initial layers and persist through to the output. This pattern partly aligns with findings by Wendler et al. (2024) and Schut et al. (2025). Language entropy in these models increases across layers: the model appears confident in its language prediction in early layers, but entropy rises from around layer 20 onward, suggesting that the decision about which language to generate is made in the final layers.

In contrast, more multilingual models such as Aya-Expanse-8B and 32B exhibit English token

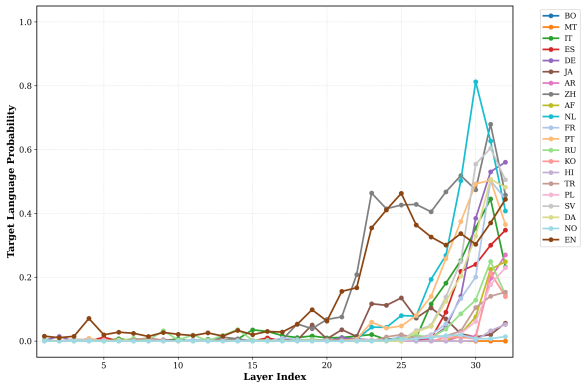


Figure 4: Predicted probabilities of target languages for Llama-3.1 using logit lens outputs from each layer and FastText for language identification. Results for other models are shown in Appendix G.

presence across nearly all layers, while the target language is also weakly activated throughout and peaks near the end. These models show an opposite entropy trend: entropy starts high, indicating early uncertainty about the generation language, and gradually decreases toward the final layers, where the model settles on a specific language.

4 Language Neuron Manipulation

4.1 Method

We adopt a manipulation approach inspired by Tang et al. (2024), but instead of **replacing neuron activations with fixed values**, as done in Tang et al. (2024) (using mean activations) and Kojima et al. (2024) (using median activations), we perform an **additive intervention**. This additive strategy is less destructive and allows the model to retain more of its original contextual dynamics while still shifting its behavior toward the target language.

Let $\mathbf{a} \in \mathbb{R}^{L \times D}$ be the post-activation tensor after the gated projection layer for a single sequence, where L is the sequence length, and D is the hidden size. Let $\mathcal{I} \subset \{1, \ldots, D\}$ be the set of neuron indices identified as language-specific. For each $i \in \mathcal{I}$, we define a boost value b_i corresponding to the average activation of neuron i when processing that language.

We apply the following update to the activation tensor:

$$\mathbf{a}_{:,i} \leftarrow \mathbf{a}_{:,i} + b_i \quad \text{for all } i \in \mathcal{I}$$

This additive manipulation shifts the activation patterns toward those typical of the desired language, while preserving the original contextual information encoded in the activations.

4.2 Tasks and Data

We evaluate our neuron selection and manipulation method across five multilingual tasks to test its effectiveness in both generative and classification settings. For each task, we experiment with five different fractions of language-specific neurons.

First, we design a novel controlled language-forcing task using six simple questions ("How are you today?", "What is your name?", "What year is it now?", "What is your favorite color?", "What is the weather like?", "Where are you from?") translated into 21 languages using Google Translate (Wu et al., 2016). For each input question, we deactivate the neurons associated with the source language by setting their activations to zero or a small negative value (see Appendix H) and activate the neurons associated with a desired target language. We then use FastText (Joulin et al., 2016) to identify the language of the generated output. This setup evaluates the model's ability to override the input language and generate responses in the specified target language.

We further evaluate our neuron manipulation method across four multilingual downstream tasks covering both generation and classification outputs. For generation, we use FLORES-200 (Costa-Jussà et al., 2022) for machine translation, activating only target-language neurons during decoding to test whether neuron-level steering can enforce correct-language outputs, and XQuAD (Artetxe et al., 2020) for extractive question answering. For classification, we use XNLI (Conneau et al., 2018b) for natural language inference and Belebele (Bandarkar et al., 2024) for multiple-choice reading comprehension. We vary the ratio of manipulated neurons to assess how language-specific activation impacts multilingual performance. All evaluations are performed in a zero-shot setting, with prompts and generation hyperparameters detailed in Appendix A.

Additionally, we investigate whether models exhibit a **"fallback"** mechanism in language selection during generation. To this end, we use a set of 70 English questions from Vicuna (Chiang et al., 2023) and progressively deactivate neurons associated with high-resource languages one by one.

4.3 Results

Language Forcing: For the language forcing task, we compare two neuron manipulation strategies: (1) deactivating neurons associated with the source

Model	Intervention	Strategy	Top 1%	Top 2%	Top 3%	Top 4%	Top 5%
Llama-3.1	Additive	Activate	12.81%	16.44%	19.43%	21.62%	22.07%
	Additive	Deactivate + Activate	19.88%	25.13%	27.25%	28.04%	28.12%
	Replacement	Activate	9.67%	11.56%	12.69%	12.74%	13.53%
	Replacement	Deactivate + Activate	13.68%	15.91%	16.09%	15.04%	16.21%
	DiffMean	Activate	9.71%	10.09%	10.54%	11.11%	11.0%
	DiffMean	Deactivate + Activate	13.98%	16.48%	17.49%	18.71%	19.05%

Table 1: Overall success rates (%) of language forcing for Llama-3.1-8B using three intervention types and two manipulation strategies across different top-k% neuron thresholds.

language and simultaneously activating neurons for the target language, and (2) only activating the neurons corresponding to the desired target language. Our results show that the first approach more reliably steers the model to generate output in the target language (Table 1 and Appendix H). We additionally compare our additive language arithmetic intervention with the simple replacement methods (Tang et al., 2024; Kojima et al., 2024) and the widely used DiffMean approach (Marks and Tegmark, 2023; Panickssery et al., 2023)[2] (see Table 1 and Appendix 8). We find that the additive method performs best, likely because it is less disruptive to the model's internal representations.

Figure 1 illustrates the effects of this manipulation across 21 typologically diverse languages in Llama-3.1-8B with the top 5% neuron ratio. The results indicate that higher-resourced languages such as Chinese (zh), Korean (ko), Japanese (ja), German (de), etc. are more susceptible to successful language forcing. In contrast, languages with likely lower representation in the pretraining corpus are more difficult to control, showing inconsistent output.

Below, we provide qualitative examples demonstrating successful language forcing, where the model generates output in a specified target language despite the input being in a different source language:

Forcing Japanese

Input: Q: Wie geht es dir heute? A:
Output: 今日は元気です。

Forcing German

Input: Q: Откуда ты? A:
Output: Ich komme aus Deutschland.

[2]For DiffMean, we construct the steering vector as the difference between the mean activation for the target language (positive class) and that of all 20 other languages (negative class).

Forcing Korean

Input: Q: Comment tu t'appelles? A:
Output: 저는 김민수입니다..

Downstream Tasks: We evaluate the impact of activating language-specific neurons during inference on four multilingual benchmarks: FLORES-200 and Belebele using all languages covered in our study, XNLI and XQuAD using all available languages. We conduct systematic evaluations on Llama-3.1 and Mistral-Nemo across five neuron ratios. Selected heatmap results are shown in Figures 5, 7, 8, and 6.

Overall, we observe modest improvements (up to 10–20%) or neutral effects for most language pairs when activating neurons from various languages, particularly within the same language family, in contrast to the negative transfer effects reported by Mondal et al. (2025). For instance, activating Romance or Germanic language neurons consistently benefits other languages from the same families across all tasks. Task type influences cross-linguistic effects: translation and question answering show more consistent positive transfer, while machine comprehension and natural language inference exhibit mixed results, including occasional negative transfer from distant languages. Activating the actual target language typically improves performance across all tasks and languages. Interestingly, some Romance and Germanic languages benefit even from activating Sino-Tibetan and Other language neurons–likely due to the larger number of neurons in those groups and their partial overlap with the other families at current activation ratios. Finally, Hindi and Urdu show disproportionate gains from nearly all language activations, possibly due to polysemantic neurons–units encoding multiple concepts (Elhage et al., 2022).

Language "Fallbacks": By progressively deactivating high-resource language neurons in Llama-3.1-8B (see Appendix I), we observe that

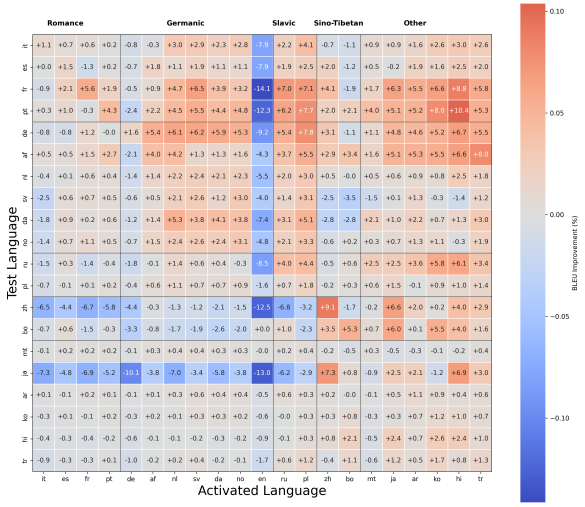


Figure 5: FLORES performance changes over the baseline (measured by BLEU score) when activating language-specific neurons for Mistral-Nemo (5%).

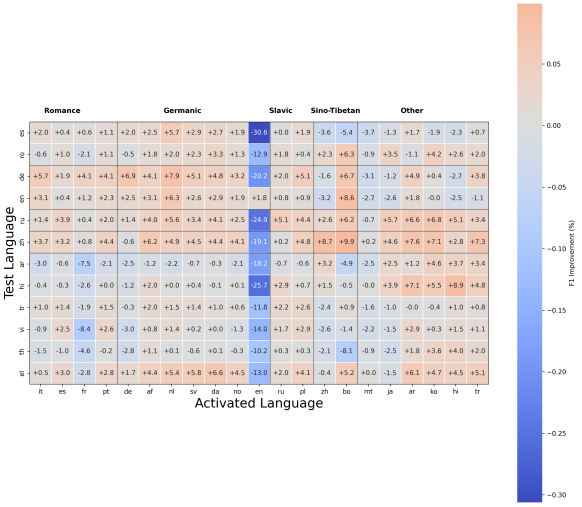


Figure 6: QXuAD performance changes over the baseline (measured by F1-score) when activating language-specific neurons for Mistral-Nemo (5%).

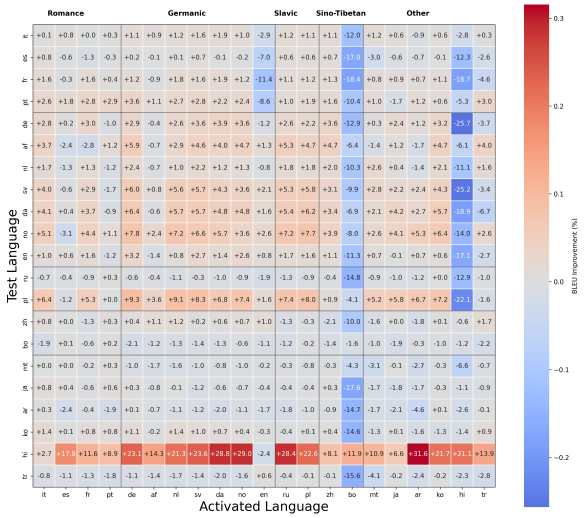


Figure 7: BELEBELE changes over the baseline (measured by accuracy score) when activating language-specific neurons for Mistral-Nemo (5%).

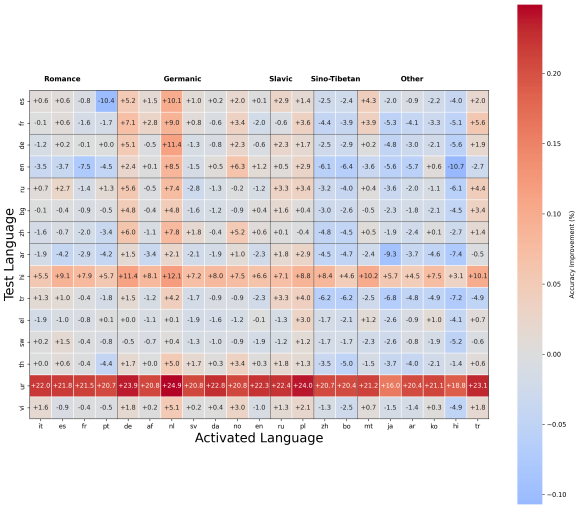


Figure 8: XNLI changes over the baseline (measured by accuracy score) when activating language-specific neurons for LLama-3.1 (4%).

the model exhibits internal "fallback" strategies for language generation. When English (en) neurons are deactivated, the model defaults to Spanish (es), French (fr), and Russian (ru) in a few cases. Deactivating English and French shifts responses primarily to Russian (ru), with some in Italian (it), Spanish (es), and Arabic (ar). Further removal of Latin-script languages–Spanish, Italian, and Portuguese (pt)–promotes generation in Russian, Arabic, and Chinese. Deactivating German, Russian, and Arabic leads to "fallback" responses in Thai, Vietnamese, and Chinese. Notably, some English, Spanish, and French responses persist despite deactivation, reflecting the model's strong priors for these languages. Similar patterns are observed for Mistral-Nemo, but with a slightly different language hierarchy. We leave a more detailed investigation of these dynamics to future work.

5 Discussion

5.1 Identification Insights

The layer-wise distribution of language-specific neurons in Llama-3.1, Mistral Nemo, and Aya-Expanse partially aligns with prior findings (Tang et al., 2024; Kojima et al., 2024; Zhao et al., 2024). We observe that these neurons are predominantly concentrated in the later layers, whereas Tang et al. (2024) and Kojima et al. (2024) report a dual-

peaked distribution in older models, with concentrations in both early and late layers. Notably, Tang et al. (2024) also show that the more multilingual BLOOM model (Le Scao et al., 2023) exhibits a similar late-layer concentration, a trend further supported by Mondal et al. (2025) with recent multilingual models. This divergence likely reflects architectural differences, or variations in training data composition. Interestingly, Chou et al. (2025) demonstrate that modifying a small number of SAE features from mid-to-late layers enables high-accuracy language steering for high-resource languages, while Andrylie et al. (2025) similarly find that their SAE-LAPE method identifies language-specific features concentrated in the same layers. These converging results, obtained through entirely different methodologies, reinforce our observation that language-specific representations are predominantly localized in the mid-to-late transformer layers of modern multilingual LLMs.

Layer-wise logit lens analysis complements these findings: in all tested models, English token probabilities emerge early on and persist, while target language tokens appear more sharply in the very final layers, which is in line with Wendler et al. (2024) and Schut et al. (2025). Entropy trends in more English-centric models rise toward the end, suggesting that language generation decisions are finalized in the later layers. In terms of cross-linguistic neuron overlap, we observe clear clustering patterns based on language families. This supports findings from Tan et al. (2024), who report that related languages share substantial neural components while maintaining distinct processing features.

5.2 Manipulation Insights

Our findings show that neuron-level interventions can reliably steer multilingual model behavior, with our additive language arithmetic method outperforming activation replacement (Tang et al., 2024; Kojima et al., 2024) and DiffMean-based approaches (Marks and Tegmark, 2023). Activating language-specific neurons improves performance across both generative and discriminative tasks, with the strongest gains observed when the activated language matches the target or is typologically related, yielding positive transfer within language families, but more limited benefits for structurally distant languages. Interestingly, Hindi and Urdu show disproportionate improvements across many interventions, potentially due to neurons encoding multiple overlapping functions (Elhage et al., 2022). Moreover, compared to Mondal et al. (2025), who report limited improvements from activation replacement or language neuron fine-tuning on tasks like XNLI (Conneau et al., 2018a) and XQuAD (Artetxe et al., 2019), our method achieves consistent gains due to three key differences. First, we evaluate across a broader and more typologically diverse set of languages. Second, our additive interventions using mean activation vectors preserve more of the original representational context. Third, all evaluations are conducted in a strict zero-shot setting, without fine-tuning on downstream tasks in any source language.

The effects are especially pronounced in our language forcing experiments. Deactivating neurons for the input language while simultaneously activating neurons for a desired target language significantly increases the likelihood of generating output in the target language–even when the input and output languages are typologically distant. These results suggest that target-language neurons contain sufficient signal to override source-language priors, especially for languages better represented in pre-training. In contrast, low-resource languages are harder to control, likely due to weaker or less distinct neural representations resulting from limited pretraining exposure. Importantly, manipulation also reveals internal "fallback" mechanisms: when dominant language neurons like English or French are deactivated, the model switches to secondary high-resource languages such as Russian or Arabic. This "fallback" hierarchy varies slightly across models but consistently exposes the model's internal language preferences. The persistence of certain languages despite deactivation–particularly English–highlights the robustness of high-priority language priors in LLMs.

6 Conclusion

Our large-scale study demonstrates that language-specific neurons encode meaningful linguistic features that can be identified, analyzed, and manipulated to steer model behavior across languages. By employing a novel additive intervention method, we show that activating these neurons allows for generation language control and improves performance on various multilingual tasks. These findings underscore the potential of neuron-level control as an interpretable mechanism for guiding multilingual model outputs.

Limitations

Our study has several limitations. First, while we demonstrate the effectiveness of neuron-level interventions across multiple tasks, a broader evaluation spanning more diverse downstream settings–such as dialogue, summarization, or code generation–is needed to assess the generality and robustness of our approach. Second, although our interventions often produce outputs in the desired language, the quality and fluency of these forced outputs across different languages–especially lower-resourced ones–remain underexplored and require more rigorous evaluation. Third, our current analysis primarily focuses on isolated neurons with high language specificity. This overlooks the possibility that language representations may be distributed across larger sub-networks or circuits; future work should investigate the interactions and dependencies between such neurons to better understand the network-level mechanisms supporting multilingual processing.

Additionally, due to computational constraints, not all experiments were run across every model in our study. In particular, some analyses (large-scale downstream evaluation) were limited to smaller models–LLama-3.1-8B and Mistral-Nemo-14B. Expanding these evaluations to include larger models could reveal further insights but requires significantly more resources.

References

Mistral AI. 2024. Mistral-nemo-base-2407. https://www.mistral.ai/news/mistral-nemo/. Base language model released in collaboration with NVIDIA, without instruction tuning or RLHF.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. 2016. Concrete problems in ai safety. *Preprint*, arXiv:1606.06565.

Lyzander Marciano Andrylie, Inaya Rahmanisa, Mahardika Krisna Ihsani, Alfan Farizki Wicaksono, Haryo Akbarianto Wibowo, and Alham Fikri Aji. 2025. Sparse autoencoders can capture language-specific concepts across diverse languages. *arXiv preprint arXiv:2507.11230*.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2019. On the cross-lingual transferability of monolingual representations. *arXiv preprint arXiv:1910.11856*.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Lucas Bandarkar, Davis Liang, Benjamin Muller, Mikel Artetxe, Satya Narayan Shukla, Donald Husa, Naman Goyal, Abhinandan Krishnan, Luke Zettlemoyer, and Madian Khabsa. 2024. The belebele benchmark: a parallel reading comprehension dataset in 122 language variants. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 749–775, Bangkok, Thailand. Association for Computational Linguistics.

Robert Belanec, Simon Ostermann, Ivan Srba, and Maria Bielikova. 2024. Task prompt vectors: Effective initialization through multi-task soft-prompt transfer. *arXiv preprint arXiv:2408.01119*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Cheng-Ting Chou, George Liu, Jessica Sun, Cole Blondin, Kevin Zhu, Vasu Sharma, and Sean O'Brien. 2025. Causal language control in multilingual transformers via sparse feature steering. In *ACL 2025 Student Research Workshop*.

Alexis Conneau, Guillaume Lample, Ruty Rinott, Adina Williams, Samuel R Bowman, Holger Schwenk, and Veselin Stoyanov. 2018a. Xnli: Evaluating cross-lingual sentence representations. *arXiv preprint arXiv:1809.05053*.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018b. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Marta R Costa-Jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, and 1 others. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Xavier Suau Cuadros, Luca Zappella, and Nicholas Apostoloff. 2022. Self-conditioning pre-trained language models. In *International Conference on Machine Learning*, pages 4455–4473. PMLR.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. 2023. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*.

John Dang, Shivalika Singh, Daniel D'souza, Arash Ahmadian, Alejandro Salamanca, Madeline Smith, Aidan Peppin, Sungjin Hong, Manoj Govindassamy, Terrence Zhao, Sandra Kublik, Meor Amer, Viraat Aryabumi, Jon Ander Campos, Yi-Chern Tan, Tom Kocmi, Florian Strub, Nathan Grinsztajn, Yannis Flet-Berliac, and 26 others. 2024. Aya expanse: Combining research breakthroughs for a new multilingual frontier. *Preprint*, arXiv:2412.04261.

Boyi Deng, Yu Wan, Yidan Zhang, Baosong Yang, and Fuli Feng. 2025. Unveiling language-specific features in large language models via sparse autoencoders. *Preprint*, arXiv:2505.05111.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, and 1 others. 2022. Toy models of superposition. *arXiv preprint arXiv:2209.10652*.

Wikimedia Foundation. Wikimedia downloads.

Iason Gabriel. 2020. Artificial intelligence, values, and alignment. *Minds and Machines*, 30(3):411–437.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2023. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*.

Mojan Javaheripi, Sébastien Bubeck, Marah Abdin, Jyoti Aneja, Sebastien Bubeck, Caio César Teodoro Mendes, Weizhu Chen, Allie Del Giorno, Ronen Eldan, Sivakanth Gopi, and 1 others. 2023. Phi-2: The surprising power of small language models. *Microsoft Research Blog*, 1(3):3.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Takeshi Kojima, Itsuki Okimura, Yusuke Iwasawa, Hitomi Yanaka, and Yutaka Matsuo. 2024. On the multilingual ability of decoder-based pre-trained language models: Finding and controlling language-specific neurons. *arXiv preprint arXiv:2404.02431*.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, and 1 others. 2023. Bloom: A 176b-parameter open-access multilingual language model.

Samuel Marks and Max Tegmark. 2023. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. *arXiv preprint arXiv:2310.06824*.

Mistral AI Team. 2024. Mistral nemo. Technical report, Mistral AI. Accessed: 2025.

Soumen Kumar Mondal, Sayambhu Sen, Abhishek Singhania, and Preethi Jyothi. 2025. Language-specific neurons do not facilitate cross-lingual transfer. In *The Sixth Workshop on Insights from Negative Results in NLP*, pages 46–62, Albuquerque, New Mexico. Association for Computational Linguistics.

Thuat Nguyen, Chien Van Nguyen, Viet Dac Lai, Hieu Man, Nghia Trung Ngo, Franck Dernoncourt, Ryan A. Rossi, and Thien Huu Nguyen. 2023. Culturax: A cleaned, enormous, and multilingual dataset for large language models in 167 languages. *Preprint*, arXiv:2309.09400.

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

Nina Panickssery, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. 2023. Steering llama 2 via contrastive activation addition. *arXiv preprint arXiv:2312.06681*.

Lisa Schut, Yarin Gal, and Sebastian Farquhar. 2025. Do multilingual llms think in english? *Preprint*, arXiv:2502.15603.

Claude Elwood Shannon. 2001. A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review*, 5(1):3–55.

Chandan Singh, Jeevana Priya Inala, Michel Galley, Rich Caruana, and Jianfeng Gao. 2024. Rethinking interpretability in the era of large language models. *Preprint*, arXiv:2402.01761.

Shaomu Tan, Di Wu, and Christof Monz. 2024. Neuron specialization: Leveraging intrinsic task modularity for multilingual machine translation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6506–6527, Miami, Florida, USA. Association for Computational Linguistics.

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, and 1 others. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

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.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2019. Ccnet: Extracting high quality monolingual datasets from web crawl data. *arXiv preprint arXiv:1911.00359*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, and 12 others. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *Preprint*, arXiv:1609.08144.

Yiran Zhao, Wenxuan Zhang, Guizhen Chen, Kenji Kawaguchi, and Lidong Bing. 2024. How do large language models handle multilingualism? *Preprint*, arXiv:2402.18815.

Appendix

A Prompt Templates

A.1 FLORES-200 Machine Translation

For the machine translation task, we use the following prompt format to instruct the model to translate from a source language into a target language:

```
Translate this {source_name} sentence into {target_name}: {source_text}. Translation:
```

Here, `{source_name}` and `{target_name}` are replaced by the names of the source and target languages (e.g., `French`, `Hindi`), and `{source_text}` is the input sentence.

A.2 XQuAD Question Answering

For extractive question answering, we prompt the model to find an answer span in a given context:

```
Answer the question based on the {language_name} context provided. Extract the exact answer from the context.

Context: {context}

Question: {question}

Answer:
```

Here, `{language_name}` is the name of the language (e.g., `Hindi`), and `{context}`, `{question}` are the inputs.

A.3 Belebele Machine Comprehension

For the multiple-choice machine comprehension task, we use:

```
Read the following {language_name} passage and answer the question.

Passage: {passage}

Question: {question}

A. {option_A}

B. {option_B}

C. {option_C}

D. {option_D}

Answer:
```

Each `{option_X}` is a possible answer choice, and the model is expected to return one of `A`, `B`, `C`, or `D`.

A.4 XNLI Natural Language Inference

To classify the relationship between a premise and hypothesis, we use:

```
Given the following {language_name} premise and hypothesis, determine the relationship between them.

Premise: {premise}

Hypothesis: {hypothesis}

Options:

1. Entailment

2. Neutral

3. Contradiction

Answer:
```

The model is expected to return one of the listed option numbers or labels.

A.5 Generation Settings

We use the same decoding configuration across all tasks unless specified otherwise. The generation is performed using the following sampling parameters:

```
SamplingParams( temperature=0, repetition_penalty=1.1, stop_token_ids=[eos_token_id] if eos_token_id is not None else [], skip_special_tokens=True )
```

We vary the `max_tokens` parameter depending on the task:

- **FLORES-200 (Machine Translation):** `max_tokens = 128`
- **XQUAD (Question Answering):** `max_tokens = 64`
- **XNLI (Natural Language Inference):** `max_tokens = 32`
- **BELEBELE (Machine Comprehension):** `max_tokens = 32`
- **Language Forcing Experiments:** `max_tokens = 256`

These settings ensure deterministic generation (due to `temperature=0`) while reducing repetition and enabling flexible truncation across task types.

B Selected Languages

Language Code	Language Name
bo	Tibetan
mt	Maltese
it	Italian
es	Spanish
de	German
ja	Japanese
ar	Arabic
zh	Chinese
af	Afrikaans
nl	Dutch
fr	French
pt	Portuguese
ru	Russian
ko	Korean
hi	Hindi
tr	Turkish
pl	Polish
sv	Swedish
da	Danish
no	Norwegian
en	English

Table 2: Language codes and corresponding full names

C Neuron Counts

Lang.	1%	2%	3%	4%	5%
bo	1492	2687	3723	4708	5687
mt	617	1578	2687	3853	5043
it	337	761	1198	1653	2104
es	298	667	1007	1358	1717
de	284	651	1065	1509	1965
ja	638	1084	1503	1887	2286
ar	400	708	1093	1472	1913
zh	681	1150	1564	1929	2286
af	455	1174	1966	2852	3803
nl	363	901	1432	2040	2680
fr	269	615	988	1376	1786
pt	310	698	1072	1449	1844
ru	394	717	985	1248	1546
ko	556	830	1054	1264	1484
hi	781	1456	2071	2644	3199
tr	476	1031	1681	2307	2916
pl	378	900	1455	2065	2744
sv	420	990	1614	2358	3089
da	431	1055	1771	2557	3344
no	433	1041	1743	2499	3268
en	61	88	116	140	151

Table 3: Number of language-specific neurons at different top-k thresholds in **Llama-3.1**.

Lang.	1%	2%	3%	4%	5%
bo	1153	1941	2547	3059	3554
mt	250	717	1283	1913	2552
it	155	343	539	766	969
es	154	343	528	730	907
de	130	291	460	656	859
ja	384	770	1113	1446	1808
ar	275	499	703	912	1131
zh	346	717	1076	1380	1723
af	219	567	1012	1549	2115
nl	184	458	791	1207	1646
fr	156	323	475	650	812
pt	175	388	589	799	1001
ru	213	387	520	653	764
ko	335	623	852	1079	1299
hi	310	619	924	1193	1443
tr	213	586	1027	1510	1982
pl	186	436	752	1106	1472
sv	160	451	807	1211	1637
da	171	486	886	1335	1812
no	148	464	836	1237	1643
en	32	55	78	100	117

Table 4: Number of language-specific neurons at different top-k thresholds in **Mistral-Nemo-Base-2407**.

Lang.	1%	2%	3%	4%	5%
bo	1636	3343	4947	6551	8085
mt	433	1169	2108	3079	4150
it	231	591	981	1393	1850
es	277	621	1016	1388	1745
de	211	509	874	1245	1674
ja	636	1171	1580	1983	2393
ar	383	662	962	1251	1566
zh	702	1348	1933	2512	3030
af	296	781	1440	2181	2925
nl	181	485	859	1243	1633
fr	211	489	842	1229	1636
pt	280	621	1007	1408	1811
ru	381	708	1052	1356	1671
ko	597	1051	1405	1763	2110
hi	459	880	1269	1668	2063
tr	322	734	1126	1512	1927
pl	260	633	1003	1364	1746
sv	282	837	1595	2437	3331
da	303	886	1688	2628	3569
no	290	866	1672	2576	3496
en	55	80	115	141	169

Table 5: Number of language-specific neurons at different top-k thresholds in **Aya-Expanse-8B**.

Lang.	1%	2%	3%	4%	5%
bo	1107	2241	3314	4403	5541
mt	568	1431	2403	3427	4461
it	353	758	1192	1629	2076
es	344	711	1100	1483	1805
de	240	631	1105	1557	1986
ja	676	1249	1703	2152	2563
ar	361	643	902	1163	1436
zh	656	1306	1920	2507	3032
af	364	978	1672	2368	3021
nl	235	607	990	1371	1691
fr	280	593	943	1332	1717
pt	362	709	1134	1548	1940
ru	355	757	1138	1514	1860
ko	549	1000	1383	1746	2091
hi	498	956	1296	1604	1899
tr	331	728	1138	1524	1902
pl	276	656	1036	1467	1823
sv	322	897	1585	2273	2925
da	343	980	1739	2488	3162
no	329	968	1705	2448	3102
en	51	85	122	148	171

Table 6: Number of language-specific neurons at different top-k thresholds in **Aya-Expanse-32B**.

D Downstream Baseline Results

Model	af	ar	bo	da	de	es	fr	hi	it	ja	ko	mt	nl	no	pl	pt	ru	sv	tr	zh
Llama-3.1	0.227	0.049	0.172	0.233	0.176	0.154	0.276	0.113	0.155	0.024	0.040	0.086	0.136	0.163	0.100	0.311	0.166	0.217	0.096	0.102
Mistral-Nemo	0.114	0.022	0.176	0.109	0.140	0.144	0.215	0.023	0.123	0.140	0.022	0.030	0.088	0.076	0.039	0.190	0.104	0.069	0.041	0.136

(a) Machine translation (BLEU scores).

Model	af	ar	bo	da	de	en	es	fr	hi	it	ja	ko	mt	nl	no	pl	pt	ru	sv	tr	zh
Llama-3.1	0.690	0.760	0.316	0.742	0.764	0.847	0.751	0.764	0.670	0.766	0.729	0.767	0.542	0.750	0.744	0.714	0.777	0.799	0.787	0.727	0.779
Mistral-Nemo	0.689	0.751	0.284	0.706	0.792	0.841	0.823	0.790	0.357	0.806	0.749	0.798	0.408	0.733	0.697	0.659	0.783	0.840	0.714	0.734	0.808

(b) Machine comprehension (accuracy).

Model	ar	bg	de	el	en	es	fr	hi	ru	sw	th	tr	ur	vi	zh
Llama-3.1	0.382	0.361	0.365	0.356	0.429	0.358	0.377	0.289	0.375	0.350	0.386	0.397	0.130	0.358	0.379
Mistral-Nemo	0.332	0.333	0.335	0.335	0.338	0.333	0.349	0.331	0.335	0.332	0.334	0.335	0.327	0.334	0.334

(c) Natural language inference (accuracy).

Model	ar	de	el	en	es	hi	ro	ru	th	tr	vi	zh
Llama-3.1	0.410	0.462	0.422	0.616	0.515	0.460	0.542	0.416	0.459	0.511	0.606	0.355
Mistral-Nemo	0.281	0.443	0.258	0.585	0.556	0.387	0.451	0.362	0.379	0.452	0.444	0.276

(d) Question answering (F1 score).

Table 7: Baseline multilingual evaluation results for translation, comprehension, NLI, and QA.

E Language Neuron Distributions

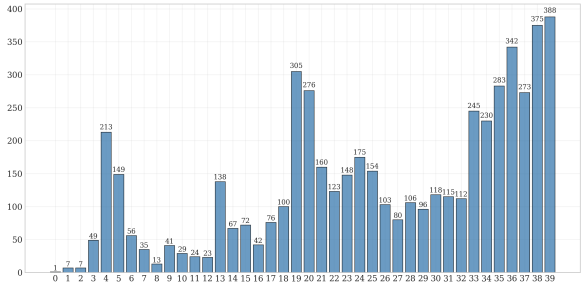


Figure 9: Distribution of identified language-specific neurons across **Mistral-Nemo** layers for all 21 evaluated languages. The neuron distributions for individual languages are further in the Appendix.

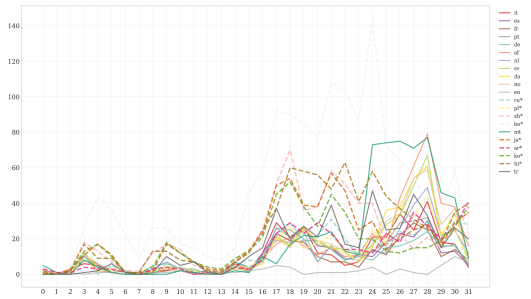


Figure 10: Distribution of **individual** language-specific neurons across **Llama-3.1** layers for all 21 languages.

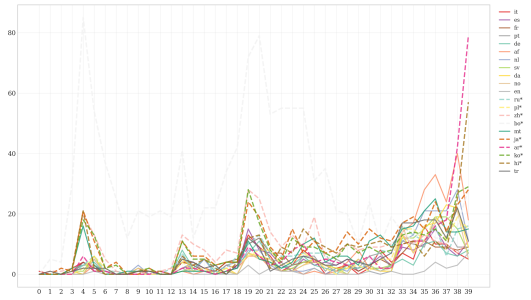


Figure 11: Distribution of **individual** language-specific neurons across **Mistral-Nemo** layers for all 21 languages.

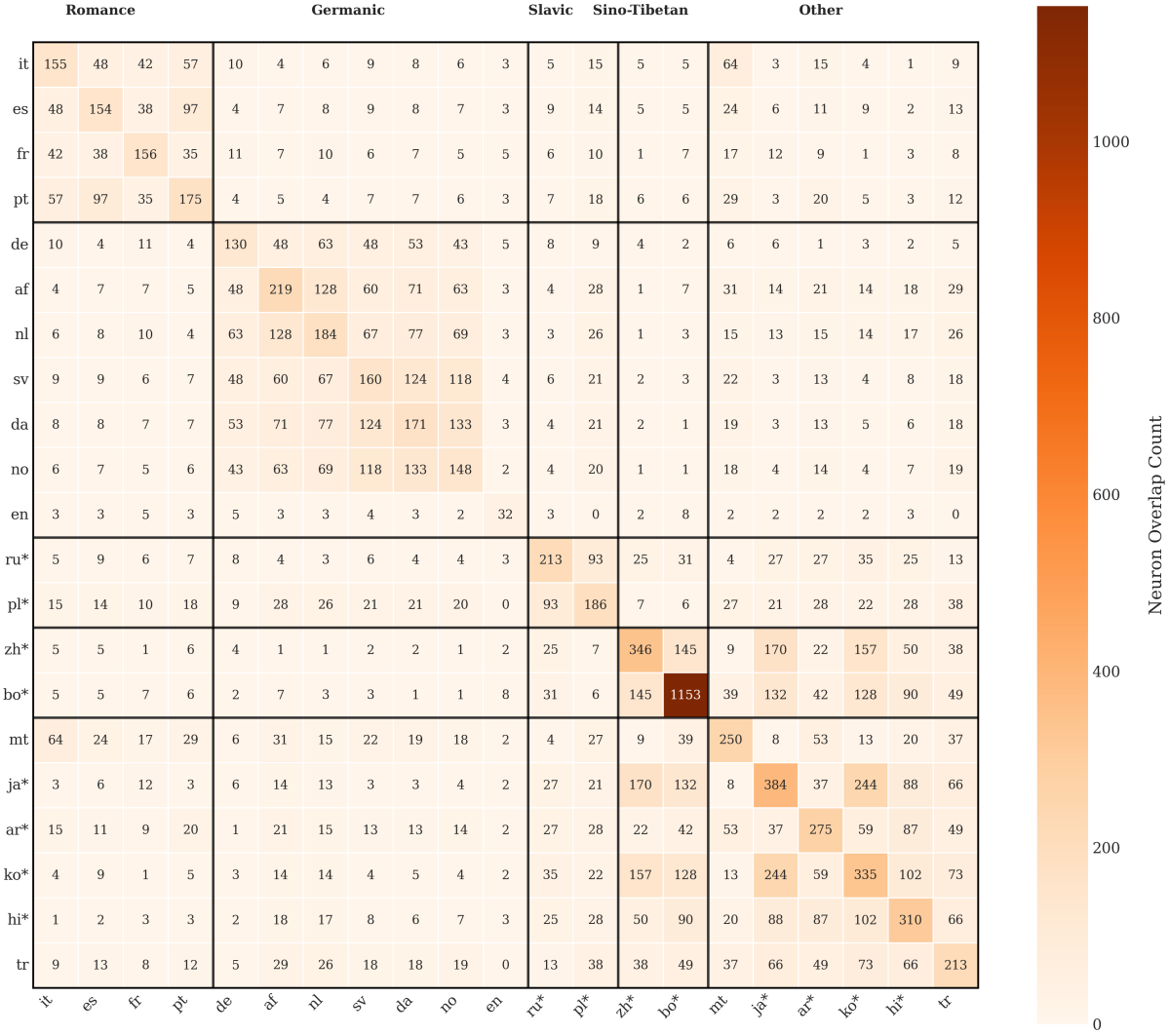


Figure 12: Overlap of language-specific neurons between individual languages and language families in Mistral-Nemo when considering top 1% of neurons as potentially language-specific. Diagonal values indicate the number of language-specific neurons for each language; off-diagonal values indicate the number of overlapping neurons. Asterisks denote languages with non-Latin scripts.

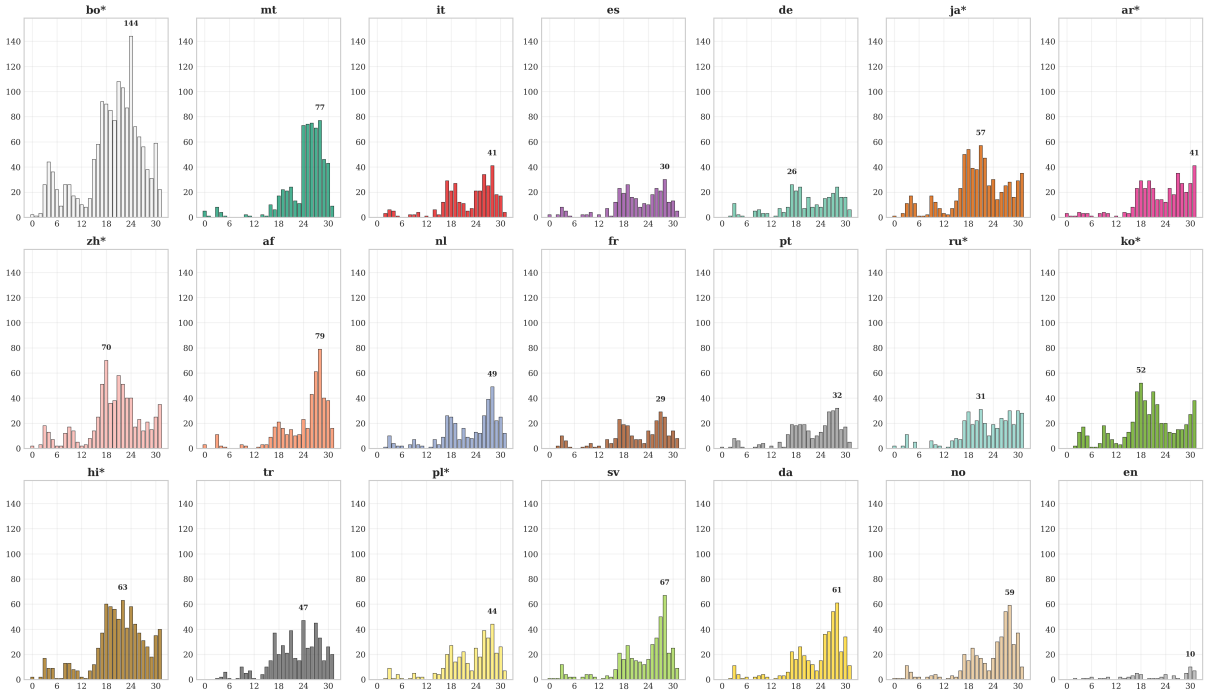


Figure 13: Distribution of **individual** language-specific neurons across **Llama-3.1-8B** layers for all 21 languages.

Figure 14: Distribution of **individual** language-specific neurons across **Mistral-Nemo** layers for all 21 languages.

F Additional Results for Aya Models

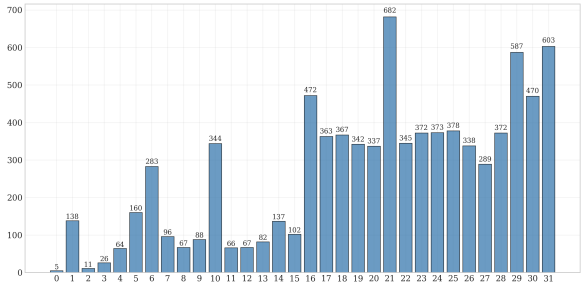


Figure 15: Distribution of identified language-specific neurons across **Aya-Expanse-8B** layers for all 21 evaluated languages.

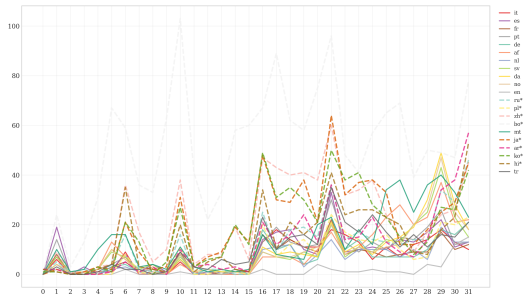


Figure 16: Distribution of **individual** language-specific neurons across **Aya-Expanse-8B** layers for all 21 languages.

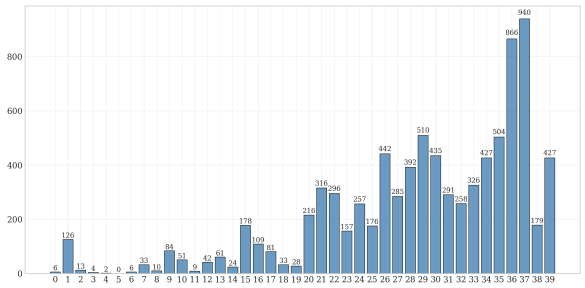


Figure 17: Distribution of identified language-specific neurons across **Aya-Expanse-32B** layers for all 21 evaluated languages. The neuron distributions for individual languages are further in the Appendix.

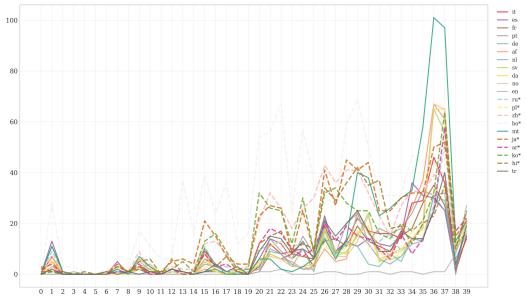


Figure 18: Distribution of **individual** language-specific neurons across **Aya-Expanse-32B** layers for all 21 languages.

Figure 19: Distribution of **individual** language-specific neurons across **Aya-Expanse–8B** layers for all 21 languages.

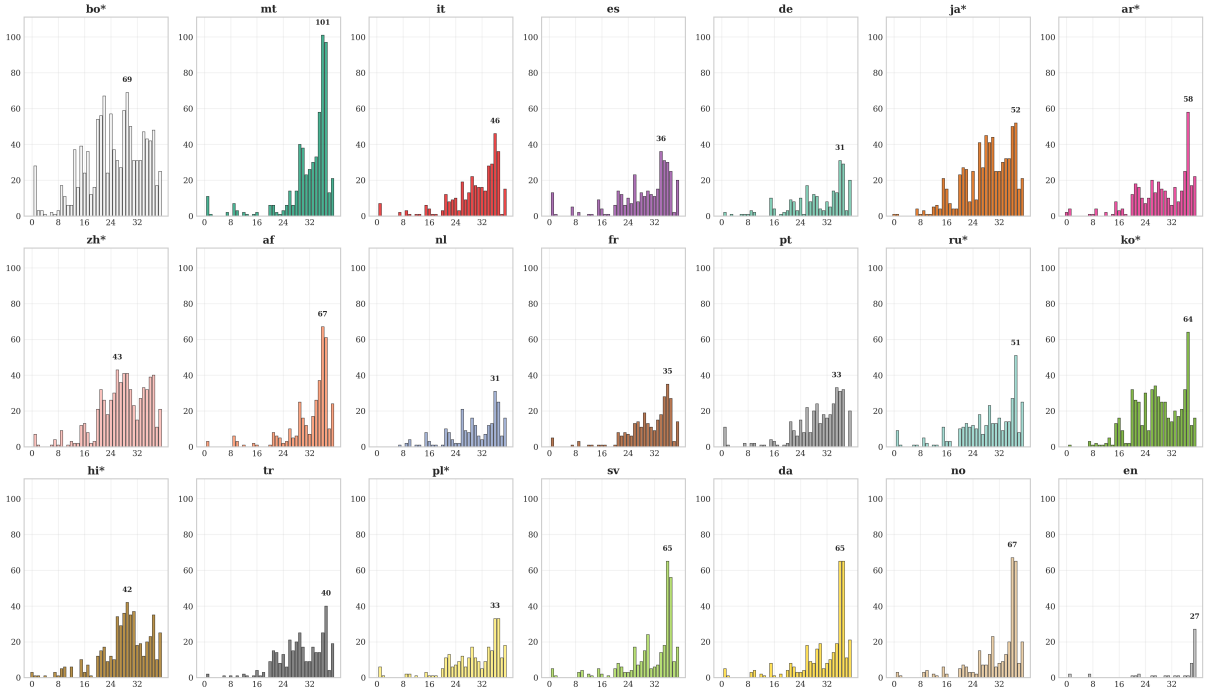


Figure 20: Distribution of **individual** language-specific neurons across **Aya-Expanse-32B** layers for all 21 languages.

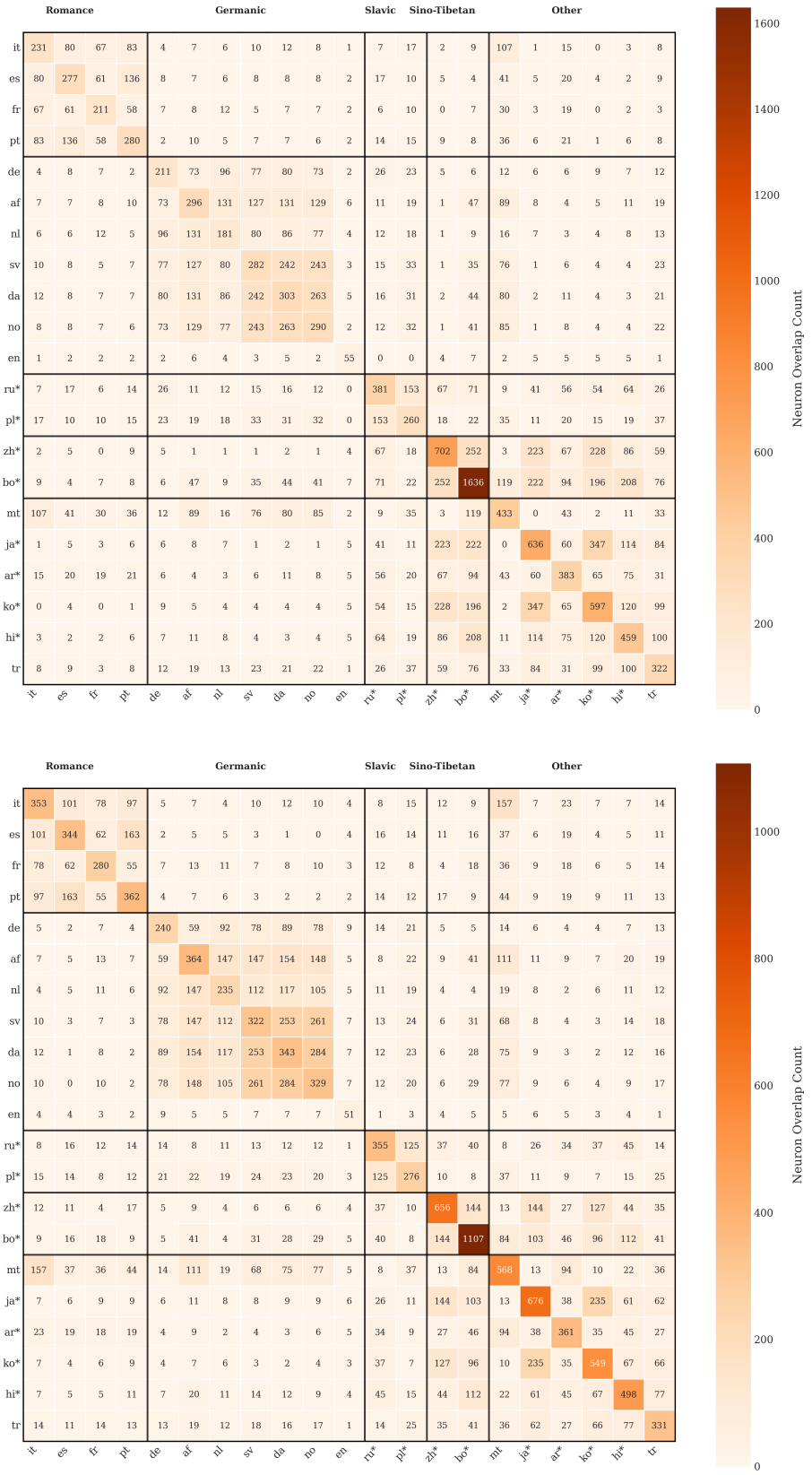
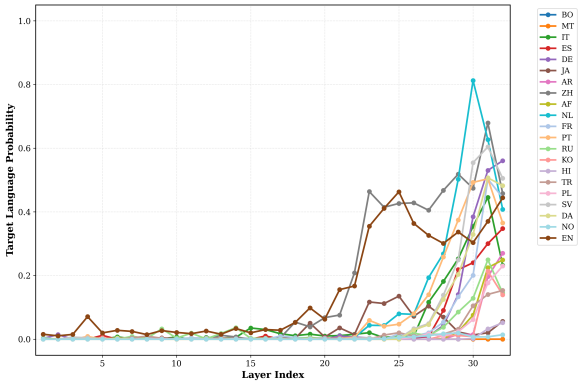
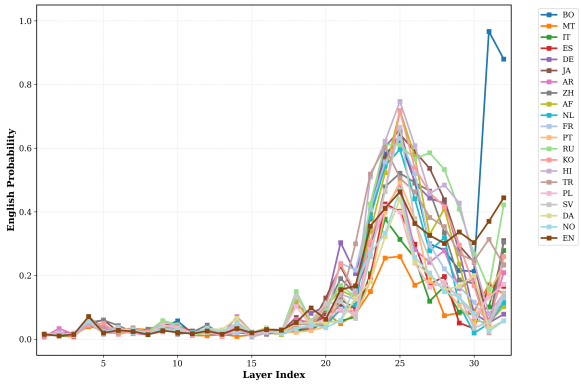


Figure 21: Overlap of language-specific neurons between individual languages and language families in Aya-Expanse-8B and 32B when considering top 1% of neurons as potentially language-specific. Diagonal values indicate the number of language-specific neurons for each language; off-diagonal values indicate the number of overlapping neurons. Asterisks denote languages with non-Latin scripts.

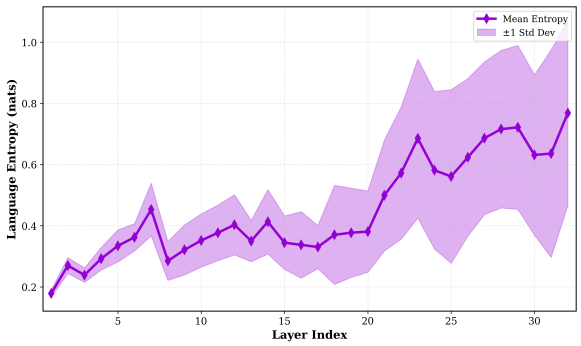
G Output Analysis with Logit Lens



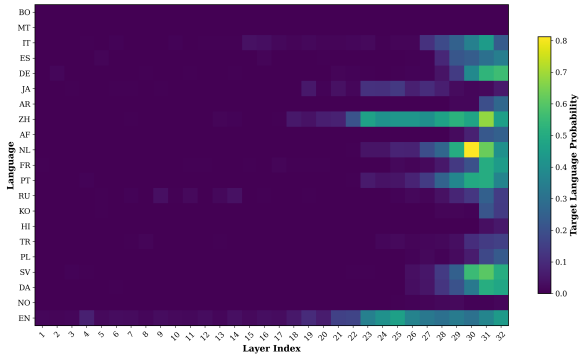
(a) Probabilities of the target languages.



(b) Probabilities of English.

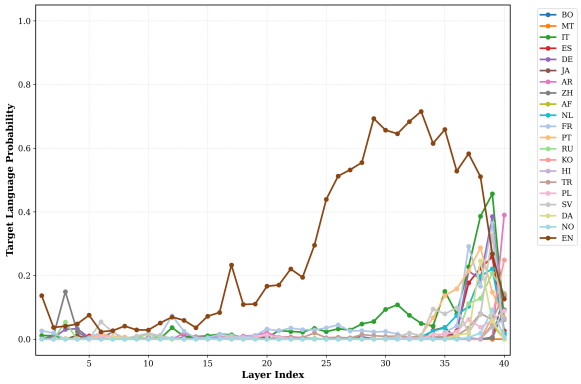


(c) Evolution of target language probability.

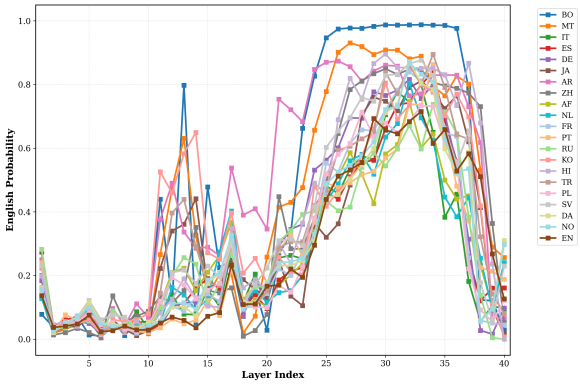


(d) Heatmap of English probabilities.

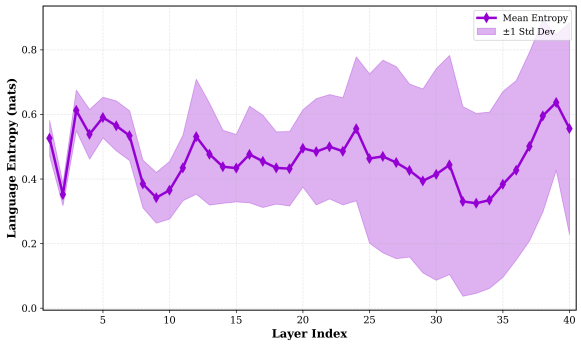
Figure 22: Peeking into language prediction across layers using the logit lens for **Llama-3.1-8B**.



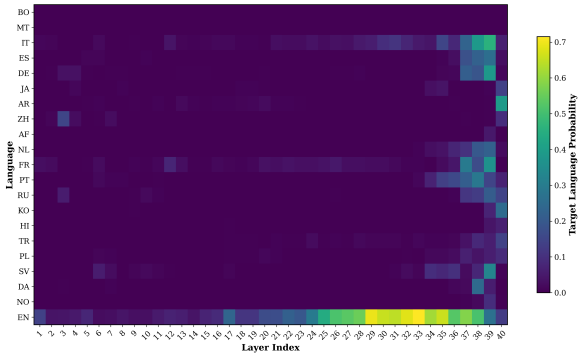
(a) Probabilities of the target languages.



(b) Probabilities of English.

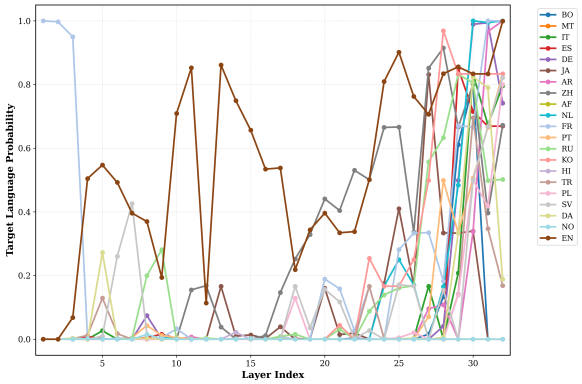


(c) Evolution of target language probability.

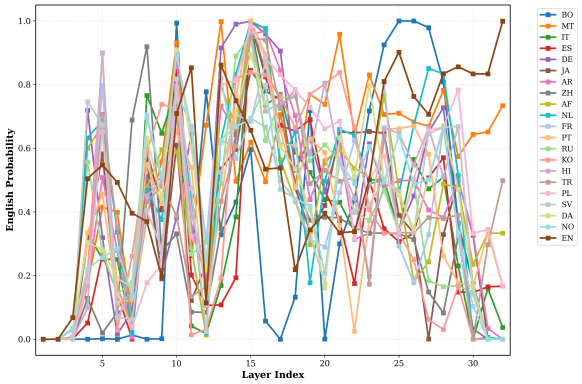


(d) Heatmap of English probabilities.

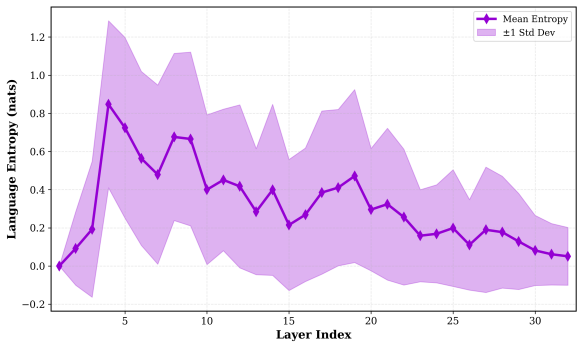
Figure 23: Peeking into language prediction across layers using the logit lens for **Mistral-Nemo**.



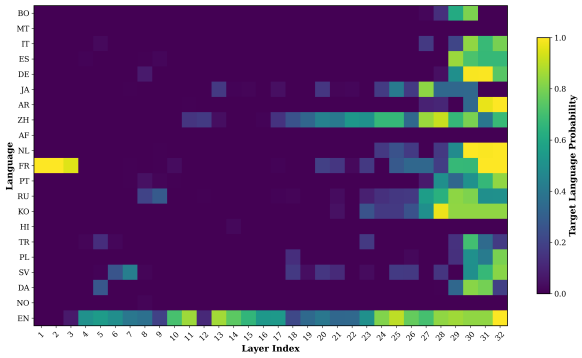
(a) Probabilities of the target languages.



(b) Probabilities of English.

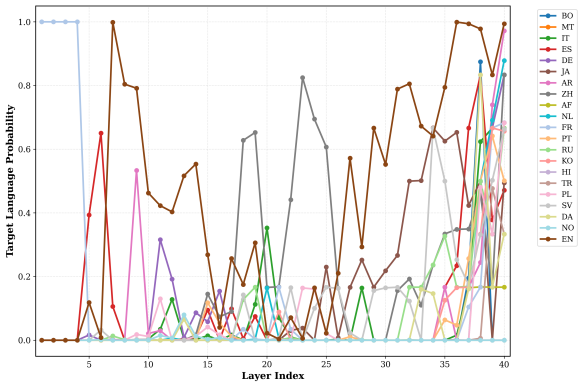


(c) Evolution of target language probability.

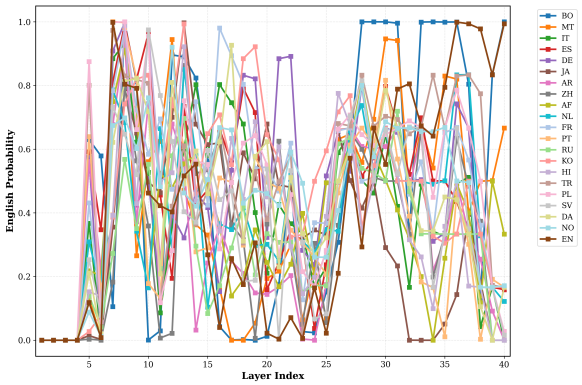


(d) Heatmap of English probabilities.

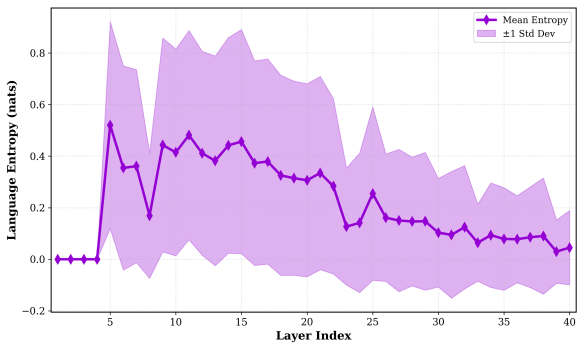
Figure 24: Peeking into language prediction across layers using the logit lens for **Aya-Expanse-8B**.



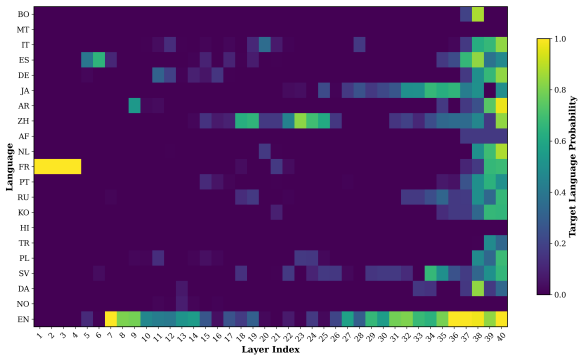
(a) Probabilities of the target languages.



(b) Probabilities of English.



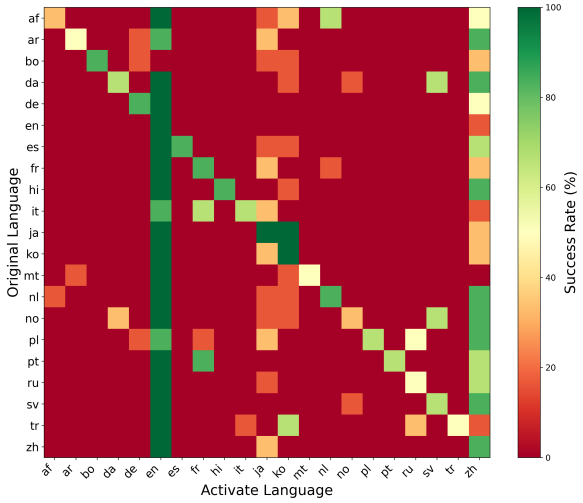
(c) Evolution of target language probability.



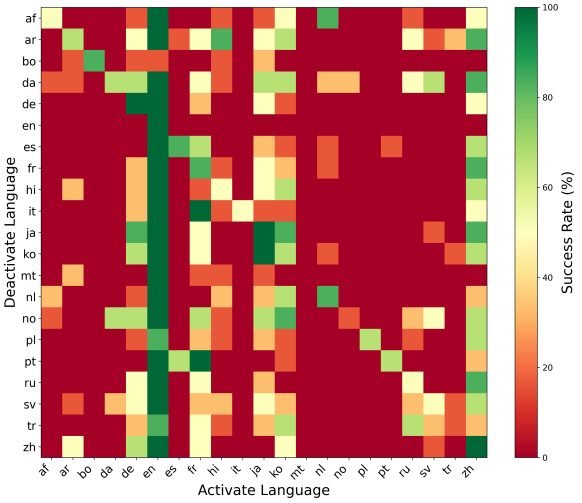
(d) Heatmap of English probabilities.

Figure 25: Peeking into language prediction across layers using the logit lens for **Aya-Expanse-32B**.

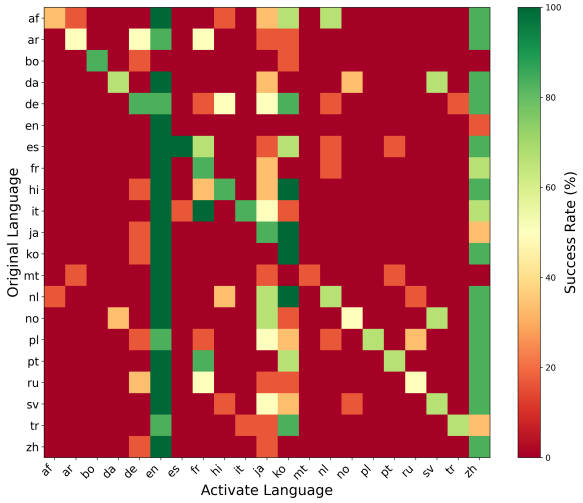
H Language Forcing



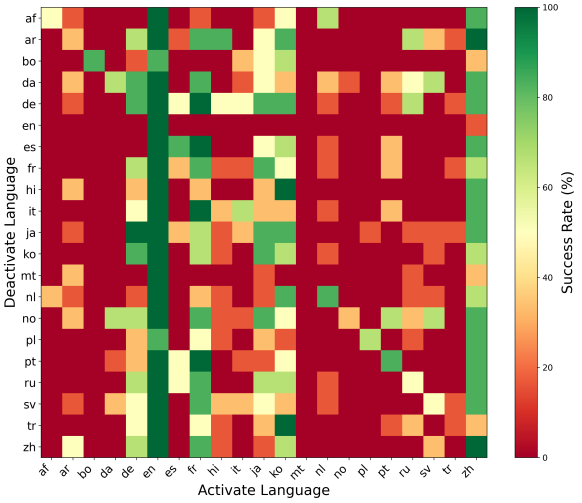
(a) Activate-only - Top 1%



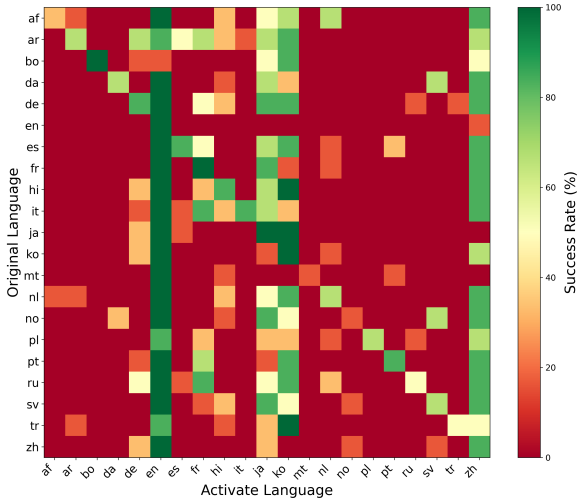
(d) Deactivate-Activate - Top 1%



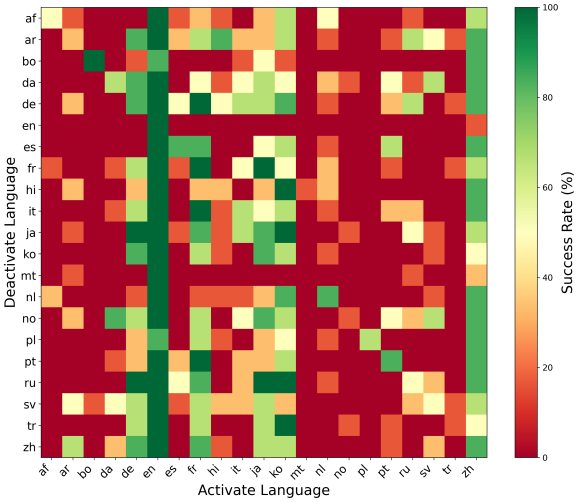
(b) Activate-only - Top 2%



(e) Deactivate-Activate - Top 2%

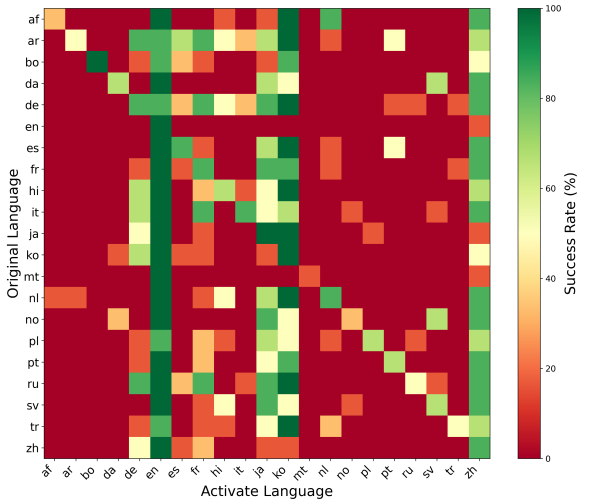


(c) Activate-only - Top 3%

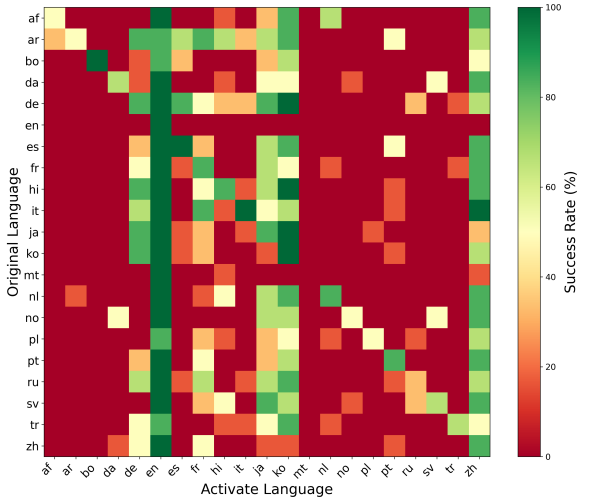


(f) Deactivate-Activate - Top 3%

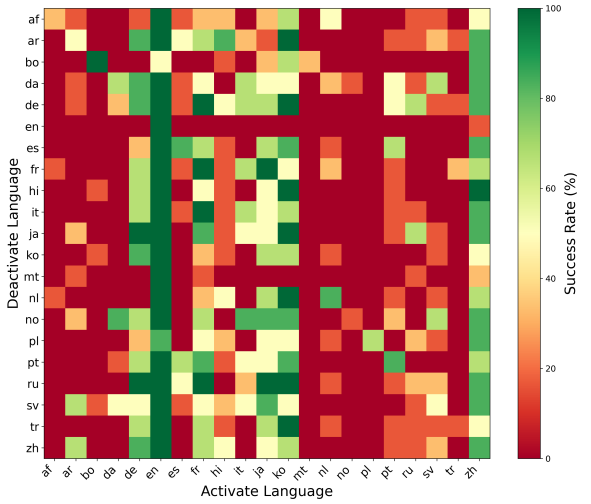
Figure 26: Page 1 - Language forcing results across 5 neuron ratios for two manipulation strategies - activate-only and deactivate-activate for **Llama-3.1-8B**. For deactivation, we set the neurons to 0.



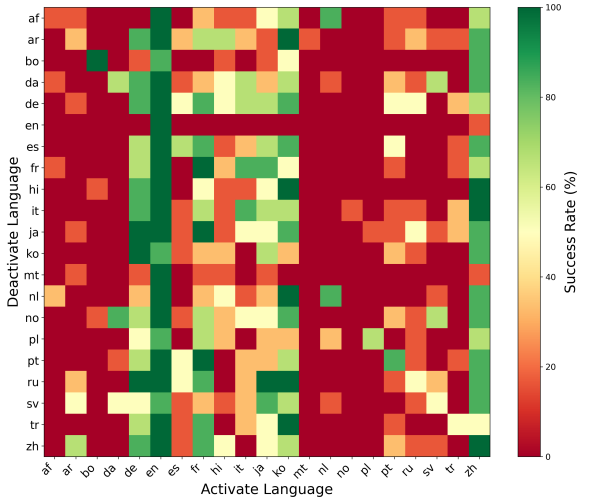
(a) Activate-only - Top 4%



(b) Activate-only - Top 5%

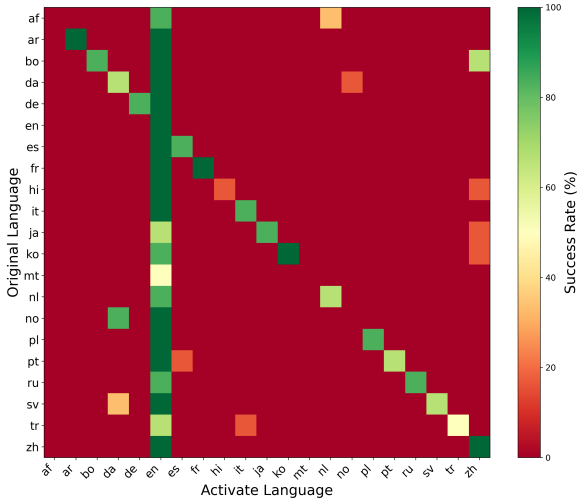


(c) Deactivate-Activate - Top 4%

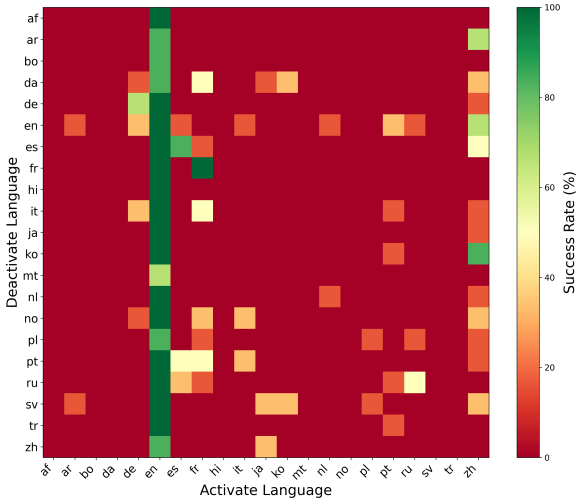


(d) Deactivate-Activate - Top 5%

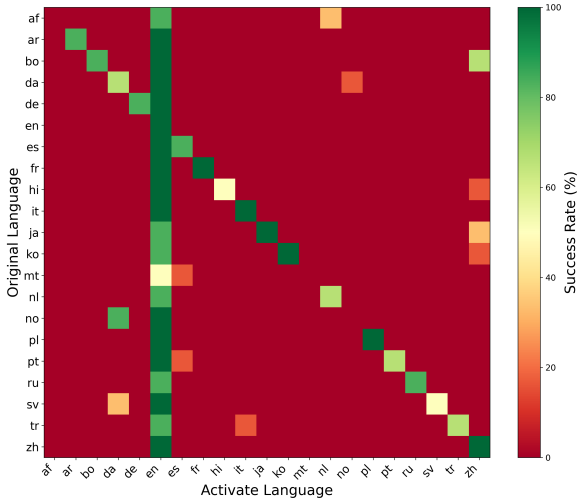
Figure 27: Page 2 - Language forcing results across 5 neuron ratios for two manipulation strategies - activate-only and deactivate-activate for **Llama-3.1-8B**. For deactivation, we set the neurons to 0.



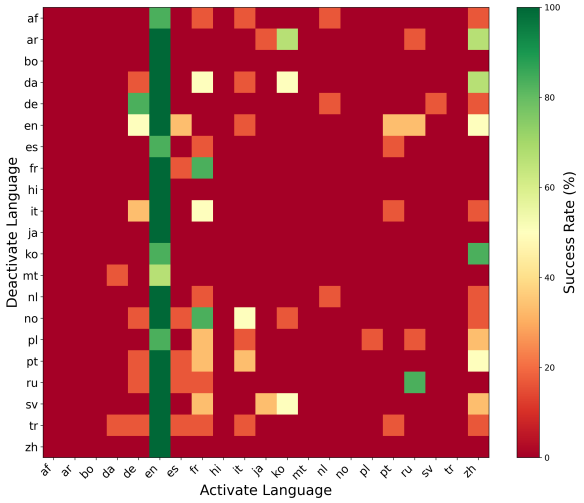
(a) Activate-only - Top 1%



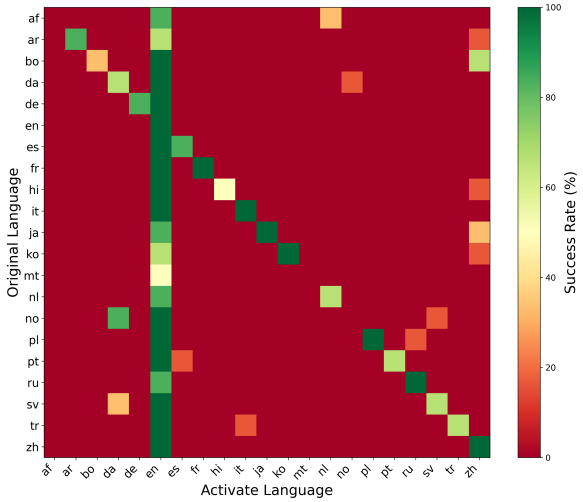
(d) Deactivate-Activate - Top 1%



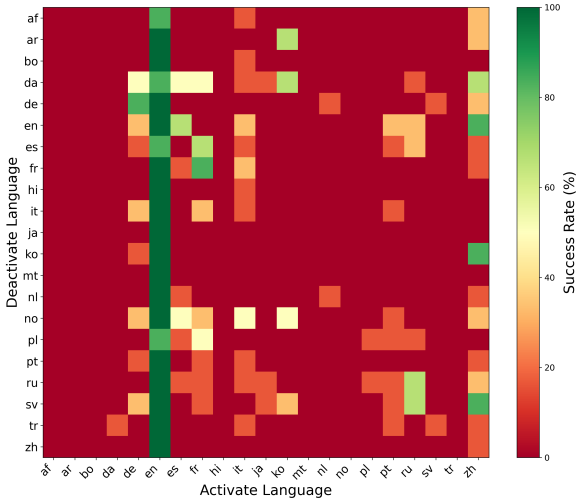
(b) Activate-only - Top 2%



(e) Deactivate-Activate - Top 2%

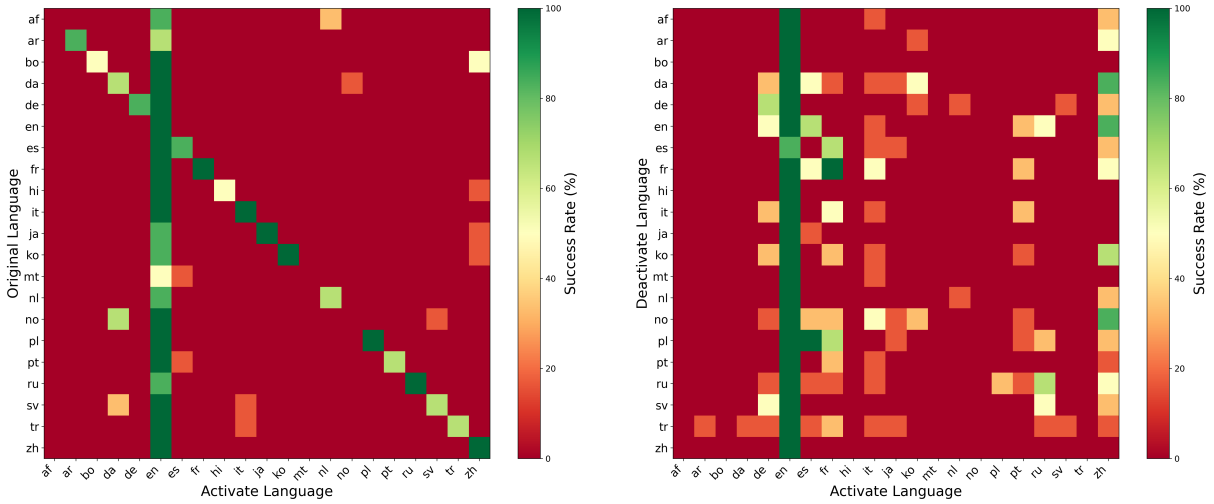


(c) Activate-only - Top 3%

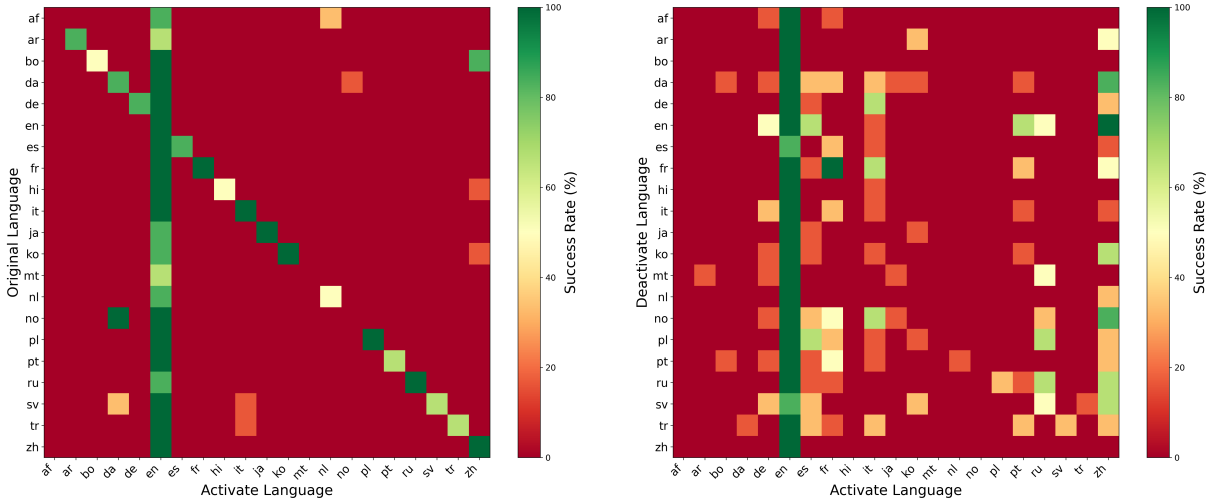


(f) Deactivate-Activate - Top 3%

Figure 28: Page 1 - Language forcing results across 5 neuron ratios for two manipulation strategies - activate-only and deactivate-activate for **Mistral-Nemo**. For deactivation, we set the neurons to -2.5. The results with 0 were slightly less effective



(a) Activate-only - Top 4%



(b) Activate-only - Top 5%

(c) Deactivate-Activate - Top 4%

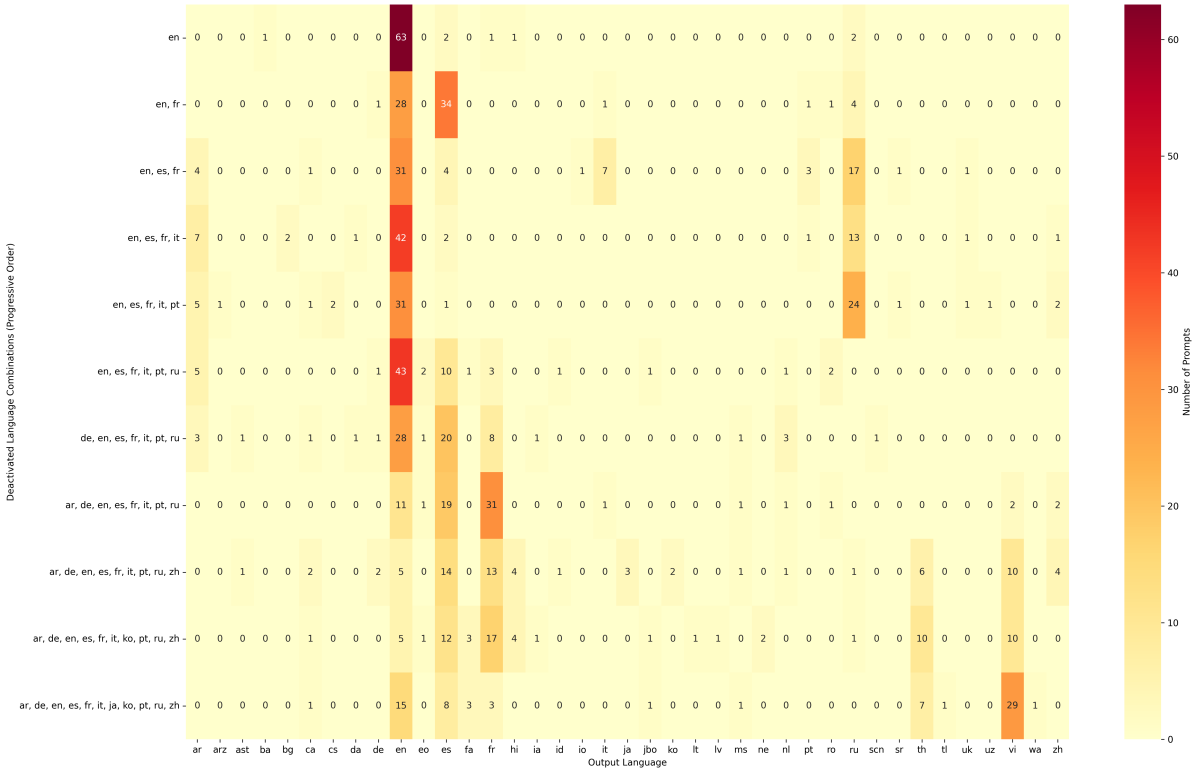
(d) Deactivate-Activate - Top 5%

Figure 29: Page 2 - Language forcing results across 5 neuron ratios for two manipulation strategies - activate-only and deactivate-activate for **Mistral-Nemo**. For deactivation, we set the neurons to -2.5. The results with 0 were slightly less effective

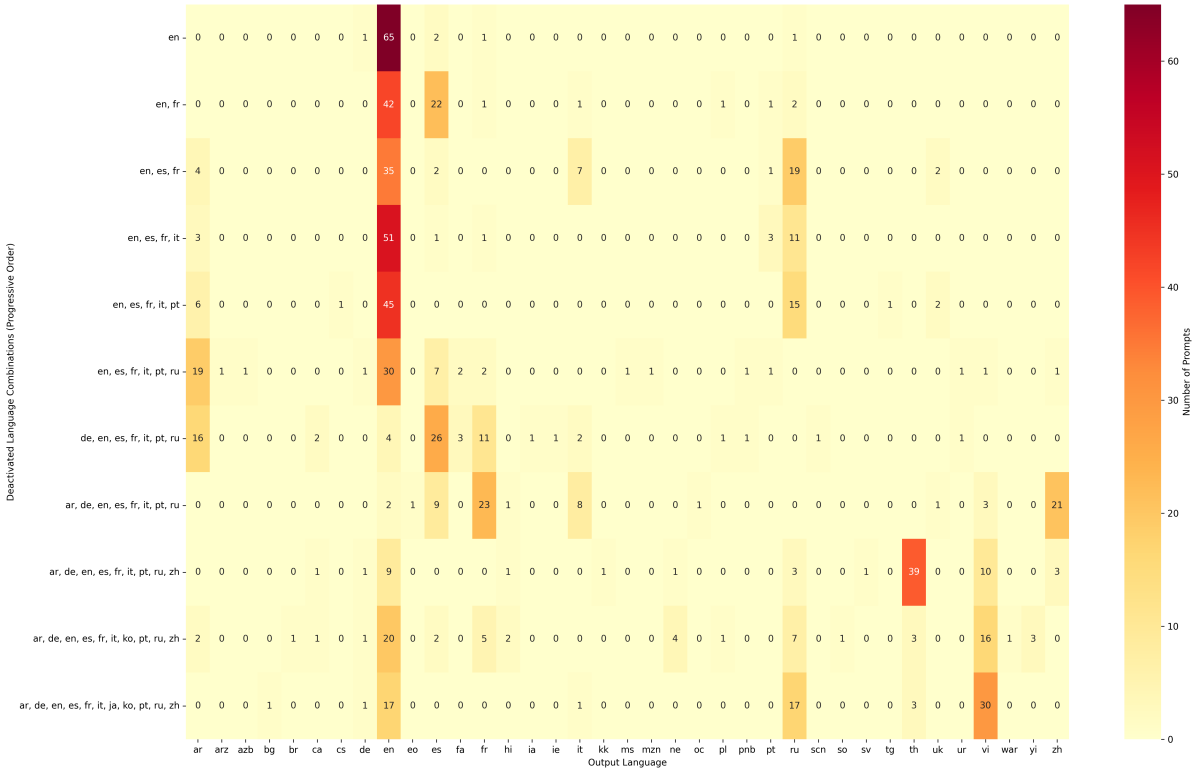
Model	Intervention	Strategy	Top 1%	Top 2%	Top 3%	Top 4%	Top 5%
Mistral-Nemo	Additive	Activate	8.05%	8.35%	8.31%	8.28%	8.31%
	Additive	Deactivate + Activate	8.28%	9.18%	10.36%	11.07%	11.41%
	Replacement	Activate	7.14%	6.92%	6.8%	6.61%	7.03%
	Replacement	Deactivate + Activate	9.94%	10.24%	11.11%	11.64%	11.83%
	DiffMean	Activate	7.82%	8.01%	8.01%	8.05%	8.13%
	DiffMean	Deactivate + Activate	7.45%	7.94%	8.62%	8.5%	8.99%

Table 8: Overall success rates (%) of language forcing for Mistral-Nemo using three intervention types and two manipulation strategies across different top-k% neuron thresholds. Among the methods, replacement performs the worst, while additive and DiffMean interventions achieve comparable results.

I Language "Fallbacks"

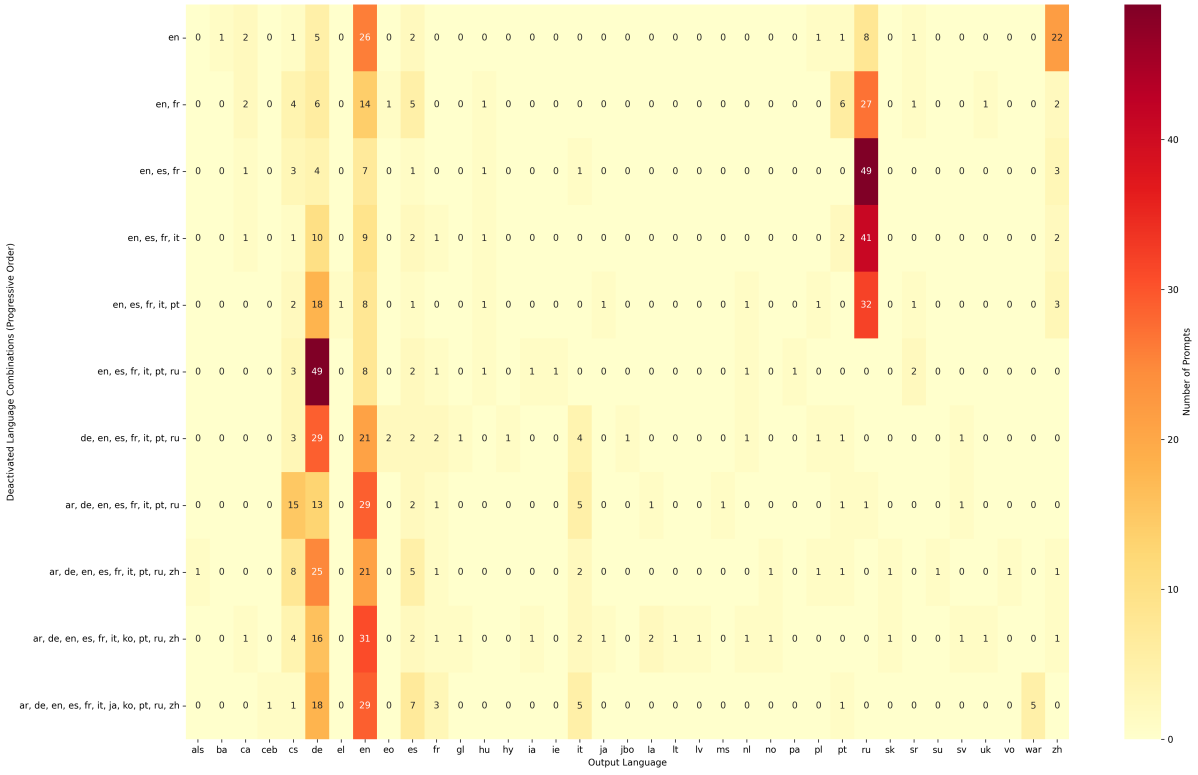


(a) Top 4% neurons

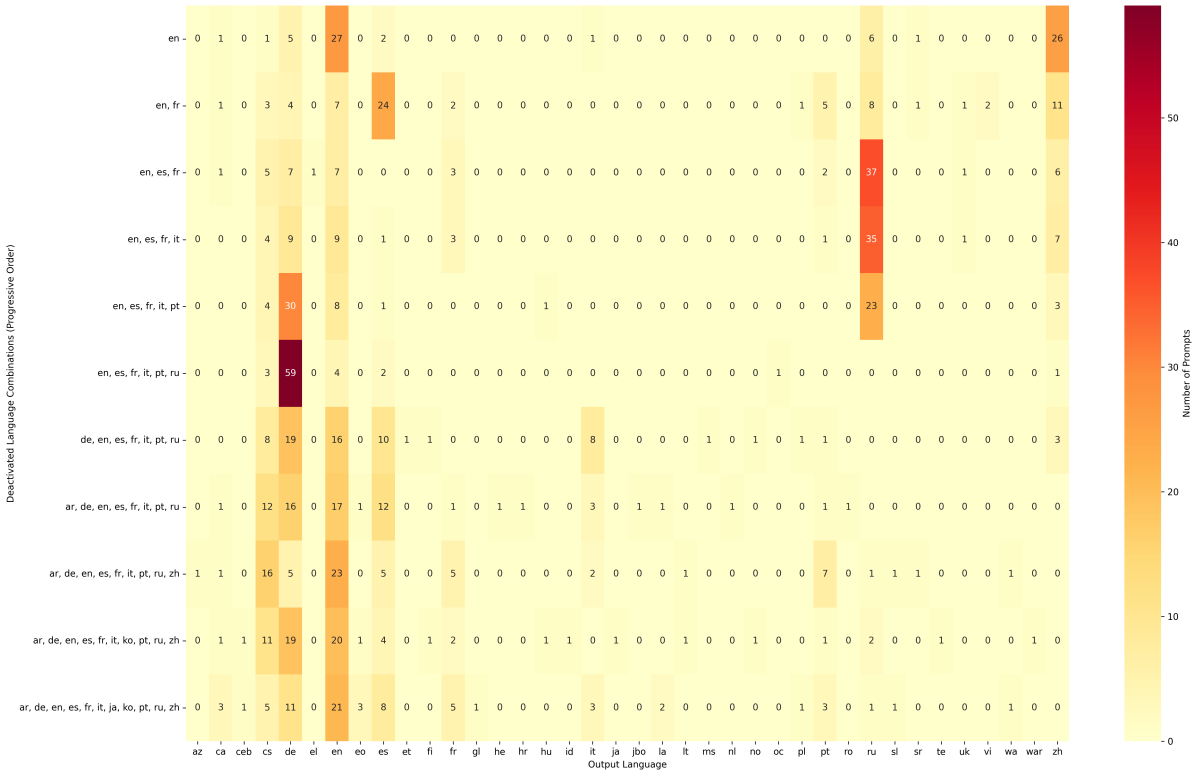


(b) Top 5% neurons

Figure 30: Progressive deactivation of language-specific neurons for high-resource languages for **Llama-3.1-8B**. We set the language neurons to -1 for deactivation in this scenario; 0 does not produce the required effects.



(a) Top 4% neurons



(b) Top 5% neurons

Figure 31: Progressive deactivation of language-specific neurons for high-resource languages for **Mistral-Nemo**. We set the language neurons to -2.5 for deactivation in this scenario; values from 0 to -2 do not produce the required effects.