Language-specific Neurons Do Not Facilitate Cross-Lingual Transfer

Soumen Kumar Mondal[‡], **Sayambhu Sen**[§], **Abhishek Singhania**[§], **Preethi Jyothi**[‡]

[‡] Indian Institute of Technology Bombay, India,
[§]Amazon Alexa
{soumenkm,pjyothi}@iitb.ac.in, {sensayam, mrabhsin}@amazon.com

Abstract

Multilingual large language models (LLMs) aim towards robust natural language understanding across diverse languages, yet their performance significantly degrades on low-resource languages. This work explores whether existing techniques to identify language-specific neurons can be leveraged to enhance cross-lingual task performance of low-resource languages. We conduct detailed experiments covering existing language-specific neuron identification techniques (such as Language Activation Probability Entropy and activation probability-based thresholding) and neuron-specific LoRA fine-tuning with models like Llama 3.1 and Mistral Nemo. We find that such neuron-specific interventions are insufficient to yield cross-lingual improvements on downstream tasks (XNLI, XQuAD) in low-resource languages. This study highlights the challenges in achieving cross-lingual generalization and provides critical insights for multilingual LLMs[1].

1 Introduction

Acquiring multilingual capabilities in LLMs remains a challenge, particularly for low-resource languages (Hangya et al., 2022; Conneau et al., 2020; Lample and Conneau, 2019). Despite their remarkable success in tasks that require cross-lingual transfer, models such as Llama 3.1 (Grattafiori et al., 2024) and Mistral Nemo (MistralAI, 2024) do not perform consistently across languages, particularly underperforming on low-resource languages (Touvron et al., 2023; Hu et al., 2020). This is largely due to the imbalance in high-quality training data across languages, thus limiting the ability of multilingual models to effectively scale to low-resource languages (Touvron et al., 2023; Xue et al., 2021).

A tool that has recently emerged to better understand the nature of multilinguality in these LLMs is the use of *language-specific neurons* (Duan et al., 2025a; Tang et al., 2024; Zhang et al., 2024). These neurons are claimed to encode unique language-specific features pertaining to each language, thus potentially enabling targeted language interventions. Previous studies (Kojima et al., 2024a; Zhao et al., 2024a; Tang et al., 2024) have demonstrated that these neurons play an important role in language generation tasks. However, the extent to which these neurons contribute to or affect cross-lingual transfer to low-resource languages when evaluated on downstream tasks such as natural language inference (XNLI) and question answering (XQuAD) remains unclear.

In this study, we systematically probe the role of language-specific neurons in facilitating cross-lingual transfer within multilingual LLMs. By utilizing existing techniques to identify language-specific neurons such as Language Activation Probability Entropy (LAPE) (Tang et al., 2024) and Low Rank Adaptation (LoRA)-based fine-tuning (Hu et al., 2021), we aim to identify and analyze neurons that mainly contribute towards language-specific representations. Our experiments span two popular cross-lingual benchmarks, XNLI for NLI (Conneau et al., 2018) and XQuAD for QA (Artetxe et al., 2020). After identifying language-specific neurons using existing techniques for a target language, we modify the activations of these language-specific neurons using different aggregation schemes in an attempt to amplify their role in cross-lingual transfer.

Our results show that such test-time (training-free) interventions via language-specific neurons are not very effective in enabling cross-lingual transfer, yielding very modest overall performance improvements of less than 1 absolute point in accuracy for low-resource languages. Fine-tuning strategies like neuron freezing and activation sub-

[1]Code is available at GitHub: https://github.com/csalt-research/LangSpecificNeurons

stitution were shown to significantly impact generation (Lai et al., 2024; Kojima et al., 2024a) but do not show any consistent impact on cross-lingual task performance. A deeper analysis revealed that language-specific neurons often lack independence and we hypothesize that this polysemantic nature of neuron activations limits the effectiveness of targeted adjustments in multilingual LLMs (Elhage et al., 2022).

2 Methodology

The goal of this work is to explore whether targeting language-specific neurons in multilingual LLMs can be used to improve downstream performance on tasks such as XNLI and XQuAD. Previous studies (Zhao et al., 2024b; Kojima et al., 2024a; Tang et al., 2024; Duan et al., 2025b) have shown that distinct neuron subsets exist in multilingual models that encode language-specific features. Prior work (Bhattacharya and Bojar, 2023) further indicates that language-specific representations are largely prevalent within feedforward networks.

While prior work focused on how deactivating language-specific neurons degrades the quality of language generation, there has been little investigation into whether activating or fine-tuning these neurons can positively influence task performance (Zhao et al., 2024c; Lai et al., 2024). This forms the main motivation for our work. We aim to evaluate the role of language-specific neurons by aiming to enhance cross-lingual task performance through targeted neuron interventions. Our results indicate that manipulating language-specific neurons, either by activating or fine-tuning them, does not lead to significant improvements in downstream task performance.

2.1 Language-Specific Neuron Identification

In LLMs, a *neuron* corresponds to the output of the non-linear activation function within a feedforward layer. Let L be the total number of feedforward layers and d_f be the dimensionality of each feedforward layer. Each neuron is uniquely identified by a pair of indices (i, j), where $i \in \{1, 2, \ldots, L\}$ denotes the layer index and $j \in \{1, 2, \ldots, d_f\}$ denotes the position within the hidden dimension of the feedforward network. As our main approach, we employ the LAPE method (Tang et al., 2024) to identify language-specific neurons. For a given language l and a neuron indexed by (i, j), let $h^l_{i,j}(x)$ denote the activation of that neuron for an input sentence s. We define the activation probability of this neuron as:

$$\mathbb{P}\left(h^l_{i,j}(s) > 0\right) := \mathbb{E}_{s\sim D_l}\left[\mathbb{I}(h^l_{i,j}(s) > 0)\right],$$

where D_l represents the corpus in language l and $\mathbb{I}(\cdot)$ is the indicator function that equals 1 if the condition is satisfied and 0 otherwise. Formally, the LAPE score for a neuron (i, j) is defined as:

$$\text{LAPE}(i, j) = -\sum_{l=1}^{k} P^l_{i,j} \log P^l_{i,j},$$

$$P^l_{i,j} = \frac{\mathbb{P}\left(h^l_{i,j}(x) > 0\right)}{\sum_{l'\in\mathcal{L}} \mathbb{P}\left(h^{l'}_{i,j}(x) > 0\right)}$$

where $P^l_{i,j}$ represents the normalized activation probability of neuron (i, j) for language l, and k denotes the total number of languages in the set $\mathcal{L}$. Neurons with low LAPE values are deemed to be language-specific since they exhibit high activation probabilities for only a limited subset of languages. We note here that the LAPE method is dependent on the choice of the language set $\mathcal{L}$ used for calculating the activation probability distributions. To address this limitation, we propose a simple alternative that does not have such a dependency.

Existing methods (Tang et al., 2024; Xie et al., 2021a) often consider neurons to be relevant to a language if their activation is greater than 0, and quantify this as a relevance score computed as $r^l_{i,j} = \mathbb{E}[\mathbb{I}(h^l_{i,j} > 0)]$ where $h^l_{i,j}$ is the activation of neuron (i, j) for language l. However, it overlooks the possibility that negative activations can also carry meaningful information. To account for this, we propose an activation statistics-based approach. Instead of relying on a threshold of 0, we consider neurons as relevant if their activation exceeds a chosen percentile threshold of the overall activation distribution. For example, the relevance of a neuron based on the 90th percentile is defined as $r^l_{i,j} = \mathbb{E}[\mathbb{I}(h^l_{i,j} > P_{90}(h^l_{i,j}))]$ where $P_{90}(h^l_{i,j})$ is the 90th percentile of the activation values for neuron (i, j) in language l. We call this technique *Activation Probability 90p* which is entirely based on neuron activations and avoids the language set dependency issue inherent to LAPE. Neurons are then ranked based on their relevance scores, and the top m neurons are selected as being language-specific. More details on language neuron identification can be found in Appendix A.

2.2 Neuron Fine-Tuning using LoRA

LoRA (Hu et al., 2021) is employed to efficiently fine-tune only the neurons identified as language-specific in the MLP layers. Let $\mathbf{W} \in \mathbb{R}^{d\times k}$ be a pre-trained weight matrix; LoRA adds a trainable update $\Delta\mathbf{W}$ such that $\mathbf{W}' = \mathbf{W}+\Delta\mathbf{W}$. To remain parameter-efficient, $\Delta\mathbf{W}$ is factorized into two low-rank matrices $\mathbf{B} \in \mathbb{R}^{d\times r}$ and $\mathbf{A} \in \mathbb{R}^{r\times k}$: $\Delta\mathbf{W} = \mathbf{BA}$, $r \ll \min(d, k)$. In the forward pass, the feedforward layer computes

$$\mathbf{y} = (\mathbf{W} + \Delta\mathbf{W})\,\mathbf{x} = \mathbf{W}\mathbf{x} + \mathbf{BA}\mathbf{x},$$

with only $\mathbf{B}$ and $\mathbf{A}$ being trainable. To restrict updates to language-specific neurons, we define a binary mask $\mathbf{M} \in \{0, 1\}^{d\times k}$. If $\mathbf{M}_{i,j} = 1$, the j-th neuron of layer i is considered language-specific and thus it will be trained; otherwise, it will remain frozen. The effective LoRA update thus becomes:

$$\Delta\mathbf{W} \leftarrow \mathbf{M} \otimes (\mathbf{BA}),$$

where $\otimes$ denotes an element-wise multiplication. Therefore, $\Delta\mathbf{W}_{i,j} = 0$ if $\mathbf{M}_{i,j} = 0$. Hence, the forward pass is given by:

$$\mathbf{y} = (\mathbf{W} + \mathbf{M} \otimes (\mathbf{BA}))\,\mathbf{x},$$

and only those sub-blocks of $\mathbf{B}$ and $\mathbf{A}$ associated with masked entries of 1 are trainable. In addition to these masked LoRA updates, the classification head and attention layers are fine-tuned to maintain overall task performance, while all remaining parameters (including $\mathbf{W}$ itself) remain frozen.

3 Experimental Setup

3.1 Datasets, Tasks, and Models

To identify language-specific neurons, we use a subset of the Wikipedia (Foundation, 2024) dataset spanning 16 languages: *en, fr, es, vi, id, ja, zh, bn, hi, ta, te, mr, ur, kn, ml, pa* [2]. However, only a subset of these languages will be used for evaluation as mentioned in Section 4. The dataset creation process is outlined in Appendix A. For fine-tuning experiments aimed at evaluating task performance, we use two popular multilingual benchmarks: the XNLI dataset (Conneau et al., 2018) for NLI and the XQuAD dataset (Artetxe et al., 2020) for QA. In our experiments, we use two pretrained LLMs: Llama 3.1 (8B) (Grattafiori et al., 2024) and Mistral Nemo (12B) (MistralAI, 2024). More details such as tasks, models, optimizer and hyper-parameters used in LoRA can be found in Appendix B.

[2] https://en.wikipedia.org/wiki/List_of_ISO_639_language_codes

3.2 Experiment Design

The primary goal of this work is to improve the zero-shot performance of the model on target languages, without using target language training data.

Zero-Shot Transfer. The model is fine-tuned on task-specific data from a source language and evaluated on task-specific test data for a target language. We assume access only to task-specific training data in the source language, and no target language task-specific data. Our goal is to improve over zero-shot transfer using (1) test-time language-specific neuron intervention and (2) language-specific neuron fine-tuning, detailed below.

(1) Test-time Neuron Intervention. We train the LLM on the task-specific training dataset in the source language and evaluate its performance on the task-specific test dataset in the target language. During evaluation, we modify the activations of the target language neurons in the forward pass using a range of statistical aggregates computed based on the Wikipedia dataset of target languages.

(2) Language Neuron Fine-Tuning. We fine-tune the language-specific neurons as detailed in Section 2.2. We explore three different setups for fine-tuning: (a) Fine-tuning only the source language-specific neurons, (b) Fine-tuning only the target language-specific neurons, (c) Fine-tuning both the source and target language-specific neurons. After fine-tuning, we evaluate the model by performing test-time interventions on the target language-specific neurons.

4 Results and Analysis

4.1 Zero Shot Transfer Performance

In all our experiments, we use English (*en*) as the source language. For the XNLI task, we evaluate the model's zero-shot performance on Vietnamese (*vi*), Hindi (*hi*), and Urdu (*ur*), while for the XQuAD task, we consider Vietnamese (*vi*), Hindi (*hi*), and Chinese (*zh*) as target languages. These target languages are selected due to their relatively lower performance in the XNLI and XQuAD benchmark results (Artetxe et al., 2020; Conneau et al., 2018), making them strong candidates for evaluating improvements in cross-lingual transfer. For the XNLI task, we use a subset of 100,000 training samples, which corresponds to 25% of the full training dataset (Asai et al., 2024). For the XQuAD task, we utilize the entire training dataset. Table 1 and Table 2 present the zero-shot results for both tasks.

EL	No Int	Int μ	Int P90	Int 0	Int P10
Llama 3.1 with LAPE					
vi	**80.5**	79.5	79.0	79.8	77.7
hi	75.0	**75.2**	74.9	74.4	75.1
ur	70.0	**70.4**	69.3	68.5	68.7
Llama 3.1 with Act Prob 90p					
vi	**80.5**	78.2	79.3	79.0	77.4
hi	**75.0**	74.1	71.8	73.7	74.6
ur	**70.0**	69.7	69.3	69.6	69.5
Mistral Nemo with LAPE					
vi	80.5	80.4	**80.6**	79.2	80.5
hi	**76.1**	69.8	66.9	74.9	72.4
ur	66.8	66.5	67.0	**66.9**	65.4
Mistral Nemo with Act Prob 90p					
vi	80.5	67.4	**81.1**	79.8	40.7
hi	**76.1**	72.2	74.5	74.5	66.3
ur	**66.8**	65.9	61.3	66.4	61.6

Table 1: XNLI performance across different models and intervention methods. "No Int" represents zero-shot performance without intervention, while "Int μ", "Int P90", "Int 0", and "Int P10" denote test-time interventions using mean, 90th percentile, zero, and 10th percentile activations, respectively. The best performance for each evaluation language (EL) is highlighted in bold.

EL	No Int	Int μ	Int P90	Int 0	Int P10
Llama 3.1 with LAPE					
vi	**41 (73.5)**	40 (72.9)	31 (69.5)	32 (69.2)	10 (43.2)
hi	38 (64.1)	**40 (65.5)**	36 (65.4)	23 (49.9)	37 (62.8)
zh	**56 (77.5)**	10 (62.8)	3 (56.1)	33 (63.2)	33 (63.2)
Llama 3.1 with Act Prob 90p					
vi	41 (73.6)	39 (73.0)	23 (64.9)	**42 (73.8)**	36 (70.3)
hi	**38 (64.1)**	34 (60.7)	36 (62.8)	38 (62.9)	31 (58.6)
zh	56 (77.5)	**61 (80.7)**	56 (78.8)	55 (78.5)	50 (73.6)
Mistral Nemo with LAPE					
vi	39 (74.6)	**42 (76.8)**	40 (75.0)	13 (45.0)	11 (41.2)
hi	**38 (66.9)**	35 (65.9)	37 (66.6)	22 (51.7)	36 (66.1)
zh	**47 (74.9)**	24 (74.0)	0 (61.6)	14 (53.3)	24 (68.9)
Mistral Nemo with Act Prob 90p					
vi	**39 (74.6)**	11 (43.3)	29 (63.9)	39 (74.5)	0 (6.5)
hi	**38 (66.9)**	26 (54.4)	37 (68.9)	33 (63.8)	0 (11.9)
zh	47 (74.9)	46 (77.4)	20 (59.8)	**48 (76.2)**	0 (17.0)

Table 2: XQuAD performance across different models and intervention methods. The intervention strategies are the same as described in Table 1. The values indicate Exact Match (EM) scores, with F1 scores in parentheses.

4.2 Impact of Test-Time Intervention

Test-Time Interventions Do Not Improve Performance. Tables 1 and 2 show that test-time interventions fail to consistently improve zero-shot transfer performance. Instead, they often disrupt the task-specific information encoded in the activations. This suggests that language-specific neurons in LLMs are not purely language-dependent but also contribute to task-relevant computations. Overwriting their activations with statistical values removes essential information required for solving the task due to the polysemantic nature of neuron activations (Elhage et al., 2022). We also experiment with different approaches for identifying language neurons, including LAPE and activation probability-based methods (e.g., 90th percentile); no significant improvements are observed. From the results for Chinese (*zh*) in XQuAD shown in Table 2, we observe that the *Act Prob 90p* method outperforms LAPE. This difference in performance can be attributed to the fact that the neurons identified by LAPE and *Act Prob 90p* are largely disjoint, as shown in Figures 18 and 19.

Deactivation of Zero Does not Degrade Performance Significantly: Prior studies (Kojima et al., 2024a; Tang et al., 2024) commonly deactivate neurons by setting their activations to zero. However, we argue that zero is not necessarily a true indicator of deactivation. While replacing activations with far lower percentiles (such as the 10th percentile) leads to a clear drop in performance (Table 2), setting activations to zero does not show a similar degradation. Figure 1 illustrates the perplexity change (PPXC(i, j)), defined as the difference in perplexity for language j when language neurons for language i are deactivated versus when they remain active, thereby quantifying the impact of targeted neuron deactivation on language understanding and their role in cross-lingual performance. As illustrated in Figure 1, deactivation at zero significantly increases perplexity (thus degrading generation quality); however, this degradation in perplexity does not directly translate to a decline in task performance. This suggests that setting activations to zero may not be an effective choice for deactivation. Detailed experimentation results can be found in Appendix C.

4.3 Impact of Neuron Fine-Tuning

We fine-tuned the identified language-specific neurons using LoRA but observed no improvement in performance (Table 3). When applying test-time interventions to the fine-tuned models, the results remained consistent with the zero-shot transfer (Table 1), reinforcing that fine-tuning language neurons does not enhance task performance. We also fine-tuned randomly selected neurons in the MLP layers. The results were similar to both language neuron fine-tuning and the original model (Table 8), indicating that LoRA applied to attention layers is already effective for task-specific tuning.

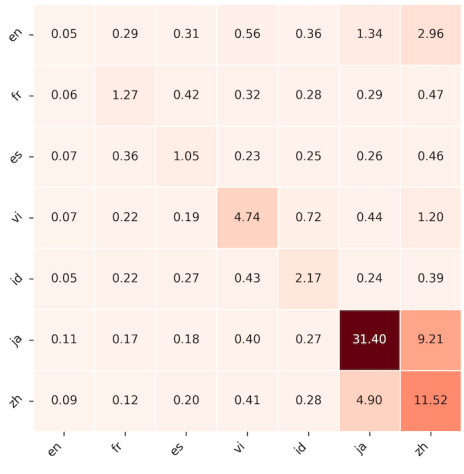

Figure 1: Perplexity Change (PPXC): Measures the effect of interventions on target language perplexity, defined as $\text{PPXC}(i, j) = \text{PPX}(j \mid \text{Intervention by 0 at } i) - \text{PPX}(j)$. Lower $\text{PPXC}(i, j)$ values indicate minimal interference, while higher values signify a significant impact on the model's understanding of language j (on 1 Million tokens).

FTL	EL	No Int	Int μ	Int P90	Int 0	Int P10
Llama 3.1 with LAPE						
en	vi	**80.2**	79.6	78.5	79.2	78.0
vi	vi	**80.1**	79.5	78.6	79.2	78.0
en+vi	vi	**80.1**	79.4	78.5	79.1	78.0
en	hi	**74.9**	74.6	74.6	74.1	74.6
hi	hi	**74.9**	74.6	74.5	74.3	74.6
en+hi	hi	**74.9**	74.5	74.5	74.3	74.7
en	ur	69.8	**70.4**	69.6	70.2	69.0
ur	ur	**69.8**	70.5	69.5	70.4	69.1
en+ur	ur	70.0	**70.6**	69.5	70.3	68.9

Table 3: Fine-tuning results for language-specific neurons on XNLI. The results follow the same format as Table 1, comparing zero-shot performance with test-time interventions across different fine-tuning language neuron (FTL) as per Section 2.2. A complete version is provided in Table 7.

5 Related Works

Other from Tang et al. (2024), Zhu et al. (2024) also introduce LANDeRMT that routes language-aware neurons to mitigate catastrophic forgetting, improving translation quality. Similarly, Xie et al. (2021b) propose a neuron allocation strategy to balance general and language-specific knowledge, thereby enhancing translation without increasing complexity. Lai et al. (2024) present Neuron-TST, which enhances text style transfer by identifying and deactivating source-style neurons to guide target-style generation. Kojima et al. (2024b) analyze language-specific neurons in decoder PLMs, showing that manipulating a small subset can control output language. Huo et al. (2024) study domain-specific neurons in Multimodal LLMs, showing a 10% accuracy gain in domain-specific tasks through neuron manipulation, akin to language-specific neuron use. Durrani et al. (2020) analyze encoder models, and find small neuron subsets capture linguistic tasks, with lower-level tasks requiring fewer neurons. No prior work has examined the effect of language-specific neurons on cross-lingual downstream tasks, which we attempt in this work.

6 Conclusion

In this work, we investigate if language-specific neurons in multilingual LLMs could be manipulated to improve cross-lingual task performance. Our results show that test-time interventions and fine-tuning of language-specific neurons do not yield meaningful improvements. Altering these activations often disrupt task-relevant information likely due to the polysemantic nature of LLM neurons. We found the same behaviour across different methods for identifying language neurons, such as LAPE and activation probability. Additionally, setting activations to zero did not significantly degrade performance, suggesting that zero is not a true indicator of deactivation. These findings indicate that language-specific neurons do not function independently but interact with broader model components, and need further investigation as tools of cross-lingual transfer.

Limitations

This work focuses on language-specific neurons in the MLP layers of multilingual LLMs, excluding attention mechanisms, which may also play a significant role. The experiments use a limited number of languages and datasets, limiting the generalizability of the findings. The interventions rely on statistical activations computed from Wikipedia text, which might not fully capture task-specific behavior. Additionally, the study does not explore alternate methods of fine-tuning techniques that might yield better results. Factors beyond language-specificity in neurons such as training data quality and architectural details of models should also be closely examined for effective cross-lingual transfer.

Acknowledgments

We are grateful to the anonymous reviewers for their insightful feedback. The last author gratefully acknowledges the generous support provided by the joint AI/ML initiative of Amazon and the Indian Institute of Technology Bombay.

References

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.

Akari Asai, Sneha Kudugunta, Xinyan Yu, Terra Blevins, Hila Gonen, Machel Reid, Yulia Tsvetkov, Sebastian Ruder, and Hannaneh Hajishirzi. 2024. BUFFET: Benchmarking large language models for few-shot cross-lingual transfer. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1771–1800, Mexico City, Mexico. Association for Computational Linguistics.

Sunit Bhattacharya and Ondrej Bojar. 2023. Unveiling multilinguality in transformer models: Exploring language specificity in feed-forward networks. *Preprint*, arXiv:2310.15552.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

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

Xufeng Duan, Xinyu Zhou, Bei Xiao, and Zhenguang Cai. 2025a. Unveiling language competence neurons: A psycholinguistic approach to model interpretability. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 10148–10157, Abu Dhabi, UAE. Association for Computational Linguistics.

Xufeng Duan, Xinyu Zhou, Bei Xiao, and Zhenguang Cai. 2025b. Unveiling language competence neurons: A psycholinguistic approach to model interpretability. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 10148–10157, Abu Dhabi, UAE. Association for Computational Linguistics.

Nadir Durrani, Hassan Sajjad, Fahim Dalvi, and Yonatan Belinkov. 2020. Analyzing individual neurons in pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4865–4880, Online. Association for Computational Linguistics.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. 2022. Toy models of superposition. *Transformer Circuits Thread*.

Wikimedia Foundation. 2024. Wikimedia downloads.

Aaron Grattafiori, Abhimanyu Dubey, and Abhinav Jauhri. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Viktor Hangya, Hossain Shaikh Saadi, and Alexander Fraser. 2022. Improving low-resource languages in pre-trained multilingual language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11993–12006, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. *Preprint*, arXiv:2003.11080.

Jiahao Huo, Yibo Yan, Boren Hu, Yutao Yue, and Xuming Hu. 2024. MMNeuron: Discovering neuron-level domain-specific interpretation in multimodal large language model. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6801–6816, Miami, Florida, USA. Association for Computational Linguistics.

Takeshi Kojima, Itsuki Okimura, Yusuke Iwasawa, Hitomi Yanaka, and Yutaka Matsuo. 2024a. On the multilingual ability of decoder-based pre-trained language models: Finding and controlling language-specific neurons. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6919–6971, Mexico City, Mexico. Association for Computational Linguistics.

Takeshi Kojima, Itsuki Okimura, Yusuke Iwasawa, Hitomi Yanaka, and Yutaka Matsuo. 2024b. On the multilingual ability of decoder-based pre-trained language models: Finding and controlling language-specific neurons. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6919–6971, Mexico City, Mexico. Association for Computational Linguistics.

Wen Lai, Viktor Hangya, and Alexander Fraser. 2024. Style-specific neurons for steering LLMs in text style transfer. In *Proceedings of the 2024 Conference on*

Empirical Methods in Natural Language Processing, pages 13427–13443, Miami, Florida, USA. Association for Computational Linguistics.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *Preprint*, arXiv:1901.07291.

MistralAI. 2024. Mistral nemo.

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. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5701–5715, Bangkok, Thailand. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Wanying Xie, Yang Feng, Shuhao Gu, and Dong Yu. 2021a. Importance-based neuron allocation for multilingual neural machine translation. *Preprint*, arXiv:2107.06569.

Wanying Xie, Yang Feng, Shuhao Gu, and Dong Yu. 2021b. Importance-based neuron allocation for multilingual neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5725–5737, Online. 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.

Xue Zhang, Yunlong Liang, Fandong Meng, Songming Zhang, Yufeng Chen, Jinan Xu, and Jie Zhou. 2024. Multilingual knowledge editing with language-agnostic factual neurons. *Preprint*, arXiv:2406.16416.

Xin Zhao, Naoki Yoshinaga, and Daisuke Oba. 2024a. Tracing the roots of facts in multilingual language models: Independent, shared, and transferred knowledge. *Preprint*, arXiv:2403.05189.

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

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

Shaolin Zhu, Leiyu Pan, Bo Li, and Deyi Xiong. 2024. LANDeRMT: Dectecting and routing language-aware neurons for selectively finetuning LLMs to machine translation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12135–12148, Bangkok, Thailand. Association for Computational Linguistics.

A Language Neuron Identification

A.1 Dataset Collection and Preprocessing

The dataset is constructed from publicly available Wikipedia dumps, specifically from the Foundation (2024) dataset. For each language, the following preprocessing steps are applied:

- The dataset is randomly shuffled to ensure diverse text coverage.
- Only the first 100 million tokens per language are retained for computational efficiency.
- Each sequence is truncated to a maximum context length of $T_{\max} = 512$ tokens.

A.2 Activation Computation

To identify language-specific neurons, we compute activation statistics from the Wikipedia dataset for each language. This involves calculating both the mean activation and the 90th percentile activation (*P90*) for every neuron in the model. These statistics provide insights into how neurons behave across different languages and form the basis for selecting language-specific neurons.

A.2.1 Mean Activation Computation

For a given language l, let the activation of neuron (i, j) at token position t in sequence s be denoted as $h_{i,j}^{(s,t)} \in \mathbb{R}$. The mean activation of a neuron across all token positions for a single sequence is computed as:

$$\bar{h}_{i,j}^{(s)} = \frac{1}{T} \sum_{t=1}^{T} h_{i,j}^{(s,t)},$$

where T represents the sequence length (maximum of 512 tokens). To obtain the overall mean activation for a language l, we aggregate across all sequences S_l in the Wikipedia dataset:

$$\mu_{i,j}^{l} = \frac{1}{|S_l|} \sum_{s \in S_l} \bar{h}_{i,j}^{(s)}.$$

This provides a language-specific average activation for each neuron, which helps in identifying neurons that consistently activate for a particular language.

A.2.2 90th Percentile Activation Computation

In addition to mean activation, we compute the 90th percentile activation (*P90*) to capture the upper range of neuron activity. The 90th percentile is useful in determining neurons that are highly responsive in a given language. The *P90* activation for neuron (i, j) in language l is computed as:

$$P_{90}(h_{i,j}^{l}) = \inf \left\{ x \mid F_{h_{i,j}^{(s)}}(x) \geq 0.90, s \in S_l \right\},$$

where $F_{h_{i,j}^{(s)}}(x)$ is the cumulative distribution function (CDF) of the activations of neuron (i, j) for all sequences in language l. In practice, this is computed by sorting activation values for all sequences and selecting the value at the 90th percentile position.

A.3 LAPE and Act Prob 90p Details

In our experiments, we focus on two specific language sets: *Set1* and *Set6*. Although we have explored different combinations of language sets during our analysis, for clarity and brevity we present results corresponding only to *Set1* and *Set6*.

Set1: Core Languages. This set consists of languages that were part of the original LAPE analysis (Tang et al., 2024). It includes: {en, fr, es, vi, id, ja, zh}. This selection covers a broad range of linguistic families, including *Indo-European (en, fr, es), Austroasiatic (vi), Austronesian (id), Japonic (ja), and Sino-Tibetan (zh)*. These languages are well-represented in large-scale multilingual corpora and serve as strong candidates for evaluating multilingual neuron activations.

Set6: Indian Language-Dominant Set. While *Set1* includes a mix of global languages, *Set6* is specifically designed to focus on *Indian languages*: {en, bn, hi, ta, te, mr, ur, kn, ml, pa}. The motivation behind selecting *Set6* is to investigate how LLMs encode representations for typologically and script-wise diverse Indian languages. The inclusion of *Bengali (bn), Hindi (hi), Tamil (ta), Telugu (te), Marathi (mr), Urdu (ur), Kannada (kn), Malayalam (ml), and Punjabi (pa)* ensures a wide coverage of Indo-Aryan and Dravidian language families.

The LAPE method is evaluated on both *Set1* and *Set6* to determine how neuron activations vary across these two distinct sets. Since *Set1* was originally introduced in prior work, our experiments on *Set6* extend the understanding of LAPE to Indian languages, which are underrepresented in pretrained LLMs.

For the *Activation Probability 90p* (Act Prob 90p) method, we select {en, vi, hi, ur, zh}. This selection was based on language diversity, cross-lingual representation, and performance disparities in downstream tasks. Since Act Prob 90p is a set-independent method, we focus on selecting languages that exhibit poor performance in task-specific evaluations. Specifically, for the XNLI task, the lowest-performing languages were *Hindi (hi), Urdu (ur), and Vietnamese (vi)*, leading to their inclusion. Similarly, for the XQuAD task, the weakest-performing languages were *Vietnamese (vi), Hindi (hi), and Chinese (zh)*, which motivated their selection.

B Task, Models and Experiment Details

In this section, we provide detailed descriptions of the two evaluation tasks used in our experiments, namely XNLI and XQuAD, as well as the two large language models (LLMs) used for our study: Llama 3.1 and Mistral Nemo. We also formalize the task setup using mathematical notations.

B.1 Tasks

B.1.1 XNLI

The XNLI dataset (Conneau et al., 2018) is a cross-lingual extension of the MultiNLI dataset, designed for evaluating natural language inference (NLI) across multiple languages. Given a premise p and a hypothesis h, the task is to determine whether the hypothesis is *entailment*, *contradiction*, or *neutral* with respect to the premise. Formally, given a dataset $\mathcal{D} = \{(p_i, h_i, y_i)\}_{i=1}^{N}$, where: $p_i \in \mathcal{X}$ is the premise, $h_i \in \mathcal{X}$ is the hypothesis, $y_i \in \{0, 1, 2\}$ represents the label: entailment (0), contradiction (1), or neutral (2). We conduct zero-shot evaluation on target languages (*vi*, *hi*, *ur*), using English (*en*) as the source language. We limit the training dataset to 100,000 samples (25% of the full dataset) for efficiency.

B.1.2 XQuAD: Cross-lingual Question Answering

XQuAD (Artetxe et al., 2020) is a multilingual question-answering dataset based on the Stanford Question Answering Dataset (SQuAD). The task requires extracting an answer span a from a given context c for a question q. Given a dataset $\mathcal{D} = \{(c_i, q_i, a_i)\}_{i=1}^{M}$, where: $c_i \in \mathcal{X}$ is the passage (context), $q_i \in \mathcal{X}$ is the question, $a_i \in \mathcal{X}$ is the ground-truth answer. We evaluate on target languages (*vi*, *hi*, *zh*) and use the full training dataset for fine-tuning.

B.2 Models

B.2.1 Llama 3.1

Llama 3.1[3] is an 8 billion parameter multilingual model from Meta, trained on diverse text corpora across multiple languages (Grattafiori et al., 2024). The model consists of stacked transformer layers, each comprising self-attention and feedforward MLP components. Llama 3.1 is optimized for computational efficiency and supports a wide range of languages, making it a strong candidate for evaluating multilingual transfer performance.

B.2.2 Mistral Nemo

Mistral Nemo[4] is a 12 billion parameter transformer based model designed for multilingual tasks, with a particular emphasis on high-performance fine-tuning capabilities (MistralAI, 2024). Similar to Llama 3.1, it consists of transformer layers with self-attention and MLP modules.

B.3 Implementation Details for LoRA Fine-Tuning

In this section, we provide an overview of the implementation details for our fine-tuning experiments on the XNLI and XQuAD tasks using LoRA.

For model configuration, our experiments were conducted using the Meta-Llama-3.1-8B and Mistral-Nemo-Base-2407 models. Both models were loaded in 4-bit precision to optimize efficiency. Specifically, we employed the `nf4` quantization type, used `bfloat16` as the compute data type, and enabled double quantization.

Regarding task-specific dataset preparation, for the XNLI dataset—which involves natural language inference by predicting entailment, contradiction, or neutral relationships—we used 25% of the training data and 100% of the evaluation data. The maximum context length was set to 256 tokens. For the XQuAD dataset, which focuses on question answering by extracting answer spans from a given context, we utilized the full training data (100%) along with 100% of the evaluation data, with a maximum context length of 512 tokens.

LoRA was applied to fine-tune specific layers in the attention module of the models for both tasks. The fine-tuning was performed with a LoRA rank $r = 64$ and a LoRA scaling factor $\alpha = 128$. The learning rate was set to 1×10^{-6} for XNLI and 5×10^{-5} for XQuAD, with a weight decay of 0.1 and gradient clipping at a threshold of 10.0. The AdamW optimizer was used with parameters $\beta_1 = 0.95$ and $\beta_2 = 0.999$.

For the training configuration, we trained the model for 2 epochs on XNLI using a batch size of 8, and for 10 epochs on XQuAD using a batch size of 4. A linear warm-up was employed for 1% of the total steps, followed by a linear decay of the learning rate. Mixed precision training was enabled with `bfloat16` to improve memory efficiency.

C Results of Language Neuron Analysis

Figure 2 to 7 presents the number of neurons assigned per language, comparing the LAPE and Activation Probability 90p methods. The overlap of language-specific neurons across different languages is illustrated in Figure 8 to 13, highlighting the extent of shared neurons between languages. The layer-wise distribution of these neurons is shown in Figure 14 to 19, providing insights into where language-specific representations are most prominent in the model. Finally, the impact of neuron interventions on perplexity is analyzed in Figure 20 to 24, which displays the perplexity change across different languages when language neurons are manipulated. These figures collectively summarize the key findings from our language neuron analysis.

Table 4 presents the full XNLI results, extending the analysis from Table 1, incorporating additional statistical interventions, including percentiles at P75, P90, P95, P5, P10, and P25. Similarly, Table 5 provides the full XQuAD results, expanding upon Table 2, detailing both exact match (EM) and F1 scores across various intervention methods. The complete activation statistics for both Llama 3.1 and Mistral Nemo are listed in Table 6, offering a

[3] https://huggingface.co/meta-llama/Llama-3.1-8B

[4] https://huggingface.co/mistralai/Mistral-Nemo-Base-2407

breakdown of mean activations and quantiles, capturing the variations in neuron activations across languages. Finally, Table 7 details the full language neuron fine-tuning results, extending Table 3, comparing zero-shot performance with fine-tuning on different language neuron setups, and evaluating test-time interventions across multiple configurations.

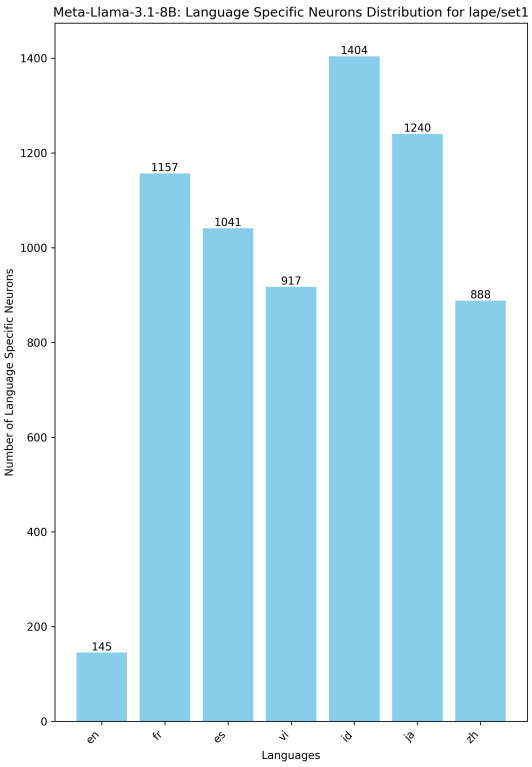

Figure 2: Llama 3.1: Number of language neurons assigned per language for LAPE in a set of languages {*en*,*es*,*fr*,*vi*,*id*,*zh*,*ja*}.

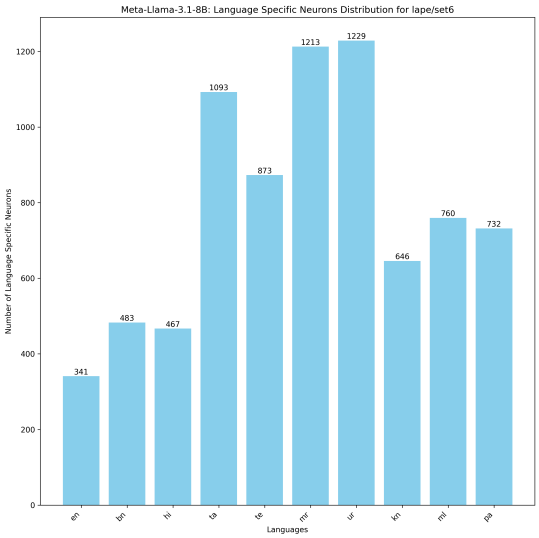

Figure 3: Llama 3.1: Number of language neurons assigned per language for LAPE in a set of languages {*en*,*bn*,*hi*,*ur*,*mr*,*pa*,*ta*, *te*, *ml*, *kn*}.

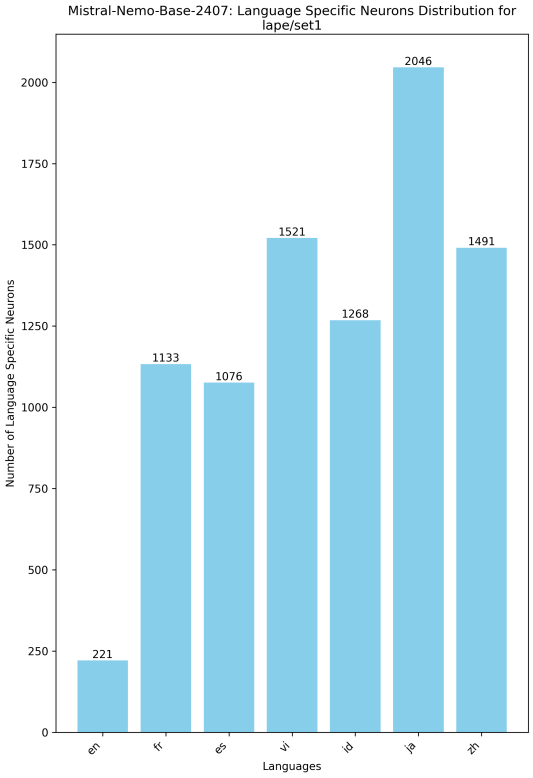

Figure 4: Mistral Nemo: Number of language neurons assigned per language for LAPE in a set of languages {*en*,*es*,*fr*,*vi*,*id*,*zh*,*ja*}.

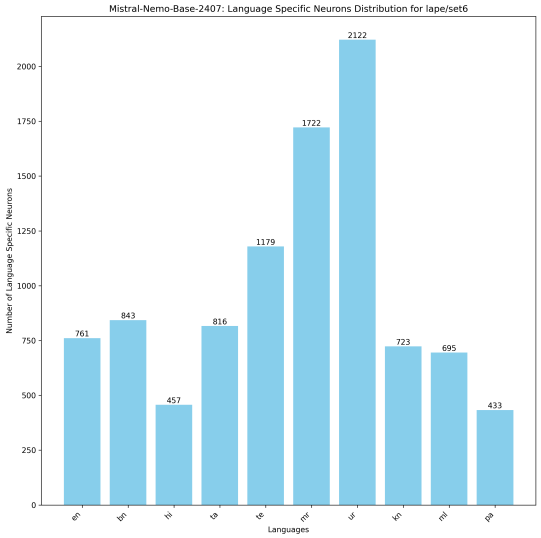

Figure 5: Mistral Nemo: Number of language neurons assigned per language for LAPE in a set of languages {*en*,*bn*,*hi*,*ur*,*mr*,*pa*,*ta*, *te*, *ml*, *kn*}.

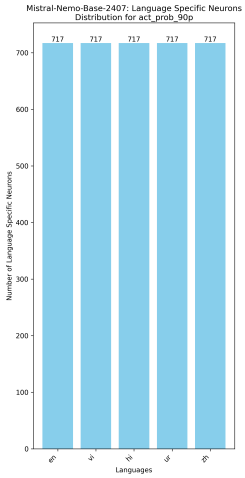

Figure 7: Mistral Nemo: Number of language neurons assigned per language for Activation Probability 90p which is same for all the languages.

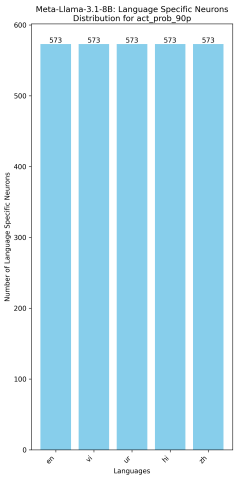

Figure 6: Llama 3.1: Number of language neurons assigned per language for Activation Probability 90p which is same for all the languages.

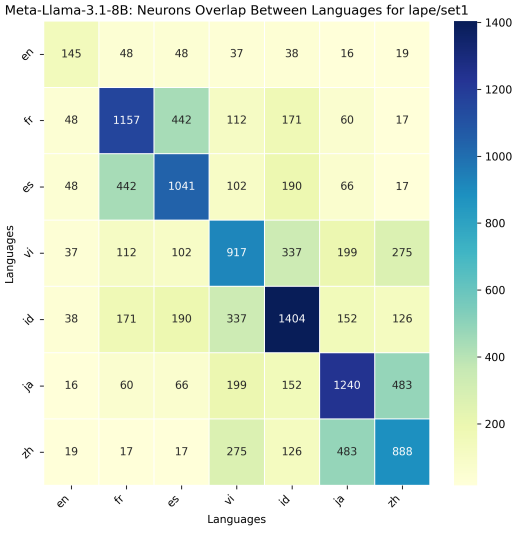

Figure 8: Llama 3.1: Language neuron overlap between languages using LAPE in a set of languages {*en*,*es*,*fr*,*vi*,*id*,*zh*,*ja*}.

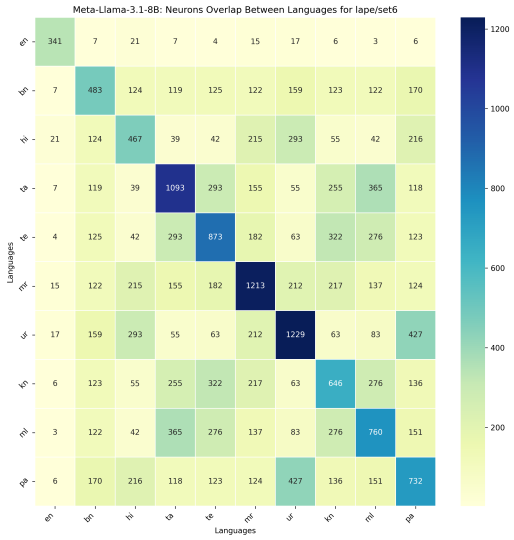

Figure 9: Llama 3.1: Language neuron overlap between languages using LAPE in a set of languages {*en*,*bn*,*hi*,*ur*,*mr*,*pa*,*ta*, *te*, *ml*, *kn*}.

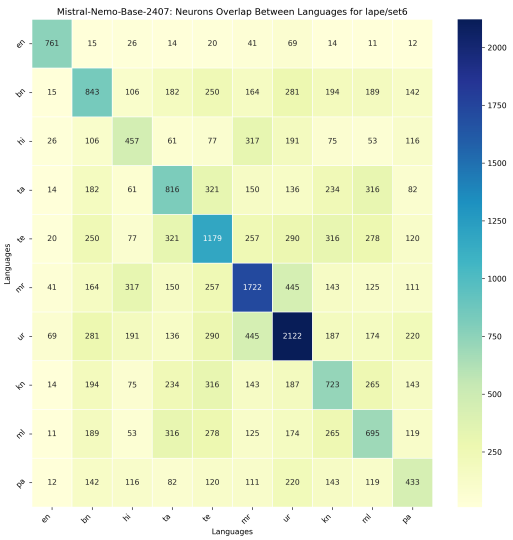

Figure 11: Mistral Nemo: Language neuron overlap between languages using LAPE in a set of languages {*en*,*bn*,*hi*,*ur*,*mr*,*pa*,*ta*, *te*, *ml*, *kn*}.

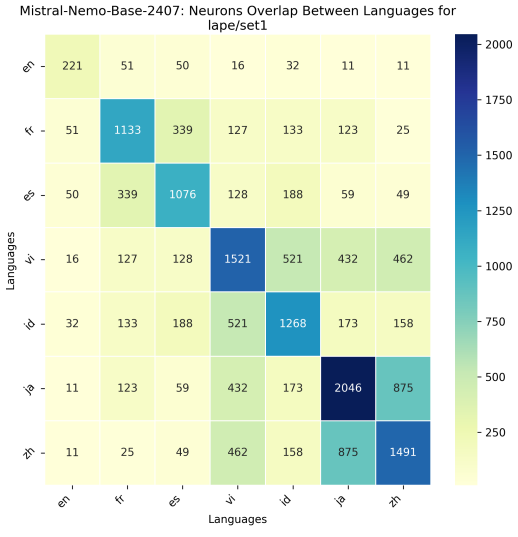

Figure 10: Mistral Nemo: Language neuron overlap between languages using LAPE in a set of languages {*en*,*es*,*fr*,*vi*,*id*,*zh*,*ja*}.

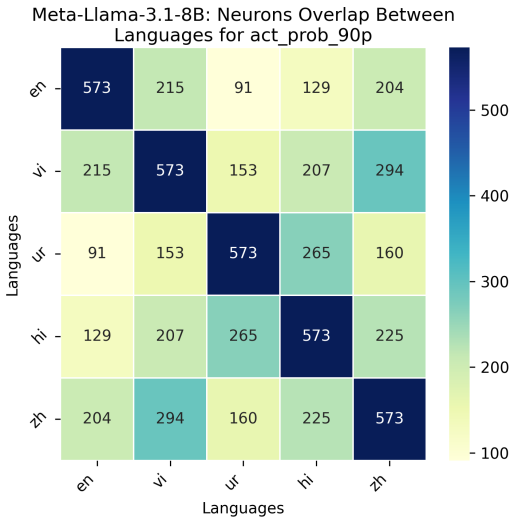

Figure 12: Llama 3.1: Language neuron overlap between languages using Activation Probability 90p.

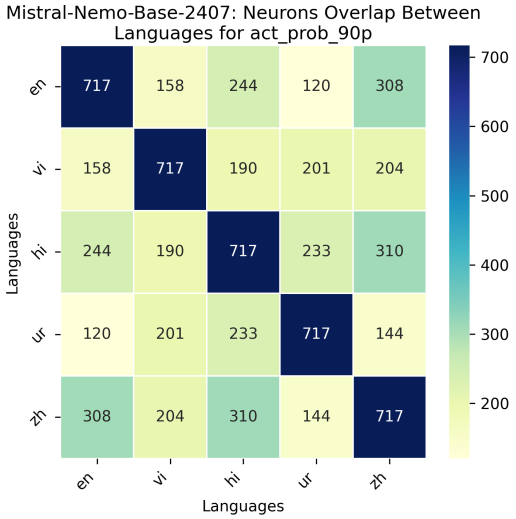

Figure 13: Mistral Nemo: Language neuron overlap between languages using Activation Probability 90p.

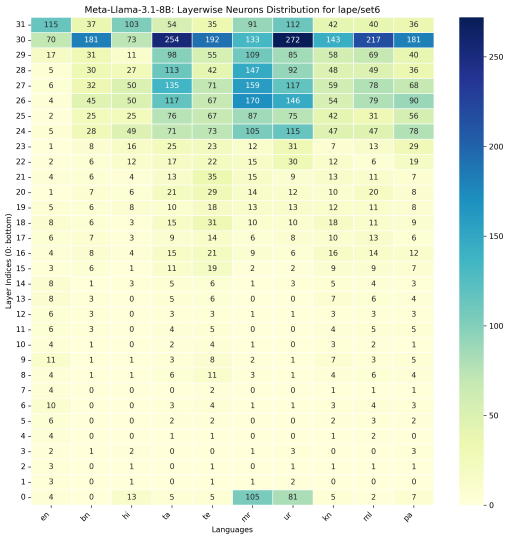

Figure 15: Llama 3.1: Layer-wise distribution of language neurons for LAPE in a set of languages {*en*,*bn*,*hi*,*ur*,*mr*,*pa*,*ta*, *te*, *ml*, *kn*}.

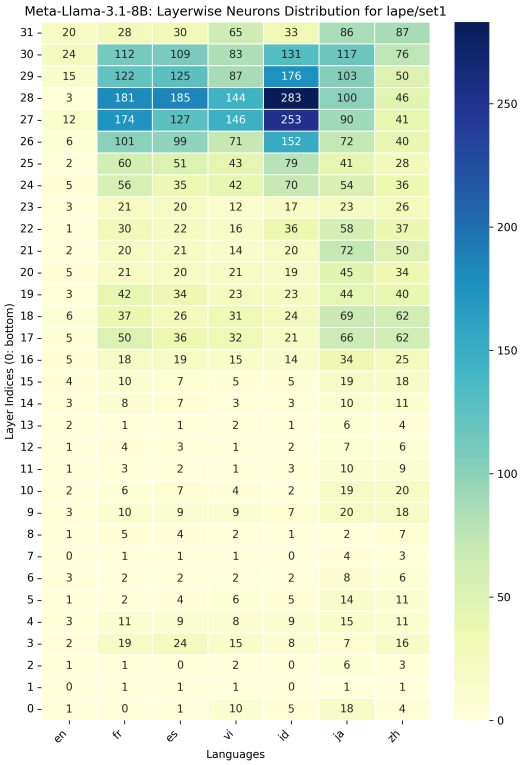

Figure 14: Llama 3.1: Layer-wise distribution of language neurons for LAPE in a set of languages {*en*,*es*,*fr*,*vi*,*id*,*zh*,*ja*}.

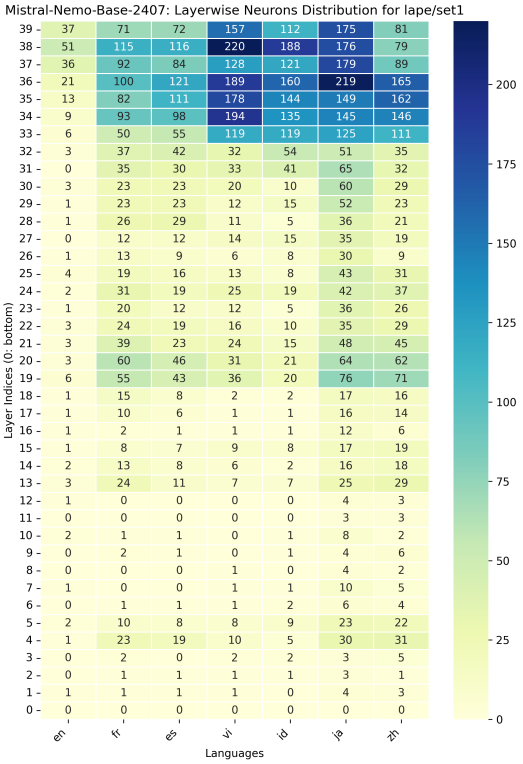

Figure 16: Mistral Nemo: Layer-wise distribution of language neurons for LAPE in a set of languages {*en*,*es*,*fr*,*vi*,*id*,*zh*,*ja*}.

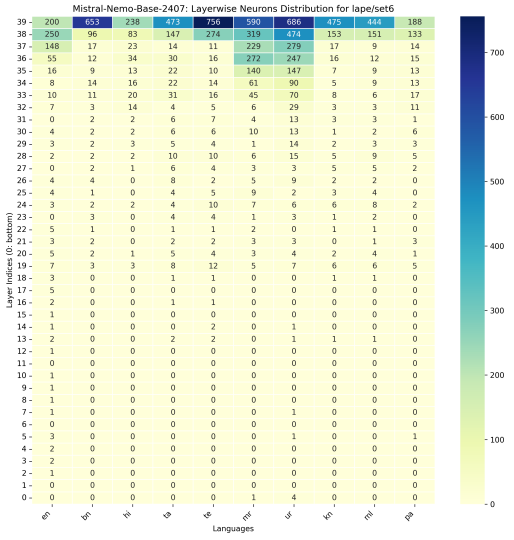

Figure 17: Mistral Nemo: Layer-wise distribution of language neurons for LAPE in a set of languages {*en*,*bn*,*hi*,*ur*,*mr*,*pa*,*ta*, *te*, *ml*, *kn*}.

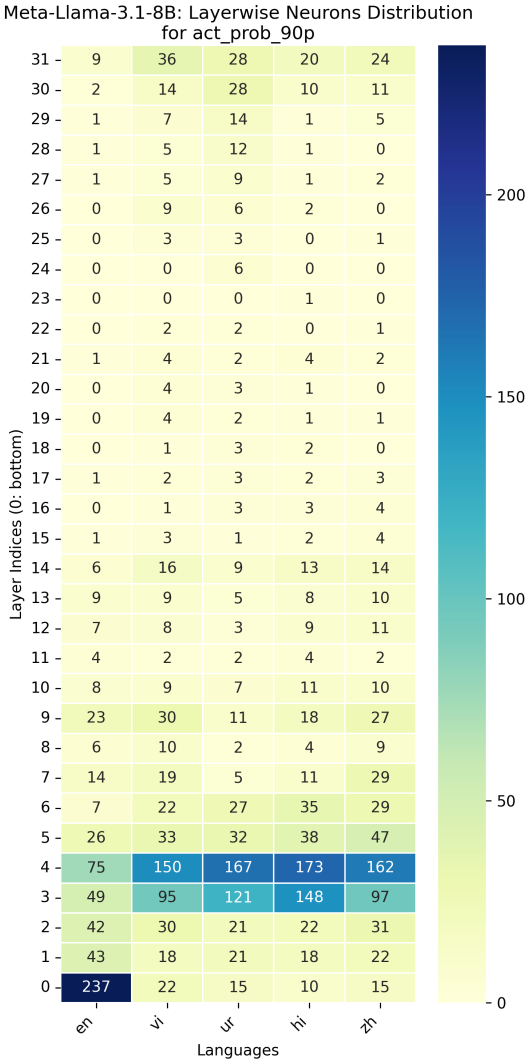

Figure 18: Llama 3.1: Layer-wise distribution of language neurons for Activation Probability 90p.

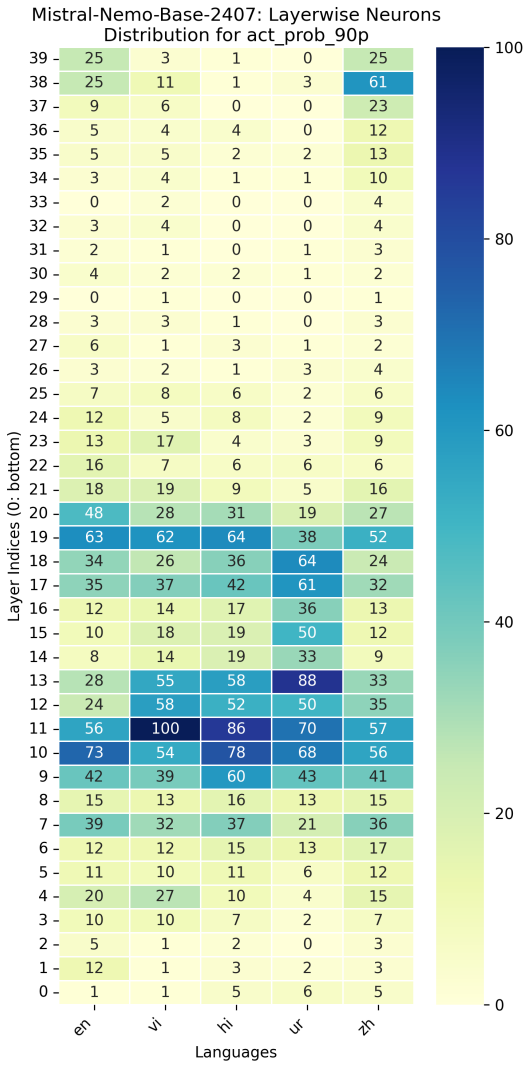

Figure 19: Mistral Nemo: Layer-wise distribution of language neurons for Activation Probability 90p.

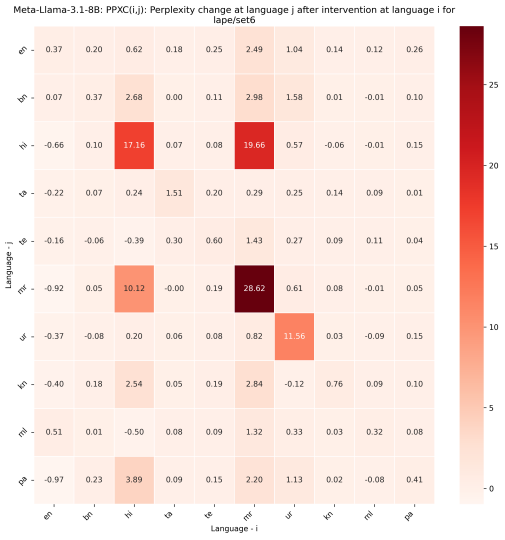

Figure 20: Llama 3.1: Perplexity change for LAPE in a set of languages {*en*,*bn*,*hi*,*ur*,*mr*,*pa*,*ta*, *te*, *ml*, *kn*} (on 0.1 Million tokens).

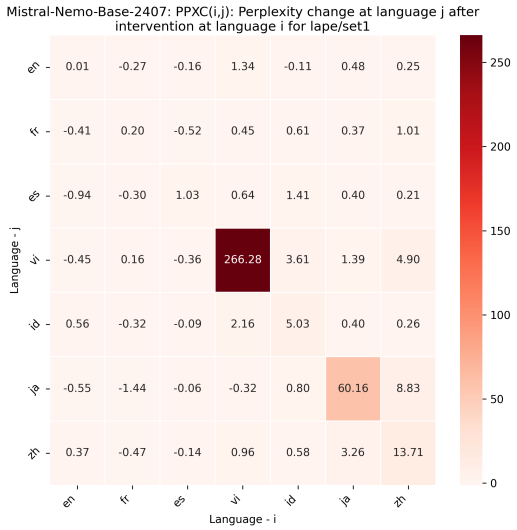

Figure 21: Mistral Nemo: Perplexity change for LAPE in a set of languages {*en*,*es*,*fr*,*vi*,*id*,*zh*,*ja*} (on 0.1 Million tokens).

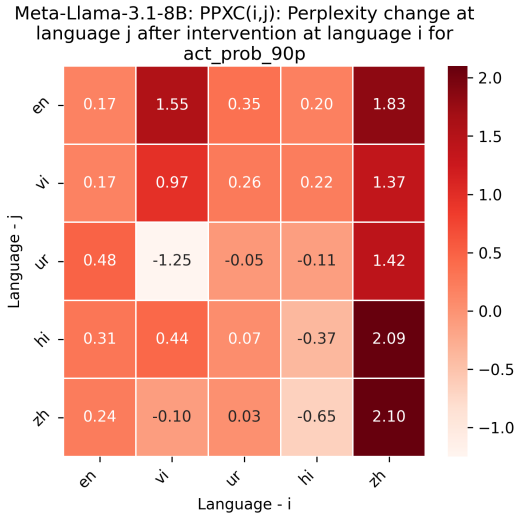

Figure 23: Llama 3.1: Perplexity change for Activation Probability 90p (on 0.1 Million tokens).

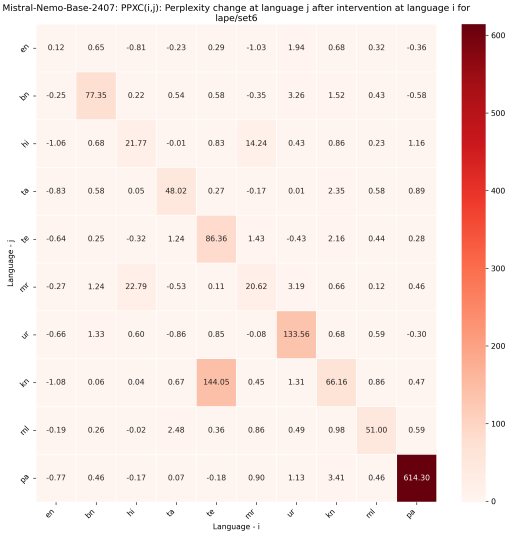

Figure 22: Mistral Nemo: Perplexity change for LAPE in a set of languages {*en*,*bn*,*hi*,*ur*,*mr*,*pa*,*ta*, *te*, *ml*, *kn*} (on 0.1 Million tokens).

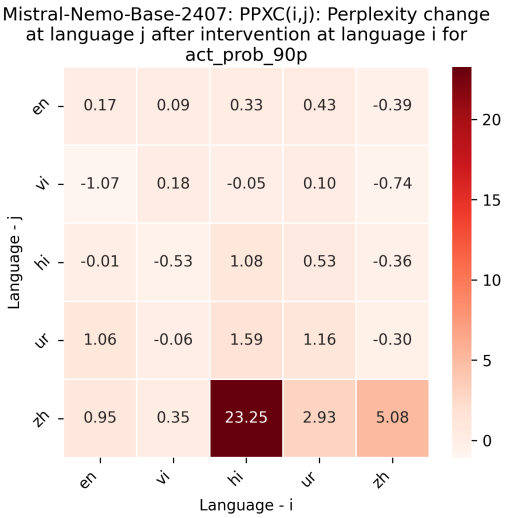

Figure 24: Mistral Nemo: Perplexity change for Activation Probability 90p (on 0.1 Million tokens).

Model & Method	Eval Lang	No Int	Int μ	Int P75	Int P90	Int P95	Int 0	Int P5	Int P10	Int P25
Llama 3.1 + LAPE	vi	**80.5**	79.5	79.1	79.0	78.9	79.8	73.6	77.7	**80.5**
	hi	75.0	**75.2**	74.9	74.9	75.0	74.4	74.8	75.1	74.9
	ur	70.0	**70.4**	70.2	69.3	69.0	68.5	68.6	68.7	69.7
Llama 3.1 + Act Prob 90p	vi	**80.5**	78.2	78.9	79.3	79.5	79.0	76.5	77.4	78.0
	hi	**75.0**	74.1	72.6	71.8	71.2	73.7	75.0	74.6	74.2
	ur	**70.0**	69.7	69.7	69.3	68.7	69.6	69.4	69.5	69.5
Mistral Nemo + LAPE	vi	80.5	80.4	80.5	**80.6**	80.5	79.2	**80.6**	80.5	80.3
	hi	**76.1**	69.8	67.8	66.9	66.6	74.9	73.0	72.4	71.0
	ur	66.8	66.5	67.2	67.0	66.8	**66.9**	65.8	65.4	65.7
Mistral Nemo + Act Prob 90p	vi	80.5	67.4	79.0	**81.1**	**81.1**	79.8	37.4	40.7	47.8
	hi	**76.1**	72.2	73.7	74.5	74.4	74.5	62.4	66.3	69.3
	ur	**66.8**	65.9	64.0	61.3	59.7	66.4	54.6	61.6	65.0

Table 4: Full XNLI performance results, including additional statistical interventions. This table extends Table 1 by incorporating multiple activation percentile-based interventions, including P75, P90, P95, P5, P10, and P25, alongside the mean and zero activation interventions.

Model & Method	Eval Lang	No Int	Int μ	Int P75	Int P90	Int P95	Int 0	Int P5	Int P10	Int P25
Llama 3.1 + LAPE	vi	41 (73.5)	40 (72.9)	36 (76.7)	31 (69.5)	28 (66.8)	32 (69.2)	4 (35.8)	10 (43.2)	39 (71.5)
	hi	38 (64.1)	40 (65.5)	38 (66.4)	36 (65.4)	36 (66.3)	23 (49.9)	38 (62.8)	37 (62.8)	39 (65.3)
	zh	56 (77.5)	10 (62.8)	3 (58.1)	3 (56.1)	4 (52.9)	33 (63.2)	20 (49.1)	33 (63.2)	22 (67.8)
Llama 3.1 + Act Prob 90p	vi	41 (73.6)	39 (73.0)	26 (65.8)	23 (64.9)	24 (64.8)	42 (73.8)	33 (67.8)	36 (70.3)	39 (72.0)
	hi	38 (64.1)	34 (60.7)	35 (62.3)	36 (62.8)	32 (61.9)	38 (62.9)	23 (54.9)	31 (58.6)	35 (60.7)
	zh	56 (77.5)	61 (80.7)	59 (79.5)	56 (78.8)	56 (78.4)	55 (78.5)	40 (61.7)	50 (73.6)	56 (79.3)
Mistral Nemo + LAPE	vi	39 (74.6)	42 (76.8)	40 (76.4)	40 (75.0)	39 (73.6)	13 (45.0)	10 (40.4)	11 (41.2)	37 (73.3)
	hi	38 (66.9)	35 (65.9)	38 (67.1)	37 (66.6)	34 (66.5)	22 (51.7)	36 (65.8)	36 (66.1)	36 (67.1)
	zh	47 (74.9)	24 (74.0)	4 (65.6)	0 (61.6)	0 (59.3)	14 (53.3)	1 (33.6)	24 (68.9)	35 (77.0)
Mistral Nemo + Act Prob 90p	vi	39 (74.6)	11 (43.3)	26 (62.8)	29 (63.9)	15 (48.5)	39 (74.5)	0 (3.8)	0 (6.5)	2 (12.9)
	hi	38 (66.9)	26 (54.4)	34 (65.1)	37 (68.9)	37 (68.7)	33 (63.8)	0 (6.0)	0 (11.9)	12 (35.2)
	zh	47 (74.9)	46 (77.4)	26 (69.9)	20 (59.8)	15 (54.9)	48 (76.2)	0 (8.5)	0 (17.0)	9 (45.6)

Table 5: Full XQuAD performance results, with exact match (EM) and F1 scores across various interventions. This table builds upon Table 2, expanding the analysis with additional intervention strategies. The results further validate the findings on test-time interventions and their impact on cross-lingual task performance.

Model	Lang	Act μ	Act P75	Act P90	Act P95	Act P5	Act P10	Act P25
Llama 3.1	en	-0.2621	-0.0493	0.1693	0.3106	-0.7943	-0.6801	-0.4892
	vi	-0.2599	-0.049	0.1648	0.3012	-0.782	-0.6709	-0.4841
	hi	-0.2728	-0.0648	0.144	0.2784	-0.7893	-0.6788	-0.4934
	ur	-0.254	-0.0498	0.1521	0.2804	-0.7604	-0.6511	-0.4684
	zh	-0.2603	-0.0505	0.1611	0.2958	-0.7797	-0.6689	-0.4826
Mistral Nemo	en	-0.3938	-0.0798	0.2477	0.4629	-1.1888	-1.0149	-0.7294
	vi	-0.399	-0.0807	0.2454	0.4564	-1.1975	-1.0237	-0.737
	hi	-0.4265	-0.1147	0.2053	0.4133	-1.2092	-1.0392	-0.7588
	ur	-0.397	-0.0881	0.2252	0.4279	-1.1722	-1.0029	-0.7238
	zh	-0.3962	-0.0715	0.2568	0.4678	-1.2088	-1.032	-0.7388

Table 6: Activation statistics for Llama 3.1 and Mistral Nemo across different languages for Wikipedia dataset. It provides an in-depth analysis of activation values, including mean activation values and key quantiles (P75, P90, P95, P5, P10, P25).

FTL	EL	No Int	Int μ	Int P90	Int 0	Int P10
Llama 3.1 with LAPE						
en	vi	**80.2**	79.6	78.5	79.2	78.0
vi	vi	**80.1**	79.5	78.6	79.2	78.0
en+vi	vi	**80.1**	79.4	78.5	79.1	78.0
en	hi	**74.9**	74.6	74.6	74.1	74.6
hi	hi	**74.9**	74.6	74.5	74.3	74.6
en+hi	hi	**74.9**	74.5	74.5	74.3	74.7
en	ur	69.8	**70.4**	69.6	70.2	69.0
ur	ur	**69.8**	70.5	69.5	70.4	69.1
en+ur	ur	70.0	**70.6**	69.5	70.3	68.9
Llama 3.1 with Act Prob 90p						
en	vi	**80.3**	78.0	79.1	78.5	77.5
vi	vi	**80.1**	78.0	79.0	78.7	77.4
en+vi	vi	**80.1**	78.0	78.9	78.8	77.4
en	hi	**74.9**	73.9	72.1	73.3	74.5
hi	hi	**74.9**	73.8	72.1	73.9	74.6
en+hi	hi	**74.9**	73.7	72.1	73.8	74.6
en	ur	69.9	**70.1**	69.4	70.0	69.5
ur	ur	69.9	70.0	69.4	**70.1**	69.5
en+ur	ur	69.8	69.9	69.3	**70.1**	69.6

Table 7: Full language neuron fine-tuning results for XNLI. This table extends Table 3, presenting fine-tuning experiments where language-specific neurons are updated in different configurations. It includes results for models fine-tuned on the source language alone, the target language alone, and both together, with evaluation of test-time interventions across multiple setups.

Model	FTL	EL	No Int
Llama 3.1	rand	vi	80.1
Llama 3.1	rand	hi	74.8
Llama 3.1	rand	ur	69.8
Mistral Nemo	rand	vi	80.4
Mistral Nemo	rand	vi	75.6
Mistral Nemo	rand	ur	65.5

Table 8: Random neuron fine-tuning results for Llama 3.1 and Mistral Nemo. The table reports zero-shot performance without intervention (No Int) after fine-tuning randomly selected 10 neurons per layers instead of language-specific neurons.