How do Large Language Models Handle Multilingualism?

Anonymous ACL submission

Abstract

Large language models (LLMs) demonstrate remarkable performance across a spectrum of languages. In this work, we delve into the question: How do LLMs handle multilingualism? We introduce a framework that depicts LLMs' processing of multilingual inputs: In the first several layers, LLMs understand the question, converting multilingual inputs into English to facilitate the task-solving phase. In the intermediate layers, LLMs engage in problem-solving by thinking in English and incorporating multilingual knowledge to obtain factual content, leveraging the self-attention and feed-forward structures, respectively. In the last several layers, LLMs generate responses that align with the original language of the query. In addition, we investigate the existence of language-specific neurons when processing a certain language. To detect neurons activated by the input language, even without labels, we innovatively design a Parallel Language specific Neuron Detection (`PLND`) method that effectively measures the significance of neurons when handling multilingual inputs. By comprehensive ablation analysis through deactivating neurons of different layers and structures, we verify the framework that we propose. Additionally, we demonstrate that we can utilize such a framework to effectively enhance the multilingual ability with much less training effort.[1]

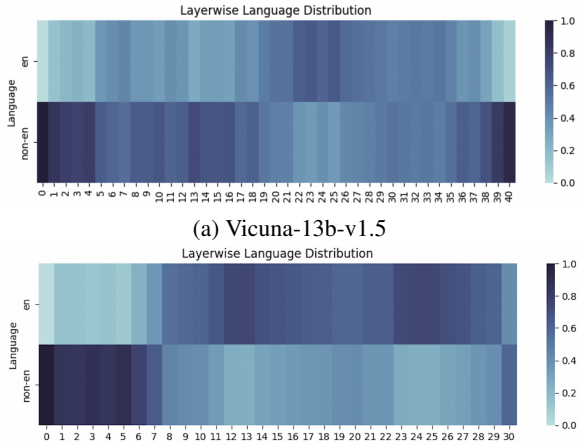
1 Introduction

Recent advancements in large language models (LLMs) such as PaLM (Chowdhery et al., 2022), GPT-4 (OpenAI, 2023), LLaMA (Touvron et al., 2023), and Mistral (Jiang et al., 2023) have dramatically transformed traditional natural language processing (NLP) tasks, integrating seamlessly into daily and professional uses. Thanks to their extensive pretraining on massive corpora mixed with different languages, these models demonstrate remarkable capabilities in understanding and generating text across multiple languages (Huang et al., 2023; Zhu et al., 2023; Zhang et al., 2023; Zhao et al., 2024). Despite their proven effectiveness, the intricate workings of their multilingual processing mechanisms remain largely unclear, which leads to an important research question: *How do large language models handle multilingualism?*

Some studies explore the multilingual capabilities of language models, focusing on their cross-lingual performance or structural commonalities between languages (K et al., 2020; Doddapaneni et al., 2021; Chai et al., 2022). More recently, to understand the relation of certain abilities with specific model architecture especially the dominant Transformer (Vaswani et al., 2017) architecture, studies such as Hou et al. (2023); Stolfo et al. (2023); Friedman et al. (2023) explore the reasoning abilities of LLMs with self-attention layers. While others like Geva et al. (2021); Dai et al. (2022a); Meng et al. (2022) focus on the feed-forward layers, regarding them as key-value memories for storing factual knowledge. However, these works often examine components in isolation and lack a comprehensive framework that encompasses the entire end-to-end process. More importantly, the working mechanism of multilingualism is largely unaddressed in these interpretations.

In this study, to gain an initial understanding of the multilingual mechanisms within LLMs, we analyze the decoded embeddings after each layer when processing inputs in various languages other than English. We then classify these embeddings as corresponding to either English or non-English tokens (see details in Appendix A). As illustrated in Figure 1, the models initially represent non-English user instructions in non-English form. However, as the instructions are processed through the layers of the model, the representation surprisingly becomes English-centric. In the final layers, we observe a reversion to a predominance of non-English embed-

[1] Code will be publicly available.



(a) Vicuna-13b-v1.5

(b) BLOOMZ-7b1

Figure 1: Distribution of English and non-English tokens among layers given non-English instructions.

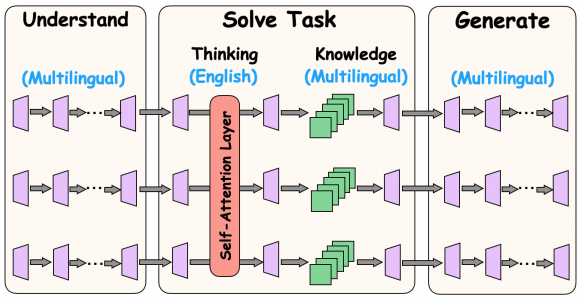


Figure 2: Our proposed multilingual workflow of LLMs.

dings, as the initial instructions are posted in non-English languages. This pattern suggests a complex interplay between languages within LLMs, as they negotiate between instruction fidelity and a possible English-oriented processing bias.

Motivated by these observations, we propose a new framework shown in Figure 2 that conceptualizes the operational stages of LLMs when processing multilingual inputs: In the first several layers, LLMs *understand* the user input and convert the diverse linguistic features into a unified representation. Transitioning into the *task-solving* phase, LLMs engage in solving the tasks by thinking in English and incorporating multilingual knowledge to obtain factual contents, leveraging the self-attention and feed-forward structures, respectively. Finally, models *generate* responses that align with the original language of the query. Such a framework echoes the observations we have above and also some findings in previous studies, and thus reveals the LLMs' ability to shift among languages, allowing them to comprehend, reason, and respond effectively to multilingual tasks.

The subsequent question that arises is whether LLMs engage all available neurons of certain components or only a particular subset when processing a single language, which essentially investigates the presence of "language-specific neurons". To tackle this question, it is crucial to pinpoint which neurons are activated by the input, even without explicit labels of certain tasks. Therefore, we further develop a novel approach called Parallel Language-specific Neuron Detection (`PLND`), which effectively measures the significance of individual neurons in relation to the input provided. With `PLND`, we can identify language-specific neurons by feeding free text corpus of that language into the model and isolating neurons that consistently activate in responses. Experimental results show that by deactivating language-specific neurons which account for only 0.13% of all neurons, LLMs' performance on a summarization task could drop by 99%.

With the proposed `PLND` method, we then conduct extensive experiments to validate our hypothesis illustrated in Figure 2. We selectively disable neuron groups to study their effect on LLM multilingualism. Notably, when deactivating the language-specific neurons in the understanding layer leaves English XQuAD performance stable but decreases non-English performance by 14%. Other tasks including reasoning, knowledge extraction and NLG also exhibit the same characteristics. Moreover, enhancing the multilingual capabilities of LLMs can be achieved by fine-tuning language-specific neurons with merely 200 contextual examples. This method results in a substantial increase in model performance, with a 7.4% relative improvement on the XQuAD and an 8.9% enhancement on XLSum, reducing the size of the required training dataset significantly.

2 Parallel Language-specific Neuron Detection (`PLND`)

In this section, we introduce a neuron detection method called `PLND` for investigating the presence of language-specific neurons.

2.1 Sequential Neuron Detection

To identify neurons responsible for a specific language, it is crucial to discern the significance of a neuron in relation to the inference of a given input. We denote the input of i-th layer in Transformer (Vaswani et al., 2017) as h_i, with the corresponding output represented as $h_{i+1} = T_i(h_i)$, where T_i represents the parameters of the i-th layer.

For a specific neuron, denoted as $N_k^{(i)}$, within the i-th layer—whether located in the attention or feed-forward layer—the importance is quantified as the difference between output when $N_k^{(i)}$ is either activated or deactivated. Formally, it is defined as

$$\text{Imp}(N_k^{(i)}|h_i) = \|T_i \backslash N_k^{(i)}(h_i) - T_i(h_i)\|_2, \quad (1)$$

where $T_i \backslash N_k(\cdot)$ denotes deactivating $N_k^{(i)}$ in T_i. Then, with a set of the corpus in the specific language, denoted as $\mathcal{C} = \{c_1, \cdots, c_l, \cdots, c_n\}$, we can calculate the importance of each neuron in each layer to each corpus. Furthermore, we can select neurons that are important to all corpus in $\mathcal{C}$, i.e.,

$$\text{Imp}(N_k^{(i)}|c_l) \geq \epsilon, \ \forall c_l \in \mathcal{C}, \quad (2)$$

where ϵ is the pre-defined threshold. However, it is super time-consuming to traverse all neurons and all inputs sequentially. Therefore, we need to design an approximating algorithm for acceleration.

2.2 Parallel Neuron Detection

Feed-Forward Layer In Llama2 (Touvron et al., 2023), the FFN(x) is defined as

$$\Big(\text{SiLU}\big(W_{gate}(x)\big) \cdot W_{up}(x)\Big) W_{down}, \quad (3)$$

where $x \in \mathbb{R}^{l \times d_{model}}$, $W_{gate} \in \mathbb{R}^{d_{model} \times d_{inter}}$, $W_{down} \in \mathbb{R}^{d_{inter} \times d_{model}}$. We denote hidden embedding before W_{down} as h_{ffn}. When deactivating the k-th neuron of W_{up},

$$\begin{aligned} &\text{Imp}(W_{up}[:,k]|x) = \|\hat{\text{FFN}}(x) - \text{FFN}(x)\|_2 \\ &= \Big\|\big(h_{\text{ffn}} \cdot \texttt{Mask}[k]\big) W_{down}(x)\Big\|_2, \end{aligned} \quad (4)$$

where $\texttt{Mask}[k]$ is a vector of length d_{inter} with the k-th element as 1 and others as 0. For calculating $\text{Imp}(W_{up}[:,k]|x)$ for all neurons in W_{up} parallelly, we introduce a diagonal mask matrix of size (d_{inter}, d_{inter}), denoted as $\texttt{Mask}$. Therefore,

$$\text{Imp}(W_{up}|x) = \|(h_{\text{ffn}} \cdot \texttt{Mask}) W_{down}(x)\|_2. \quad (5)$$

Furthermore, we find that deactivating the k-th neuron of W_{down} is equivalent to deactivating the k-th neuron in W_{up} as they all set $h_{\text{ffn}}[k] = 0$. Therefore $\text{Imp}(W_{down}|x)$ can be obtain by Equation (5).

Self-Attention Layer For the input x of length l, the self-attention layer is defined as

$$\text{Softmax}\Big(\frac{W_Q(x)W_K^T(x)}{\sqrt{d}}\Big) W_V(x), \quad (6)$$

where $W_Q \in \mathbb{R}^{d_{model} \times d_{mid}}$, $W_K \in \mathbb{R}^{d_{model} \times d_{mid}}$, $W_V \in \mathbb{R}^{d_{model} \times d_{mid}}$.[2] As $W_V(x)$ is a linear layer, $\text{Imp}(W_V|x)$ can be obtained following Equation (5). In the case of W_Q, when deactivating the k-th neuron, $\hat{W}_Q \leftarrow W_Q[:,k] = 0$, we aim to obtain $\text{Imp}(W_Q[:,k]|x)$. Firstly, we calculate the difference in attention weight, i.e., $W_Q(x)W_K^T(x)$.

$$\begin{aligned} \Delta_k &= \hat{W}_Q(x)W_K^T(x) - W_Q(x)W_K^T(x) \\ &= W_Q(x)[:,k]W_K(x)[k,:] \in \mathbb{R}^{l \times l} \end{aligned} \quad (7)$$

Then, the importance of $W_Q[:,k]$ can be defined as

$$\begin{aligned} &\text{Imp}(W_Q[k,:]|x) \\ &\approx \|\hat{\text{attention}}(x) - \text{attention}(x)\|_2 \\ &\approx \Big\|\text{softmax}\Big(\frac{W_Q(x)W_K^T(x) - \Delta_k}{\sqrt{d}}\Big) - \\ &\quad \text{softmax}\Big(\frac{W_Q(x)W_K^T(x)}{\sqrt{d}}\Big)\Big\|_2 \end{aligned} \quad (8)$$

This process can also be calculated parallelly, i.e.,

$$\begin{aligned} \Delta &= \hat{W}_Q(x)W_K^T(x) - W_Q(x)W_K^T(x) \\ &= W_Q(x).resize(l,1,d_{mid}) \times \\ &\quad W_K(x).resize(1,l,d_{mid}) \in \mathbb{R}^{l \times l \times d_{mid}} \end{aligned} \quad (9)$$

Then, the importance of W_Q can be defined as

$$\text{Imp}(W_Q|x) \approx \Big\|\text{softmax}\Big(\frac{W_Q(x)W_K^T(x) - \Delta}{\sqrt{d}}\Big) - \text{softmax}\Big(\frac{W_Q(x)W_K^T(x)}{\sqrt{d}}\Big)\Big\|_2.$$

$\text{Imp}(W_K|x)$ can be calculated the same way.

3 Investigate Language-Specific Neurons

In this section, we apply the PLND method to selected languages and models in order to confirm the existence of language-specific neurons and investigate the relationships between languages.

3.1 Experimental Setup

Models We test two open-source models that perform well on multilingual tasks, including *Vicuna-7b-v1.5* (Chiang et al., 2023) and *Mistral-7b-v1.0* (Jiang et al., 2023). Vicuna-7b-v1.5 is an open-source chat model fine-tuned on top of LlaMa 2 (Touvron et al., 2023) via instruction fine-tuning

[2] In Vicuna and Mistral, $d_{model} = d_{mid}$, but we use different notations to avoid ambiguity.

and has shown good multilingual instruction following ability.[3] Mistral-7b-v1.0 is fine-tuned on Mistral-7b. For simplicity, we abbreviate them as Vicuna and Mistral hereafter to represent the two models respectively.

Languages We adopt 6 languages including English (En), German (De), French (Fr), Chinese (Zh), Spanish (Es), and Russian (Ru). These languages were selected because their training corpora represent over 0.1% of the total pre-training data for Llama2. Furthermore, both Vicuna and Mistral are designed to tackle problems across these languages effectively, and their performance in these languages is much higher than in other low-resource languages such as Thai, Swahili, and Greek.

Corpus To compile a language-specific corpus without task-specific considerations, we utilize OSCAR (Caswell et al., 2020) corpus which contains web crawling texts for each language. Our selection criterion for the number of contexts is based on achieving substantial coverage of each language's vocabulary, ensuring that the selected contexts provide a representative sample of the language. Details are shown in Table 1, where "corpus size" indicates the number of contexts selected, "corpus vocab" represents the vocabulary coverage within the selected contexts, "vocab size" refers to the number of vocabularies of that language.

Language	En	De	Fr	Zh	Es	Ru
Corpus Size	180k	30k	50k	20k	20k	20k
Corpus Vocab	249k	154k	134k	198k	90k	144k
Vocab Size	273k	148k	135k	329k	93k	150k

Table 1: Corpus details across languages are tailored to encompass the majority of each language's vocabulary.

3.2 Existence of Language-Specific Neurons

Using PLND, we feed a corpus in a specific language to LLMs and identify the neurons that are consistently activated. These language-specific neurons are responsible for processing queries in that language. To ascertain whether these neurons are genuinely language-specific, we assess the performance of LLMs in corresponding languages when these neurons are deactivated versus when the same number of randomly sampled neurons is deactivated. We utilize the summarization task with XLSum dataset (Hasan et al., 2021) to showcase the capabilities of LLMs, aiming to demonstrate that the resulting output can be devoid of meaning.

	Method	Fr	Zh	Es	Ru	Avg.
Vicuna	Original	14.2	61.1	10.4	20.8	26.6
	Deact-Rand.	14.1	61.6	10.4	20.8	26.7
	Deact-Lang.	**0.83**	**0.00**	**0.24**	**0.42**	**0.37**
Mistral	Original	15.2	56.4	10.6	21.0	25.8
	Deact-Rand.	15.4	55.9	10.2	21.2	25.7
	Deact-Lang.	**0.21**	**0.39**	**0.15**	**0.07**	**0.21**

Table 2: Multilingual performance of LLMs upon deactivating neurons, where "Lang." denotes language-specific neurons and "Rand." indicates an equivalent number of neurons selected at random.

Table 2 demonstrates the decline of multilingual capabilities when deactivating language-specific neurons. Although just deactivating around 0.13% neurons, LLMs lose multilingual capabilities including understanding multilingual inputs, handling multilingual questions, and generating multilingual outputs. In contrast, deactivating the same number of randomly selected neurons does not yield any difference. Therefore, the detected neurons are language-specific and related to handling corresponding multilingual inputs.

3.3 Interrelation of Language-Specific Neurons Across Languages

Using neurons identified by PLND, we investigate the relationships of two languages via the degree of overlap between their language-specific neurons:

$$\text{overlap}(x, y) = \frac{|\mathcal{N}_x \cap \mathcal{N}_y|}{|\mathcal{N}_y|}, \qquad (10)$$

where $\mathcal{N}_{language}$ represents the set of detected language-specific neurons. Figure 3 shows the neuron overlapping ratio $\text{overlap}(x, y)$ of any two languages in different structures of two models.

We can observe that in both Mistral and Vicuna, the intersection with English from other languages is relatively limited, suggesting that English possesses a predominant number of language-specific neurons. Additionally, there is a pronounced tendency for languages belonging to the same family to demonstrate a higher degree of overlap with each other. Moreover, the feed-forward structure typically exhibits a higher degree of consistency in overlap across various languages, due to the shared world knowledge embedded within the neurons that is accessible to multiple languages.

[3]We do not directly utilize Llama2-chat as it does not follow multilingual instructions, consistently responding in English regardless of the language of the query.

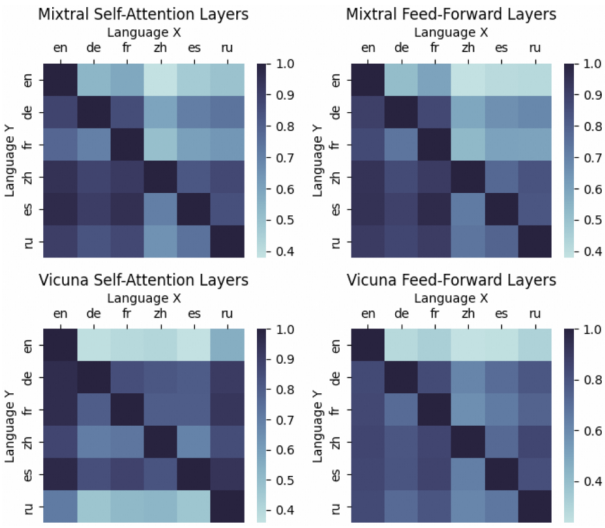


Figure 3: Overlapping ratio of language-specific neurons in self-attention and feed-forward structures.

4 How do LLMs Handle Multilingualism?

In this section, we validate the workflow presented in Figure 2 by systematically deactivating neurons with PLND across different layers and components to examine their impact on a range of tasks.

4.1 Experimental Setup

Dataset To comprehensively understand how LLMs work with different abilities, we employ four different kinds of tasks including MGSM (Shi et al., 2022) for reasoning task, XQuAD (Artetxe et al., 2020) for NLU task, XLSum (Hasan et al., 2021) for NLG task, and X-CSQA (Lin et al., 2021) for knowledge question answering task. For XLSum, we randomly sample 500 data points from the whole test set for each language, while for other tasks, we employ the whole test set.

Original Results We first evaluate the vanilla performance of Vicuna and Mistral on these datasets for later comparison as presented in Table 3.

	Task	En	De	Fr	Zh	Es	Ru
Vicuna	XQuAD	57.5	50.3	–	55.7	55.7	–
	MGSM	20.4	14.8	14.8	12.8	13.2	10.0
	X-CSQA	57.8	43.8	40.1	43.2	44.3	26.0
	XLSum	13.1	–	14.2	61.1	10.4	20.8
Mistral	XQuAD	57.1	48.5	–	64.3	54.1	–
	MGSM	46.0	21.2	26.0	31.6	31.2	21.6
	X-CSQA	61.7	40.0	40.4	47.1	45.7	14.1
	XLSum	13.5	–	15.2	56.4	10.6	21.0

Table 3: Assessing the baseline performance of Vicuna and Mistral across four representative multilingual tasks in selected languages.

Experiment Details For reasoning and NLU tasks, we adopt accuracy as the metric. As for the NLG tasks, we adopt ROUGE-L as the metric. Furthermore, for the concrete numbers of different layers, we tune hyperparameters by XQuAD in Chinese. Details are explained in Appendix B.

Notations Tables 4 to 7 present the results of deactivating certain neurons, where "Under" denotes the understanding layers, "S-ATTN" and "S-FFN" correspond to the self-attention and the feed-forward within the task-solving layers respectively, "Gen" refers to the generation layers. The term "Random" is used to describe deactivating randomly chosen neurons, whereas "Lang-Spec" refers to the deactivation of language-specific neurons. We also present the gap between the original performance and performance after deactivation for English (Δ_{Eng}) and averaged non-English languages ($\Delta_{\text{n-Eng}}$), respectively. A single metric Δ is then introduced as $\Delta_{\text{Eng}} - \Delta_{\text{n-Eng}}$, where a high value indicates such deactivation operation does not bring much impact to the English performance but lead to performance drop in non-English.

4.2 Analysis on Understanding

Deactivating Method Table 4 shows the results of the understanding task following the deactivation of five distinct sets of neurons: (i) neurons randomly selected from the understanding layers; (ii) neurons randomly chosen across all layers; (iii) language-specific neurons within the understanding layers; (iv) language-specific neurons in the task-solving layers; (v) language-specific neurons in the generation layers. For a fair comparison, we ensure the numbers of deactivated neurons in all settings are the same. Full results of each language are listed in Appendix C.

Findings We find that by deactivating randomly sampled neurons, no matter in the understanding layer or all layers, the performance of LLMs in both English and non-English languages is almost unaffected compared to other deactivating methods. Note that in some cases, deactivating randomly sampled neurons may even increase the performance because irrelevant neurons are removed, which also aligns with the finding from Sharma et al. (2023). When assessing the differential impact on English and non-English language performance post-deactivation, specifically the difference calculated as $\Delta_{\text{Eng}} - \Delta_{\text{n-Eng}}$, it is evident that the

Model	Deactivating Method					Performance				
	Under	S-ATTN	S-FFN	Gen	Neuron	Eng	n-Eng	Δ_{Eng}	$\Delta_{\text{n-Eng}}$	$\Delta\uparrow$
Vicuna	✓	✗	✗	✗	Random	57.8	53.9	+0.3	−0.1	+0.4
	✓	✓	✓	✓	Random	57.9	54.2	+0.4	+0.3	+0.1
	✓	✗	✗	✗	Lang-Spec	56.5	46.0	−0.5	−7.9	+7.4
	✗	✓	✓	✗	Lang-Spec	40.9	38.6	−15.9	−15.3	−0.6
	✗	✗	✗	✓	Lang-Spec	57.9	52.8	−0.4	−1.1	+0.7
Mistral	✓	✗	✗	✗	Random	58.1	55.5	+1.0	−0.2	+1.2
	✓	✓	✓	✓	Random	57.6	55.5	+0.5	−0.2	+0.7
	✓	✗	✗	✗	Lang-Spec	56.2	48.3	−0.9	−7.4	+6.5
	✗	✓	✓	✗	Lang-Spec	53.2	47.0	−3.9	−8.7	+4.8
	✗	✗	✗	✓	Lang-Spec	56.4	54.6	−0.7	−1.0	+0.3

Table 4: Results of the understanding task, where '✓' indicates that neurons in the corresponding layer are deactivated, and '✗' signifies they are not. Δ is defined as the difference between the reduction in performance in English, denoted as Δ_{Eng}, and the reduction in performance in non-English languages, denoted as $\Delta_{\text{n-Eng}}$.

Model	Deactivating Method					Performance				
	Under	S-ATTN	S-FFN	Gen	Neuron	Eng	n-Eng	Δ_{Eng}	$\Delta_{\text{n-Eng}}$	$\Delta\uparrow$
Vicuna	✗	✓	✗	✗	Random	20.0	11.3	−0.4	−1.8	+1.4
	✗	✓	✓	✗	Random	18.4	12.2	−2.0	−1.0	−1.0
	✓	✓	✓	✓	Random	19.6	12.5	−0.8	−0.7	−0.1
	✗	✓	✓	✗	Lang-Spec	7.2	3.4	−13.2	−9.8	−3.4
	✓	✗	✗	✓	Lang-Spec	18.1	8.3	−2.3	−4.9	+2.6
	✓	✗	✓	✓	Lang-Spec	19.0	7.8	−1.4	−5.4	+4.0
Mistral	✗	✓	✗	✗	Random	40.8	23.4	−5.2	−2.9	−2.3
	✗	✓	✓	✗	Random	39.2	24.0	−6.8	−2.3	−4.5
	✓	✓	✓	✓	Random	45.2	26.8	−0.8	+0.5	−1.3
	✗	✓	✓	✗	Lang-Spec	38.2	18.4	−7.8	−7.9	+0.1
	✓	✗	✗	✓	Lang-Spec	44.0	18.1	−2.0	−8.2	+6.2
	✓	✗	✓	✓	Lang-Spec	46.2	18.3	+0.2	−8.0	+8.2

Table 5: Results of the reasoning task. The highest performance reduction difference (Δ) is achieved by disabling all language-specific neurons, except for those involved in self-attention structure within the task-solving layer.

deactivation of random neurons within the understanding layer amplifies this effect. This observation lends partial support to the hypothesized role of the understanding layer in language processing.

Furthermore, we find that deactivating language-specific neurons in the understanding layer influences the performance in English a little while significantly decreasing the performance in non-English languages. When deactivating language-specific neurons in the task-solving layer, both English and non-English languages are significantly reduced while deactivating language-specific neurons in the generation layer influences a little for both English and non-English languages. Therefore, we prove that the first several layers are responsible for understanding because deactivated neurons just disable LLMs on the NLU task in non-English languages. Furthermore, disabling language-specific neurons in the task-solving layer shows that LLMs rely on English, as their performance drops across all languages.

4.3 Analysis on Reasoning

Deactivating Method Table 5 shows the result of the reasoning task, where we deactivate 6 sets of neurons. Details are listed in Appendix C.

Findings We find that deactivating randomly sampled neurons in task-solving layers disables the capabilities of LLMs in reasoning to a greater extent than deactivating randomly sampled neurons in all layers, which verifies the function of the task-solving layer. Furthermore, comparing three deactivating language-specific neuron methods, we find that deactivating the task-solving layer decreases performance in both English and non-English. On the contrary, when we only deactivate language-specific neurons not in the task-solving layer, non-English is influenced more seriously than English. Moreover, eliminating interference from the feed-forward layer achieves better results, which verifies the function of attention structure in the task-solving layer.

Model	Deactivating Method					Performance				
	Under	S-ATTN	S-FFN	Gen	Neuron	Eng	n-Eng	Δ_{Eng}	$\Delta_{\text{n-Eng}}$	$\Delta\uparrow$
Vicuna	✗	✗	✓	✗	Random	57.5	39.5	−0.3	+0.0	−0.3
	✗	✓	✓	✗	Random	56.0	38.7	−1.8	−0.8	−1.0
	✓	✓	✓	✓	Random	57.7	39.6	−0.1	+0.1	−0.2
	✗	✓	✗	✗	Lang-Spec	33.7	30.3	−24.1	−9.2	−14.9
	✗	✗	✓	✗	Lang-Spec	57.5	37.5	−0.3	−2.0	+1.7
Mistral	✗	✗	✓	✗	Random	61.0	37.0	−0.3	−0.5	+0.2
	✗	✓	✓	✗	Random	60.7	36.3	−0.6	−1.2	+0.6
	✓	✓	✓	✓	Random	61.8	37.4	+0.1	−0.1	+0.2
	✗	✓	✗	✗	Lang-Spec	51.2	28.9	−10.1	−8.6	−1.5
	✗	✗	✓	✗	Lang-Spec	61.2	35.1	−0.1	−2.4	+2.3

Table 6: Results of the knowledge extraction. The highest performance reduction difference (Δ) is achieved by disabling all language-specific neurons in the feed-forward structure within the task-solving layer.

Model	Deactivating Method					Performance				
	Under	S-ATTN	S-FFN	Gen	Neuron	Eng	n-Eng	Δ_{Eng}	$\Delta_{\text{n-Eng}}$	$\Delta\uparrow$
Vicuna	✗	✗	✗	✓	Random	13.2	26.8	+0.1	+0.1	+0.0
	✓	✓	✓	✓	Random	13.0	26.7	−0.1	+0.0	−0.1
	✗	✗	✗	✓	Lang-Spec	13.1	25.7	+0.0	−1.1	+1.1
Mistral	✗	✗	✗	✓	Random	13.6	25.9	+0.1	+0.1	+0.0
	✓	✓	✓	✓	Random	13.6	25.7	+0.1	−0.2	+0.3
	✗	✗	✗	✓	Lang-Spec	13.8	24.3	+0.3	−1.5	+1.8

Table 7: Results of the generation task following neuron deactivation. The highest performance reduction difference (Δ) is achieved by disabling all language-specific neurons in the generation layer.

4.4 Analysis on Knowledge Extraction

Deactivating Method Table 6 shows the result of the knowledge extraction task, where we deactivate 5 sets of neurons. Details are listed in Appendix C.

Findings Likewise, the deactivation of randomly selected neurons has a lesser impact compared to language-specific neurons, validating the efficiency of `PLND` in identifying neurons pertinent to a particular language. Targeted deactivation of language-specific neurons within the feed-forward structure of the task-solving layer predominantly affects non-English languages. This implies that processing multilingual queries necessitates accessing the multilingual information embedded within the relevant structures. However, disabling the self-attention structure compromises the ability to solve tasks across all languages.

4.5 Analysis on Generation

Deactivating Method Table 7 shows the result of the knowledge extraction task, where we deactivate 3 sets of neurons. Details are listed in Appendix C.

Findings Similar to other tasks, the disabling of language-specific neurons within the generation layer attenuates their capacity to generate content in the respective languages. By selectively deactivating neurons that are not associated with English, we do not completely eliminate the models' multilingual generation abilities. However, as demonstrated in Table 2, the complete deactivation of all language-specific neurons results in the total loss of the LLMs' multilingual generation capabilities.

5 Further Analysis

5.1 Enhance Multilingual Ability

We have verified the proposed framework for explaining the multilingual working mechanism of LLMs in the above section via deactivating certain neurons. While opposite to employing deactivation, we can also enhance their multilingual ability, especially the understanding and generating ability, by fine-tuning these language-specific neurons.

We employ Llama2-7b-base model for enhancement to eliminate the interference of instruction fine-tuning. We select causal language modeling as our fine-tuning task and create a dataset comprising 200 documents for each language, extracted from the Wikipedia corpus.[4] It is important to note that our enhancements are focused on augmenting the model's capabilities in understanding and genera-

[4] https://dumps.wikimedia.org/

tion only; we do not extend its reasoning faculties or broaden its knowledge base as it may require more specific data preparation. Accordingly, we evaluate the efficacy of our enhancements through targeted understanding and generation tasks. Detailed experiment results are shown in Table 8.

Our findings indicate that with just 10 minutes of fine-tuning on 200 contexts, LLMs exhibit significant enhancements in multilingual understanding and generation abilities. Notably, there is a relative performance boost of 7.4% on the XQuAD benchmark. Similarly, for XLSum, we observe an 8.9% relative improvement in performance.

Task	Method	De	Fr	Zh	Es	Ru
XQuAD	Original	42.2	–	47.3	39.6	–
	Enhance	**45.3**	–	**49.7**	**43.7**	–
XLSum	Original	–	8.5	48.2	9.1	19.7
	Enhance	–	**9.4**	**51.6**	**11.0**	**21.1**

Table 8: Enhancement is achieved by fine-tuning the Llama2-7b-base model through a causal language modeling task, utilizing 200 contexts from each language.

5.2 Analysis on Different Multilingual LLMs

To further verify our proposed framework, we examine two more types of multilingual LLMs, including BLOOMZ (Muennighoff et al., 2022), a *hyper-multilingual* LLM claiming to support 46 languages, and Chinese Llama (Cui et al., 2023), a *bilingual* LLM focusing on English and Chinese.

Hyper-Multilingual LLMs From Figure 1b we observe that BLOOMZ also tends to default to English as its "thinking" language. Therfore, it can support multiple languages mainly because it has the ability to understand and generate these languages while using English as the language of thinking. Additionally, Figure 4 illustrates the degree of neuron overlap among languages within both the attention and feed-forward structures of BLOOMZ. In contrast to the findings shown in Figure 3, there is a marked reduction in overlap, indicating that individual languages maintain a higher degree of independence and do not extensively share neurons with one another.

Bilingual LLMs We employ Chinese Llama (Cui et al., 2023), which extends existing vocabulary and incorporate secondary pre-training using Chinese data and fine-tune the model with Chinese instruction datasets. However, this intensive training can lead to a degradation in performance for languages other than Chinese. As depicted in Figure 5, Chinese predominates as the primary language for thinking processing and knowledge extraction across all languages. Consequently, the absence of language-specific neurons results in the transformation of it into a Chinese-centric LLM.

6 Related Work

Interpretability Conventional interpretability research investigates the significance of input features with their corresponding outputs (Vig, 2019; Hewitt and Liang, 2019; Qiu et al., 2020).In the era of LLMs, one brunch of work includes efforts to understand knowledge storage, with Geva et al. (2021) initiating the study of the feed-forward layer as a knowledge base. Subsequent work has furthered this by altering neuron values (Dai et al., 2022b), mapping embeddings to words (Geva et al., 2022), modifying inputs to recover embeddings (Meng et al., 2022), and analyzing attention heads (Li et al., 2023a). Another line of research centers on the self-attention layer, examining its connection to reasoning (Hou et al., 2023; Stolfo et al., 2023; Friedman et al., 2023) by contrasting the reasoning tree based on attention weights.

Multilingualism Various studies have been undertaken to construct benchmarks (Zhang et al., 2023), enhance performance through translation (Liang et al., 2023; Huang et al., 2023), aligning representations (Nguyen et al., 2023; Salesky et al., 2023), prompting (Li et al., 2023b; Tanwar et al., 2023). These efforts underscore the importance and complexity of enabling LLMs to operate effectively across multiple languages.

7 Conclusion

In this work, we introduce a hypothesis suggesting that LLMs address multilingualism by first translating queries into English, processing them using English with the help of multilingual knowledge, and then translating the responses back into the original language. To validate this framework, we introduce a novel method for detecting language-specific neurons and conduct extensive ablation studies. These studies involve selectively deactivating various neuron sets to observe their impact on the multilingual capabilities of LLMs. Additionally, we refine the multilingual performance of LLMs by fine-tuning these language-specific neurons, which only take up a small part of the neurons.

Limitations

Our experiments are mainly conducted on models with a size of approximately 7 billion parameters (i.e., vicuna-7b-v1.5 and mistral-7b-v1.0), primarily due to computational resource constraints. Testing our methods on larger models could potentially yield additional insights, particularly in understanding the scalability of our proposed framework.

Furthermore, our exploration into enhancing the multilingual abilities of language models through our framework was preliminary. Expanding our experiments to include a broader array of languages, especially those considered low-resource, could better demonstrate the effectiveness of our proposed framework. Moreover, extending the scope of our experiments to evaluate other capabilities such as reasoning and multilingual knowledge extraction with specific datasets could provide a more comprehensive picture of the potential benefits of our approach.

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*, pages 4623–4637.

Isaac Caswell, Theresa Breiner, Daan van Esch, and Ankur Bapna. 2020. Language id in the wild: Unexpected challenges on the path to a thousand-language web text corpus. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6588–6608.

Yuan Chai, Yaobo Liang, and Nan Duan. 2022. Cross-lingual ability of multilingual masked language models: A study of language structure. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022*, pages 4702–4712.

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.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Yiming Cui, Ziqing Yang, and Xin Yao. 2023. Efficient and effective text encoding for chinese llama and alpaca. *arXiv preprint arXiv:2304.08177*.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022a. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502, Dublin, Ireland. Association for Computational Linguistics.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022b. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502.

Sumanth Doddapaneni, Gowtham Ramesh, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh M. Khapra. 2021. A primer on pretrained multilingual language models. *CoRR*, abs/2107.00676.

Dan Friedman, Andrew Lampinen, Lucas Dixon, Danqi Chen, and Asma Ghandeharioun. 2023. Interpretability illusions in the generalization of simplified models.

Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. 2022. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 30–45.

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

Tahmid Hasan, Abhik Bhattacharjee, Md Saiful Islam, Kazi Mubasshir, Yuan-Fang Li, Yong-Bin Kang, M Sohel Rahman, and Rifat Shahriyar. 2021. Xl-sum: Large-scale multilingual abstractive summarization for 44 languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4693–4703.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. *arXiv preprint arXiv:1909.03368*.

Yifan Hou, Jiaoda Li, Yu Fei, Alessandro Stolfo, Wangchunshu Zhou, Guangtao Zeng, Antoine Bosselut, and Mrinmaya Sachan. 2023. Towards a mechanistic interpretation of multi-step reasoning capabilities of language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4902–4919, Singapore. Association for Computational Linguistics.

Haoyang Huang, Tianyi Tang, Dongdong Zhang, Xin Zhao, Ting Song, Yan Xia, and Furu Wei. 2023. Not

all languages are created equal in LLMs: Improving multilingual capability by cross-lingual-thought prompting. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12365–12394, Singapore. Association for Computational Linguistics.

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

Karthikeyan K, Zihan Wang, Stephen Mayhew, and Dan Roth. 2020. Cross-lingual ability of multilingual BERT: an empirical study. In *8th International Conference on Learning Representations, ICLR 2020*.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023a. Inference-time intervention: Eliciting truthful answers from a language model. *arXiv preprint arXiv:2306.03341*.

Shuang Li, Xuming Hu, Aiwei Liu, Yawen Yang, Fukun Ma, Philip S Yu, and Lijie Wen. 2023b. Enhancing cross-lingual natural language inference by soft prompting with multilingual verbalizer. *arXiv preprint arXiv:2305.12761*.

Yaobo Liang, Quanzhi Zhu, Junhe Zhao, and Nan Duan. 2023. Machine-created universal language for cross-lingual transfer. *arXiv preprint arXiv:2305.13071*.

Bill Yuchen Lin, Seyeon Lee, Xiaoyang Qiao, and Xiang Ren. 2021. Common sense beyond english: Evaluating and improving multilingual language models for commonsense reasoning. 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 1274–1287.

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

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, et al. 2022. Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786*.

Hoang H Nguyen, Chenwei Zhang, Tao Zhang, Eugene Rohrbaugh, and Philip S Yu. 2023. Enhancing cross-lingual transfer via phonemic transcription integration. *arXiv preprint arXiv:2307.04361*.

OpenAI. 2023. Gpt-4 technical report.

Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10):1872–1897.

Elizabeth Salesky, Neha Verma, Philipp Koehn, and Matt Post. 2023. Pixel representations for multilingual translation and data-efficient cross-lingual transfer. *arXiv preprint arXiv:2305.14280*.

Pratyusha Sharma, Jordan T Ash, and Dipendra Misra. 2023. The truth is in there: Improving reasoning in language models with layer-selective rank reduction. *arXiv preprint arXiv:2312.13558*.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, et al. 2022. Language models are multilingual chain-of-thought reasoners. In *The Eleventh International Conference on Learning Representations*.

Alessandro Stolfo, Yonatan Belinkov, and Mrinmaya Sachan. 2023. A mechanistic interpretation of arithmetic reasoning in language models using causal mediation analysis. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7035–7052, Singapore. Association for Computational Linguistics.

Eshaan Tanwar, Manish Borthakur, Subhabrata Dutta, and Tanmoy Chakraborty. 2023. Multilingual llms are better cross-lingual in-context learners with alignment. *arXiv preprint arXiv:2305.05940*.

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

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

Jesse Vig. 2019. A multiscale visualization of attention in the transformer model.

Wenxuan Zhang, Sharifah Mahani Aljunied, Chang Gao, Yew Ken Chia, and Lidong Bing. 2023. M3exam: A multilingual, multimodal, multilevel benchmark for examining large language models. *CoRR*, abs/2306.05179.

Jun Zhao, Zhihao Zhang, Luhui Gao, Qi Zhang, Tao Gui, and Xuanjing Huang. 2024. Llama beyond english: An empirical study on language capability transfer.

Wenhao Zhu, Yunzhe Lv, Qingxiu Dong, Fei Yuan, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2023. Extrapolating large language models to non-english by aligning languages.

A English and Non-English Tokens

We employ `cld3` package to detect the language of each token in the embeddings of

each layer, which is a language detection library based on the Compact Language Detector 3 model developed by Google. Furthermore, if the detection result is reliable, i.e., cld3.get_language(token).is_reliable $== True$, we adopt the detection results, otherwise the token is categorized as a non-word.

B Hyper-parameters

We adopt the performance on XQuAD in Chinese as the validation set to all languages and all tasks. Specifically, Table 9 shows the result on Vicuna when deactivating language-specific neurons in the understanding layer ($D_{\mathcal{U}}$) and generation layer ($D_{\mathcal{G}}$), where N_1 is the number of understanding layers and N_2 is the number of generation layer. We find that when setting $N_1 = 8$ and $N_2 = 2$, performance in English is influenced the least while performance in Chinese decreases the most. As for Mistral, the number is $N_1 = 6$ and $N_2 = 3$.

Method	$D_{\mathcal{U}}$		$D_{\mathcal{G}}$	
	N_1	ACC	N_2	ACC
En-Vanilla		57.5		
Zh-Vanilla		55.5		
En-Deact	**8**	**57.7** (↑ 0.2)	4	54.7 (↓ 2.8)
Zh-D-Deact		**44.9** (↓ 10.6)		54.6 (↓ 0.9)
En-Deact	6	58.6 (↑ 1.1)	3	57.7 (↑ 0.2)
Zh-Deact		55.1 (↓ 0.4)		54.5 (↓ 1.0)
En-Deact	4	57.3 (↓ 0.2)	**2**	**58.4** (↑ 0.9)
Zh-Deact		53.9 (↓ 1.6)		**54.1** (↓ 1.4)

Table 9: XQuAD with Chinese on Vicuna-7b-v1.5.

Method	$D_{\mathcal{U}}$		$D_{\mathcal{G}}$	
	N_1	ACC	N_2	ACC
En-Vanilla		57.1		
Zh-Vanilla		64.3		
En-Deact	8	53.3 (↓ 3.8)	4	55.8 (↓ 1.3)
Zh-Deact		52.6 (↓ 11.7)		62.9 (↓ 1.4)
En-Deact	**6**	**56.8** (↓ 0.3)	**3**	**56.3** (↓ 0.8)
Zh-Deact		**54.9** (↓ 9.4)		**62.7** (↓ 1.6)
En-Deact	4	57.6 (↑ 0.5)	2	55.7 (↓ 1.4)
Zh-Deact		61.8 (↓ 2.5)		63.8 (↓ 0.5)

Table 10: XQuAD with Chinese on Mistral-7b-v1.0.

C Detailed Experiment Results

C.1 Detailed Experiment Settings

Reasoning Task Deactivation methods: (i) randomly sampled neurons in the attention structure of task-solving layer. (ii) randomly sampled neurons in the task-solving layer. (iii) randomly sampled neurons in all layers. (iv) language-specific neurons in the task-solving layer. (v) language-specific neurons in the understanding layer and generation layer. (vi) language-specific neurons not in the attention structure of task-solving layers.

Knowledge Extraction Task Deactivation methods: (i) randomly sampled neurons in the feed-forward structure of task-solving layers. (ii) randomly sampled neurons in the task-solving layer. (iii) randomly sampled neurons in all layers. (iv) language-specific neurons in the attention structure of task-solving layers. (v) language-specific neurons in the feed-forward structure of task-solving layers.

Generation Task Deactivation methods: (i) randomly sampled neurons in the generating layers. (ii) randomly sampled neurons in all layers. (iv) language-specific neurons in the generating layers.

C.2 Detailed Result

Due to the limited space, we employ a more concise notation. We denote deactivating neurons in the self-attention layer of the i-th layer as $D_i^{(A)}$, while deactivating neurons in the feed-forward layer of the i-th layer is denoted as $D_i^{(F)}$. We denote $\mathcal{U} = \{1, \cdots, N_1\}$ as the set of layers that take charge of understanding as shown in Figure 2. Similarly, we denote $\mathcal{S} = \{N_1 + 1, \cdots, N_2\}$ as the set of layers that take charge of task solving and $\mathcal{G} = \{N_2 + 1, \cdots, 32\}$ as the set of layers that take charge of generation[5]. Furthermore, $D_{\mathcal{U}}^{(A)}$ represents deactivating neurons in self-attention layers of $\mathcal{U}$. Similarly, we introduce $D_{\mathcal{U}}^{(F)}$, $D_{\mathcal{S}}^{(A)}$, $D_{\mathcal{S}}^{(F)}$, $D_{\mathcal{G}}^{(A)}$ and $D_{\mathcal{G}}^{(A)}$.

D Hyper-Multilingual LLMs

We present the interrelation of language-specific neurons of BLOOMZ in Figure 4.

E Bilingual LLMs

We present the distribution of languages among layers in Chinese Llama in Figure 5.

[5]Vicuna-7b-v1.5 and Mistral-7b-v1.0 both have 32 layers.

	Method	German				Chinese				Spanish			
		En-D	De-D	$\Delta_{\text{En-D}}$	$\Delta_{\text{De-D}}$	En-D	Zh-D	$\Delta_{\text{En-D}}$	$\Delta_{\text{Zh-D}}$	En-D	Es-D	$\Delta_{\text{Es-D}}$	$\Delta_{\text{Es-D}}$
Vicuna	$D^R_{\mathcal{U}}$	57.8	49.7	+0.3	−0.6	57.8	55.8	+0.3	+0.1	57.8	56.1	+0.3	+0.4
	D^R_{All}	57.9	50.8	+0.4	+0.5	57.9	55.8	+0.4	+0.1	57.9	55.9	+0.4	+0.2
	$D_{\mathcal{U}}$	55.7	40.7	−2.0	−9.6	57.7	44.9	+2.0	−10.8	56.1	52.4	−1.4	−3.2
	$D_{\mathcal{S}}$	48.3	41.7	−7.2	−8.6	45.0	45.4	−12.5	−10.3	29.5	28.6	−28.0	−27.1
	$D_{\mathcal{G}}$	57.5	50.1	0.0	−0.2	58.4	54.1	+0.9	−1.6	57.7	54.1	+0.2	−1.6
Mistral	$D^R_{\mathcal{U}}$	58.1	48.2	+1.0	−0.4	58.1	63.9	+1.0	−0.4	58.1	54.3	+1.0	+0.2
	D^R_{All}	57.6	48.3	+0.5	−0.3	57.6	63.6	+0.5	−0.7	57.6	54.5	+0.5	+0.4
	$D_{\mathcal{U}}$	56.5	42.4	−0.6	−6.2	56.8	54.9	−0.3	−9.4	55.4	47.5	−1.7	−6.6
	$D_{\mathcal{S}}$	54.3	43.2	−2.8	−5.4	54.9	52.9	−2.2	−11.4	50.3	44.9	−6.8	−9.2
	$D_{\mathcal{G}}$	56.7	47.9	−0.4	−0.7	56.3	62.7	−0.8	−1.6	56.2	53.2	−0.9	−0.8

Table 11: Understanding task.

	Method	German				French				Chinese				Spanish				Russian			
		En-D	De-D	$\Delta_{\text{En-D}}$	$\Delta_{\text{De-D}}$	En-D	Fr-D	$\Delta_{\text{En-D}}$	$\Delta_{\text{Fr-D}}$	En-D	Zh-D	$\Delta_{\text{En-D}}$	$\Delta_{\text{Zh-D}}$	En-D	Es-D	$\Delta_{\text{Es-D}}$	$\Delta_{\text{Es-D}}$	En-D	Ru-D	$\Delta_{\text{En-D}}$	$\Delta_{\text{Ru-D}}$
Vicuna	$D^R_{\mathcal{S}(A)}$	20.0	12.4	−0.4	−2.4	20.0	13.6	−0.4	−1.2	20.0	13.2	−0.4	+0.4	20.0	12.4	−0.4	−0.8	20.0	4.8	−0.4	−5.2
	$D^R_{\mathcal{S}}$	18.4	12.4	−2.0	−2.4	18.4	14.0	−2.0	−0.8	18.4	14.4	−2.0	+1.6	18.4	15.2	−2.0	+2.0	18.4	4.8	−2.0	−5.2
	D^R_{All}	19.6	14.0	−0.8	−0.8	19.6	13.8	−0.8	−1.0	19.6	14.8	−0.8	+2.0	19.6	12.4	−0.8	−0.8	19.6	7.6	−0.8	−2.4
	$D_{\mathcal{S}}$	3.6	2.0	−16.8	−12.8	8.4	3.2	−12.0	−11.6	4.8	4.0	−15.6	−8.8	8.8	4.0	−11.6	−9.2	10.4	4.0	−10.0	−6.0
	$D_{\mathcal{U}\&\mathcal{G}}$	16.4	5.6	−4.0	−9.2	19.2	9.6	−1.2	−5.2	20.0	9.2	−0.4	−3.6	17.6	11.6	−2.8	−1.6	17.2	5.6	−3.2	−4.4
	$\bar{D}_{\mathcal{S}(A)}$	16.8	4.4	−3.6	−10.4	19.6	8.8	−0.8	−4.4	21.6	9.6	+1.2	−3.2	19.6	10.4	−0.8	−2.8	17.2	5.6	−3.2	−4.4
Mistral	$D^R_{\mathcal{S}(A)}$	40.8	18.0	−5.2	−3.2	40.8	25.6	−5.2	−0.4	40.8	24.0	−5.2	−7.6	40.8	29.2	−5.2	−2.0	40.8	20.4	−5.2	−1.2
	$D^R_{\mathcal{S}}$	39.2	20.0	−6.8	−1.2	39.2	25.2	−6.8	−0.8	39.2	25.6	−6.8	−6.0	39.2	29.6	−6.8	−1.6	39.2	19.6	−6.8	−2.0
	D^R_{All}	45.2	24.0	−0.8	+2.8	45.2	27.6	−0.8	+1.6	45.2	31.2	−0.8	−0.4	45.2	30.4	−0.8	−0.8	45.2	20.8	−0.8	−0.8
	$D_{\mathcal{S}}$	38.4	12.0	−7.6	−9.2	40.8	24.8	−5.2	−1.2	37.9	19.6	−8.1	−12.0	40.4	24.4	−5.6	−6.8	33.6	11.2	−12.4	−10.4
	$D_{\mathcal{U}\&\mathcal{G}}$	42.4	9.2	−3.6	−12.0	41.2	21.6	−4.8	−4.4	46.4	19.6	+0.4	−12.0	44.0	28.0	−2.0	−3.2	46.0	12.0	+0.0	−9.6
	$\bar{D}_{\mathcal{S}(A)}$	43.6	9.6	−2.4	−11.6	44.8	19.2	−1.2	−6.8	46.4	18.8	+0.4	−12.8	47.6	27.6	+1.6	−3.6	48.4	16.4	+2.4	−5.2

Table 12: Reasoning task.

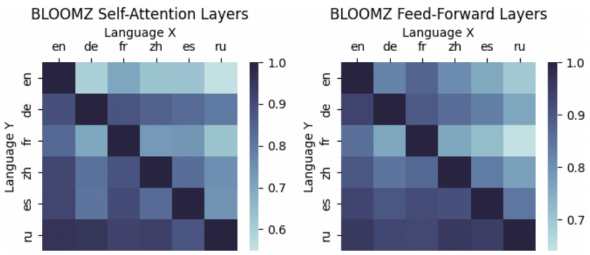


Figure 4: Interrelation of language-specific neurons in BLOOMZ.

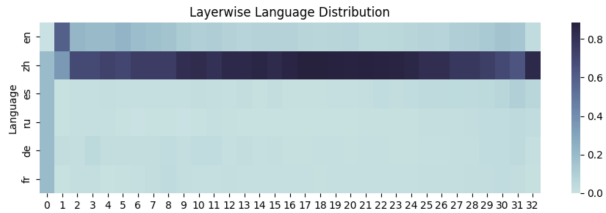


Figure 5: Distribution of languages among layers in Chinese Llama.

	Method	German				French				Chinese				Spanish				Russian			
		En-D	De-D	$\Delta_{\text{En-D}}$	$\Delta_{\text{De-D}}$	En-D	Fr-D	$\Delta_{\text{En-D}}$	$\Delta_{\text{Fr-D}}$	En-D	Zh-D	$\Delta_{\text{En-D}}$	$\Delta_{\text{Zh-D}}$	En-D	Es-D	$\Delta_{\text{Es-D}}$	$\Delta_{\text{Es-D}}$	En-D	Ru-D	$\Delta_{\text{En-D}}$	$\Delta_{\text{Ru-D}}$
Vicuna	$D^R_{\mathcal{S}(F)}$	57.5	43.8	−0.3	+0.0	57.5	40.3	−0.3	+0.2	57.5	43.2	−0.3	+0.0	57.5	44.6	−0.3	+0.3	57.5	25.5	−0.3	−0.5
	$D^R_{\mathcal{S}}$	56.0	44.0	−1.8	+0.2	56.0	38.6	−1.8	−1.5	56.0	43.4	−1.8	+0.2	56.0	43.5	−1.8	−0.8	56.0	24.0	−1.8	−2.0
	D^R_{All}	57.7	43.6	−0.1	−0.2	57.7	40.5	−0.1	+0.4	57.7	43.2	−0.1	+0.0	57.7	44.5	−0.1	+0.2	57.7	26.0	−0.1	+0.0
	$D_{\mathcal{S}(A)}$	34.8	43.4	−23.0	−0.4	32.6	31.1	−25.2	−12.7	32.6	28.9	−25.2	−14.3	20.4	25.0	−37.1	−19.3	48.3	22.9	−9.5	−3.1
	$D_{\mathcal{S}(F)}$	57.8	41.5	+0.0	−2.5	57.2	37.8	−0.6	−6.0	56.9	39.6	−0.9	−3.6	57.6	43.0	−0.2	−1.3	57.8	25.6	+0.0	−0.4
Mistral	$D^R_{\mathcal{S}(F)}$	61.0	40.2	−0.7	+0.2	61.0	40.1	−0.7	−0.3	61.0	46.7	−0.7	−0.4	61.0	45.2	−0.7	−0.5	61.0	12.7	−0.7	−1.4
	$D^R_{\mathcal{S}}$	60.7	40.4	−1.0	+0.4	60.7	36.9	−1.0	−3.5	60.7	46.9	−1.0	−0.3	60.7	46.3	−1.0	+0.7	60.7	11.1	−1.0	−3.0
	D^R_{All}	61.8	40.1	+0.1	+0.1	61.8	40.7	+0.1	+0.3	61.8	47.2	+0.1	+0.1	61.8	44.7	+0.1	−1.0	61.8	14.1	+0.1	+0.0
	$D_{\mathcal{S}(A)}$	50.4	32.3	−11.3	−7.7	55.3	27.4	−6.4	−13.0	54.7	42.4	−7.0	−4.7	44.5	34.1	−17.2	−11.6	51.1	8.3	−10.6	−5.8
	$D_{\mathcal{S}(F)}$	61.5	38.1	−0.2	−1.9	61.2	38.1	−0.5	−2.3	61.3	43.5	−0.4	−3.6	61.0	43.9	−0.7	−1.8	60.8	11.8	−0.4	−2.3

Table 13: Knowledge Extraction task.

	Method	French				Chinese				Spanish				Russian			
		En-D	Fr-D	$\Delta_{\text{En-D}}$	$\Delta_{\text{Fr-D}}$	En-D	Zh-D	$\Delta_{\text{En-D}}$	$\Delta_{\text{Zh-D}}$	En-D	Es-D	$\Delta_{\text{Es-D}}$	$\Delta_{\text{Es-D}}$	En-D	Ru-D	$\Delta_{\text{En-D}}$	$\Delta_{\text{Ru-D}}$
Vicuna	$D^R_{\mathcal{G}}$	13.2	14.2	+0.1	+0.0	13.2	61.6	+0.1	+0.5	13.2	10.4	+0.1	+0.0	13.2	20.8	+0.1	+0.0
	D^R_{All}	13.0	14.1	−0.1	−0.1	13.0	61.6	−0.1	+0.5	13.0	10.4	−0.1	+0.0	13.0	20.8	−1.0	+0.0
	$D_{\mathcal{G}}$	13.0	13.8	−0.1	−0.4	13.1	59.5	+0.0	−1.6	13.0	9.1	−0.1	−1.3	13.1	20.3	+0.0	−0.5
Mistral	$D^R_{\mathcal{G}}$	13.6	15.2	+0.1	+0.0	13.6	56.7	+0.1	+0.3	13.6	10.3	+0.1	−0.3	13.6	21.2	+0.1	+0.2
	D^R_{All}	13.6	15.4	+0.1	+0.2	13.6	55.9	+0.1	−0.5	13.6	10.2	+0.1	−0.4	13.6	21.1	+0.1	+0.1
	$D_{\mathcal{G}}$	14.3	14.2	+0.8	−1.0	13.6	52.8	+0.1	−3.6	13.7	10.2	+0.2	−0.4	13.5	20.2	−0.1	−0.8

Table 14: Generation task.