Can Neuron Activation be Predicted? A New Lens of Neuron Predictability for Analyzing Transformer-based Large Language Models

Anonymous ACL submission

Abstract

Transformer-based large language models (LLMs) play a vital role in various NLP tasks, but the internal neurons are rather unpredictable, functioning in a black box style. Thus, in this work, we introduce the *Neuron Predictability Lens (NPL)*, an analytical framework that focuses on the way how neurons work through feed-forward networks (FFNs), to understand and analyze transformer-based LLMs. Based on the proposed framework, we conduct extensive experiments on LLaMA-2 and GPT-J. Firstly, we show that neuron activations are predictable and for the first time we introduce the concept of *Neuron Predictability*. Secondly, we apply NPL to both global and local analysis. For global analysis, we investigate how FFNs contribute to model behaviors explicitly and implicitly with the aid of NPL. For local analysis, we explore the connection between neuron predictability and neuron interpretability. We examine various functional neurons under NPL and uncover the existence of "background neurons." With the findings mentioned above, we demonstrate the value of NPL as a novel analytical tool and shed light on its future application on model efficiency and/or effectiveness for improved language modeling.

1 Introduction

Large Language Models (LLMs) exhibit human-level proficiency in completing multiple natural language tasks (Vaswani et al., 2017; OpenAI, 2022; Touvron et al., 2023). However, these models are often regarded as "black boxes" since how their inner neuron function is mysterious (Bommasani et al., 2021). Insufficient understanding of LLMs hinders further optimization and responsible deployment of such powerful tools. Thus, paving the way towards a more transparent internal structure of LLMs becomes increasingly important.

Efforts to understand and analyze LLMs range from global examinations of model behaviors to local dissections of specific modules (Luo and Specia, 2024). From a global view, researchers delve into comprehending the model's output and decision-making processes, e.g. detect how the activations in feed-forward neural network (FFN) contribute to the logits(Geva et al., 2021). In contrast, the local analysis seeks to unravel the mysteries of specific modules. For example, neuron interpretability research has dived into the relationship between individual neurons and specific linguistic tasks or functions (Dai et al., 2022a). Bridging these two perspectives, our work introduces a novel concept called *Neuron Predictability Lens*, which potentially encapsulates both the broader granularity and the finer granularity of LLM analysis. Figure 1 is an illustration of *Neuron Predictability*.

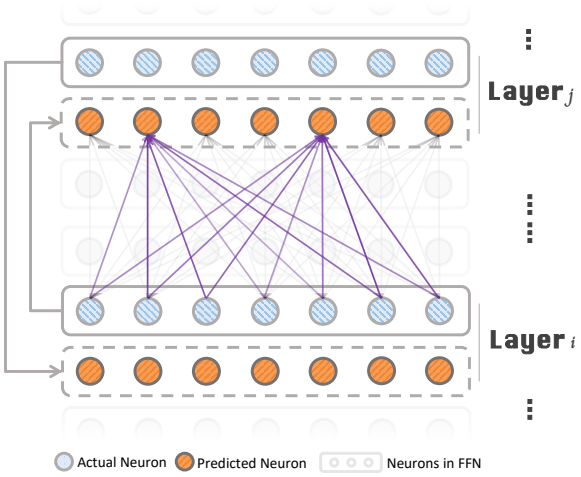

Figure 1: *Neuron Predictability*: The basis of *NPL*. The predicted version of neurons in layer j (in orange) can be extracted from the actual neurons in layer i (in blue), and vice versa.

Neuron Predictability Lens (NPL) is an analytical framework devised to provide a new perspective for understanding the behavior of transformer-based LLMs. NPL is performed through linear transformation, mapping FFN neurons across different layers. This method provides new insights when being used, renewing the interpretability of

vast concepts for transformer-based LLMs, such as logits contribution (i.e. the contribution of specific modules to the final logits, same hereafter) and neuron activation.

To make it clearer, we use *neuron activation* to denote the intermediate representation of the FFN module (further demonstrated in Sec.2). We establish mappings between different layers and project activations in either a forward or a backward direction. We need to answer a natural research question (**RQ1**): *can neuron activation be predicted*? To answer this question, we train neuron mappings across possible layer pairs on LLaMA-2 and GPT-J. Extensive experiments demonstrate that neuron activations are indeed predictably interconnected; the predictability persists even when transferring to data distribution away from the training data.

With the feasibility of neuron predictability established, we then raise the second research question (**RQ2**): *how to use NPL for model analysis?*

We utilize NPL to analyze LLMs in both global and local ways, unveiling significant findings in both branches: (1) In the global analysis, we substitute the predicted neuron activations for the actual ones and record the corresponding performance changes. Through this analysis, explicit and implicit contributions are investigated along with various substitution strategies. The main experiment reveals that shallow layers contribute to the final logits more implicitly while deep layers contribute more explicitly. The follow-up experiment delves deeper into the phenomenon and demonstrates that neurons with lower variance are more predictable, playing a crucial role in model performance. (2) Local analysis is conducted where we explore the relationship between neuron predictability and neuron interpretability. Through the lens of neuron predictability, we examine a variety of "functional" neurons pinpointed by prior research (Dai et al., 2022a), uncovering common characteristics among these functionally specialized neurons. From this analysis, we uncover a special category of "*background neurons*" – neurons that are vital to model performance, easy to predict, but do not exhibit explicit functional roles.

Overall, our contributions are as follows:

• **The NPL framework:** we propose and verify the effectiveness of *Neuron Predictability Lens* to analyze transformer-based LLMs;

• **Findings from the global analysis** with NPL: we find that shallow layers have more implicit logits contributions while deep ones have more explicit contributions; neurons with lower variance contribute more to the final logits;

• **Findings from the local analysis** with NPL: our proposed method measures the predictability of functional neurons, and reveals the existence of "background neurons", which are discovered – to the best of our knowledge – for the first time.

2 Neuron Predictability Lens

A major LLMs family is implemented based on transformer-based auto-regressive language models, which is our primary focus in this paper. Models are comprised of layers, and each layer contains two modules: a multi-head self-attention module (MHSA), and a feed-forward network module (FFN). Let $\boldsymbol{h}^l$ denote the input vector of l^{th} layer, the computations within l^{th} layer can be formulated as follows:

$$\begin{aligned} \boldsymbol{a}^l &= \text{MHSA}^l(\boldsymbol{h}^l), \\ \boldsymbol{m}^l &= \text{FFN}^l(\boldsymbol{h}^l + \boldsymbol{a}^l), \\ \boldsymbol{h}^{l+1} &= \boldsymbol{h}^l + \boldsymbol{a}^l + \boldsymbol{m}^l. \end{aligned} \tag{1}$$

$\boldsymbol{a}^l$ and $\boldsymbol{m}^l$ represents the output of MHSA and FFN. Based on these equations, the whole computation process is (L denotes the number of layers):

$$\boldsymbol{h}^{final} = \boldsymbol{h}^1 + \sum_{l=1}^{L} \boldsymbol{a}^l + \sum_{l=1}^{L} \boldsymbol{m}^l. \tag{2}$$

We focus on the FFN module specifically, which has been proven to bear vast information (Suau et al., 2020; Geva et al., 2021, 2022; Dai et al., 2022a; Wang et al., 2022; Luo and Specia, 2024; Gurnee et al., 2024). The inner structure of FFN comprises two full-connection feed-forward layers with the activation function sandwiched between them. Formally:

$$\text{FFN}\left(\boldsymbol{x}\right) = \boldsymbol{W}^O \cdot \sigma\left(\boldsymbol{W}^I \cdot \boldsymbol{x}\right), \tag{3}$$

where σ is the activation function, and $\boldsymbol{W}^I \in \mathbb{R}^{d \times d_{\text{ffn}}}$ and $\boldsymbol{W}^O \in \mathbb{R}^{d_{\text{ffn}} \times d}$ are learnable weight matrices. d is the hidden size and d_{ffn} is the intermediate dimension of FFN. For simplicity, the bias terms of linear layers are ignored.

Neurons in FFN NPL is proposed based on the Neurons in FFN. To elaborate the neurons, we

rewrite Equation 3 as:

$$\mathrm{FFN}(\boldsymbol{x}) = \sum_{i=1}^{d_{\mathrm{ffn}}} [\boldsymbol{g}]_i \boldsymbol{W}^O_{:,i}, \quad \boldsymbol{g} = \sigma(\boldsymbol{W}^I \cdot \boldsymbol{x}). \tag{4}$$

Just like the previous studies (Dai et al., 2022a; Wang et al., 2022; Zhang et al., 2023), neurons are defined here as the column vectors $\boldsymbol{W}^O_{:,i}$. We denote $\boldsymbol{g}$ as the activation vector, indicating the activation of neurons. The i^{th} element of $\boldsymbol{g}$ is the activation of the i^{th} neuron.

The *Neuron Predictability* indicates a mapping between neurons in different FFN modules. Given two layers i and j, we establish projection $M_{i\to j} : \mathbb{R}^{d_{\mathrm{ffn}}} \to \mathbb{R}^{d_{\mathrm{ffn}}}$ which projects from the activation vector $\boldsymbol{g}_i$ of layer i to the activation vector $\boldsymbol{g}^j$ of layer j. From this projection, we could get $\tilde{\boldsymbol{g}}^j = M_{i\to j}(\boldsymbol{g}^i)$, where $\tilde{\boldsymbol{g}}^j$ is a predicted item of real $\boldsymbol{g}^j$. NPL measures how well $\tilde{\boldsymbol{g}}^j$ fits $\boldsymbol{g}^j$. We use two metrics to evaluate the prediction, the L2 distance and the Pearson Correlation (Pearson, 1895). The prediction mapping is implemented by a linear transformation and is optimized by minimizing the mean square error (MSE). Below are the corresponding equations.

$$M_{i\to j}(\boldsymbol{g}^i) := \boldsymbol{W}_{M_{i\to j}} \cdot \boldsymbol{g}^i \tag{5}$$

$$\boldsymbol{W}_{M_{i\to j}} = \arg\min_{\boldsymbol{W}} \mathbb{E}||\boldsymbol{W} \cdot \boldsymbol{g}^i - \boldsymbol{g}^j||_2 \tag{6}$$

3 Preliminary Analysis: Predictability of Neuron Activations

In this section, we implement NPL in real settings to answer **RQ1**. The results prove the existence of neuron predictability in tested models.

3.1 Experimental Setup

We establish mapping $M_{i\to j}$ across every other layer on LLaMA-2-7b (Touvron et al., 2023) and GPT-J-6b (Wang and Komatsuzaki, 2021) ($\forall i, j \in \{2k \mid 2k < L, k \in \mathbb{N}\}$; L is the number of layers). Not all layers are utilized due to constraints by computational resources. Here, i could be either smaller than, larger than, or equal to j.

We use the training set of WikiText2 (Merity et al., 2016) to train the mappings. Since a quick and consistent convergence emerges while training, we sample a subset (consisting of about 10^7 tokens) instead of utilizing the entire dataset in the real process. We employ the Adagrad optimizer (Duchi et al., 2011) and set the initial learning rate as 0.01. The training is completed for a single epoch with a batch size of 10^4 tokens. Please refer to Appendix A for more details.

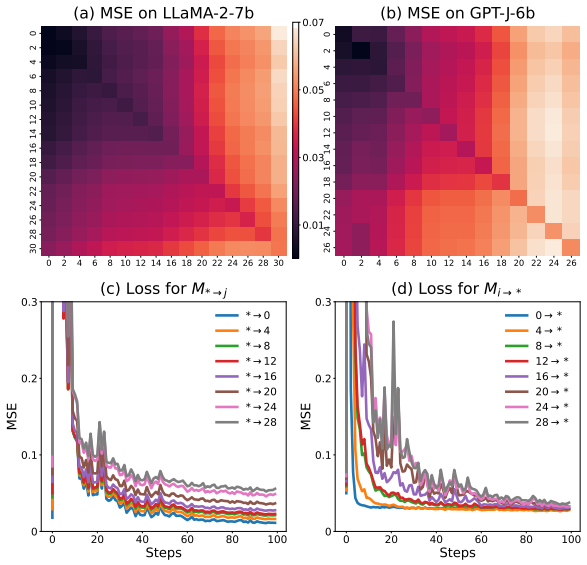

Figure 2: (a, b)Performance evaluation of learned neuron mappings for LLaMA-2 and GPT-J, respectively. (c, d) Averaged training curve on LLaMA-2, $M_{*\to j}$ denotes mapping from any layer to layer j and $M_{i\to *}$ denotes mapping layer i to any layer.

3.2 Results and Analysis

Figure 2 (a, b) is the visualization of NPL implementation. They illustrated the layer-wise neuron predictability on LLaMA-2-7b and GPT-J-6b measured by L2 distance. In the results, the overall L2 distances are around or less than 0.05, and the largest L2 distance is no more than 0.07. This decent result shows that neurons demonstrate a predicting relation between layers, and the phenomenon exists in both models.

The predictability varies among layers. Shallow layers tend to yield better predictability than deeper ones regardless of the projecting direction. Similar results are shown in the averaged training loss in Figure 2 (c, d), where shallow layers converge quicker and better in both projecting directions.

Furthermore, we calculate the average L2 error for three different cases: 0.037 for shallow-to-deep prediction ($i < j$), 0.024 for deep-to-shallow prediction ($i > j$), and 0.020 for self-prediction ($i = j$). These results indicate that deep-to-shallow prediction is more accurate than the reverse, with self-prediction yielding the best performance. This means deeper layer FFN activations encapsulate information from shallower layers, which accounts

Settings		LLaMA-2				GPT-J			
Mapping	Substitution	Chunk 1	Chunk 2	Chunk 3	Chunk 4	Chunk 1	Chunk 2	Chunk 3	Chunk 4
Random	Complete	> 1000	47.54	54.49	55.82	> 1000	403.02	45.23	580.48
	Partial	33.98	33.37	42.52	112.89	25.09	27.77	53.09	282.60
Zero	Complete	> 1000	43.56	50.07	55.14	> 1000	383.30	43.88	557.31
	Partial	33.33	33.40	42.34	109.47	24.98	27.38	49.32	286.14
Identical	Complete	> 1000	58.96	65.19	62.35	> 1000	94.99	41.20	246.34
	Partial	34.83	33.56	43.59	114.07	24.21	27.48	58.72	231.28
NPL	Complete	47.23	37.78	38.51	47.17	247.76	58.73	34.92	46.82
	Partial	33.41	33.54	37.49	49.66	23.64	24.60	32.01	50.30

Table 1: Perplexities of various mapping and substitution settings. For the random mapping, we run the evaluation 5 times and compute the average. We split all transformer layers into four equal-sized chunks and enumerate them from shallow to deep. Take the LLAMA-2-7b model as an example, with 32 layers, there are 8 contiguous layers in each chunk and Chunk 1 contains the first 8 contiguous layers.

for the greater ease of predicting shallower layer outputs from deeper FFN activations.

3.3 Follow-up: Cross-Domain Genreralization

Datasets in different domains from the training set are used to test the generalization of the NPL framework. Results show that the NPL framework performs well in different tasks (See App.B.1). More interesting experiments are listed in AppendixB.

4 Global Analysis: Analyzing the Logits Contribution of Predicted Neurons

This is our first step to answer **RQ2**. Through NPL, we evaluate how the predicted activations affect the model performance, which both provides a global LLM analysis and validates the effectiveness of NPL. Specifically, we substitute the actual neuron activations with those predicted by the *NPL Mapping*. Given a mapping $M_{i \to j}$ where the activation of layer i serves as the stimulus for predicting the response in layer j, we substitute the authentic activations in layer j with the predicted ones.

4.1 Experimental Setup

Recalling Equation 2, due to the existence of residual connection, the model's final representation $\boldsymbol{h}^{final}$ can be viewed as a summation of the outputs from each layer's FFN and MHSA modules. This final representation is normalized and projected to the "logits" over vocabulary via the language modeling head. We refer to the FFN output $\boldsymbol{m}^l$ as the **explicit contribution** from the FFN^l to the logits as $\boldsymbol{m}^l$ is explicitly added to the final output $\boldsymbol{h}^{final}$. There is also an **implicit contribution** from FFN^l, as deeper layer representations are computed based on the outputs of shallower layers. Thus, $\boldsymbol{m}^l$ also contributes to $\boldsymbol{h}^{final}$ implicitly by involving the computation of all its subsequent layers.

In this section, we conduct substitution experiments to study how the predicted neuron activations affect the explicit and implicit contributions. Figure 3 shows the two substitution settings.

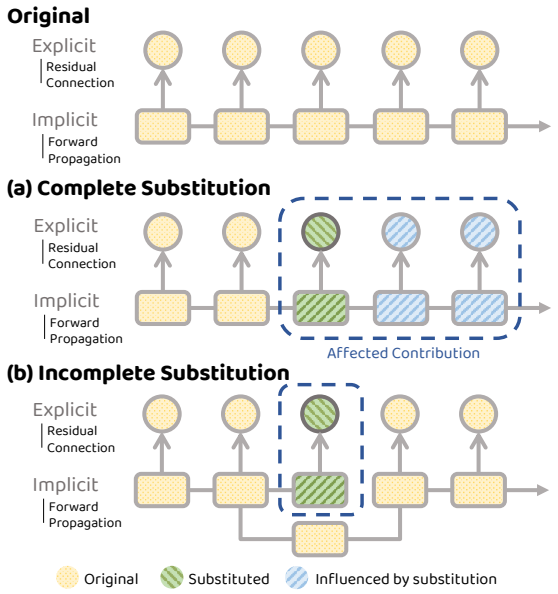

Figure 3: An illustration of the substitution settings: (a) *Complete Substitution* where both explicit and implicit contributions are substituted, (b) *Partial Substitution* where only implicit contribution is substituted.

As forward propagation proceeds from shallow layers to deep ones, we only consider the mapping $M_{i \to j}$, i.e. when $i < j$ (if not specified, $i = j - 1$). We split all layers into four chunks. In each trial,

we substitute neuron activations of one chunk of layers. For each setting, the following three types of mappings are compared with *NPL Mapping*:

- *Random Mapping* substitutes actual activations with activations obtained through a randomized mapping. (See caption in Tab.1)
- *Zero Mapping* zero-outs actual activations.
- *Identical Mapping* substitutes actual activations with activations from its previous layer.

4.2 Results and Analysis

Table 1 presents the results extracted in various substitution settings in LLaMA-2 and GPT-J. The *NPL Mapping* exerts the most negligible impact on the logits, corroborating that neuron predictability indeed captures information intrinsically linked to the model's capabilities. In contrast, the *Random Mapping* and *Zero Mapping* either introduce meaningless noise or remove the activations within certain chunks, both resulting in a substantial perturbation of the logits.

There is a strong correlation between the depth of substituted layers and the resultant effect. Substituting activations within the two middle chunks causes a relatively minor impact on the final logits, whereas substitution at either the bottom or top chunks introduces a more pronounced effect. Additionally, our findings indicate that this correlative relationship manifests differentially when assessing explicit versus implicit contributions. Here is a bulleted list of our findings:

- **Only FFN in deep layers (Chunk 4) exhibit a significant explicit contribution to the logits.** Conversely, substituting the activations in the shallow layers, particularly layers in Chunk 1, demonstrate an almost negligible explicit contribution to the logits regardless of the substitution setting.
- The trend is reversed for implicit contributions. **FFN in shallow layers (Chunk 1) contribute more implicitly than those in deep layers (Chunk 4)**. Since the shallow layers play foundational roles and influence all the subsequent computations, this phenomenon is explainable. Thus, if these layers are compromised, the ability of the model would be severely impaired. On implicit contribution, *NPL Mapping* shows an evident advantage over other substitution strategies, again suggesting that NPL captures anticipated meaningful semantic information to some extent.
- Another intriguing finding is that in Chunk 4, *complete substitution* outperforms *partial substitution* in all mappings for LLaMA-2 and in *NPL Mapping* for GPT-J. This phenomenon suggests that **in deep layers, the presence of a "fake" explicit contribution appears to elicit a negative effect on the actual implicit contribution**, which is abnormal.

4.3 Finer-grained Neuron Substitution

As different neuron performs differently, we further investigate the relation between neuron performance and the substitution results. This time, for each layer in one chunk, a subset of the neurons is substituted. The subset is sampled by the neuron performance. Four distinct metrics are utilized to guide our selection: the *Pearson correlation* and *L2 distance* between the predicted activation $\tilde{\boldsymbol{g}}$ and the actual one $\boldsymbol{g}$, coupled with the *mean* and *variance* of $\boldsymbol{g}$. Through this investigation, we can explore more fine-grained connections between neuron predictability and model behaviors.

Results As depicted in Figure 4, a strong relationship exists between neuron predictability and all measured metrics except for the mean. Comparing *top 50% predicted; bottom 50% zero* with *top 50% zero; bottom 50% predicted*, we find that preserving the information from low-variance neurons is sufficient for maintaining acceptable performance, even if the remaining neurons are masked. Besides, the *actual vs. predicted* comparison reveals the significance of neurons that are easily predicted (as indicated by a lower L2 distance) for performance, while those that are hard to predict (with a higher L2 distance) appear to be less important. In summary, concerning performance impact, we observe that *bottom 50% zero* > *top 50% zero* > *bottom 50% predicted* > *top 50% predicted*.

Insights This experiment reveals several findings: (1) Neurons with lower variance in their activations are vital for the performance. (2) These vital neurons also tend to be more predictable, as indicated by their lower L2 distance between the predicted and the real activations. (3) This also implies that *NPL Mapping* is not random but rather related to the intrinsic properties of the neuron, such as its stability and role within the network.

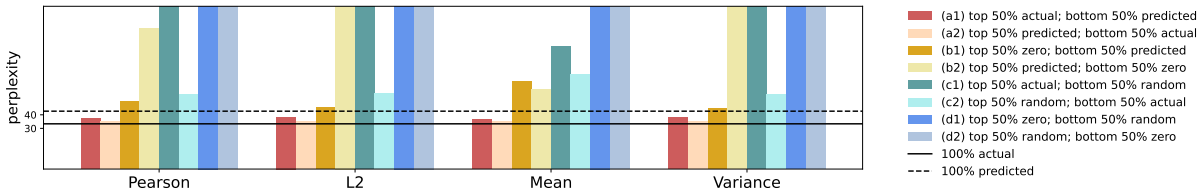

Figure 4: Comparison of evaluation performance for different neuron substitution strategies. We categorize all neurons into two groups based on four metrics and then implement varying substitution techniques in these groups. For instance, *top 50% predicted; bottom 50% zero* indicates that the top 50% of neurons, according to one of the four metrics, are replaced with predicted activations, while the rest is set to zero.

4.4 Interim Summary

With NPL, the above global analysis delves into LLMs' inner structures by detecting corresponding contribution to the model's logits. Apart from the analysis itself, this section validates the effectiveness of the NPL framework as the neuron predictability indeed captures information relevant to model capability instead of learning irrelevant features. Besides, our experiment reveals that a simple linear network is sufficient to capture certain general neuron activation patterns within LLMs, which is noteworthy.

5 Local Analysis: Analyzing the Predictability of Functional Neurons

This section demonstrates how the NPL framework could adapt to the local LLMs analysis and steps further to answer **RQ2**. We classify neurons according to their specialties and detect the predictability of functional neurons. Following the previous works, we use "functional neuron" to denote neurons whose activation patterns correlate to specific functions, including token identification, position encoding, knowledge storing, and others (Gurnee et al., 2024; Voita et al., 2023; Dai et al., 2022b). We conduct further analysis on functional neurons and examine their characteristics under NPL. To this end, we first need to locate functional neurons, and then evaluate their predictability. We follow the procedure of Gurnee et al. (2024) to locate these neurons, which assess the variance reduction when conditioned on specific functionality. For a given neuron i in layer l, we compute:

$$\mu_{\mathcal{P}}^{l,i} = 1 - \frac{(1-\beta)\sigma^2([\boldsymbol{g}^l]_i|\mathcal{P}(\boldsymbol{x})) + \beta\sigma^2([\boldsymbol{g}^l]_i|\neg\mathcal{P}(\boldsymbol{x}))}{\sigma^2([\boldsymbol{g}^l]_i)}, \tag{7}$$

where $\mathcal{P}$ represents the property function that determines whether the input token $\boldsymbol{x}$ exhibits the functionality under investigation, and β is the proportion of tokens that possess this functionality. The resulting $\mu_{\mathcal{P}}^{l,i}$ serves as the importance score of neuron i in layer l concerning the functionality $\mathcal{P}$. Then, we set a threshold $\theta_{\mathcal{P}}$ to filter the neurons that exhibit relatively high importance scores. Afterward, we compute the mean predictability score over the neurons with high importance scores:

$$S_{\mathcal{P}}^{l} = \frac{1}{|N_{\mathcal{P}}^{l}|} \sum_{i \in N_{\mathcal{P}}^{l}} s^{l,i}, \tag{8}$$

where $N_{\mathcal{P}}^{l} = \{i|\mu_{\mathcal{P}}^{l,i} > \theta_{\mathcal{P}}\}$ is the subset of filtered neurons and $s^{l,i}$ is the predictability of neuron i in layer l, measured by L2 distance. As a result, higher $S_{\mathcal{P}}^{l}$ means lower predictability for neurons with property $\mathcal{P}$. For comparison, we also compute the predictability score on all evaluation data $S_{\mathcal{O}}^{l}$ and on a random subset of tokens $S_{\mathcal{R}}^{l}$. In our experiments, we explore the setting where the presence of the specific functionality is considered as the sufficient condition for high activation. In each following section, we examine one specific kind of functional neuron.

5.1 N-gram-Sensitive Neurons $\mathcal{P}_{n-gram}$

Some neurons are found to activate exclusively when specified n-grams are present in the input, as a result, they are named as "n-gram detecting" neurons (Voita et al., 2023).

We examine n-grams with n ranging from 1 to 3 and conduct a comprehensive analysis of all n-grams present within the test corpus, filtering out meaningless ones, and selecting the 1,000 most frequent ones for each n for further investigation As shown in Figure 5(a), there is a clear distinction between n-gram sensitive neurons and the random baseline across most of the layers, while the distinctions between different choices of n are subtle. This finding verifies the feasibility of pinpointing "n-gram sensitive neurons" within LLMs, as these neurons exhibit significantly higher predictability scores compared to others, indicating a heightened level of unpredictability. Considering that our mapping network is simple, the results appear plausible.

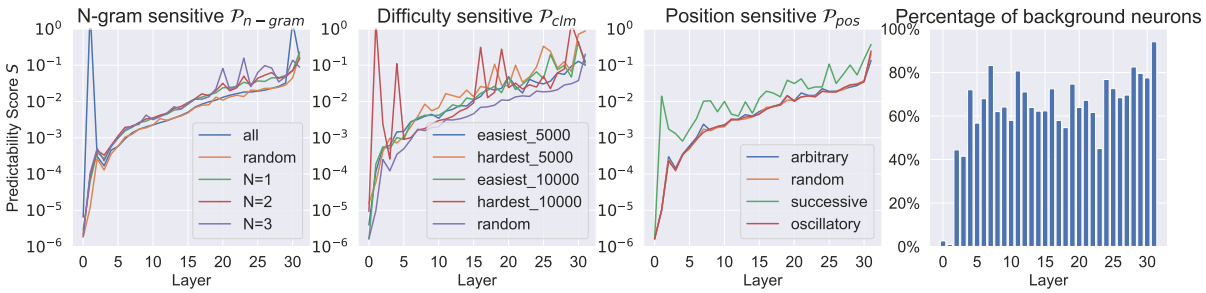

Figure 5: Summary of functional neuron experiments in LLaMA-2. (a-c) Predictability score for n-gram/difficulty/position sensitive neurons. (d) Percentage of background neurons across different layers.

5.2 Difficulty-Sensitive Neurons $\mathcal{P}_{loss}$

During the investigation, we found that the activations of certain neurons are correlated with the performance of the causal language modeling objective. Tokens that are hard to predict, manifesting in high cross-entropy loss (denoted as *hard token*), tend to activate specific neurons. Conversely, another set of neurons is similarly activated in response to tokens that are easy to predict (denoted as *easy token*). We get hard and easy tokens by filtering tokens based on their cross-entropy loss, upon which we compute the predictability score $S^l_{\mathcal{P}}$. As depicted in Figure 5(b), regardless of whether we consider easy or hard tokens, difficulty-sensitive neurons exhibit significantly lower predictability scores than the random baseline. Furthermore, it is observable that the predictability associated with neurons corresponding to hard tokens exhibits greater fluctuations across different layers. This is a logical phenomenon, as these tokens represent greater challenges for the model, and thus the information flow related to them within the model is more unstable.

The identification of difficulty-sensitive neurons is intriguing, as it suggests that the LLM possesses a form of self-awareness regarding its confidence in predicting the next tokens. By probing its internal representations, we can uncover such "mental states" of LLMs without external signals.

5.3 Position-sensitive Neurons $\mathcal{P}_{pos}$

Another branch of neurons is those associated with positional information, which activates in response to the position rather than the token or its context. Inspired by Voita et al. (2023), we hypothesize that positional neurons can work in teams and collectively respond to various positional patterns. We explored three types of positional pattern: (1) an *arbitrary* pattern, where we randomly select a subset of all positions; (2) a *successive* pattern, where we choose a fixed-length span of consecutive positions; (3) an *oscillatory* pattern, where we select positions at regular intervals. In this experiment, we clip the maximum input length to 1024 and examine positions ranging from 1 to 1024. As illustrated in Figure 5, only the predictability scores for successive patterns exhibit significant deviations from the random baseline. The rest two position patterns are too complex to be captured.

5.4 "Background" Neurons

In the above examination, all neurons associated with specific functionalities exhibit high predictability scores, indicating that they are hard to predict. Conversely, we are also interested in those who are more predictable. To this end, we set the predictability score of the random baseline as a threshold to filter out neurons whose predictability score falls below. As depicted in Figure 5(d), a substantial proportion of neurons (ranging from 40% to 80%) fall into this category, suggesting that a majority of the neurons within FFNs are relatively easy to predict. The precise function of these neurons is challenging to define, but their critical importance is evident based on the results from Figure 4. Masking highly predictable neurons (b2) leads to a significant drop in performance, whereas masking neurons with lower predictability (b1) has a milder impact. Due to their less apparent roles, we refer to these neurons as "background" neurons.

The mysterious nature of background neurons challenges our conventional approach to interpreting model behavior, which often revolves around identifying various types of functional neurons. Given that a considerable proportion of neurons contribute to model behavior while "working in the dark", it prompts us to rethink how we credit the success of the model's performance. How should we attribute the model's effectiveness solely to those functional but rare neurons amidst a backdrop of numerous "invisible" contributors?

5.5 Interim Summary

In this section, a variety of functional neurons are examined through NPL, within which difficult-sensitive neurons are first identified. Besides, we name a special kind of neuron as the "background neuron." This utilization of NPL further proves the effectiveness of the framework. See more details in Appendix (App.C.1 and Fig.8).

6 Discussions and Implications

The above analysis reveals NPL as an effective analytical tool for LLMs. Here, we discuss the following applications and implications:

Inference Acceleration by Short-Cutting Transformers Previous research has investigated accelerating inference by establishing linear shortcuts across transformer blocks (Din et al., 2023). NPL bears a resemblance to these efforts with a distinct focus – NPL is set to predict the activations of FFN neurons instead of hidden representations. Besides, we investigate the logits contributions of FFNs and conclude that employing predicted neurons in intermediary layers does not result in significant performance degradation (Sec.4), suggesting the potential of NPL as a promising avenue for bypassing the complicated computations of vanilla transformers.

Connection to Knowledge Distillation Establishing *NPL Mapping* across different FFN layers shares conceptual parallels with layer-wise knowledge distillation (Liang et al., 2023), where the original transformer blocks act as teacher models and the linear mappings act as student models. However, with NPL, we have a different target from traditional knowledge distillation, which aims at replicating the teacher model's performance with a more lightweight student model. Instead, we use the mapping as an analytical lens, through which we better understand LLMs internally.

Rethinking the Role of FFN Previous research has offered varying interpretations regarding the role of FFN within LLMs. Some posit that FFN functions as key-value memories (Geva et al., 2021), while others suggest the role of FFN is to project hidden representations onto a distribution over the output vocabulary, thus amplifying the predicted probability of some words while diminishing that of others (Geva et al., 2022; Belrose et al., 2023; Katz and Belinkov, 2023). Our investigation reveals these arguments to be incomplete. Under NPL, we observe that FFNs at various depths play diverse roles, and even within the same layer, individual neurons exhibit varied behaviors. Consequently, it is reductive to view the function of FFNs through a singular point of view.

7 Related Work

Analyzing transformers has attracted much attention in recent years. Following Luo and Specia (2024), we roughly categorize transformer analysis into two streams: local analysis, which delves into the intricacies of individual transformer components, and global analysis, which seeks a holistic understanding of the behaviors and capabilities of the model. Among local analysis, we are interested in those targeting individual neurons. For instance, Dai et al. (2022a) shed light on the storage of knowledge within model parameters by identifying specific "knowledge neurons". Similarly, Voita et al. (2023) uncover a range of functional neurons characterized by regular activation patterns. Global analysis encompasses a variety of approaches, including probing techniques (Rogers et al., 2020; Petroni et al., 2019; Li et al., 2023), mechanistic interpretability (Elhage et al., 2021; Wang et al., 2023), and more. Among these, the "Vocabulary lens", which projects weights and activations onto the vocabulary space, is a trending analytical tool (Geva et al., 2021). This lens allows researchers to explore the contributions of different modules and inputs to model performance (Belrose et al., 2023; Ram et al., 2023; Geva et al., 2023). Another direction analyzes transformers through simple mappings between modules. For example, Dar et al. (2023) learn to project parameters into a shared embedding space, while Din et al. (2023) explore linear shortcuts between layers, which bears conceptual relevance to our approach.

In contrast to previous studies, our introduction of the neuron predictability lens encompasses both the local and global facets of transformer analysis.

8 Conclusion

In this work, we present and validate the Neuron Predictability Lens (NPL) as a powerful analytical framework for examining transformer-based LLMs. By applying NPL to both global and local analysis and conducting extensive experiments, we uncover intriguing insights into the role of FFN and neuron interpretability. Implications we gained from NPL pave a new way for analyzing transformers and informing potential future applications.

Limitations

Since NPL is a newly proposed analytical framework, more applications are to be explored. Our work is an initial attempt to analyze transformers with NPL, and even at this early stage, we have already uncovered interesting insights. Due to space limitations, some experimental results are not fully elaborated. We attach part of them in the appendix. Moreover, we use linear mapping to implement the NPL framework, while other kinds of mappings could also be explored, though this would likely incur additional computational overhead. Future research may explore other mappings to further leverage the potential of NPL.

References

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. 2023. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ B. Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen Creel, Jared Quincy Davis, Dorottya Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, and et al. 2021. On the opportunities and risks of foundation models. *CoRR*, abs/2108.07258.

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.

Guy Dar, Mor Geva, Ankit Gupta, and Jonathan Berant. 2023. Analyzing transformers in embedding space. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 16124–16170. Association for Computational Linguistics.

Alexander Yom Din, Taelin Karidi, Leshem Choshen, and Mor Geva. 2023. Jump to conclusions: Short-cutting transformers with linear transformations. *arXiv preprint arXiv:2303.09435*.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7).

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2021/framework/index.html.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. Dissecting recall of factual associations in auto-regressive language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 12216–12235. Association for Computational Linguistics.

Mor Geva, Avi Caciularu, Kevin Ro Wang, and Yoav Goldberg. 2022. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. *arXiv preprint arXiv:2203.14680*.

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

Wes Gurnee, Theo Horsley, Zifan Carl Guo, Tara Rezaei Kheirkhah, Qinyi Sun, Will Hathaway, Neel Nanda, and Dimitris Bertsimas. 2024. Universal neurons in gpt2 language models. *arXiv preprint arXiv:2401.12181*.

Shahar Katz and Yonatan Belinkov. 2023. Visit: Visualizing and interpreting the semantic information flow of transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14094–14113.

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

Chen Liang, Simiao Zuo, Qingru Zhang, Pengcheng He, Weizhu Chen, and Tuo Zhao. 2023. Less is more: Task-aware layer-wise distillation for language model compression. In *International Conference on Machine Learning*, pages 20852–20867. PMLR.

Haoyan Luo and Lucia Specia. 2024. From understanding to utilization: A survey on explainability for large language models. *arXiv preprint arXiv:2401.12874*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1797–1807. Association for Computational Linguistics.

OpenAI. 2022. Openai chatgpt.

Karl Pearson. 1895. Vii. note on regression and inheritance in the case of two parents. *proceedings of the royal society of London*, 58(347-352):240–242.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473.

Giovanni Puccetti, Anna Rogers, Aleksandr Drozd, and Felice Dell'Orletta. 2022. Outlier dimensions that disrupt transformers are driven by frequency. In *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 1286–1304. Association for Computational Linguistics.

Ori Ram, Liat Bezalel, Adi Zicher, Yonatan Belinkov, Jonathan Berant, and Amir Globerson. 2023. What are you token about? dense retrieval as distributions over the vocabulary. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 2481–2498. Association for Computational Linguistics.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Xavier Suau, Luca Zappella, and Nicholas Apostoloff. 2020. Finding experts in transformer models. *arXiv preprint arXiv:2005.07647*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Elena Voita, Javier Ferrando, and Christoforos Nalmpantis. 2023. Neurons in large language models: Dead, n-gram, positional. *arXiv preprint arXiv:2309.04827*.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax.

Lean Wang, Lei Li, Damai Dai, Deli Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. 2023. Label words are anchors: An information flow perspective for understanding in-context learning. *arXiv preprint arXiv:2305.14160*.

Xiaozhi Wang, Kaiyue Wen, Zhengyan Zhang, Lei Hou, Zhiyuan Liu, and Juanzi Li. 2022. Finding skill neurons in pre-trained transformer-based language models. *arXiv preprint arXiv:2211.07349*.

Zhengyan Zhang, Zhiyuan Zeng, Yankai Lin, Chaojun Xiao, Xiaozhi Wang, Xu Han, Zhiyuan Liu, Ruobing Xie, Maosong Sun, and Jie Zhou. 2023. Emergent modularity in pre-trained transformers. *arXiv preprint arXiv:2305.18390*.

Mapping	Substitution	WikiText-2	Alpaca	XSum
Random	Complete	> 200	> 200	> 200
	Partial	55.69	13.49	12.37
NPL	Complete	42.67	12.44	12.16
	Partial	38.53	9.76	8.21
Original		33.35	8.63	6.23

Table 2: Averaged performance of Chunks 1-4 on WikiText-2, Alpaca (Taori et al., 2023), and XSum (Narayan et al., 2018). See full results in App. C.

A Additional Implementation Details

Implementing neuron mappings on LLaMA and GPT-J models needs to consider their slight architectural variations from the standard transformer design. These deviations are noteworthy and must be considered when conducting such mappings. The FFN of LLaMA contains an extra linear layer, which can be formulated as follows:

$$\begin{aligned} \mathrm{FFN}\left(\boldsymbol{x}\right) &= \boldsymbol{W}^{down} \cdot \boldsymbol{g}, \\ \boldsymbol{g} &= \sigma\left(\boldsymbol{W}^{gate} \cdot \boldsymbol{x}\right) \odot \left(\boldsymbol{W}^{up} \cdot \boldsymbol{x}\right). \end{aligned} \tag{9}$$

The transformer block for GPT-J is as follows:

$$\begin{aligned} \boldsymbol{a}^l &= \mathrm{MHSA}^l(\boldsymbol{h}^l), \\ \boldsymbol{m}^l &= \mathrm{FFN}^l(\boldsymbol{h}^l), \\ \boldsymbol{h}^{l+1} &= \boldsymbol{h}^l + \boldsymbol{a}^l + \boldsymbol{m}^l, \end{aligned} \tag{10}$$

where the MHSA and FFN within the same layer have no computation dependency.

B More Analysis on Neuron Predictability

B.1 Cross-Domain Generalization of NPL

To demonstrate that *NPL Mapping* does not just imitate the distribution of the training data, we evaluate its cross-domain generalization ability. The experimental setting is the same as Section 4. As shown in Table 2, while trained on Wikitext, *NPL Mapping* successfully generalizes to other data distributions by outperforming *Random Mapping* and closely approximating the performance of real activations. These results suggest that NPL is not limited to the specificities of the training data but rather captures broader, more universal patterns that are applicable even in contexts that diverge from the original training domain or language.

B.2 Context Length Affects Neuron Predictability

We investigate scenarios where tokens are exposed only to a constrained segment of the preceding context. To achieve this, we employ a context window, denoted by w, to limit the range of context accessible to each token. Subsequently, we train multiple NPL mappings for various w values and visualize the differences. As shown in Figure 6, a larger w extends the context scope and also results in increased predictability for neurons in shallower layers, while simultaneously decreasing predictability in deeper layers. We hypothesize that an extended context provides the NPL with more comprehensive information, aiding in the accurate prediction of neuron activations in shallow layers. Contrastingly, the semantics in deeper layers may become too complex to be captured by the NPL.

B.3 Cross-Model Neuron Predictability

Neuron mapping can be established not only within a single but also across different models. To validate this, we conduct experiments applying NPL between the LLaMA-2-7b and LLaMA-2-13b models. Figure 7 shows that the neuron mapping across models is learnable. Our observations reveal a strong correlation between the layers of the two models, with the most effective mappings establishing when layers of similar depth are used to predict each other. Additionally, based on the L2 distance metric, we have noted that shallower layers tend to be more predictable than their deeper counterparts, a similar phenomenon observed in single-model experiments.

C Additional Experimental Results

Mapping	Substitution	Chunk 1	Chunk 2	Chunk 3	Chunk 4
Alpaca					
Random	Complete	> 1000	11.98	11.44	12.12
	Partial	9.16	8.78	10.38	25.62
NPL	Complete	17.68	10.37	10.27	11.44
	Partial	8.67	8.83	9.63	11.91
XSum					
Random	Complete	> 1000	10.38	11.56	13.47
	Partial	6.94	6.68	9.08	26.78
NPL	Complete	19.57	8.45	9.57	11.06
	Partial	6.63	6.67	8.30	11.25

Table 3: Full results for the substitution experiment on Alpaca and XSum.

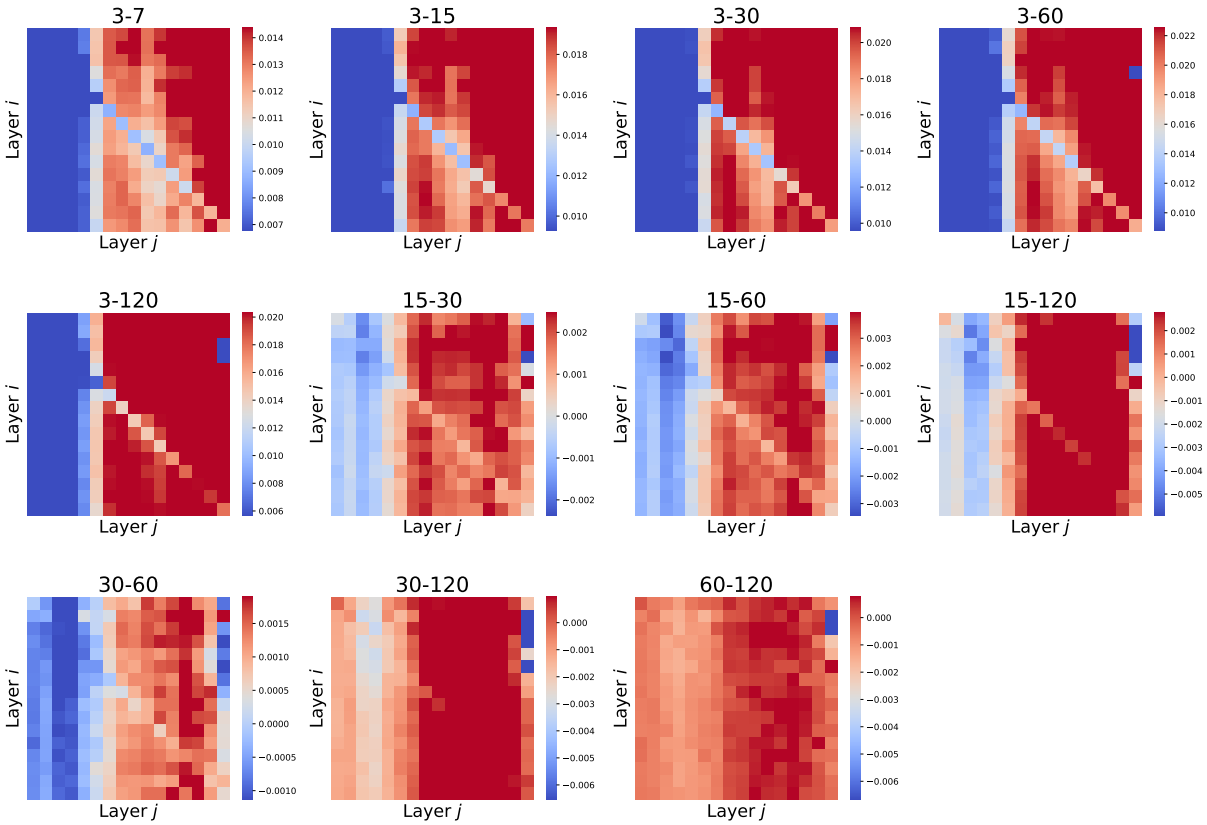

Figure 6: L2-distance difference of NPL mappings on LLaMA-2-7b under the settings of different window sizes w. The window sizes selected for this analysis include $w \in \{3, 7, 15, 30, 60, 120\}$.

C.1 Additional Details on Functional Neurons

Outlier Neurons Outlier phenomenon has been observed across various LLMs (Puccetti et al., 2022). This phenomenon refers to the persistent emergence of extreme values within the models' activations and weights which, though comprising less than 0.1% of the values, can exceed the magnitude of other values by several hundredfold and are thus termed "outliers". For LLaMA-2-7b, we find the 7890-th neuron of layer 2 (shorted as L2.7890) to be an outlier. We observe that the occurrence of outliers is associated with meaningless tokens, such as <SOS>, <UNK>. As for neuron predictability, outlier neurons are extremely hard to predict.

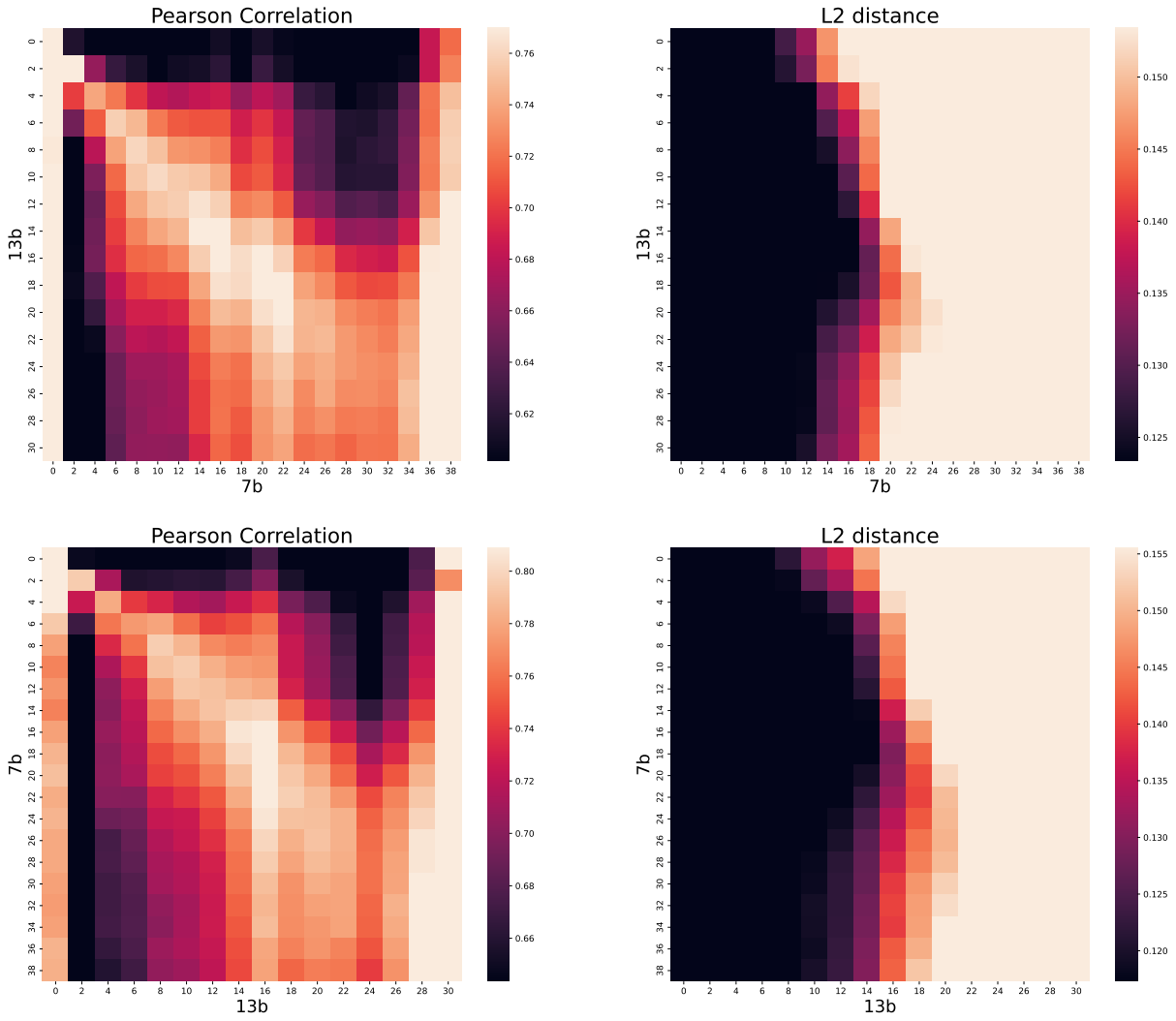

Figure 7: NPL between LLaMA-2-7b and LLaMA-2-13b.

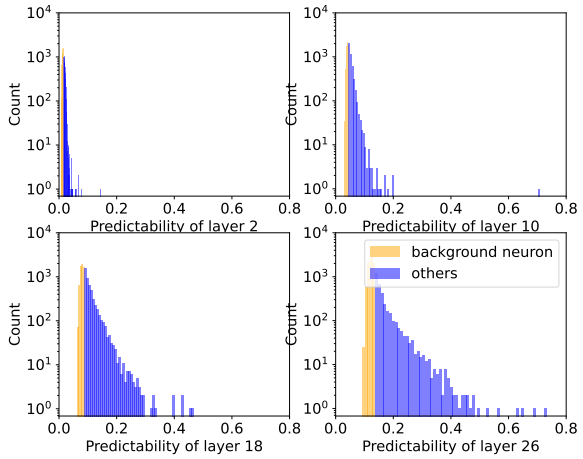

Figure 8: Statistics of background neurons in layer 2, 10, 18, 28 in LLaMA-2-7b.