Linear Effect of Neuron Activations in Transformer-based Language Models

Xin Zhao[1,a)] Zehui Jiang[1,b)] Naoki Yoshinaga[2,c)]

Abstract: Neurons in feed-forward layers of Transformers have shown the ability to store factual knowledge. However, previous analyses mostly focused on qualitative evaluation, leaving the numerical relationship between neuron activations and model outputs less understood. Our study conducts a quantitative analysis through neuron-wise intervention experiments using the knowledge probing dataset. Our findings first reveal that neurons exhibit linearity and polarity in producing output tokens probabilities, quantified by **"neuron empirical gradients."** Empirical gradients provide a direct measure of neurons' importance in representing knowledge. However, neuron-wise intervention experiments are costly, making it impractical to obtain empirical gradients in large language models. To address this, we propose **NeurGrad**, an efficient method for measuring neuron empirical gradients. Our experimental results show that NeurGrad outperforms several baseline methods in both efficiency and accuracy.

Keywords: Factual Knowledge Probing, neuron-wise intervention, Activation Editing, Large Language Models

1. Introduction

Transformer [1]-based language models (LMs) have demonstrated a strong ability to process linguistic tasks and understand factual knowledge through pre-training on large-scale real-world corpora. However, pre-trained LMs (PLMs) still suffer from the hallucination problem, where models sometimes generate fluent language while containing incorrect knowledge. This issue makes it an important topic to understand the mechanism by which PLMs store knowledge within their parameters, attracting great attention in recent years [2, 3, 4, 5].

In Transformer-based LMs, feed-forward networks (FFNs) are found to serve as key-value memory, with neurons possessing the ability to retrieve knowledge from this memory [6]. Through qualitative analysis, researchers have discovered that specific facts are highly correlated with a limited number of neurons (knowledge neurons) [2, 5]. Despite the simplification of this theory, it is commonly adopted to explain knowledge in large LMs (LLMs) [7, 8] and used for model editing [9]. However, these analyses focus on qualitative analysis of specific neurons in knowledge representation, while the numerical relationship between neuron activations and model outputs remains poorly understood.

In this study, we first conduct a quantitative analysis of how neuron activations affect model generation to knowledge inquiries(§ 2). To observe model generation under varying neuron activations, we conduct a neuron-wise intervention with factual prompts on LMs using MyriadLAMA [10], a factual knowledge probing dataset. For given changes of the neuron activations, we observe the resulting changes in probabilities of target tokens that represent correct knowledge (hereafter, "output probabilities"). Notably, we find that for some neurons, within a relatively broad range of activations, shifts in neuron activations (hereafter, "activation shifts") have a linear relationship with the output probabilities. Furthermore, we find that different neurons vary in the shifting direction of output probabilities when increasing neuron activations. We term this property of neurons as *polarity*, which we use to classify neurons as either *positive* or *negative*. Our evaluation of six PLMs, including Llama2-70B, demonstrates that the linearity and polarity of neurons generally exist. Finally, we denote a specific neuron's linear relationship for a specific token as its *neuron empirical gradient*.

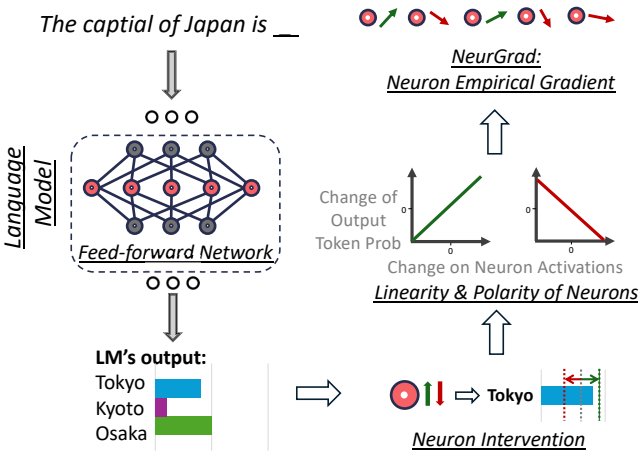

Fig. 1 Through neuron-wise intervention experiments on factual prompts, we discovered a linear correlation between shifts in neuron activations and the model's output probability for specific tokens. We denote this correlation as "neuron empirical gradients" and propose NeurGrad as an efficient method to capture it.

[1] The University of Tokyo
[2] Institute of Industrial Science, The University of Tokyo
[a)] xzhao@tkl.iis.u-tokyo.ac.jp
[b)] zjiang@tkl.iis.u-tokyo.ac.jp
[c)] ynaga@iis.u-tokyo.ac.jp

While the neuron empirical gradient provides a direct measurement of neurons' importance in reflecting models' knowledge to output, its calculation requires a high computational cost as the neuron empirical gradient changes for different combinations of prompt, neuron, and target token. This variation makes calculating empirical gradients through neuron-wise intervention for all neurons in LLMs computationally expensive.

To address this challenge, we further propose NeurGrad, a method for estimating neuron empirical gradients precisely and efficiently, and evaluate its performance using the MyriadLAMA dataset (§ 3). The proposal of NeurGrad stems from our experimental discovery of the relationship between neuron empirical gradients, computational gradients, and neuron activations. The computational gradients are calculated through backward propagation in the computational graph. Our evaluation of the PLMs demonstrates that NeurGrad is superior in both efficiency and precision compared to two baseline methods, including the integrated neurons [2] and the computational gradients. We measure the empirical gradients of all neurons for 1000 prompts per PLM. The experimental results indicate that a broad range of neurons, rather than a few specific ones [2], can influence the model's output to factual prompts.

2. Neuron Numerical Analysis

This section aims to establish a numerical understanding of how neurons in PLMs' FFN layers affect model generations. We use factual knowledge probing as our target task and conduct neuron-wise intervention experiments to observe model outputs when setting different neuron activations for the same prompt. Through experiments on six PLMs, including both encoder- and decoder-based models, we observe both linearity and polarity exist for neurons. The signed slope of the linear relationship between activation shifts and output probabilities is termed the neuron empirical gradient.

2.1 Neuron-wise Intervention Experiment Setup

Models. To develop a general and universal quantitative measurement of neuron effects, we experiment with two types of LMs: masked and causal LMs, with varied sizes. For masked LMs, we use three models from the BERT [11] family: $\text{BERT}_{\text{base}}$[*1], $\text{BERT}_{\text{large}}$[*2], and BERT_{wwm}[*3]. These models have different learning strategies and sizes. We construct masked prompts and let the model predict the masked token. We also conduct probing on recent LLMs. As causal models can only generate tokens in an auto-regressive manner, we follow the setting in [10] and utilize the instruction understanding ability of LLMs to generate single-token answers. Specifically, we examine three instruction-tuned LLMs from the Llama2 family [12], with sizes of 7B[*4], 13B[*5], and 70B[*6].

Dataset. We utilize MyriadLAMA [10], a multi-prompt knowledge probing dataset, for our experiments. MyriadLAMA offers diverse prompts for each relational factual knowledge, reducing the influence of specific linguistic expressions on probing results. For each PLM, we randomly sample 1,000 prompts from MyriadLAMA so that the PLM can correctly predict the target token. Our study focuses on single-token probing, where the target answer can be described by a single token. Since different models use different tokenizers, the probing prompts may vary across models.

[*1] https://huggingface.co/google-bert/bert-base-uncased
[*2] https://huggingface.co/google-bert/bert-large-uncased
[*3] https://huggingface.co/google-bert/bert-large-uncased-whole-word-masking
[*4] https://huggingface.co/meta-llama/Llama-2-7b-hf
[*5] https://huggingface.co/meta-llama/Llama-2-13b-hf
[*6] https://huggingface.co/meta-llama/Llama-2-70b-hf

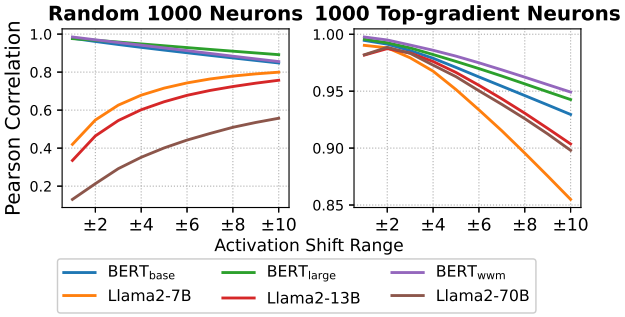

Fig. 2 The average of absolute Pearson correlations between activation shifts and output probabilities across 1000 neurons.

Protocol. We conduct neuron-wise intervention experiments to analyze how activation shift affects model outputs, establishing a numerical relationship between neuron activation and model generation. Specifically, we alter the neuron activations within a range of [-10, 10] with a step size of 0.2, to observe the resulting changes in target token output probabilities. Given that observing the effect of a single neuron on one token for one prompt requires 100 inference runs, we only perform the neuron-wise intervention on specific neurons to reduce the computational costs. We employ two strategies for neuron selection: random sampling and choosing the top-k neurons with the highest absolute computational gradients.

2.2 Linearity and Polarity of Neuron

On the basis of our observation of experimental data, we find that for some neurons, the activation shifts are largely linearly correlated with the target output probabilities. To quantify whether and under what conditions this linearity exists, we measured the Pearson correlation between the activation shifts and target output probabilities for parts of neurons.

Correlation vs. Shift range. We first investigate the range within which neurons exhibit linearity. We calculate the Pearson correlation between the shift ranges and the output probability of the correct tokens using 1000 neurons for each prompt. To focus on the correlation examination, we use the absolute value of the Pearson correlation. Finally, we average the correlations over 10 prompts with the same shift range. Figure 2 depicts the average of absolute Pearson correlations based on the two neuron selection methods. In BERT models, a large portion of neurons exhibits strong correlations between activation shifts and output probabilities, even with a broad range of 10 (the left-hand side of Figure 2).[*7]

[*7] We measure the mean, maximum, and minimum activations of 1000 prompts in $\text{BERT}_{\text{base}}$ and Llama2-7B. The average minimum, maximum,

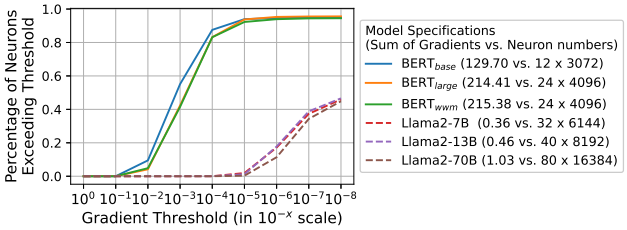

Fig. 3 Percentage of neurons exceeding threshold.

Regarding the impact of shift range on correlations, we observe consistently stronger correlations with smaller ranges in BERT models, whereas Llama2 models exhibit the opposite behavior. To understand this divergence, we analyze neuron gradient statistics. Specifically, we collect the absolute gradients of target tokens across all neurons in PLMs and measure the percentage of neurons exceeding specific gradient magnitudes. As shown in Figure 3, although the Llama2 models have significantly more neurons than BERT models, the sum of their absolute gradients is still two orders of magnitude smaller than that of BERT. This suggests that the neurons in Llama2 might be more influenced by random noise rather than true gradients. This could explain why correlations increase even with larger shift ranges: the increased number of data points (from multiple neuron-wise intervention experiments) likely reduces the impact of noise on the results.

To reduce the impact of noise on the correlation, we select neurons with high absolute gradient values. We use the gradient computed from the computational graph through network backpropagation (hereafter, "computational gradient"). Specifically, we measure the correlations from the 1,000 neurons with the highest absolute computational gradients (Figure 2, right). The result indicates that activation shifts tend to show stronger correlations with output tokens at smaller shift ranges, with consistency across six models. Specifically, when setting the range to ±2, the correlations in all models are close to 0.99, which we consider the threshold for indicating the linear relationship. Therefore, our subsequent analysis uses the top-gradient neurons within a shift range of ±2 by default.

On Neuron linearity. We then present a quantitative analysis of the prevalence of neuron linearity and the generality of these neurons across different prompts and Transformer layers. Specifically, we report the percentage of neurons exhibiting linearity, defined as having correlations equal to or greater than 0.99 within a shift range of ±2.

For neuron generality, which means the prevalence of linear neurons, we want to verify whether the linear neurons exist widely across different Transformer feed-forward layers and within different prompts. We use the metrics of layer generality (**LG**) and prompt generality (**PG**) to measure the prevalence of their existence. Intuitively, we can consider a simplified problem as follows: suppose we have many colored balls (green, blue, ...) and 10 bins, and if we want to verify whether the blue ball has "generality," it means (1) **high coverage**: the blue ball exists in most of the bins; (2) **even distribution**: the number of blue balls in each bin hardly differs from others. For our neuron generality, the "balls" are the "linear neurons," and the "bins" refer to either "feed-forward layers" (for **LG**) or "different prompt" (for **PG**). To address these two aspects simultaneously, we define **LG** and **PG** as follows:

and mean activations for BERT$_{base}$ are -0.17, 4.83, and -0.04, respectively. For Llama2-7B, these values are -21.6, 7.13, and 0.

	Linear neuron ratio	Prompt-wise gen.	Layer-wise gen.
BERT$_{base}$	89.01%	99.99%	98.22%
BERT$_{large}$	92.20%	99.99%	94.89%
BERT$_{wwm}$	99.99%	99.99%	94.93%
Llama2-7B	91.10%	99.99%	97.60%
Llama2-13B	92.90%	99.99%	98.23%
Llama2-70B	91.81%	99.99%	97.49%

Table 1 Neuron linearity statistics. We choose 1000 prompts and their corresponding 100 neurons randomly. For Llama2-70B, since the model is giant, we only chose 200 prompts and 100 neurons due to the high computational cost. The shift range is set to ± 2.

$$\textbf{LG} \triangleq \text{coverage}_{\text{layer}} \times \text{distribution}_{\text{layer}}, \tag{1}$$

$$\textbf{PG} \triangleq \text{coverage}_{\text{prompt}} \times \text{distribution}_{\text{prompt}}, \tag{2}$$

where coverage and distribution are defined as:

$$\text{coverage}_{\text{x}} = \frac{\Sigma_i \mathbb{1}(\text{linear neuron exists in } x_i)}{\#\text{ of } x}, \tag{3}$$

$$\text{distribution}_{x} = 1 - \frac{\text{Var}(\#\text{neurons in } x)}{\text{maxVar}(\#\text{neurons in } x)}, \tag{4}$$

where x refers to either layer or prompt, maxVar$(\cdot)$ denotes the max possible variance. High coverage and distribution are desirable; a perfect generality then achieves coverage of one and distribution of one.

Table 1 shows the statistics of neuron linearity: a large portion of neurons exhibit linearity, and the linear neurons are prevalent in most of the Transformer FFNs and prompts, which indicates the generality of linear neurons.

Neuron polarity. We then incorporate the direction of change in output probabilities into our numerical analysis. We denote neurons as *positive neurons* if increasing their activations could enhance the target output probabilities while decreasing it suppresses the output. In contrast, *negative neurons* have the opposite effect. Table 3 also shows that the number of positive neurons is nearly equivalent to negative neurons. This suggests that polarity is a general property of neurons in PLMs, and PLMs show no preference for either positive or negative neurons.

3. NeurGrad: Neuron Empirical Gradient

Assuming that neurons hold linearity and polarity, we quantify the neurons' ability to control output token as neuron empirical gradient. In this section, we propose **NeurGrad**, a simple yet efficient method to measure neurons' empirical gradient.

3.1 NeurGrad

The neuron empirical gradient demonstrates how significantly a neuron can alter the model's output. It offers a quantitative assessment of a neuron's importance in knowledge representation.

	G_C	IG.	$\bar{G}_E$
$\text{BERT}_{\text{large}}$	-.9307	.7360	.9998
$\text{BERT}_{\text{base}}$	-.8909	.7167	.9958
BERT_{wwm}	-.8914	.8584	.9989
Llama2-7B	.0115	.6728	.9769
Llama2-13B	-.0113	.6964	.9641
Llama2-70B	-.0391	-[*9]	.7811

Table 2 Pearson correlations between various measured gradients and empirical gradients for randomly sampled neurons.

However, conducting neuron-wise intervention experiments requires multiple inferences for a specific neuron, prompt, and output token. Consequently, the computational cost of measuring empirical gradients for all neurons across various factual prompts and answers becomes extremely high. Therefore, we introduce NeurGrad, a method for efficiently calculating neuron empirical gradients, detailed below:

$$\bar{G_E} = \text{sign}(A) \times -G_C, \tag{5}$$

$$\text{sign}(x) = \begin{cases} 1 & \text{if } x > 0, \\ 0 & \text{if } x = 0, \text{[*8]} \\ -1 & \text{if } x < 0. \end{cases} \tag{6}$$

where $\bar{G_E}, A, G_C$ represents the estimated neuron empirical gradient, activation, and computational gradient, respectively.

The proposal of NeurGrad comes from our observation that while the computational gradients can provide a close estimation of the volume of empirical gradient, it could make wrong of the neuron polarity. To verify the effectiveness of NeurGrad, we run neuron-wise intervention experiments on 1000 prompts, with 100 random neurons per prompt. The activation shift range is set to [-2, 2] according to § 2.2. We first calculate the neuron empirical gradients using the experimental data as the ground-truth data. Then, we estimate the empirical gradients using three different methods: computational gradient (G_c), integrated gradients (IG.) used for identifying knowledge neurons [2] that intervene neuron in small step sizes multiple times to simulate the gradient, and our NeurGrad method.

Table 2 reports the Pearson correlations between measured gradients and empirical gradients. The results indicate NeurGrad's superiority in accurately measuring empirical gradients. NeurGrad is also much more efficient than IG. Regarding efficiency, calculating IG requires multiple iterations, each involving changes to neuron activations. In contrast, NeurGrad completes the calculation with just one inference pass, resulting in a computational cost nearly identical to that of computational gradients.

3.2 Dynamic Knowledge Store Hypothesis

The empirical gradient of neurons reveals a perspective that differs from the existing explanations in knowledge representation [2, 13, 14, 15, 6]. These explanations, such as the knowledge neuron theory, posit that knowledge is decisively represented by a few neurons [2, 13, 14]. Some studies have also used activations as indicators of knowledge representation [15, 6], suggesting that if a neuron has a neuron activation of zero, it is not involved in representing the knowledge. We refer to this perspective as the static knowledge store hypothesis.

[*8] For neurons with zero activation, we set their empirical gradient as zero as the number of neurons with zero activation is relatively small: on average, there are 103, 1,037, and 46 neurons with zero activations in BERTbase, BERTlarge, and BERT_{wwm}, respectively. For Llama2 models, this number is consistently zero.

[*9] Due to the high memory cost of calculating integrated gradients, we cannot do the calculation on Llama2-70B under our server environment.

Model	Pos. ratio	Neg. ratio
BERT-base	50.19%	49.81%
BERT-large	50.08%	49.92%
BERT-wwm	49.96%	50.04%
Llama2-7B	46.04%	45.92%
Llama2-13B	46.64%	46.60%
Llama2-70B	44.84%	44.80%

Table 3 The ratio of positive and negative neurons over 1000 prompts in PLMs

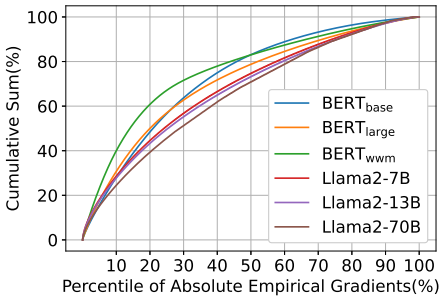

Fig. 4 Cumulative distribution of empirical gradient magnitudes, sorted by descending empirical gradient volume. The x-axis shows the percentiles of absolute empirical gradients, while the y-axis indicates the cumulative contribution of these gradients to the total magnitude.

The empirical gradient offers a dynamic knowledge store hypothesis: *the expression of knowledge in a model is not determinative but a balanced status that can be reimplemented by modifying neuron activations.* For instance, by simultaneously increasing the activations of both positive and negative neurons, the model can use different activations to achieve the same output probability. This hypothesis provides a different perspective from the statistical hypothesis. Firstly, our experiments show that setting the activations of different neurons from positive to zero yields different effects. For positive neurons, this suppresses the representation of knowledge, while for negative neurons, it activates the knowledge. We report the ratio of positive and negative neurons in Table 3. The percentage of positive and negative neurons is similar across the PLMs. All neurons in the BERT family exhibit non-zero empirical gradients, while only a small portion of neurons in Llama2 models show non-zero empirical gradients.

Secondly, we found that a substantial number of neurons are capable of altering the PLMs' output, indicating that while specific neurons can control the expression of certain knowledge, this relationship is not exclusive—other neurons also have this capacity. Figure 4 shows the cumulative distribution of empirical gradient magnitudes for all neurons in PLMs, calculated from 1,000 prompts and sorted in descending order. We can observe that although different PLM families have varying distributions of neuron empirical gradient values as shown in Figure 3, their cumulative distributions are similar. Moreover, the figure shows that the rising curves do not converge until all neurons are accounted for. This smooth, steady increase suggests that a wide range of

neurons can influence the PLMs' output. This suggests that there are no "decisive" knowledge neurons that can absolutely control knowledge representation while others have zero effect. Instead, knowledge representation in PLMs seems to emerge from the collective contributions of numerous neurons. The overall state of PLMs' ability to map factual inquiry to correct answers is balanced by the activations of many neurons rather than being dominated by a select few.

4. Related Work

The question of how to understand the underlying mechanism of LLMs has received extensive attention recently [4]. A paradigm for this kind of mechanism study is to first formulate some hypotheses to the underlying mechanisms, then use experiments to verify this hypothesis, and finally propose applications by utilizing the verified hypothesis. In what follows, we first introduce existing studies on mechanism interpretation (hypothesis) and then review methods for activation intervention (application).

4.1 Mechanism Interpretation

The great success of Transformer [1]-based PLMs [16, 17] has attracted substantial studies, and inspired the work focuses on knowledge probing [18, 2, 19], model behavior interpretation [6], and model manipulation (editing) [20, 5, 9]. Among all these papers, [6] proposed the feed-forward layers in Transformers serve as a key-value memory and store factual knowledge; following their work, [2] located in specific neurons related to certain factual knowledge, which they termed as "knowledge neuron" (KN), whose activation encodes the knowledge. They also concluded that there are only 4 to 5 KNs responsible for specific knowledge.

Inspired by these studies, rather than from a static point of view like [2], we explore the neuron activation in a dynamic view: by measuring the output probability's gradient while shifting the neuron activations. We found that most neurons demonstrate linearity, which is a general phenomenon regardless of model size or architecture.

4.2 Activation Intervention

Activation intervention is a method that modifies the activation of specific neurons in Transformer-based models, to manipulate model behavior. In previous work, [5] proposed a prompt-tuning-based method to probe certain neurons' activations that encode particular "skills"; while [21] focuses on attention heads' activations, and utilize a subset of attention heads' activations to intervene model's trustfulness. Compared with weight editing methods [9, 22, 23], which are usually related to backpropagation, activation intervention methods consume relatively less computational resources.

Our work also falls into the category of activation intervention. In this paper, we propose NeurGrad, a method for efficiently calculating neuron empirical gradients, that outperforms several baselines. We hope our work provides some insights into more efficient activation intervention research in the future.

5. Conclusion

Our study focuses on deepening the understanding of PLMs' mechanisms in storing knowledge in their parameters. Through neuron-wise neuron intervention using factual prompts, we reveal that significant neurons exhibit a linear relationship between the neuron activations and the PLM's output token probability. This linearity, observed across different PLMs, prompts, layers, and neurons, is described as neuron empirical gradients. Given the high computational cost of neuron-wise intervention, we propose an efficient and precise method, NeurGrad, to calculate these gradients. Our experiments show that NeurGrad outperforms baseline methods in both accuracy and efficiency.

6. Future Work

In our study, we only investigate the relationship between neuron activations and the model generations, while whether the neuron empirical gradients can represent the linguistic knowledge is still unexplored. Our future work involves deepening the understanding of the relationship between linguistic knowledge and neuron gradients.

7. Acknowledgements

This work was partially supported by the special fund of Institute of Industrial Science, The University of Tokyo, by JSPS KAKENHI Grant Number JP21H03494, and by JST, CREST Grant Number JPMJCR19A4, Japan.

References

[1] Vaswani, A.: Attention is all you need, *Advances in Neural Information Processing Systems* (2017).

[2] Dai, D., Dong, L., Hao, Y., Sui, Z., Chang, B. and Wei, F.: Knowledge Neurons in Pretrained Transformers, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Muresan, S., Nakov, P. and Villavicencio, A., eds.), Dublin, Ireland, Association for Computational Linguistics, pp. 8493–8502 (online), DOI: 10.18653/v1/2022.acl-long.581 (2022).

[3] Niu, J., Liu, A., Zhu, Z. and Penn, G.: What does the Knowledge Neuron Thesis Have to do with Knowledge?, *The Twelfth International Conference on Learning Representations*, (online), available from ⟨https://openreview.net/forum?id=2HJRwwbV3G⟩ (2024).

[4] Wang, M., Yao, Y., Xu, Z., Qiao, S., Deng, S., Wang, P., Chen, X., Gu, J.-C., Jiang, Y., Xie, P., Huang, F., Chen, H. and Zhang, N.: Knowledge Mechanisms in Large Language Models: A Survey and Perspective, *Findings of the Association for Computational Linguistics: EMNLP 2024* (Al-Onaizan, Y., Bansal, M. and Chen, Y.-N., eds.), Miami, Florida, USA, Association for Computational Linguistics, pp. 7097–7135 (online), available from ⟨https://aclanthology.org/2024.findings-emnlp.416⟩ (2024).

[5] Wang, X., Wen, K., Zhang, Z., Hou, L., Liu, Z. and Li, J.: Finding Skill Neurons in Pre-trained Transformer-based Language Models, *Proceedings of the 2022 Con-*

ference on Empirical Methods in Natural Language Processing (Goldberg, Y., Kozareva, Z. and Zhang, Y., eds.), Abu Dhabi, United Arab Emirates, Association for Computational Linguistics, pp. 11132–11152 (online), DOI: 10.18653/v1/2022.emnlp-main.765 (2022).

[6] Geva, M., Schuster, R., Berant, J. and Levy, O.: Transformer Feed-Forward Layers Are Key-Value Memories, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (Moens, M.-F., Huang, X., Specia, L. and Yih, S. W.-t., eds.), Online and Punta Cana, Dominican Republic, Association for Computational Linguistics, pp. 5484–5495 (online), DOI: 10.18653/v1/2021.emnlp-main.446 (2021).

[7] Bills, S., Cammarata, N., Mossing, D., Tillman, H., Gao, L., Goh, G., Sutskever, I., Leike, J., Wu, J. and Saunders, W.: Language models can explain neurons in language models, https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html (2023).

[8] Huang, J., Geiger, A., D'Oosterlinck, K., Wu, Z. and Potts, C.: Rigorously Assessing Natural Language Explanations of Neurons (2023).

[9] Meng, K., Bau, D., Andonian, A. and Belinkov, Y.: Locating and editing factual associations in GPT, *Advances in Neural Information Processing Systems*, Vol. 35, pp. 17359–17372 (2022).

[10] Zhao, X., Yoshinaga, N. and Oba, D.: What Matters in Memorizing and Recalling Facts? Multifaceted Benchmarks for Knowledge Probing in Language Models, *Findings of the Association for Computational Linguistics: EMNLP 2024* (Al-Onaizan, Y., Bansal, M. and Chen, Y.-N., eds.), Miami, Florida, USA, Association for Computational Linguistics, pp. 13186–13214 (online), available from ⟨https://aclanthology.org/2024.findings-emnlp.771⟩ (2024).

[11] Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (Burstein, J., Doran, C. and Solorio, T., eds.), Minneapolis, Minnesota, Association for Computational Linguistics, pp. 4171–4186 (online), DOI: 10.18653/v1/N19-1423 (2019).

[12] Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S. et al.: Llama 2: Open Foundation and Fine-Tuned Chat Models (2023).

[13] Geva, M., Caciularu, A., Wang, K. and Goldberg, Y.: Transformer Feed-Forward Layers Build Predictions by Promoting Concepts in the Vocabulary Space, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing* (Goldberg, Y., Kozareva, Z. and Zhang, Y., eds.), Abu Dhabi, United Arab Emirates, Association for Computational Linguistics, pp. 30–45 (online), DOI: 10.18653/v1/2022.emnlp-main.3 (2022).

[14] Yu, Z. and Ananiadou, S.: Neuron-Level Knowledge Attribution in Large Language Models, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing* (Al-Onaizan, Y., Bansal, M. and Chen, Y.-N., eds.), Miami, Florida, USA, Association for Computational Linguistics, pp. 3267–3280 (online), available from ⟨https://aclanthology.org/2024.emnlp-main.191⟩ (2024).

[15] Voita, E., Ferrando, J. and Nalmpantis, C.: Neurons in Large Language Models: Dead, N-gram, Positional, *Findings of the Association for Computational Linguistics: ACL 2024* (Ku, L.-W., Martins, A. and Srikumar, V., eds.), Bangkok, Thailand, Association for Computational Linguistics, pp. 1288–1301 (online), DOI: 10.18653/v1/2024.findings-acl.75 (2024).

[16] Devlin, J.: Bert: Pre-training of deep bidirectional transformers for language understanding, *arXiv preprint arXiv:1810.04805* (2018).

[17] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W. and Liu, P. J.: Exploring the limits of transfer learning with a unified text-to-text transformer, *Journal of machine learning research*, Vol. 21, No. 140, pp. 1–67 (2020).

[18] Petroni, F., Rocktäschel, T., Lewis, P., Bakhtin, A., Wu, Y., Miller, A. H. and Riedel, S.: Language models as knowledge bases?, *arXiv preprint arXiv:1909.01066* (2019).

[19] Zhao, X., Yoshinaga, N. and Oba, D.: Tracing the Roots of Facts in Multilingual Language Models: Independent, Shared, and Transferred Knowledge, *arXiv preprint arXiv:2403.05189* (2024).

[20] Mitchell, E., Lin, C., Bosselut, A., Finn, C. and Manning, C. D.: Fast model editing at scale, *arXiv preprint arXiv:2110.11309* (2021).

[21] Li, K., Patel, O., Viégas, F., Pfister, H. and Wattenberg, M.: Inference-time intervention: Eliciting truthful answers from a language model, *Advances in Neural Information Processing Systems*, Vol. 36 (2024).

[22] Orgad, H., Kawar, B. and Belinkov, Y.: Editing Implicit Assumptions in Text-to-Image Diffusion Models (2023).

[23] Meng, K., Sen Sharma, A., Andonian, A., Belinkov, Y. and Bau, D.: Mass Editing Memory in a Transformer, *The Eleventh International Conference on Learning Representations (ICLR)* (2023).