The Importance of Prompt Tuning for Automated Neuron Explanations

Justin Lee[*] Mt. Carmel High School
Tuomas Oikarinen[*] UC San Diego
Arjun Chatha Canyon Crest Academy
Keng-Chi Chang[†] UC San Diego
Yilan Chen[†] UC San Diego
Tsui-Wei Weng UC San Diego

Abstract

Recent advances have greatly increased the capabilities of large language models (LLMs), but our understanding of the models and their safety has not progressed as fast. In this paper we aim to understand LLMs deeper by studying their individual neurons. We build upon previous work showing large language models such as GPT-4 can be useful in explaining what each neuron in a language model does. Specifically, we analyze the effect of the prompt used to generate explanations and show that reformatting the explanation prompt in a more natural way can significantly improve neuron explanation quality and greatly reduce computational cost. We demonstrate the effects of our new prompts in three different ways, incorporating both automated and human evaluations.

1 Introduction

Large language models (LLMs) have exhibited remarkable capabilities across a variety of domains and tasks, such as text generation, question answering, and language translation. As an example, GPT-4 [10] exhibits human-level performance on various professional and academic benchmarks, including passing a simulated bar exam with a score around the top 10% of test takers [10]. Even more, [3] shows that GPT-4 can solve novel and difficult tasks that span mathematics, coding, vision, medicine, law, psychology, and more, viewing it as an early version of artificial general intelligence.

With the popularity of LLMs and their use in safety-critical and fairness-related applications such as healthcare [13, 14], education [6, 15], law [10, 19] and finance [17, 18], it is crucial to understand the models better and ensure their safety. Understanding how LLMs make their decisions can help us decide when to trust model predictions, detect bias in a network, and allow for greater control of the behavior the model exhibits. Recently, [2] used GPT-4 to automatically write explanations for the behavior of neurons in large language models and to score those explanations, scaling an interpretability technique to all the neurons in a LLM. While impressive, their approach is still preliminary, and vast majority of the neurons cannot be explained well using this approach. This is in part due to some neurons simply not having a simple function that can be explained, but in part failures of the method to detect these roles.

In this paper, we improve on existing methods [2] to provide more accurate automated neuron explanations. Specifically, we propose 4 new and cost-effective prompts to improve the quality of single neuron explanations in LLMs. Our extensive experiments show that our proposed methods outperform the state-off-the-art [2] and can improve explanation performance over [2] in terms of both automated and human evaluations, while being 2-3$\times$ more efficient and cost-effective.

[*]Shared First-authorship
[†]Equal contribution

ATTRIB: Workshop on Attributing Model Behavior at Scale at NeurIPS 2023.

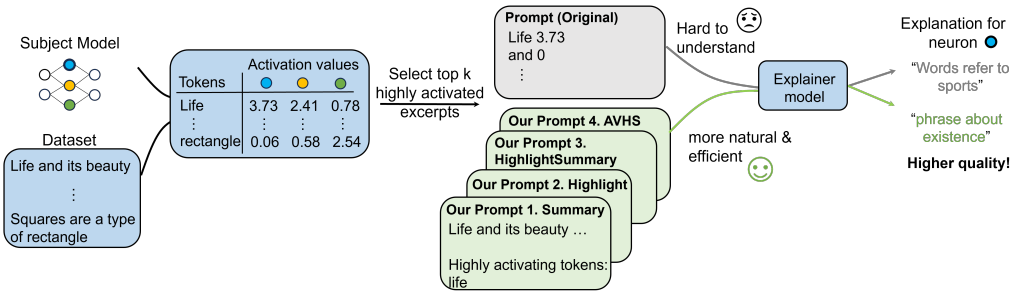


Figure 1: Overview of the neuron explanation pipeline and our proposed prompts (in green) to highly improve the explanation quality and efficiency.

2 Background and related work

Mechanistic and Neuron-level Interpretability.

A growing literature attempts to reverse engineer deep neural networks to understand the principles behind their operation [8, 4, 9]. One foundation behind this approach is for humans to understand the function of individual neurons. However, given the sheer number of neurons in modern networks, having humans generate descriptions for each neuron is extremely labour intensive, even if the neurons are interpretable. To address this, some methods for generating automatic explanations have been proposed for vision models [1, 5, 7]. Recently, a similar approach was proposed for language models, where [2] use GPT-4 to automatically write explanations for the behavior of neurons in large language models (GPT-2) and score those explanations. In addition to [2], a host of previous method have been developed for understanding individual neurons in LLMs, but many of them are not automated or can only detect simpler concepts like single tokens. See [12] for a comprehensive overview of previous approaches.

Explaining neurons in LLMs with GPT.

In [2], the team from OpenAI showcases that GPT-4 can be useful in describing the roles of individual neurons in language models. Specifically, they focused on explaining the neurons in MLP layers of GPT2-XL[11], which is called the subject model (i.e. the model to be dissected and interpreted). The basic idea is to run a large text corpus $\mathcal{D}$ of text excerpts through the model, and record how highly each individual neuron activates for each token. In particular they used 60,000 random excerpts of 64 tokens each from the model's training data. They then find the 5 excerpts with the highest individual token activations for a specific neuron. These excerpts are then fed to the explainer model(GPT-4) together with the neuron's activation pattern following the prompt pattern shown in Fig.2. The explainer model uses this information to generate a simple description of this neurons behavior.

GPT models. GPT models are part of the broader family of transformer architectures [16], which utilize attention mechanisms to process and generate text. A decoder-only transformer, such as GPTs, starts with a token embedding, followed by a series of "residual blocks". Each residual block contains an attention layer, followed by an MLP layer. The attention layer computes weights for the model about which part of the input tokens to focus on and adjust the residuals streams accordingly. The MLP layer then calculates activation based on the updated residual stream. Finally, the residual stream is projected back to get the probability of next tokens. In this paper we focus on analyzing the neurons in the MLP layers.

3 Methodology

Motivation. As introduced in the previous section, [2] propose to use the *explainer model* (GPT-4) to explain the neurons in the *subject model* (GPT-2). To allow the explainer model to generate explanations of subject model neurons, [2] sends the explainer model a list of (token, activation) pairs separated by tabs and newlines as shows in Figure 2, where we call their method **Original prompt**, abbreviated as **Original**.

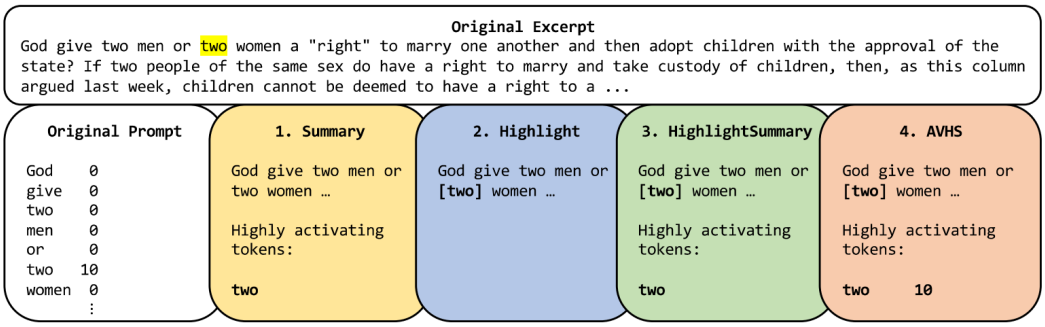


Figure 2: An overview of our proposed prompting methods, compared to the original prompt. Boldfacing is only done for the visual clarity, not part of actual prompt.

However, we find that this kind of prompt has several drawbacks: (i) this requires 4 times as many tokens as the original text from $\mathcal{D}$, which results in large overhead and (ii) the text becomes unnatural/hard for a human to follow as it is interspersed with activation values all the time.

To address these limitations, in this work we propose several simpler prompting methods, which have *higher computation efficiency* and are more *natural*. More importantly, we show in the experiments that our proposed prompts can improve the quality of neuron explanations compared to [2].

3.1 Proposed Approach

Our approach builds on [2] described in the previous section. Our goal is to test modifications to their pipeline to improve the quality and/or efficiency of the generated explanations. Specifically, we focused on changing the neuron explanation prompt given to explainer model.

Below we propose 4 new prompting strategies, which are showcased in Figure 2 along with the **Original prompt** from [2]:

1. **Summary**: This prompt greatly simplifies the presentation, by just showing the original text excerpt and repeating a list of highly activating tokens (90% quantile or above).
2. **Highlight:** This is an alternative simplification, where we only show the original text but add square brackets around any highly activating tokens.
3. **HS (Highlight Summary)**: In this prompt, we combine our two approaches above, adding square brackets to highlight highly activating tokens in the text as well as a list of the highly activating tokens after the text excerpt.
4. **AVHS** (Activation Value + Highlight Summary): A combination of the previous prompts (with variations). This approach is similar to highlight summary, but also provides the values for the highly activating tokens – which is similar but more compact than the original prompt as the original prompt also provides 0 activation values. Hence, this prompt provides the same amount of information as original but in a more concise and readable form.

3.2 Computational efficiency

In Table 1, we display the average prompt length needed for each neuron explanation(averaged over 50 neurons), including the few-shot examples given in the prompt. As we can see, our prompts require 2-3× less tokens than original. This is important for a few reasons: first, the cost of generating explanations is calculated per token when using the API, so a 2× reduction in tokens per neuron mean you can evaluate 2× as many neurons for the same budget.

Second, this could also lead to improvement in explanation performance. The prompt length is bounded by the model's context window, which is around 4096 tokens for most of

	Tokens per prompt	Improvement
Original [2]	2338	-
Summary	959	2.44×
Highlight	**886**	**2.64**×
HS	1032	2.27×
AVHS	1360	1.72×

Table 1: Average prompt length of different prompting methods. Our proposed prompts lead to 1.72-2.64× lower computatinal cost.

the models we used. Since the length of original prompt is already close to this limit, it cannot be made much longer. With our prompts however, the prompt could be made longer by for example including additional few-shot examples, or giving the model more and/or longer text excerpts to summarize. This could improve the overall quality of the explanations.

3.3 Evaluation methods

Method 1: Simulate and Score. The original OpenAI work [2] mainly relied on simulation to test how good their explanations are. In this approach, after an explanation is generated a simulator model (also GPT-4 in their case) uses this explanation to predict a neuron's activations on unseen text excerpts. These predicted activations are then compared against actual neuron activations, and the explanation is scored based on the correlation between predicted activations and real activations. See [2] for more details. However, this method has a high computational cost, and we are unable to use GPT-4 as the simulator model as token probabilities are not available through the API. Nevertheless, we used simulation as one of our evaluation methods, utilizing the recently released *GPT-3.5-Turbo-Instruct* model as the simulator, which provides good simulation accuracy at a reasonable cost.

Method 2: Similarity to baseline explanation (AdaCS). As an alternative to the high computational cost of simulation, we developed a cheap and fast method so that we can compare different prompting methods at scale. To accomplish this, we decided to measure how similar our generated explanations are to provided explanations from OpenAI [2], available on NeuronViewer. Since these explanations are not ground truth but generated by their method, we mostly restricted our attention to neurons with high simulation scores, which indicates their descriptions are highly accurate. The descriptions on NeuronViewer are generated using GPT-4 and include additional refining steps, so we expect them to be of higher quality than the descriptions we generate using GPT-3.5. Therefore, similarity to NeuronViewer description indicates the description is better. To compare the similarity of the descriptions, we use similarity in sentence embedding space. This is done using AdaV2 sentence embedding model from OpenAI, which we found to provide high quality similarity scores. Different prompting methods were then ranked based on how similar their descriptions are to the NeuronViewer explanation for that neuron, measured by cosine similarity of their embeddings. In addition to the real neurons, we used AdaCS to evaluate answers the different prompts provided to **Neuron Puzzles**. Neuron Puzzles are a set of artificial activation patterns created by [2], which were handcrafted to correspond to an interesting ground truth role for a neuron. In this case we are able to use AdaCS to measure the similarity of the generated description to the ground truth description.

Method 3: Human evaluation. Finally we conducted a randomized study where the authors evaluated explanations for different neurons and compared explanations from different prompting methods. During the experiment, users had no way of knowing which explanation came from which method. The users were asked to evaluate 5 different explanations, and rate how good they were on a scale of 1-5, based on the neuron's activations visualized on NeuronViewer, as well as select which explanation was the best. An overview of our interface is shown in Figure 4.

4 Experiments and Results

Setup. In this section we evaluate the different prompting strategies using 4 different comparison methods which complement each other. As baseline, we used OpenAI's original code and prompt formatting to generate the explanations. We modified the explaining code available on GitHub to experiment with our 4 new prompt formats in addition to the original. All prompts used the same few-shot examples as the original (but formatted according to the proposed new prompt).

Explainer model: We used both GPT-4 and GPT-3.5-turbo models as the explainer. While GPT-4 provides better explanations, it wasn't available at the start of this project and has a much higher usage cost. Thus we used GPT-3.5 for large scale experiments, and GPT-4 with smaller scale tests. See Table 7 for an overview of our costs.

Neurons evaluated: We followed [2] and described neurons in the MLP layers of GPT-2(XL). This network contains 48 layers each with 6400 MLP neurons each, amounting to 307,200 neurons total. Due to limited computational budget, we evaluated several subsets of these neurons:

1. **Random**: We randomly sampled an equal number of neurons from each of the 48 layers (20 per layer for GPT-3.5, 10 for GPT-4).

2. **Random interpretable**: We randomly sampled neurons that had a score > 0.35 on NeuronViewer, indicating these neurons are more interpretable (20 per layer for GPT-3.5, 10 for GPT-4).
3. **Top20 per layer**: We found the 20 neurons in each layer that had the highest simulation score, which is a proxy for most interpretable neurons of each layer.
4. **Top 1k**: The 1,000 neurons in the model that had the highest simulation scores on NeuronViewer. These were mostly in the early layers as shown in Figure 3 in Appendix.

In addition to randomly sampled neurons, we chose to focus on neurons that were somewhat well explained by [2] as indicated by a high score on NeuronViewer. This was done to focus on more interesting and interpretable neurons.

	Explainer: GPT-3.5			Explainer: GPT-4		
Prompt:	Random	Random Interp.	Avg	Random	Random Interp.	Avg
Original [2]	0.0962	0.2420	0.1718	0.1211	**0.2284**	**0.1745**
Summary	**0.1249**	**0.2933**	**0.2123**	0.1238	0.2252	0.1742
Highlight	0.1129	0.2603	0.1893	**0.1241**	0.2237	0.1737
HS	0.1239	0.2833	0.2065	0.1193	0.2173	0.1681
AVHS	0.1093	0.2791	0.1974	0.1219	0.2216	0.1715

Table 2: Simulate and score results. We can see summary performs clearly best for GPT-3.5, while GPT-4 results are quite similar across methods. Standard error of the mean was 0.0030-0.0041 for GPT-3.5 results and 0.0039-0.0055 for GPT-4 results.

4.1 Simulate and score

Table 2 displays the average scores across different settings when using simulate and score as described in section 3.3. We can see Summary performs clearly the best when GPT-3.5 is the explainer model, with all our methods outperforming Original with statistical significance. When using GPT-4 on the other hand, Original performs the best on Random Interpretable neurons. However this is biased because Random Interpretable only selects neurons that originally received high simulation score (Using GPT-4 as the explainer and Original Prompt), and the effect goes away on complete random neurons. Finally, we note that the different methods perform more similarly when using GPT-4, with no statistically significant differences overall.

4.2 AdaCS

We generated explanations for neurons from all the different settings above, explaining a total of 3880 neurons with GPT-3.5 as the explainer model. We then compared these to the GPT-4 produced explanations from [2] using AdaCS, and report the results in Table 3. We can see all our proposed prompts produce explanations significantly closer to GPT-4 explanations than original in all settings, with Summary and HighlightSummary performing the best.

Table 4 shows the results on neuron puzzles, where the ground truth function is known, measured in AdaCS similarity of the produced explanation to ground truth. We can see our methods, especially

Prompt\Neurons	Random	Random interpretable	Top20 per layer	Top 1k	**Avg**
Original [2]	0.8167 ± 0.0016	0.8369 ± 0.0020	0.8521 ± 0.022	0.8820 ± 0.0025	0.8469
Summary	0.8471 ± 0.0016	$\mathbf{0.8790 \pm 0.0015}$	0.8904 ± 0.0017	$\mathbf{0.9026 \pm 0.0019}$	0.8798
Highlight	0.8460 ± 0.0017	0.8700 ± 0.0017	0.8818 ± 0.0016	0.8930 ± 0.0021	0.8727
HS	$\mathbf{0.8496 \pm 0.0016}$	0.8782 ± 0.0015	$\mathbf{0.8924 \pm 0.0014}$	0.8995 ± 0.0020	**0.8799**
AVHS	0.8422 ± 0.0016	0.8667 ± 0.0018	0.8826 ± 0.0016	0.8961 ± 0.0015	0.8719

Table 3: Average cosine similarity of generated explanations to baseline, using GPT-3.5 as the explainer model, as well as standard error of the mean. We can see all our methods noticeably outperform original.

Summary and HighlightSummary outperform Original quite clearly using both explainers. However due to only few puzzles being available, these results are on the edge of statistical significance.

4.3 Human evaluation

We had 5 authors evaluate the descriptions provided by different prompts in a randomized setting as described in section 3.3. We had authors evaluate as judging description quality is often quite complex and requires close attention and expertise, making it not suitable for crowdsourcing. The results are shown in Table 5. Each user rated 1 Randomly Interpretable neuron (>0.35 score) per layer, for a total of 240 evaluations per explainer model. We see that when using GPT-3.5 as the explainer, users largely preferred our new prompts, Summary being the best, followed HS, Highlight, AVHS and finally Original. When GPT-4 was used as the explainer instead, Summary was still rated the highest, but no explanation was better than others with statistical significance at this sample size. These findings indicate that simplifying the information provided to the model can lead to large gains, at least with less powerful explainer models, while the difference between prompts reduces when using GPT-4.

Prompt:	Explainer: GPT-3.5	Explainer: GPT-4
Original [2]	0.8418	0.8560
Summary	0.8495	**0.8657**
Highlight	0.8414	0.8601
HS	**0.8514**	0.8636
AVHS	0.8490	0.8640

Table 4: Results on Neuron puzzles. Each score is average cosine similarity to ground truth explanation, across all 19 puzzles, with 3 explanations generated per puzzle per method. The standard error of the mean was 0.0042-0.0054 for all entries.

	Explainer: GPT-3.5		Explainer: GPT-4	
Prompt:	Avg rating:	% chosen best	Avg rating:	% chosen best
Original [2]	3.487 ± 0.075	14.17%	4.367 ± 0.048	20.42%
Summary	$\mathbf{4.308 \pm 0.048}$	**32.50%**	$\mathbf{4.396 \pm 0.047}$	20.42%
Highlight	4.054 ± 0.056	17.92%	4.271 ± 0.054	17.92%
HS	4.196 ± 0.058	20.42%	4.350 ± 0.050	**21.67%**
AVHS	3.842 ± 0.066	15.00%	4.312 ± 0.052	19.58%

Table 5: User study ratings averaged over evaluated neurons. We can see Summary performs the best for both explainer models, with large margin when using GPT-3.5. Avg rating shown together with standard error of the mean.

5 Conclusions

We have summarized all our experimental results into Table 6. We can see that on average, the simple *Summary* functions performs clearly the best, ranking first or second on every evaluation metric. Based on these findings, and Table 1 showing *Summary* is the second most computationally efficient and $2.44\times$ more computationally efficient than original, we think it is the best choice for neuron explanations. Interestingly additional information from HS or AVHS was somewhat harmful to model performance, suggesting that explainer models struggle to handle too much complexity.

	GPT-3.5 Simulate	GPT-3.5 AdaCS	GPT-3.5 Puzzles	GPT-3.5 H. eval.	GPT-4 Simulate	GPT-4 Puzzles	GPT-4 H. eval.	Avg
Original	5	5	4	5	**1**	5	2	3.86
Summary	**1**	2	2	**1**	2	**1**	**1**	**1.43**
Highlight	4	3	5	3	3	4	5	3.86
HS	2	**1**	**1**	2	5	3	3	2.43
AVHS	3	4	3	4	4	2	4	3.43

Table 6: Summary of our results. Each value is the rank of the method according to that evaluation. We can see that across all evaluations, *Summary* performs the best out of all prompting methods.

Acknowledgement

The authors would like to thank San Diego Supercomputer Center, Halicioglu Data Science Institute and NSF's ACCESS program for computing support. T. Oikarinen and Y. Chen are supported by National Science Foundation under Grant No. 2107189. T.-W. Weng is supported by National Science Foundation under Grant No. 2107189 and 2313105.

References

[1] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Computer Vision and Pattern Recognition*, 2017.

[2] Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. Language models can explain neurons in language models. `https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html`, 2023.

[3] Sebastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

[4] 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. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. https://transformer-circuits.pub/2021/framework/index.html.

[5] Evan Hernandez, Sarah Schwettmann, David Bau, Teona Bagashvili, Antonio Torralba, and Jacob Andreas. Natural language descriptions of deep visual features. In *International Conference on Learning Representations*, 2022.

[6] Kamil Malinka, Martin Peresini, Anton Firc, Ondrej Hujnak, and Filip Janus. On the educational impact of chatgpt: Is artificial intelligence ready to obtain a university degree? In *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1*, pages 47–53, 2023.

[7] Tuomas Oikarinen and Tsui-Wei Weng. CLIP-dissect: Automatic description of neuron representations in deep vision networks. In *The Eleventh International Conference on Learning Representations*, 2023.

[8] Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 2020. https://distill.pub/2020/circuits/zoom-in.

[9] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads. *Transformer Circuits Thread*, 2022. https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.

[10] OpenAI. Gpt-4 technical report. arXiv preprint arXiv:2303.08774, 2023.

[11] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 2019.

[12] Hassan Sajjad, Nadir Durrani, and Fahim Dalvi. Neuron-level interpretation of deep nlp models: A survey, 2022.

[13] Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, et al. Large language models encode clinical knowledge. *Nature*, pages 1–9, 2023.

[14] Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Le Hou, Kevin Clark, Stephen Pfohl, Heather Cole-Lewis, Darlene Neal, et al. Towards expert-level medical question answering with large language models. *arXiv preprint arXiv:2305.09617*, 2023.

[15] Kehui Tan, Tianqi Pang, and Chenyou Fan. Towards applying powerful large ai models in classroom teaching: Opportunities, challenges and prospects. *arXiv preprint arXiv:2305.03433*, 2023.

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[17] Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. Bloomberggpt: A large language model for finance. *arXiv preprint arXiv:2303.17564*, 2023.

[18] Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. Fingpt: Open-source financial large language models. *arXiv preprint arXiv:2306.06031*, 2023.

[19] Fangyi Yu, Lee Quartey, and Frank Schilder. Legal prompting: Teaching a language model to think like a lawyer. *arXiv preprint arXiv:2212.01326*, 2022.

A Appendix

A.1 Total cost and computation

Table 7 displays an estimated computational cost and time for different steps of our research, when using the OpenAI API. These costs are for running all 5 prompting methods.

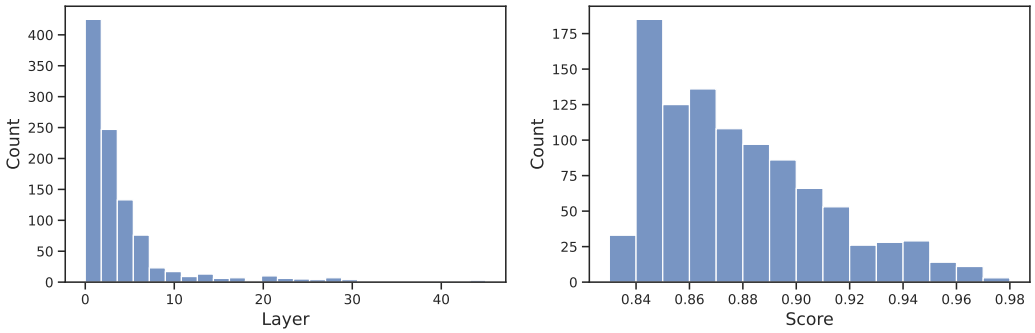
Model	Setting	Cost per 1k neurons	Runtime per 1k neurons
GPT-3.5-Turbo	Explain	$2.50	3 hr
GPT-3.5-Turbo-Instruct	Simulate+Score	$80	6 hr
GPT-4	Explain	$200	12 hr
Ada	Compare	$0.02	1 hr

Table 7: Cost and runtime estimates in different settings

A.2 Additional figures

Figure 3a shows the distribution of topk1k neurons, i.e. the ones with highest explanations scores according to [2]. The majority of Top 1k neurons comes from the first few layers of GPT according. There is a notable spike in Layer 1 (layers are start from 0) meaning that the activating texts in Layer 1 are on average easier to explain. The scores for top1k range from 0.83 to 0.98.

Figure 4 shows the interface used for our user study.



(a) Distribution of top 1k neurons by layers

(b) Distribution of top 1k neurons by scores

Figure 3: Distribution of the Top 1k neurons by layers and scores

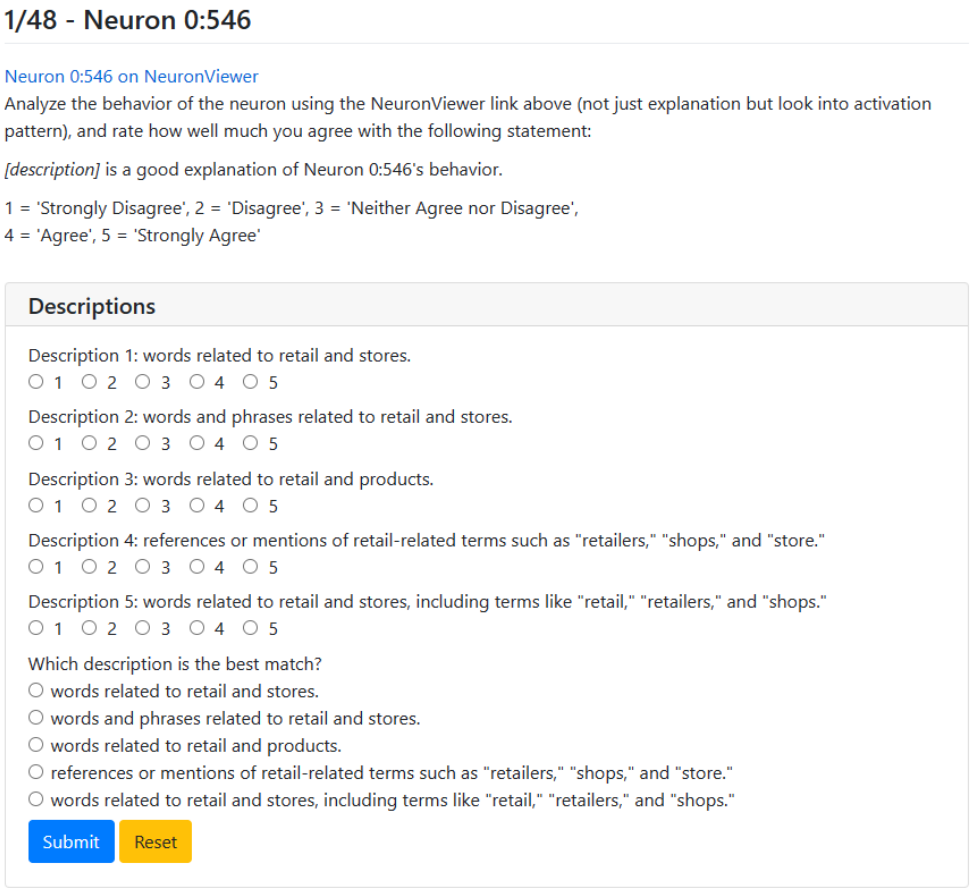


Figure 4: User study interface. To be viewed together with Neuron's activation patterns on NeuronViewer.