Predicting Paragraphs from LLM Token Activations

Nicholas Pochinkov
Independent
work@nicky.pro

Angelo Benoit
Independent
angelogbenoit@gmail.com

Lovkush Agarwal
Independent
lovkush@gmail.com

Zainab Ali Majid
Independent
zainab_majid@hotmail.com

Lucile Ter-Minassian
University of Oxford
lucile.ter-minassian@spc.ox.ac.uk

Abstract

Generative large language models (LLMs) excel in natural language processing tasks, yet their inner workings remain underexplored beyond token-level predictions. This study investigates the degree to which these models decide the content of a paragraph at its onset, shedding light on their contextual understanding. By examining the information encoded in single-token activations, specifically the "\n\n" double newline token, we demonstrate that patching these activations can transfer significant information about the context of the following paragraph, providing further insights into the model's capacity to plan ahead.

1 Introduction

Recent advancements in large language models have revolutionized Natural Language Processing, enabling unprecedented performance on a wide range of tasks, including machine translation [22; 16], question answering [3; 2], and text generation [15; 2]. Despite these successes, our understanding of how these models internally process and represent information remains limited [13; 24].

Previous studies have demonstrated that internal model representations can reveal how models plan ahead in text generation. By *intervening* on neural activations—specifically by patching them between different locations at inference time - we can uncover existing causal relationships [29; 26; 21; 19; 7]. For instance, *Pal et al.* use causal intervention methods in their *Future Lens* approach [14] to show that individual hidden states at position t contain signals rich enough to predict future tokens at $t + 2$ or beyond, and this insight has been used to improve performance of models [6; 1]. However, existing interpretability research predominantly focuses on token-level predictions by examining how models predict individual words or tokens [11], rather than exploring broader contexts such as the thematic coherence of a sentence or paragraph.

Our work aims to bridge the gap between token-level and paragraph-level understanding by investigating whether the information content of single-token activations remains relevant when we consider sequences of tokens, with a specific focus on the "\n\n" double newline token. We hypothesize that these activations contain information about the structure and content of the following paragraph, providing insight into the model's comprehension of larger textual units.

In section 2, we demonstrate through a preliminary experiment that text structure is embedded in a language model's attention scores. In section 3, we examine the extent to which a model, at the start of a paragraph, has already planned the rest of the generated text. To explore this, we patch activations onto a model with a neutral prompt – a double newline – and investigate whether the future paragraph contains information transferred at the hidden representation level. The code for our experiments is available anonymously. Compute details can be found in Appendix B.

Preprint. Under review.

2 Is Text Structure Encoded in the Model's Attention Patterns?

To motivate our approach, we first demonstrate that sequences of paragraphs can be identified through the analysis of an LLM's attention activations. We generate texts by prompting a model with instructions phrased as: "Tell me about topic 1 in k words \n\n tell me about topic 2 in k words." These generated texts, referred to as *original* contexts, are structured uniformly by instructing the model not to generate headings and additional comments. We then extract and inspect the combined attention patterns across all heads for each model-generated text. To observe the context switch, we conduct two key analyses, averaging across the textual generations: (1) the distribution of attention weights close to the topic change, and (2) the cosine similarity of attention output activations inside and across paragraphs, or topics. Experiment (1) checks to what extent attention heads focus on the current paragraph, whilst (2) investigates if attention outputs differ between paragraphs.

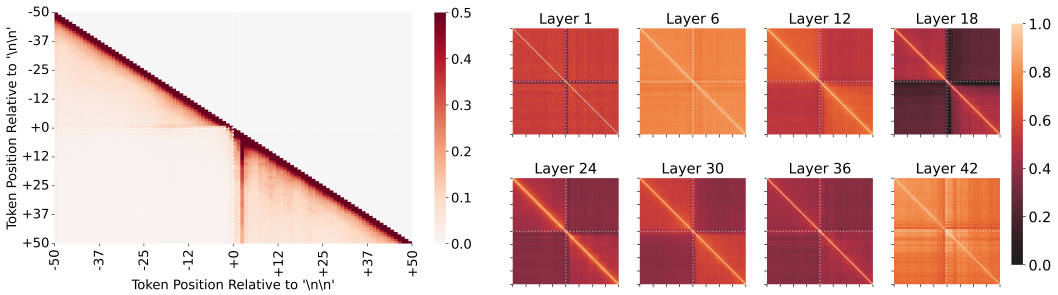


Figure 1: (Left): Heat map of the average attention weights around the topic change. (Right): Cosine similarity between attention activations. Results averaged over 1,000 model-generated original contexts, sharing a common structure.

Figure 1 shows the results of our attention pattern analysis. In our study, we used 20 prompts (i.e., pairs of topics), generating 50 texts per prompt, for a total of 1,000 generated texts. The generations were implemented with the Gemma 2, 9b model at a temperature of 0.3, and activations were retrieved using the HuggingFace Transformers library [25]. On the left, the attention weights indicate that the model tends to attend to previous tokens almost exclusively from the same paragraph. On the right, the cosine similarities of attention outputs show how strongly text structure is encoded across various layers. In the first 18 layers, the cosine similarities of attention activations increase within paragraphs and decrease across paragraphs, suggesting that the model is learning abstract representations in early layers, where it gradually develops an understanding of the paragraph topic. Another consistent finding across all experimental settings is that distinctions between paragraphs diminish in the final layers, from layer 30 onwards. We conjecture that this may be due to the model eventually producing text of a very similar overall form for both topics. An additional plot displaying the cosine similarities for all 42 model layers can be found in Appendix A. Altogether, our preliminary experiments suggest that our model maintains a strong contextual awareness during text generation, in line with research allowing consistent text embed fine-tuning [28], and "planning" [9; 26; 8]. These results also confirm that the context switch at the start of a paragraph is encoded in the activation space.

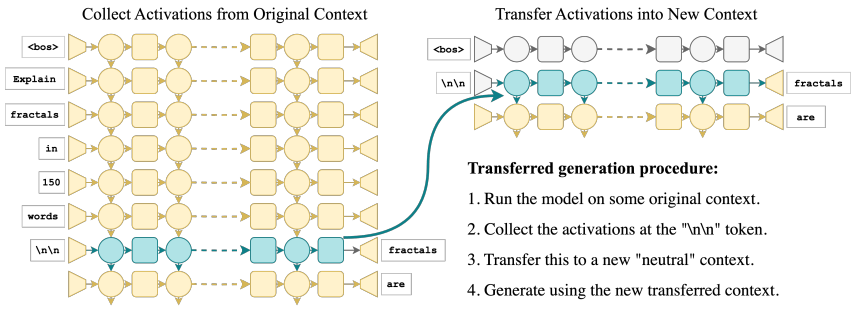


Figure 2: Diagram describing our approach. After collecting activations at the transition token on the original context model, we transfer these to all layers of the neutrally-prompted model.

3 Generation Experiments with Transferred Activations

To investigate how models plan ahead for a new section, we conduct a series of generation experiments, illustrated in Figure 2. We first prompt a model with the *original* contexts (i.e., pairs of topics) and extract the activations of the double newline token between topics. These activations are then transferred to the corresponding double newline token of a neutral context, i.e. a fresh model "neutrally" prompted with "<bos>\n\n". This transfer occurs across all layers of the model, i.e. including both attention and Multi-Layer Perceptron (MLP) layers. Doing so effectively "seeds" the neutral context with information encoded solely in the activation vector of the double newline token, without additional context. Our goal is to analyze how much of the second paragraph's information is contained in this token, assessing the extent to which the model has planned the rest of the generated text at the start of a new paragraph.

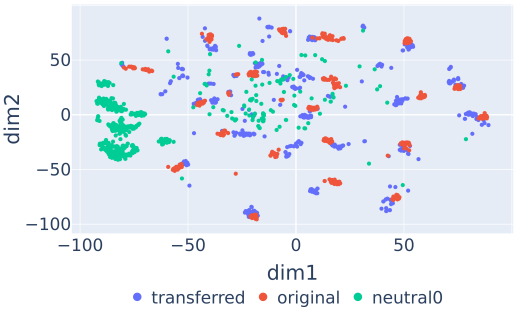
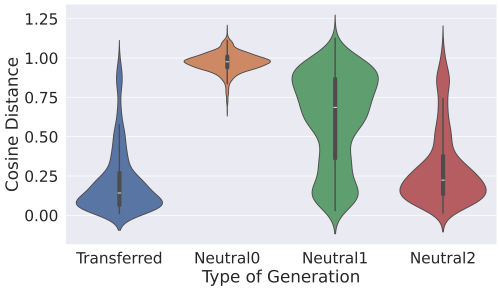


Figure 3: Context similarity visualised with T-SNE. Results over 1,000 original contexts.

To analyze the context of the transferred generations—i.e., texts generated from neutrally prompted models with transferred activations—we use state-of-the-art sentence embedding techniques [20; 12; 10]. We convert the output sequence of tokens into a single activation vector using ALL MPNET Base v2 [17; 18], and compare the semantic similarity between the *original* generations and those produced from the *transferred* activations. Additionally, we compare these with texts generated by the model using the same neutral prompt without activation transplantation, referred to as the *neutral0* generations. (The relevance of '0' is explained two paragraphs below.)

Figure 3 shows the first two dimensions of a T-Distributed Stochastic Neighbor Embedding (T-SNE) for the neutral0, original, and transferred generated texts. Our findings reveal a remarkable degree of semantic similarity between the original paragraphs and those generated from the transferred double newline token activations. For each prompt, the *transferred* cluster aligns well with the *original* cluster. In contrast, texts generated from the unaltered, neutrally prompted model are randomly scattered, showing low similarity with the original generations. This confirms that the activations of the double newline token hold a lot of information about the upcoming paragraph despite it being a separate topic from the previous one. An additional plot comparing the generations with PHATE can be found in the Appendix A.



Type of Generation	Neutral2	Transferred
Mean cosine distance to original (Std)	0.303 (0.239)	0.214 (0.210)

Figure 4: Distribution of cosine distances to the original generation. Contexts are summarized using sentence transformers, and distributions are taken over 1,000 original contexts.

Given that we transfer the activations from every layer, we in particular transfer the activations of the final layer. This means that we are effectively telling the model what the next token is, so comparing this case with the neutral prompt may unfairly advantage our transferred generations. To address this, we add "cheat" tokens to assist the neutral baseline by hinting at the context of the next paragraph and removing the next-token prediction advantage of the transferred generations. Specifically, we create two additional sets of generations—*neutral1* and *neutral2*—where the neutral prompt is concatenated with one or two "cheat" tokens, which are the first words of the following paragraph in the original generation. We also use the same sentence transformer to retrieve the embeddings of the generations.

Figure 4 displays the distribution of cosine distances to the original generation, comparing the transferred generations with *neutral0*, *neutral1*, and *neutral2*. The transferred generations significantly

outperform *neutral0* and *neutral1*, being much closer to the original generations. Notably, *neutral2* generations are much closer to the original than *neutral0* and *neutral1*, and share a similarly shaped distribution to the transferred generations. However, as shown in the table comparing the average cosine distances, the transferred generations still outperform *neutral2* in terms of closeness to original - with a t-test of the two sets of cosine distances rendering $p < 0.001$. Exhaustive results can be found in Table 1 in the Appendix.

4 Related work

Our work leverages activation patching, a standard technique in mechanistic interpretability introduced by *Vig et al.* [23; 27; 4]. This is in line with the work of *Jenner et al.*, which uses activation patching to understand the look-ahead behavior in Leela Chess Zero's policy network. However, contrasting with their focus on chess strategy optimization, we explore how language models anticipate the context of future paragraphs. Our work also builds upon recent studies examining lookahead in causal language models, although they adopt a different approach to the question. For example, the Future Lens approach [14] investigates how much signal individual hidden states contain by using them to predict subsequent embeddings, and *Wu et al.* [26] modify training procedures to gain deeper insights into token planning. Addressing context planning from another perspective, the *Patchscope* approach [5] combines inspection prompts and activation engineering to investigate how context is *read*, demonstrating that this process occurs predominantly in early layers.

While these studies offer valuable perspectives, our research question diverges by examining the phenomenon at a different scale. We focus on the *context* of an entire section of generated text, notably using sentence embedding. This approach allows us to explore broader contextual relationships and planning mechanisms within language models. [8] .

5 Discussion

Conclusion

In this work, we investigate how an LLM plans for future context. By using a specific set of prompts, we observed the model's attention allocation between the current paragraph and the previous paragraph on a different topic. We further examined the extent of "pre-planned" information the model holds for the subsequent paragraph by performing activation transfers on a neutrally prompted model. Our findings suggest that a single token encodes a substantial amount of contextual information about the forthcoming section, and that most (but not all) of this information seems to be contained in the first two tokens of generation.

Limitations While our framework provides valuable insights, it has several limitations. First, it is designed specifically for autoregressive (or "causal") models and does not apply to word2word models, which lack the same sequential generation process. Thus, its utility is tied to autoregressive model architectures. Second, this study is an experimental investigation rather than a comprehensive solution. The methods are not foolproof and are not suited for explaining sensitive or high-stakes models. This work should be viewed as a foundational step for future research aimed at developing more robust methods. Additionally, our experiments have been conducted using only a specific language model. However, we reasonably expect that our findings generalize to other transformer-based language models, which are widely used today.

Future work Our experiments focus on abrupt context switches, which, while useful for certain analyses, may not fully represent realistic scenarios. Future work could explore applying our approach to cohesive texts without abrupt context switches, though this poses its own challenges. Specifically, distinguishing between the information the model "remembers" across sections and what it knows at the onset of a paragraph is complex. Currently, our framework examines the model's planning for the "next paragraph", but future research could extend this to predict further sections ahead. Evaluating whether the framework can anticipate not just the immediate next paragraph but also subsequent sections could provide insights into its ability to construct and maintain a global narrative structure.

References

[1] Nikhil Bhendawade, Irina Belousova, Qichen Fu, Henry Mason, Mohammad Rastegari, and Mahyar Najibi. Speculative streaming: Fast llm inference without auxiliary models. *arXiv preprint arXiv:2402.11131*, 2024.

[2] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2019.

[4] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 5484–5495. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.446. URL https://doi.org/10.18653/v1/2021.emnlp-main.446.

[5] Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. Patchscope: A unifying framework for inspecting hidden representations of language models. *arXiv preprint arXiv:2401.06102*, 2024.

[6] Fabian Gloeckle, Badr Youbi Idrissi, Baptiste Rozière, David Lopez-Paz, and Gabriel Synnaeve. Better & faster large language models via multi-token prediction. *arXiv preprint arXiv:2404.19737*, 2024.

[7] Evan Hernandez, Belinda Z Li, and Jacob Andreas. Inspecting and editing knowledge representations in language models. *arXiv preprint arXiv:2304.00740*, 2023.

[8] janus. How LLMs are and are not myopic. Alignment Forum, jul 2023. URL https://www.alignmentforum.org/posts/c68SJsBpiAxkPwRHj. Accessed on 5th September 2024.

[9] Margaret Li, Weijia Shi, Artidoro Pagnoni, Peter West, and Ari Holtzman. Predicting vs. acting: A trade-off between world modeling & agent modeling. *arXiv preprint arXiv:2407.02446*, 2024.

[10] Xianming Li and Jing Li. Angle-optimized text embeddings. *arXiv preprint arXiv:2309.12871*, 2023.

[11] Tianyi Men, Pengfei Cao, Zhuoran Jin, Yubo Chen, Kang Liu, and Jun Zhao. Unlocking the future: Exploring look-ahead planning mechanistic interpretability in large language models. *arXiv preprint arXiv:2406.16033*, 2024.

[12] Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*, 2022. doi: 10.48550/ARXIV.2210.07316. URL https://arxiv.org/abs/2210.07316.

[13] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2(11): e7, 2017.

[14] Koyena Pal, Jiuding Sun, Andrew Yuan, Byron C Wallace, and David Bau. Future lens: Anticipating subsequent tokens from a single hidden state. *arXiv preprint arXiv:2311.04897*, 2023.

[15] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

[16] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

[17] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. URL https://arxiv.org/abs/1908.10084.

[18] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding. *arXiv preprint arXiv:2004.09297*, 2020.

[19] Nishant Subramani, Nivedita Suresh, and Matthew E Peters. Extracting latent steering vectors from pretrained language models. *arXiv preprint arXiv:2205.05124*, 2022.

[20] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*, 2021.

[21] Alex Turner, Lisa Thiergart, David Udell, Gavin Leech, Ulisse Mini, and Monte MacDiarmid. Activation addition: Steering language models without optimization. *CoRR*, abs/2308.10248, 2023. doi: 10.48550/arXiv.2308.10248. URL https://doi.org/10.48550/arXiv.2308.10248.

[22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[23] Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. *Advances in neural information processing systems*, 33:12388–12401, 2020.

[24] Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. *arXiv preprint arXiv:2211.00593*, 2022.

[25] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45, 2020.

[26] Wilson Wu, John X Morris, and Lionel Levine. Do language models plan ahead for future tokens? *arXiv preprint arXiv:2404.00859*, 2024.

[27] Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. *arXiv preprint arXiv:2309.16042*, 2023.

[28] Xin Zhang, Zehan Li, Yanzhao Zhang, Dingkun Long, Pengjun Xie, Meishan Zhang, and Min Zhang. Language models are universal embedders. *arXiv preprint arXiv:2310.08232*, 2023.

[29] Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*, 2023.

A Further Experimental Results

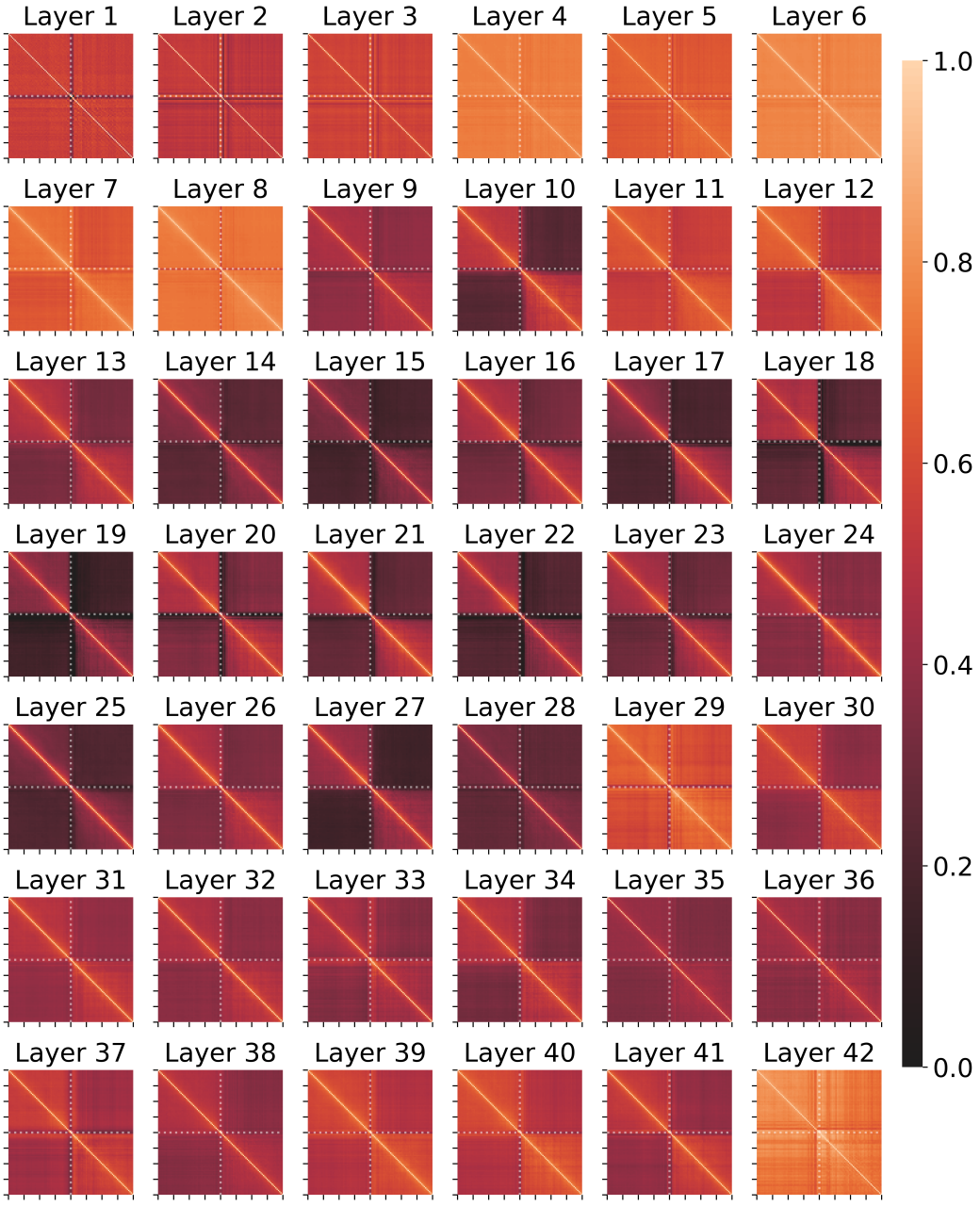


Figure 5: Cosine similarity between attention activations across all 42 layers of the model. Results averaged over 1,000 model-generated original contexts, sharing a common structure.

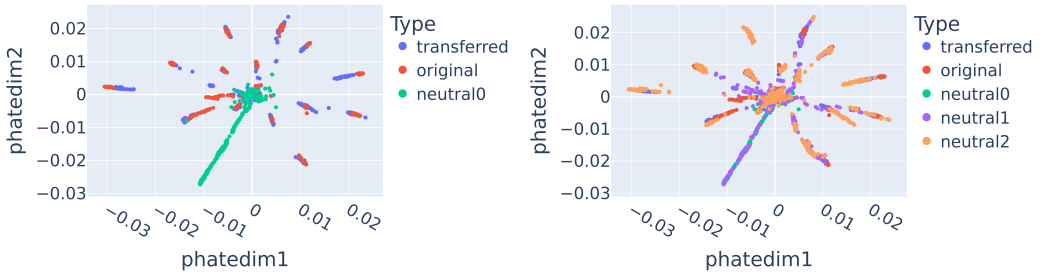


Figure 6: Comparison of context similarity between (left) original, neutral0, and activation-transferred text generations and (right) adding neutral1 and neutral2. Contexts were summarized using sentence transformer embeddings and visualized with PHATE, with results shown over 1,000 model-generated original contexts.

Below is a table that summarizes the results in Figure 4. To compare the neutral2 and transferred, we conducted a T-test on the two sets of cosine distance. This returned a t-statistic of -8.79, with $p = 3.19 \times 10^{-18}$.

Table 1: Mean cosine distances to the original generation with standard deviations. Contexts are summarized using sentence transformers, and distributions were taken over 1,000 original contexts.

Type of Generation	**Neutral0**	**Neutral1**	**Neutral2**	**Transferred**
Mean cosine distance to the original generation (Std)	0.973 (0.06)	0.616 (0.293)	0.303 (0.239)	0.214 (0.210)

B Experimental details

For our observational experiment in Section 2, note that a few generations were excluded from our analysis as the outputs were not in line with the expected structure. (e.g. text wasn't about two different topics, one blended text about two topics).

To make the attention weights plot in figure 1, we collected the attention weights in each head of a layer and summed them up. We then average the result from each layer and each text to get a combined plot.

Table 2: Computing details

Experiment	**Compute specification**	**Compute time**
Observational (Section 2)	1x RTX A4000 16GB	6 mins
Transferred generations (Section 3)	2x RTX A4000 16GB	24 hours for generating the 5 types of texts: (original, transferred, neutral, neutral1, neutral2).

We used the Huggingface Transformers library [25] to implement the extraction of attention weights.

C Social Impact

Understanding the inner workings of large language models (LLMs) through mechanistic interpretability has both positive and negative societal implications. On the positive side, this research can significantly enhance the safety and reliability of LLMs. By deciphering how these models plan ahead and make decisions, we can develop better methods for detecting and mitigating harmful outputs, such as disinformation, biased decision-making, or unintended behaviors. Improved interpretability

can also aid in the development of more robust safety measures, ensuring that LLMs align with human values and ethical standards, and help build trust in AI systems deployed in sensitive applications like healthcare, education, and critical infrastructure.

However, this work also presents potential risks. A deeper understanding of LLMs' mechanisms can be exploited to design targeted adversarial attacks or to manipulate model outputs in malicious ways. For instance, insights gained could be misused to bypass existing safeguards, generate more convincing disinformation, or optimize models for malicious tasks such as creating fake profiles or enhancing surveillance tools. Additionally, as interpretability techniques become more advanced, there is a risk that they could be used to reverse-engineer proprietary or confidential models, leading to intellectual property theft or unauthorized replication of models.

To mitigate these risks, it is important to develop safeguards alongside interpretability advancements, such as gated access to sensitive findings, rigorous monitoring of model usage, and collaboration with policymakers to create frameworks that ensure the ethical application of these insights. Balancing transparency with security will be crucial in leveraging the benefits of mechanistic interpretability while minimizing its potential misuse.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Yes, we do demonstrate that patching token activations can transfer significant information about the context of the following paragraph.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: A subsection is dedicated to Limitations in the Discussion 5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA] .

Justification: There are no theoretical results in this paper.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide a link to an anonymous repository with our code along with the prompts for all our experiments. The language model we used -Gemma 2, 9b- is cited and open-source.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Code and data are provided or open-source.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so ???No??? is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes] ,

   Justification: All experimental details can be found in the repository.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Yes, in our main experiment 3 we report the standard errors across 1000 generations.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide these details in Section B of the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The NeurIPS Code of Ethics was read and respected.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Yes, we discuss these in the Appendix, under section C.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This is outside of the scope of our paper.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes] ,

    Justification: All code and models are explicitly credited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The assets in our repository are well documented.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA] .

    Justification: Our paper is not related to human data.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA] .

    Justification: There are no human study participants in this study.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.