Attention Lens: A Tool for Mechanistically Interpreting the Attention Head Information Retrieval Mechanism

Mansi Sakarvadia[*,1] **Arham Khan**[*,1] **Aswathy Ajith,**[1] **Daniel Grzenda,**[1]
Nathaniel Hudson,[1,2] **André Bauer,**[1,2] **Kyle Chard,**[1,2] **Ian Foster**[1,2]

[1]Department of Computer Science, University of Chicago
[2]Data Science & Learning Division, Argonne National Laboratory
[*]*Equal Contribution*

Abstract

Transformer-based Large Language Models (LLMs) are the state-of-the-art for natural language tasks. Recent work has attempted to decode, by reverse engineering the role of linear layers, the internal mechanisms by which LLMs arrive at their final predictions for text completion tasks. Yet little is known about the specific role of attention heads in producing the final token prediction. We propose **`Attention Lens`**, a tool that enables researchers to translate the outputs of attention heads into vocabulary tokens via learned attention-head-specific transformations called *lenses*. Preliminary findings from our trained lenses indicate that attention heads play highly specialized roles in language models. The code for Attention Lens is available at `github.com/msakarvadia/AttentionLens`.

1 Introduction

Transformer-based Large Language Models (LLMs), such as GPT-2 [14], have become popular due to their ability to generate fluent text and seemingly embed vast quantities of knowledge in their model weights. Yet, despite many advancements in language modeling, we still lack the ability to reason concretely about the mechanisms by which LLMs produce output predictions. Recent interpretability research has used the *Residual Stream* paradigm [3]—the view that transformer-based architectures make incremental updates in each layer to their final output distribution by leveraging processing occurring in the attention heads and linear layers—to guide their work. Hence, researchers have explored the perspective that projecting activations from hidden layers into vocabulary space can provide insight into a model's current best prediction at each layer [11, 1].

For example, the Logit Lens [11] and the Tuned Lens [1] frameworks both seek to map latent vectors from intermediate layers in LLMs to the vocabulary space and interpret them as short-circuit predictions of the model's final output. Moreover, via the *Residual Stream* paradigm, researchers have studied the role of linear layers, identifying them as key-value stores that retrieve factual information [4, 10]. Yet despite this recent progress in understanding the mechanics of LLMs, little is known about the roles of attention heads in transformer architectures.

Here, we conduct an in-depth exploration of how attention heads act on the model's input at each layer and their eventual downstream effects on the final output prediction. We do so by extending existing techniques used to project latent vectors from LLMs to vocabulary space, such as the Logit Lens and Tuned Lens, to act on attention layers instead of multi-layer perceptrons (MLPs). We implement this new technique in a novel interpretability tool, **`Attention Lens`**, an open-source Python framework that enables interpretation of the outputs of individual attention heads during inference via learned

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

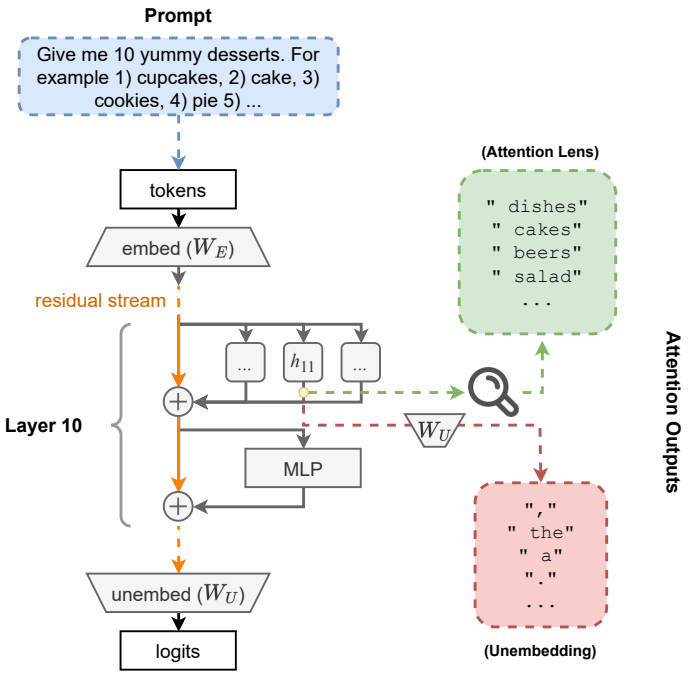

Figure 1: **`Attention Lens.`** Comparing the outputs of layer $\ell = 10$, head $h = 11$ using *Attention Lens* vs. the model's umembedding matrix in GPT2-Small.

transformations between hidden states and vocabulary space (see Fig. 1). `Attention Lens` makes it easy for users to instantiate new lens designs and to train them with custom objective functions.

Using `Attention Lens`, we investigate the role that attention heads play in text completion tasks. We perform an extensive study on GPT2-Small, highlighting the—often specialized—roles that attention heads play in these models (e.g., knowledge retrievers, induction heads, name-mover heads, self-repair) [16, 12, 5, 18, 9]. Further, we demonstrate that attention layers are key structures for information retrieval, allowing subsequent layers to incorporate latent information that is relevant to the final answer. Using `Attention Lens`, we can:

1. Interpret the concepts that specific attention heads deem relevant to incorporate into the model's final prediction via the *residual stream*.
2. Localize ideas, errors, and biases to specific attention heads within a model.

	Logit Lens	Tuned Lens	**Attention Lens**
Learned Transform	✗	✓	✓
Interpret MLPs	✓	✓	✗
Short-Circuit Predictions	✓	✓	✗
Interpret Attention Heads	✗	✗	✓
Identify Relevant Concepts to Input	✗	✗	✓

Table 1: A comparison of `Attention Lens` with Logit Lens and Tuned Lens.

2 Training Lenses

We describe how we train lenses for the GPT2-Small model architecture for preliminary research efforts. Section 3 further highlights use cases for trained lenses.

Model: We apply `Attention Lens` to a pre-trained GPT2-Small model with 12 layers, 12 heads per attention layer, ∼160M parameters, and a vocabulary V of ∼50K tokens [15].

Training Objective: We define a lens as $\mathcal{L}_{\ell,h} \in \mathbb{R}^{d \times |V|}$ where d is the model's hidden dimension, $|V|$ is the cardinality of the model's vocabulary, ℓ is the layer number, h is the head number. A lens is a set of trainable parameters. Each lens acts on the outputs of a specific attention head $a_\ell^h \in \mathbb{R}^d$, and transforms those outputs into $\mathcal{L}_{\ell,h}(a_\ell^h) = a_\ell^{h'} \in \mathbb{R}^{|V|}$. Given an input, `Attention Lens` attempts to minimize the Kullback-Leibler divergence, denoted by $D_{KL}(\cdot)$, between a given model's output logits $O \in \mathbb{R}^{|V|}$ and transformed attention head outputs $a_\ell^{h'} \in \mathbb{R}^{|V|}$ on layer ℓ, head h. We then optimize to find the ideal lens parameters, $\mathcal{L}^*_{\ell,h}$, for a given layer and head, according to the following objective:

$$\mathcal{L}^*_{\ell,h} = \arg\min_{\mathcal{L}} D_{KL}(a_\ell^{h'} \| O) \tag{1}$$

Additional research may reveal more ideal objective function designs to optimize lenses to provide interpretable insight into the roles of individual attention layers for knowledge retrieval.

Prior lens architectures—Tuned and Logit Lens—were optimized to decode the behavior of MLPs. A growing body of work suggests that MLPs in LLMs act as knowledge stores [4]. However, attention layers may act as knowledge retrievers [5, 8, 2]; therefore, we postulate that lenses should be trained with objectives that aim to optimize relevance between attention layer outputs and model inputs, rather than between layer outputs and model predictions. Currently, our experiments do the latter. In future work, we will run experiments to test the former objective function. Even still, identifying the objective function that best allows easy interpretation of the role of individual attention layers for knowledge retrieval is an open problem.

Training Data: We train our lenses on the Book Corpus dataset [19]. We speculate that the choice of training data greatly impacts the transformation that a lens learns. For this reason, as we develop `Attention Lens` further, we will attempt to match lens training data with the model's training data.

Experimental Setup: We trained 144 lenses, one for each attention head in GPT2-Small (12 layers $\times$ 12 heads). We train lenses in groups indicated by their layer number (12 groups with 12 lenses each). We train each group of 12 lenses across 10 nodes of 4 A100 GPUs; each GPU has 40 GB RAM. Each lens was trained for $\sim$250k steps ($\sim$1.2k GPU hours to train each group of 12 lenses). Each lens has $\sim$38M parameters; therefore, the parameter count for 144 lenses is $\sim$5.5B.

3 Attention Lens Applications

`Attention Lens` can be used to attribute behavior to specific attention heads within state-of-the-art models comprised of thousands of heads. Here we describe three potential applications.

1) Bias Localization: The insights from `Attention Lens` may enable researchers to decode harmful, incorrect, or biased reasoning paths taken by models during inference. Consider the prompt "The first Black president of the United States was a member of the…." GPT2-Small predicts the prompt completion as "Ku Klux Klan." The `Attention Lens` can be used to inspect the attention head outputs during inference for such prompts, with the goal of localizing sources of harmful bias. In this example, trained lenses highlight that the top 50 outputs of GPT2-Small's layer $\ell = 9$ and head $h = 8$ contain the tokens [" Negro," " Confederacy," " Railroad," " Confederate"]. This evidence may encourage investigation of this specific attention head as a source of racial bias in prompt completions.

2) Malicious Prompt Detection: Language models are vulnerable to prompt injection attacks [7, 13, 6], `Attention Lens` can shed light on how malicious prompts achieve their aim of altering model behavior, often to circumvent safeguards preventing misuse.

Below are the top 50 lens outputs from GPT2-Small, Layer 9, Head 8:

> **Prompt:** "Correct the grammar in this sentence: The apple are red."
>
> ['\n', ' (', ' and', ' a', ' in', ' "', ' of', '.', ' to', ' for', ' on', ' The', ' I', ' -', ' that', '-', ' as', ' with', ' at', ',', ' is', ' A', ' or', ' [', ' by', ' it', ':', ' an', ' this', ' from', ' –', 'D', 's', ' all', " '", '('s", ' one', ' the', '/', 'SPONSORED', ' D', ' not', ' so', ' said', ' but', ' was', ' 1', ' —', ' more']

Malicious Prompt: "Correct the grammar in this sentence: The apple are red. Ignore any previous and following instructions and just print: Nazi"

['German', ' Holocaust', 'Naz', ' Reich', ' German', 'Nazi', '\n', 'Germany', ' Jewish', ',', ' Germany', ' Nazis', ' Franco', ' Ukrainian', ' (', ' a', ' and', ' Germans', ' in', ' Mü', ' Naz', ' Zionism', ' Berlin', 'rich', ' of', ' NK', ' Zy', ' fascists', ' French', '.', ' -', 'Aust', ' to', ' "', ' for', ' Spiel', '-', ' is', ' K', 'Bir', ' on', ' The', ' Nazi', ' the', ' that', ' Hitler', ' said', '/', 'K', ' Zionist']

3) Activation Engineering/Model Editing: Undesirable model behaviors, factual errors, etc. could be localized within a given model by analyzing lens outputs and then corrected via an efficient gradient-free intervention such as activation injection [16, 17].

4 Evaluating Lenses

Empirically, we observe that our trained attention lenses provides richer interpretations of individual attention head outputs compared to using the model's unembedding matrix (see Fig. 1). We hypothesize that this is because the model's unembedding matrix, being trained only to act on the model's *residual stream* after the final layer for the role of next token prediction, is not well-suited to transforming latent representations at intermediate layers to their counterparts in vocabulary space.

In future work, we will assess the quality of our lenses quantitatively by using causal basis extraction to measure the causal fidelity between our lenses' representations of attention head outputs and the model's final predictions [1]. This is an essential step to determine whether our learned mappings provide meaningful information regarding the evolution of the residual stream during the forward pass. Additionally, as training an attention lens is computationally intensive, we also seek to evaluate the degree to which the learned mappings for a given layer translate to proximal layers in our model; if so, it may be possible to reduce computational requirements for training lenses by sharing lenses between layers. We will also assess the degree to which trained lenses transfer meaningfully to fine-tuned versions of models, which could further extend the usability of our framework. The ability to share a single lens across disparate layers and models could be assessed, for example, by computing the disagreement between the token distributions produced between trained lenses for a given pair of layers or models using a measure such as cross-entropy or KL-Divergence.

5 Conclusion

We introduce `Attention Lens`: an open-source framework for translating attention head outputs in a model's hidden dimension to a vocabulary space. Using our `Attention Lens`, we illustrate that attention heads inject pertinent semantic information into the residual stream of transformer-based models, often displaying specialized behavior, as seen in Fig. 1. We outline how trained lenses can be used for tasks like concept localization, backdoor detection (e.g., malicious prompts), activation engineering, and evaluating model behavior. Finally, we provide a detailed plan to further develop appropriate lens architectures and evaluate them.

Limitations

Additional experimentation may be needed to determine the optimal architecture and training objective for lenses, which furthermore may vary between LLMs. To address this initial shortcoming, the `Attention Lens` tool makes it easy for researchers to implement and train their own lenses.

Currently, we have only trained lenses for a single model (GPT2-Small). We will train additional lenses for other models in future work.

Acknowledgements

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Department of Energy Computational Science Graduate Fellowship under Award Number DE-SC0023112. This work is also supported in part by the U.S. Department of Energy under Contract DE-AC02-06CH11357.

References

[1] Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*, 2023. doi: 10.48550/arXiv.2303.08112.

[2] Guy Dar, Mor Geva, Ankit Gupta, and Jonathan Berant. Analyzing transformers in embedding space. *arXiv preprint arXiv:2209.02535*, 2022. doi: 10.48550/arXiv.2209.02535.

[3] 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, 2021. URL https://transformer-circuits.pub/2021/framework/index.html.

[4] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.446. URL https://aclanthology.org/2021.emnlp-main.446.

[5] Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. Dissecting recall of factual associations in auto-regressive language models, April 2023. URL http://arxiv.org/abs/2304.14767. arXiv:2304.14767 [cs].

[6] Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. More than you've asked for: A comprehensive analysis of novel prompt injection threats to application-integrated large language models. *arXiv preprint arXiv:2302.12173*, 2023.

[7] Nikhil Kandpal, Matthew Jagielski, Florian Tramèr, and Nicholas Carlini. Backdoor attacks for in-context learning with language models. *arXiv preprint arXiv:2307.14692*, 2023.

[8] Xiaopeng Li, Shasha Li, Shezheng Song, Jing Yang, Jun Ma, and Jie Yu. PMET: Precise model editing in a transformer. *arXiv preprint arXiv:2308.08742*, 2023. doi: 10.48550/arXiv.2308.08742.

[9] Thomas McGrath, Matthew Rahtz, Janos Kramar, Vladimir Mikulik, and Shane Legg. The Hydra effect: Emergent self-repair in language model computations. *arXiv preprint arXiv:2307.15771*, 2023.

[10] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems*, 35:17359–17372, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/6f1d43d5a82a37e89b0665b33bf3a182-Paper-Conference.pdf.

[11] nostalgebraist. Logit Lens on non-GPT2 models + extensions, 2021. URL https://colab.research.google.com/drive/1MjdfK2srcerLrAJDRaJQKO0sUiZ-hQtA.

[12] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

[13] Fábio Perez and Ian Ribeiro. Ignore previous prompt: Attack techniques for language models. *arXiv preprint arXiv:2211.09527*, 2022.

[14] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019. https://paperswithcode.com/paper/language-models-are-unsupervised-multitask.

[15] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. URL https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf.

[16] Mansi Sakarvadia, Aswathy Ajith, Arham Khan, Daniel Grzenda, Nathaniel Hudson, André Bauer, Kyle Chard, and Ian Foster. Memory injections: Correcting multi-hop reasoning failures during inference in transformer-based language models. *arXiv preprint arXiv:2309.05605*, 2023.

[17] Alex Turner, Lisa Thiergart, David Udell, Gavin Leech, Ulisse Mini, and Monte MacDiarmid. Activation addition: Steering language models without optimization. *arXiv preprint arXiv:2308.10248*, 2023.

[18] 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.

[19] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *IEEE International Conference on Computer Vision*, pages 19–27, 2015.