Investigating Language Model Dynamics using Meta-Tokens

Alok N. Shah[*]
University of Pennsylvania
alokshah@sas.upenn.edu

Khush Gupta[*]
University of Pennsylvania
khushg@upenn.edu

Keshav Ramji[*†]
IBM Research AI
keshav.ramji@ibm.com

Vedant Gaur[*]
University of Pennsylvania
vedantg@upenn.edu

Abstract

Transformers have achieved remarkable success across various domains, but much remains unknown about their internal reasoning and training dynamics. This paper presents a novel approach using meta-tokens, special tokens injected into the input sequence, and a dedicated meta-attention mechanism to improve model performance and interpretability. We hypothesize that meta-tokens store and retrieve global contextual information by interacting through meta-attention. We test this by pretraining modified GPT-2 architecture equipped with meta-attention, in addition to causal multi-headed attention, and demonstrate its efficacy through empirical gains on the MMLU benchmark. Furthermore, we explore the distribution of attention scores and residual stream alterations by visualizing model internals. By applying the language model head at key points in the residual stream, we find that meta-tokens accelerate layer-wise logit convergence to the correct output token. These results suggest that meta-tokens effectively capture global dependencies, providing enhanced performance on long-context tasks while offering new insights into the flow of attention scores and subsequently training behavior in transformers.

1 Introduction

The transformer architecture has revolutionized natural language processing, demonstrating impressive capabilities across a wide range of tasks and domains [Vaswani et al., 2017, Lin et al., 2021]. However, much about their underlying reasoning processes and training dynamics remain unknown [Jain and Wallace, 2019]. Recent advances in the field of *mechanistic interpretability* have furthered our understanding of model internals in toy settings and at scale [Olah et al., 2020]. These techniques have proven particularly effective in elucidating the computational processes within both small-scale and large language models. Circuit analysis methods, for instance, aim to identify and comprehend specific computational circuits within neural networks [Nanda, 2023]. Furthermore, the application of these interpretability methods to large-scale models has yielded insights into emergent behaviors [Bricken et al., 2023].

Concurrently, recent work has explored the use of auxiliary tokens or registers as a cheap trick to provably boost model performance, both empirically and theoretically, on various language and vision tasks [Goyal et al., 2024, Pfau et al., 2024, Darcet et al., 2024]. Inspired by this success, we posit a novel approach that leverages these *meta tokens*. Meta-tokens are inserted into the input sequence

[*]Equal contribution.
[†]Work done while at the University of Pennsylvania.

ATTRIB: 2nd Workshop on Attributing Model Behavior at Scale at NeurIPS 2024.

during training and are designed to interact with a dedicated *meta-attention* mechanism, allowing them to store and retrieve useful context. As such, our meta-tokens serve a dual purpose: enhancing reasoning capability and providing guideposts for understanding model internals.

Our contributions are twofold:

1. A pretrained GPT-2 model (152M parameters) modified with a meta-attention mechanism to handle injected meta-tokens, which we empirically validate through improved reasoning performance on the Massive Multitask Language Understanding (MMLU) benchmark.
2. A detailed analysis of transformer dynamics, examining changes in attention scores and residual stream distributions before and after the meta-attention layers, using interpretability tools to provide insights into model behavior.

2 Methods

2.1 Preliminaries

Let $\mathbf{x} = \{x_1, x_2, \ldots, x_T\}$ denote an input sequence of tokens of length T, $\mathcal{V}$ denote the vocabulary size of V, and $E : \mathcal{V} \to \mathbb{R}^d$ represent the the token embedding function mapping each token to a d-dimensional vector. Each x_t is embedded into some continuous representation where $\mathbf{e}_t = E(x_t) + \mathbf{p}_t$, such that $\mathbf{p}_t$ is the positional encoding for t. This is often a sinusoidal projection of the inputs tokens and their corresponding sequence position into a higher-dimensional feature space to induce relative sequence positions, given the the transformer processes $\mathbf{x}$ in parallel.

In the context of a decoder-only architecture, we utilize causal self-attention to ensure that predictions for a given token are only based on preceding tokens in the sequence. The causal self-attention mechanism modifies the attention computation by masking future positions in the attention weights. Formally, the operation is computed as:

$$\text{Causal Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}} + M\right)$$

where M is a mask matrix that prevents access to future tokens, ensuring that the model can only attend to current and past tokens. In particular, if A is the matrix of attentions scores then

$$A_{ij} = \begin{cases} \text{softmax}(A_{ij}) & \text{if } i \geq j \\ 0 & \text{if } i < j \end{cases}$$

This masking ensures that the attention scores for future tokens are zeroed out, allowing only the relevant past tokens to influence the current token's representation.

2.2 Meta Tokens

We introduce a set of M meta-tokens (denoted as m); given a context length or block size of the model, n, we take $M = kn$ for some constant fraction $k \in [0, 1]$[3]. These meta-tokens are designed to store and process global contextual information that can enhance the model's reasoning capabilities, and serve the same as adding a filler token to the model's vocabulary. We simply inject these M tokens into the input sequence uniformly at random; this was determined on the premise of two key reasons. Firstly, while we desire distinguishability for greater interpretability and control in applying these tokens, this is challenging to do without fixing a downstream task. The second consideration was how they should be injected: while [Zelikman et al., 2024] introduced <|startofthought|> and <|endofthought|> tokens interleaved between reasoning steps near punctuation (serving as natural break), we feared that this might trap the fine-tuning into local minima in the optimization landscape. That is, we hypothesized that this may create a rough periodicity between tokens and therefore unevenly calibrating the weights during pre-training. We instead chose the follow random injection, supported by the pause token pre-training approach outlined in [Goyal et al., 2024] to take advantage of noise, which has provably been shown to improve generalization [Srivastava et al., 2014b]. Additionally, the model incurs no loss for predicting meta tokens – they are simply shifted and removed when computing the binary cross-entropy (BCE) loss. This raises a very interesting question of token-level implicit regularization guiding the improvement of our method.

[3] we take $k = 0.1$ for our results in this work

2.3 Meta Attention Mechanism

We augment our transformer H to take an additional argument P which contains the positions of the meta tokens. We introduce an augmented attention mechanism, called meta-attention, which selectively modifies attention scores for specially marked "meta tokens" within a sequence. This mechanism allows the model to simulate selective attention, influencing the final behavior by focusing on these meta tokens.

Suppose we have indices of special "meta tokens" denoted by positions $\in \mathbb{R}^{B\times T'}$, where T' is the number of meta tokens in a batch. We construct a meta mask $P \in \mathbb{R}^{B\times T\times T}$ to influence the attention mechanism. For each batch element b and token positions i, j:

$$P[b,i,j] = \begin{cases} 0 & \text{if both } i \text{ and } j \text{ are meta tokens (i.e., } i,j \in \text{positions}[b,:]) \\ -\infty & \text{otherwise} \end{cases}$$

The meta attention operation is defined as:

$$\text{MetaAttention}(Q,K,V) = \text{softmax}\left(\left(\frac{QK^\top}{\sqrt{d_k}} + M\right) + P\right)V$$

Where M is the same causal mask as before. Here, the meta mask P allows attention to flow only among the meta tokens in the sequence, introducing a distinct interaction compared to regular attention. This meta attention layer selectively modifies the attention by influencing the flow of information to and from these meta tokens, distinguishing itself from the standard causal attention.

In particular, if A is the matrix of attentions scores then

$$A_{ij} = \begin{cases} \text{softmax}(A_{ij}) & \text{if i and j are meta tokens} \\ -\infty & \text{otherwise} \end{cases}$$

We borrow the underlying principles from dual cross-attention from [Jiang et al., 2024], where we perform operations higher on the abstraction hierarchy than the feature space resembling a meta-learning-like approach.

To assemble the architecture used for our model, we insert the meta-attention mechanism after the causal masked self-attention computation, to specifically attend to the injected meta tokens, as defined above. We provide a complete breakdown of the architecture in Appendix B.

3 Experiments + Results

3.1 Pretraining

We conducted our experiments using 4 NVIDIA A100 GPUs, training the meta attention transformer for 15 steps using Distributed Data Parallel (DDP) on the Colossal Cleaned Crawl Corpus (C4) [Raffel et al., 2023]. The configuration and hyperparameters used in our pre-training are included in Appendix C. As a baseline, we also pre-train GPT-2 (124M) on C4, with identical hyperparameters.

3.2 Benchmarks

We evaluated our model on the Massive Multitask Language Understanding (MMLU) [Hendrycks et al., 2021] benchmark using a 5-shot setting. To ensure a fair comparison, we retrained a 124 million parameter GPT-2 model on the C4 dataset for 30,000 iterations or 15 steps. Our model outperformed the C4-trained GPT-2 baseline, achieving a 26.0% average MMLU score compared to 25.1%. Our results are also comparable to GPT-3 (13B parameters) [Hendrycks et al., 2021], which achieved approximately 26.0% on MMLU under similar conditions.

3.3 A Glimpse into Model Internal Representations

We use the logit lens [Nostalgebraist, 2020] to visualize the internals of our meta-attention transformer. We examine, three key quantities – logits, rank with respect to the output token, and Kullback-Leilber (KL) Divergence from the output distribution.

Model	GPT-2 (C4)	Our Model	GPT-3 (2.7B)	GPT-3 (13B)
MMLU (5-shot)	25.1%	26.0%	25.9%	26.0%

Table 1: MMLU results comparing the performance of our model, GPT-2 (C4), and GPT-3.

Additionally, we attach hooks before and after the residual stream to visualize fine-grained differences and quantify the impact of the meta-attention layer.

We produced four key figures by applying the linear head of the meta-attention transformer to internal layers on the abstract of the GPT-3 Paper as the authors of logit lens do [Nostalgebraist, 2020]; Appendix E. For visualization purposes, we inject the meta-tokens every 5 tokens, though all observations will hold for the random injection case.

3.3.1 Logits

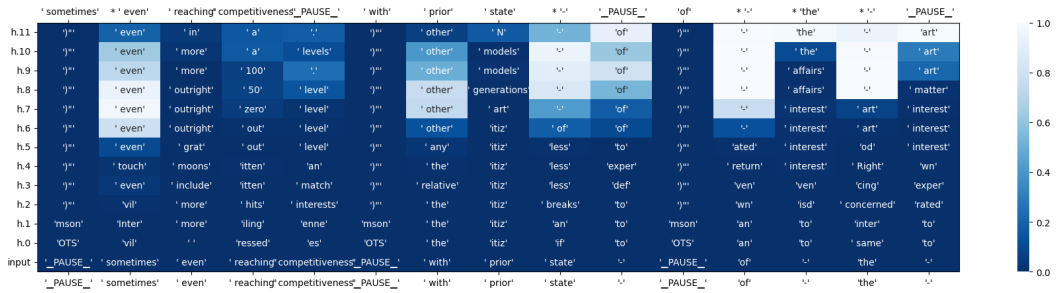


Figure 1: Logits

The heatmap in Figure 1 shows how logits evolve across transformer layers. In early layers, logits are dispersed, indicating low confidence in token predictions. As the model progresses, confidence increases, with higher logit values for specific tokens in later layers. For meta-tokens, we observe a gradual convergence and increased probability, suggesting they help focus the model's attention and store context for downstream reasoning. This aligns with the equi-learning law, which posits that each layer improves next-token prediction by a multiplicative factor [He and Su, 2024].

3.3.2 Block Rank

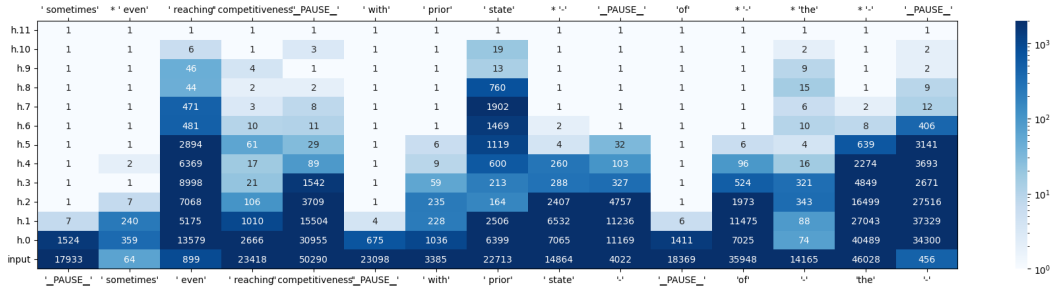


Figure 2: Block Rank

We liken 2 to relaxed version of Figure 1, showing the rank of predicted tokens relative to the model's final prediction. Lower ranks indicate closer alignment with the final prediction, while higher ranks reflect uncertainty. In early layers, most tokens have high ranks, consistent with dispersed logits. As the model progresses, ranks drop, signaling convergence. Meta-tokens visibly accelerate this convergence, dropping earlier and maintaining ranks closer to the final prediction, highlighting their role in storing context and enhancing reasoning.

3.3.3 Subblock Rank

See Appendix F.1 for the full figure. Since each block consists of attention and feedforward layers, we can decompose and apply the LM head again. While meta-tokens still accelerate rank drop, meta-attention position ranks are extremely high unless paired with a meta token. Still, they exhibit no obvious helpful impact on next-token prediction

Instead, meta-attention likely acts as a buffer for long-range dependencies, storing global or structural information that becomes useful in later layers. In particular, we hypothesize that their role seems to be smoothing attention scores and stabilizing the model's focus over longer sequences, via by distributing scores around their guidepost meta-tokens. This smoothing effect not only prevents attention divergence but also supports higher-level reasoning, enhancing performance in tasks like coreference resolution and maintaining coherence in long-context scenarios by ensuring that key information is retained across layers.

3.3.4 Attention Score Distribution

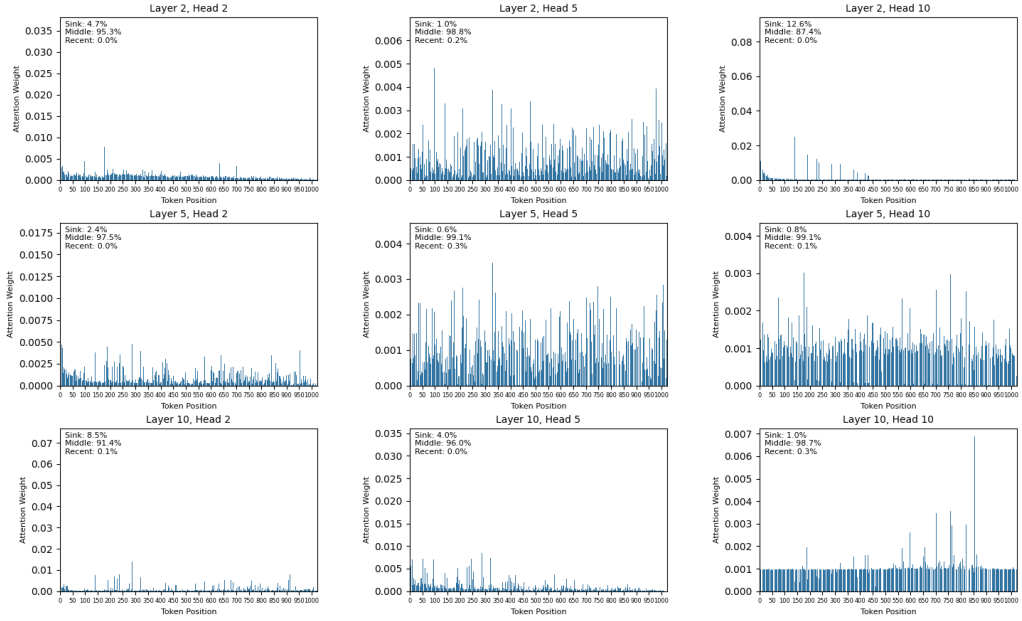


Figure 3: Attention Score Distribution Across Selected Layers and Heads: The grid visualizes attention distributions for layers 2, 5, and 10 across heads 2, 5, and 10. Each subplot highlights the proportion of attention allocated to 'sink,' 'middle,' and 'recent' token positions, revealing distinct patterns of focus that inform the model's processing dynamics. Notably, the attention score pattern is different than what prior works note about the distribution of scores in most transformers; while often U-shaped [Fu, 2024] (low attention scores over the middle of the context), we find that the scores are now largely similar (or follow similar patterns) over the entire span of the context window.

The visualization in Figure 3 elucidates the distribution of attention scores across selected layers and heads within the transformer model. Each subplot in the 3x3 grid corresponds to a specific combination of layer and head, providing a detailed view of how attention is allocated across token positions.

In examining Layer 2, we observe a diverse distribution pattern where certain heads concentrate on initial tokens, referred to as 'sinks,' while others distribute attention more uniformly across middle tokens. This suggests that Layer 2 is engaged in establishing foundational context early in the processing pipeline. Moving to Layer 5, the attention distribution becomes more balanced, with a pronounced focus on middle tokens. This indicates an intermediate stage of processing where context is maintained and refined, facilitating coherent information flow. Layer 10 presents varied patterns, with some heads heavily focusing on initial tokens and others spreading attention more evenly. This reflects advanced processing stages where global context integration occurs.

The analysis of head-specific roles reveals that Head 2 consistently allocates attention to initial tokens across layers, suggesting its role in capturing foundational context. Head 5 shows an increased focus on middle tokens, indicating its function in maintaining contextual coherence. Conversely, Head 10 exhibits significant variability, with some layers emphasizing initial tokens, which might be crucial for specific reasoning tasks.

Attention allocation insights highlight that the 'sink' section represents attention directed towards the first few tokens, which is crucial for setting context. The 'middle' section receives the majority of attention, underscoring its importance in maintaining continuity and coherence. The 'recent' section receives minimal attention in these configurations, suggesting that recency might not be prioritized by these specific heads.

These findings align with our hypothesis that meta-tokens enhance reasoning by storing global context. The varied distribution patterns suggest specialized roles for different components in processing long-context information. Heads focusing on initial tokens may serve as anchors for context retention, aiding in tasks requiring long-term dependency tracking. This visualization provides valuable insights into the internal mechanisms of transformer models and potential pathways for optimization and interpretability enhancements.

4 Discussion

In this work, we introduced a method of injecting meta-tokens into the input sequence, and devised a meta-attention mechanism to guide context to be stored in these tokens. We pre-train a 152M parameter decoder-only model to test the effectiveness of this method, and show that in addition to improving MMLU performance over pre-trained GPT-2 (the same architecture without the meta-attention mechanism), meta-tokens affect the distribution of attention scores. Furthermore, we apply the LM head to intermediate steps on the residual stream to visualize the logit and rank predictions at critical junctions on the residual stream and observe that meta-tokens experience faster convergence to the correct output while meta-attention lacks demonstrable proof of contribution towards logit/rank layerwise convergence. Instead, we conjecture that the meta-attention layers retain long-term dependencies – responsible for smoothing attention scores. Moving towards proof, we visualize the attention scores for a single head under this framework and observe that exhibit relative smoothness centered around meta-token indices – showing that meta-tokens indeed function as guideposts for attention scores.

Given the shape of the training and validation loss curves and random injection of the meta-tokens, we consider the possibility that meta-tokens and meta-attention behave like implicit token-level regularizers. This agrees with the marginal boost in performance on MMLU, but remains to be shown by other other benchmarks. Deep learning models have been proven to enjoy generalization benefits from noise injection in the training process [Srivastava et al., 2014a, Müller et al., 2019], hence opening the door of provable duality regarding the function of meta-tokens in the pretraining process.

While our study demonstrates that compact models can perform comparably to larger models, several limitations should be noted. Limited computational resources restricted us to training a 124M and 152M parameter model for 30,000 iterations, preventing exploration of larger models or longer training durations, which could further boost performance. Additionally, resource constraints limited hyperparameter tuning and experimentation with diverse architectures, potentially leaving untapped improvements in accuracy and generalization. Our evaluation was confined to the MMLU benchmark in a 5-shot setting, limiting broader assessment. Future work with greater computational resources could scale model size, extend training, and explore other benchmarks to better evaluate the scalability and generalizability of our approach.

Acknowledgments

The authors would like to thank Surbhi Goel for valuable discussions which have formed the basis of our study and support with computational resources. We would also like to thank Ben Keigwin for helpful conversations and suggestions towards designing our analysis.

References

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, 2023. URL https://arxiv.org/abs/2303.08112.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Shan Carter, Zan Armstrong, Ludwig Schubert, Ian Johnson, and Chris Olah. Activation atlas. *Distill*, 4(3):e15, 2019.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers, 2024. URL https://arxiv.org/abs/2309.16588.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Drain, et al. Superposition, memorization, and double descent. *arXiv preprint arXiv:2205.10630*, 2022.

Yao Fu. How do language models put attention weights over long context?, 2024.

Sachin Goyal, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, and Vaishnavh Nagarajan. Think before you speak: Training language models with pause tokens, 2024. URL https://arxiv.org/abs/2310.02226.

Hangfeng He and Weijie J. Su. A law of next-token prediction in large language models, 2024. URL https://arxiv.org/abs/2408.13442.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding, 2021. URL https://arxiv.org/abs/2009.03300.

Sarthak Jain and Byron C Wallace. Attention is not explanation. *arXiv preprint arXiv:1902.10186*, 2019.

Wentao Jiang, Jing Zhang, Di Wang, Qiming Zhang, Zengmao Wang, and Bo Du. Lemevit: Efficient vision transformer with learnable meta tokens for remote sensing image interpretation, 2024. URL https://arxiv.org/abs/2405.09789.

Shohei Kobayashi and Ken-ichi Kawarabayashi. Attention visualization via linear decomposition of self-attention heads. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5161–5180, 2020.

Tianyang Lin, Yuxin Wang, Xiangyang Liu, and Xipeng Qiu. A survey of transformers, 2021. URL https://arxiv.org/abs/2106.04554.

Rafael Müller, Simon Kornblith, and Geoffrey Hinton. *When does label smoothing help?* Curran Associates Inc., Red Hook, NY, USA, 2019.

Neel Nanda. A mechanistic interpretability analysis of grokking, 2023. URL https://www.alignmentforum.org/posts/N6WM6hs7RQMKDhYjB/a-mechanistic-interpretability-analysis-of-grokking. Accessed: 2024-10-05.

Nostalgebraist. Interpreting gpt: The logit lens, 2020. URL https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens. Accessed: 2024-10-05.

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

Chris Olah et al. Zoom in: An introduction to mechanistic interpretability. *Distill*, 5(3):e19, 2020.

Jacob Pfau, William Merrill, and Samuel R. Bowman. Let's think dot by dot: Hidden computation in transformer language models, 2024. URL https://arxiv.org/abs/2404.15758.

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, 2023. URL https://arxiv.org/abs/1910.10683.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Workshop at International Conference on Learning Representations*, 2013.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014a. URL http://jmlr.org/papers/v15/srivastava14a.html.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958, 2014b.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Jesse Vig. A multiscale visualization of attention in the transformer model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 37–42, 2019.

Eric Zelikman, Georges Harik, Yijia Shao, Varuna Jayasiri, Nick Haber, and Noah D. Goodman. Quiet-star: Language models can teach themselves to think before speaking, 2024. URL https://arxiv.org/abs/2403.09629.

A Related Work

Our work draws upon two key areas of recent work: (1.) contextualizing and visualizing internal representations, and (2.) special tokens.

Visualizing Transformer Representations. Recent advancements in mechanistic interpretability have provided novel techniques for visualizing and understanding the internal representations of transformer models. These methods offer insights into how transformers process information and make predictions. For visualizing internal representations, the Logit Lens [Nostalgebraist, 2020] remains a fundamental technique for understanding how transformer models evolve their predictions across layers. Belrose et al. [2023] improved upon this with the Tuned Lens, addressing biased estimates and compatibility issues with certain model families. Attention visualization techniques are crucial for understanding how transformers process information. Vig [2019] introduced tools like head_view to visualize multi-head attention across layers, while Kobayashi and Kawarabayashi [2020] proposed methods for linear decomposition of self-attention heads. Circuit analysis aims to identify and understand specific computational circuits within neural networks. Nanda [2023] demonstrated the potential for understanding complex operations by reverse-engineering modular addition in transformers. Concurrently, Elhage et al. [2022] explored how models represent more features than they have neurons through the superposition hypothesis. Gradient-based methods offer insights into which parts of the input are most influential for the model's decisions. Sundararajan et al. [2017] introduced Integrated Gradients, attributing the prediction of a deep network to its input features. Simonyan et al. [2013] proposed saliency maps to highlight influential input parts for model decisions. There is ongoing work in understanding individual neurons and activation patterns. Carter et al. [2019] introduced Activation Atlases for visualizing high-dimensional neuron activations in 2D space. Olah et al. [2017] developed methods to generate inputs that maximally activate specific neurons or layers.

Special Tokens. Our approach of using meta-tokens to investigate model internals is motivated by the efficacy of special tokens in various tasks. Zelikman et al. [2024] introduced special tokens to signal rationale generation, while Jiang et al. [2024] demonstrated how meta tokens can sparsely represent dense image tokens in vision transformers. Finally, Darcet et al. [2024] showed that empty register tokens help store intermediate computations in vision transformers.

Building on these prior works, we introduce meta-tokens as an abstraction within a modified attention mechanism. These tokens are continuously updated across layers, creating meaningful representations that provide insights into the model's internals. By masking other tokens during meta-attention, we smooth attention distribution, preventing attention sinks and offering stable anchors. Our approach draws inspiration most heavily from [Goyal et al., 2024] and [Jiang et al., 2024] achieves comparable benchmark performance.

B Full Architecture Details

We provide a full outline of the architecture design out method uses. Our architecture is equivalent to the NanoGPT (GPT-2) architecture, while introducing the meta-attention block after the initial causal masked attention and layer normalization computation.

1. **Input Layer:** Given an input sequence of tokens $\mathbf{x} = \{x_1, x_2, \ldots, x_T\}$, we first embed each token into a continuous representation. The embedding process is defined as:
$$\mathbf{e}_t = E(x_t) + \mathbf{p}_t,$$
where $\mathbf{p}_t$ is the positional encoding for the t^{th} token, producing the embedded sequence $\mathbf{E} = \{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_T\}$.
2. **Causal Masked Self-Attention:** The first layer consists of the causal masked self-attention mechanism. For each head h, the attention operation is computed as:
$$\text{CausalAttention}_h(Q, K, V) = \text{softmax}\left(\frac{QK_h^\top}{\sqrt{d_k}} + M\right)V_h,$$
where Q, K, V are the query, key, and value matrices derived from the input embeddings $\mathbf{E}$, and M is the mask matrix.

3. **Meta Attention Layer:** After the causal masked self-attention, we integrate the meta-attention mechanism to specifically attend to the injected meta tokens. This operation is defined as:

$$\text{MetaAttention}(Q, K, V, P) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}} + M_{\text{causal}} + P\right)V,$$

where P is the meta mask constructed from the indices of the meta tokens.

4. **Feedforward Layer:** Following the attention layers, we pass the output through a feedforward neural network defined by:

$$\text{FFN}(x) = \text{ReLU}(xW_1 + b_1)W_2 + b_2,$$

where W_1, W_2 are weight matrices, and b_1, b_2 are bias vectors.

5. **Layer Normalization:** After both the causal self-attention and meta-attention operations, we apply layer normalization:

$$\text{LayerNorm}(x) = \frac{x - \mu}{\sigma + \epsilon},$$

where μ and σ are the mean and standard deviation of the features, and ϵ is a small constant for numerical stability.

6. **Final Output Layer:** The final layer projects the output of the last feedforward layer back to the vocabulary size to produce the s for the next token prediction:

$$\text{s} = \text{softmax}(xW_{\text{out}} + b_{\text{out}}),$$

where W_{out} and b_{out} are the output weight matrix and bias vector, respectively.

C Pre-training Hyperparameters and Model Details

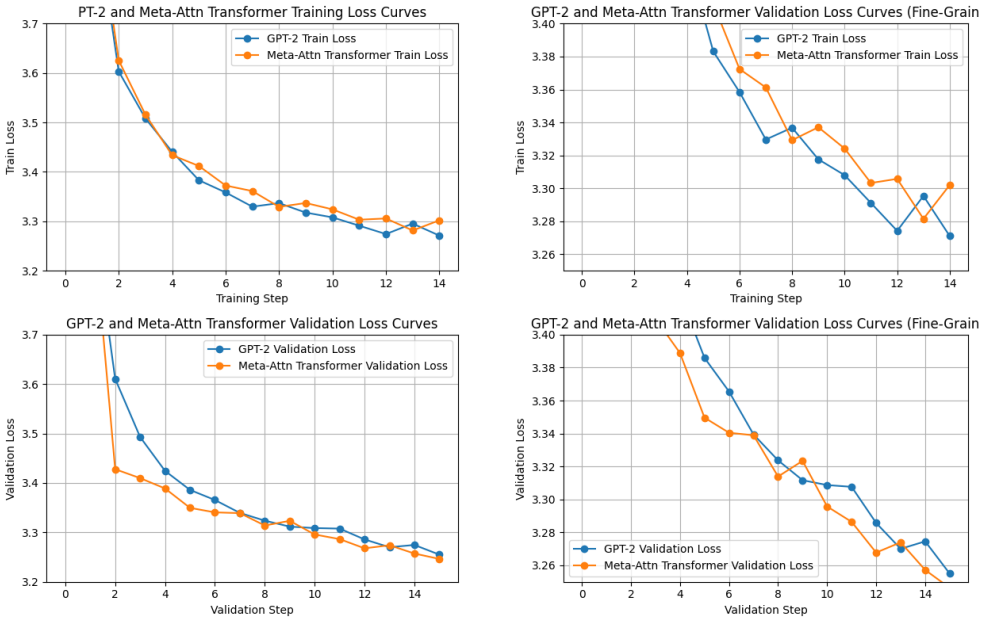
Our decoder-only modified GPT-2 model was pre-trained on the C4 dataset with the following configuration and hyperparameters:

Table 2: Pretraining Configuration Parameters

Parameter	**Value**
Batch Size	12
Gradient Accumulation Steps	40
Block Size	1024
Number of Layers	12
Number of Heads	12
Embedding Size	768
Learning Rate	6e-4
Weight Decay	1e-1
Max Iterations	600,000
Warmup Iterations	2,000
Minimum Learning Rate	6e-5
Dropout Rate	0.0
Initial Model	Resume

D Loss Curves

Below are the training and validation curves while pre-training our modified GPT-2 model with meta-attention.



(a) Training and Validation

(b) Training and Validation Loss (Fine-Grained)

Figure 4: Comparison of Loss Curves for GPT-2 and Meta-Attention Transformer at Different Scales

E GPT-3 Abstract Prompt

The following is the GPT-3 abstract prompt used in our logit-lens analysis:

> Recent work has demonstrated substantial gains on many NLP tasks and benchmarks by pre-training on a large corpus of text followed by fine-tuning on a specific task. While typically task-agnostic in architecture, this method still requires task-specific fine-tuning datasets of thousands or tens of thousands of examples. By contrast, humans can generally perform a new language task from only a few examples or from simple instructions – something which current NLP systems still largely struggle to do. Here we show that scaling up language models greatly improves task-agnostic, few-shot performance, sometimes even reaching competitiveness with prior state-of-the-art finetuning approaches. Specifically, we train GPT-3, an autoregressive language model with 175 billion parameters, 10x more than any previous non-sparse language model, and test its performance in the few-shot setting. For all tasks, GPT-3 is applied without any gradient updates or fine-tuning, with tasks and few-shot demonstrations specified purely via text interaction with the model. GPT-3 achieves strong performance on many NLP datasets, including translation, question-answering, and cloze tasks, as well as several tasks that require on-the-fly reasoning or domain adaptation, such as unscrambling words, using a novel word in a sentence, or performing 3-digit arithmetic. At the same time, we also identify some datasets where GPT-3's few-shot learning still struggles, as well as some datasets where GPT-3 faces methodological issues related to training on large web corpora. Finally, we find that GPT-3 can generate samples of news articles which human evaluators have difficulty distinguishing from articles written by humans. We discuss broader societal impacts of this finding and of GPT-3 in general.

F Logit Lens to Reveal KL Divergence With Respect to the Output

The KL divergence heatmap provides further evidence for the rank and logit behavior, showing that meta-tokens help reduce KL divergence earlier in the network. This indicates that meta-tokens stabilize predictions faster, refining the prediction distribution more effectively across layers. Strengthening the claim that meta tokens accelerate convergence to the proper output token.

F.1

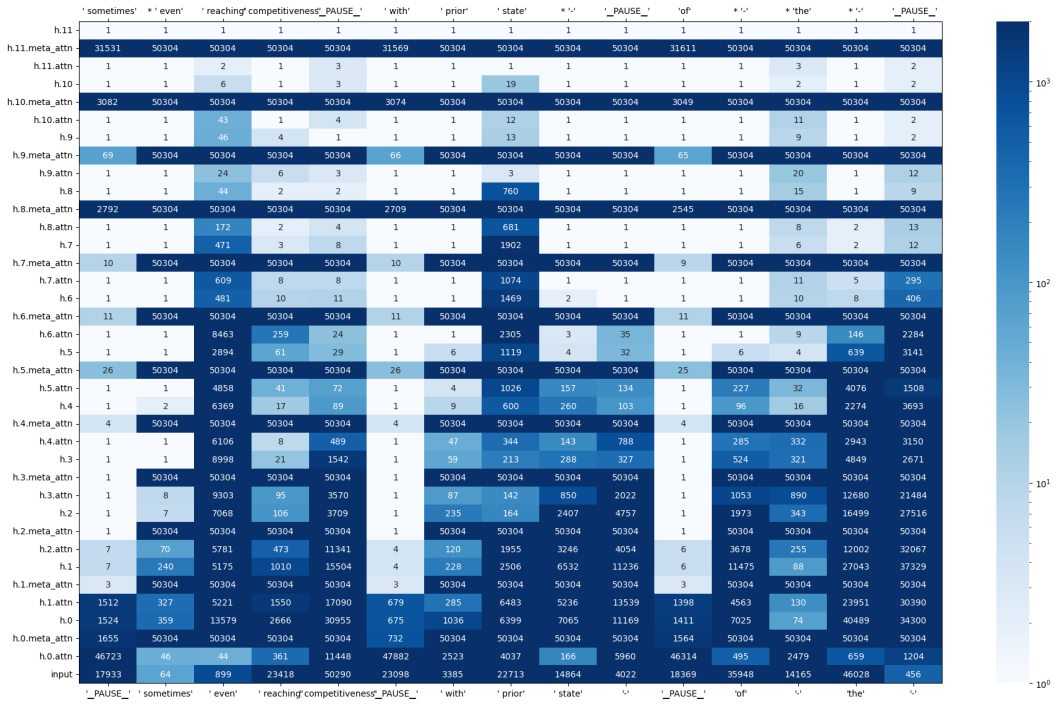


Figure 5: Sub-Block Rank

F.2

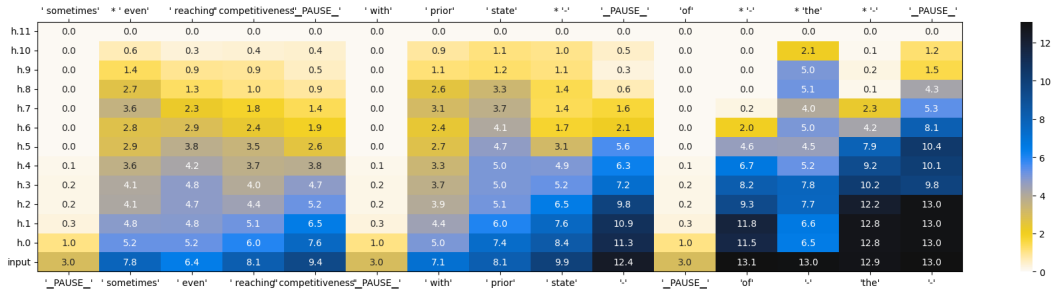


Figure 6: KL Divergence with respect to the output distribution