ZeroTuning: Unlocking the Initial Token's Power to Enhance Large Language Models Without Training

Feijiang Han [1] **Xiaodong Yu** [2 1] **Jianheng Tang** [3] **Qingyun Zeng** [4 1] **Licheng Guo** [1] **Lyle Ungar** [1]

Abstract

Training-free methods for enhancing large language models (LLMs) have attracted growing interest recently, with token-level attention tuning emerging as an interpretable and promising direction. However, existing methods typically rely on auxiliary mechanisms to identify important or irrelevant task-specific tokens, introducing potential bias and limiting applicability. In this work, we uncover a surprising and elegant alternative: the semantically empty initial token (e.g., <BOS> in Llama) serves as a powerful and underexplored control point for optimizing model behavior. Through theoretical analysis, we show that tuning the initial token's attention sharpens or flattens the attention distribution over subsequent tokens, and its role as an attention sink amplifies this effect. Empirically, we find that: (1) tuning its attention improves LLM performance across tasks more effectively than tuning other task-specific tokens; (2) the effect follows a consistent trend across layers, with earlier layers having greater impact, but varies across attention heads, with different heads showing distinct preferences in how they attend to this token. Based on these findings, we propose **ZeroTuning**, a training-free approach that improves LLM performance by applying head-specific attention adjustments to this special token. Despite tuning only one token, ZeroTuning achieves higher average performance on text classification, multiple-choice QA, and multi-turn conversation tasks across models such as LLama, Qwen, and DeepSeek. For example, ZeroTuning improves Llama-3.1-8B by 11.71% on classification tasks, 2.64% on QA tasks, and raises its multi-turn score from 7.804 to 7.966. The method is also robust to limited resources, few-shot settings, long contexts, quantization, decoding strategies, and prompt variations. Our work sheds light on a previously overlooked control point in LLMs, offering new insights into both inference-time tuning and model interpretability.

1. Introduction

Recent research has explored the potential of lightweight and training-free methods to adapt pre-trained large language models (LLMs) for downstream applications. One promising direction is **token-level attention tuning**, which modifies the attention distribution at inference time without changing any parameters. Unlike fine-tuning (Hu et al., 2021; Dettmers et al., 2023; Hayou et al., 2024; Lester et al., 2021; Liu et al., 2022; Li & Liang, 2021) or prompt engineering (Wei et al., 2022; Wang et al., 2022; Chen et al., 2022), which treat LLMs as a black box, token-level attention tuning introduces a more transparent way to guide the model's focus by increasing attention to key tokens or downplaying irrelevant ones. These methods can even outperform prompt-based approaches in certain scenarios, such as open-domain question answering (Zhang et al., 2024a). Examples include PASTA (manual token selection) (Zhang et al., 2023a), AutoPASTA (LLM-guided token identification) (Zhang et al., 2024a), and ACT (attention-thresholding-based sink token identification) (Yu et al., 2024). Similar ideas have also been applied to vision-language models to mitigate hallucinations by increasing attention to image tokens (Liu et al., 2024b; Zhu et al., 2024; Wei & Zhang, 2024).

While effective, these methods require external mechanisms to identify task-specific tokens, which may introduce potential bias (e.g., emphasizing misleading tokens) and limit applicability when the importance of tokens is unclear or attention maps are not accessible. This motivates a question: **Is it possible to improve model performance by tuning the attention to a universal and task-agnostic token without relying on task-specific token identification?**

In this paper, we reveal that the answer lies in a widely existing yet often overlooked token: the initial token (e.g., <BOS> in

*Equal contribution [1]University of Pennsylvania, Philadelphia, USA [2]AMD, USA [3]Peking University, Beijing, China [4]Microsoft, USA. Correspondence to: Lyle Ungar <ungar@cis.upenn.edu>.

Accepted at Methods and Opportunities at Small Scale (MOSS), ICML 2025, Vancouver, Canada.

LLaMA). Though it is commonly researched and regarded as an attention sink (Xiao et al., 2023; Kaul et al., 2024; Gu et al., 2024; Barbero et al., 2025), prior tuning methods have largely ignored it, focusing instead on visible input tokens. Our theoretical analysis shows that tuning the attention of this special token can sharpen or flatten the attention distribution across subsequent task-relevant tokens, while preserving their relative importance. Its role as an attention sink further amplifies this effect (Appendix A.1). Additionally, we demonstrate that modifying the initial token's key or query states achieves similar effects when direct tuning of attention weights is infeasible.

We conducted a series of experiments to investigate the impact of tuning the initial token's attention and found the following:

1. Despite lacking semantic meaning, the initial token acts as an effective control point for steering model behavior: tuning its attention consistently improves LLM performance more effectively than tuning any other token (Appendix A.2).
2. The tuning effect propagates consistently across layers, with the shallow and middle layers contributing more than the deeper ones; jointly tuning all layers yields the best results (Appendix A.3).
3. The impact also varies across attention heads: some heads respond positively to increased attention on this token, while others respond negatively. We categorize these as up-effective and down-effective heads, and show that selectively tuning them outperforms uniform tuning across all heads (Appendixs A.4 and A.5).

These findings motivate **ZeroTuning**, a simple and effective training-free method that recalibrates the initial token's attention to improve the LLM's performance without requiring task-specific token identification (Section 2). Experiments demonstrate that, despite tuning only one token, ZeroTuning achieves higher average performance on text classification (e.g., SST2, BoolQ), multiple-choice QA (e.g., MMLU, LogiQA), and multi-turn conversation (e.g., MT-Bench) benchmarks across LLMs including Llama-3.1-8B-Instruct, Llama-2-13B-Instruct, Qwen-2-7B, and Deepseek-R1-14B. For example, ZeroTuning improves Llama-3.1-8B by 11.71% on classification tasks, 2.64% on QA tasks, and raises its multi-turn score from 7.804 to 7.966. The method is also robust under various conditions, including limited resources, few-shot settings, long contexts, quantization, decoding strategies, and prompt variations.

Our work sheds light on a previously overlooked control point in LLMs, offering new insights into both inference-time tuning and model interpretability.

2. ZeroTuning Methodology

Building on our empirical findings, we propose **ZeroTuning**, a simple yet effective method that adjusts the initial token's attention in a head-specific manner to enhance LLM performance, without requiring task-specific token identification. ZeroTuning comprises three steps:

1. **Head Behavior Profiling:** Using a small calibration set (e.g., a few validation examples), we assess each attention head's sensitivity to scaling the initial token's attention. A head is classified as *up-effective* if increased attention improves performance, and *down-effective* otherwise.
2. **Selective Rescaling:** Based on the dominant head type (i.e., the group with the highest proportion), we apply a scaling factor γ exclusively to that group to adjust the initial token's attention.[1]
3. **Renormalization:** The scaled attention scores are re-normalized using the softmax function to maintain a valid distribution.

For optimized attention implementations (e.g., SDPA, Flash Attention) where direct modification of attention scores is infeasible, ZeroTuning applies scaling to the query or key states before attention computation. We demonstrate that tuning key states yields similar effects to attention-based tuning, but induces steeper changes in the attention distribution of subsequent tokens (see Appendix C). The following sections evaluate the performance and robustness of ZeroTuning across different tasks and model settings.

3. Experimental Results

3.1. Experimental Setup

Models, Tasks, and Datasets. Models: We evaluate ZeroTuning on four LLMs with distinct attention implementations: Llama-3.1-8B-Instruct (Grattafiori et al., 2024) and Llama-2-13B-Chat (Touvron et al., 2023) with eager attention, Qwen-

[1]We explore the effect of scaling different subsets of heads; results (Figure 8, Appendix H) show that tuning the top 20% to 70% yields better performance.

2-7B (Yang et al., 2024) with SDPA attention, and DeepSeek-R1-14B (DeepSeek-AI et al., 2025) with Flash attention.[2] Tasks and Datasets: Our experiments encompass three task types across 15 datasets: (1) Text Classification and Reasoning, including SST-2 (binary sentiment classification) (Socher et al., 2013), SST-5 (fine-grained sentiment analysis) (Socher et al., 2013), MR (movie review polarity detection) (Pang & Lee, 2005), SUBJ (subjectivity classification) (Pang & Lee, 2004), TREC (question type classification) (Li & Roth, 2002), CB (commitment detection) (De Marneffe et al., 2019), and BoolQ (boolean question answering) (Clark et al., 2019); (2) Domain-Specific Multiple-Choice, including MMLU (cross-domain knowledge testing) (Hendrycks et al., 2020), AQUA (math word problems) (Zheng et al., 2024a), MathQA (algebraic reasoning) (Amini et al., 2019), LogiQA (logical reasoning) (Liu et al., 2023), CQA (commonsense reasoning) (Talmor et al., 2018), PIQA (physical commonsense QA) (Bisk et al., 2020), and ARCC (scientific reasoning) (Clark et al., 2018); and (3) Multi-Round Conversation, using MT-Bench (Zheng et al., 2024a).

Baselines and Evaluation Metrics.

Baselines: We benchmark ZeroTuning against three baselines: (1) vanilla inference, which performs standard inference without any modifications; (2) ACT (Yu et al., 2024), which identifies sink tokens using an attention score threshold and reduces their attention weights; and (3) Auto-PASTA (Zhang et al., 2024a), which leverages an LLM to locate important tokens and enhance their attention weights. For ACT, we use the original hyperparameters ($\beta = 0.4$) and filter attention heads on a 10% validation set. For AutoPASTA, the evaluated LLM identifies key input regions based on prompts provided in Appendix L. Since ACT relies on explicit attention maps, it is only evaluated on LLaMA-3.1-8B-Instruct with eager attention implementation. Evaluation Metrics: We assess performance using accuracy for text classification and multiple-choice tasks. For the multi-round conversation task, we report average quality scores as evaluated by GPT-4, following the methodology outlined in (Zheng et al., 2024a).

Implementation Details. ZeroTuning is implemented using PyTorch and the Hugging Face Transformers library. We select zero-shot setting with standard greedy decoding for baseline consistency, and use prompt templates from (Ouyang et al., 2022; Sanh et al., 2021; Hao et al., 2022) across datasets (see Appendix L for details). Unless otherwise specified, we tune the top 40% of the heads. Each experiment uses a randomly selected validation set of 500 samples with the random seed fixed to 42.

3.2. Overall performance of ZeroTuning

Text Classification We first evaluate ZeroTuning on various text classification datasets using different LLMs, as shown in Table 2. Despite tuning only a single token, ZeroTuning consistently outperforms baselines and methods that require tuning more tokens. With Llama-3.1-8B-Instruct, it achieves an average improvement of 11.71% over vanilla, with peaks of 22.00% on SUBJ and 18.40% on SST-2. It outperforms AutoPASTA by an average of 7.71%. On Qwen-2-7B, ZeroTuning gains 13.09%, and on Deepseek-R1-14B, it improves by 4.20%, with a notable increase of 11.20% on TREC.

Domain-Specific Multiple Choice Next, we evaluate ZeroTuning on common domain-specific multiple-choice datasets under various settings, as shown in Table 3. For Llama-3.1-8B-Instruct, it increases the average accuracy by 2.64%, with gains of 3.40% on LogiQA and 1.40% on MMLU. Qwen-2-7B gains 1.74%, and Deepseek-R1-14B gains 2.15%, with an outstanding 7.80% on LogiQA.

Multi-Round Conversation We further demonstrate ZeroTuning's effectiveness in multi-round conversations using MT-Bench (Zheng et al., 2023), with results in Table 1. For Llama-3.1-8B-Instruct, ZeroTuning improves the average score by 0.162 points (7.966 vs. 7.804). For Llama-2-13B-Chat, it achieves a 0.266 points gain (6.916 vs. 6.650), showing its effectiveness in interactive settings.

Table 1. MT-Bench Performance Scores for Multi-Round Conversation Across Models

Model	First Turn	Second Turn	Average
gpt-4	8.956	9.025	8.991
Llama-3.1-8B-ZeroTuning	**8.294 (+0.029)**	**7.638 (+0.282)**	**7.966 (+0.162)**
Gpt-3.5-turbo	8.075	7.813	7.944
claude-instant-v1	7.800	8.013	7.906
claude-v1	8.150	7.650	7.900
Llama-3.1-8B-vanilla	8.265	7.353	7.804
Llama-2-13B-Chat-ZeroTuning	**7.106 (+0.043)**	**6.725 (+0.487)**	**6.916 (+0.266)**
Llama-2-13B-Chat-vanilla	7.063	6.238	6.650

[2]Eager, SDPA, and Flash are official attention implementations in modern Transformer libraries. Eager computes the full attention map; SDPA uses PyTorch's efficient API to select the optimal implementation; Flash relies on fused CUDA kernels from the FlashAttention library.

Table 2. Performance Comparison of Classification Tasks Across Models. The best performance in each dataset is **bolded**, the second best is underlined, and the ZeroTuning method is highlighted in gray.

Model	Method	Datasets							
		SST-2	SST-2	MR	BoolQ	CB	TREC	SUBJ	Avg.
Llama-3.1-8B-Instruct	Vanilla	73.20	45.40	89.20	69.60	82.14	14.00	44.60	59.59
	ACT	85.00	43.80	90.80	58.60	82.14	15.80	44.60	60.11
	Auto-PASTA	89.60	47.20	91.40	72.60	83.93	16.00	45.40	63.73
	ZeroTuning	**91.60**	**52.00**	**92.00**	**82.40**	**89.29**	**26.20**	**66.60**	**71.44**
Qwen-2-7B (SDPA)	Vanilla	78.80	45.40	72.40	85.00	78.50	12.60	13.00	55.10
	ACT	/	/	/	/	/	/	/	/
	Auto-PASTA	89.00	47.00	77.70	85.00	**89.29**	14.00	**57.00**	65.57
	ZeroTuning	**89.60**	**47.20**	**87.40**	**86.40**	85.71	**26.60**	54.40	**68.19**
Deepseek-R1-14B (Flash)	Vanilla	91.20	49.40	89.20	83.40	89.29	20.80	50.40	67.67
	ACT	/	/	/	/	/	/	/	/
	Auto-PASTA	92.00	**52.20**	89.80	83.40	**92.86**	22.60	50.40	69.04
	ZeroTuning	**93.00**	51.20	**90.20**	**88.00**	**92.86**	**32.00**	**55.80**	**71.87**

Table 3. Performance Comparison of Multiple-Choice Tasks Across Models

Model	Method	Datasets							
		MMLU	AQUA	MathQA	LogiQA	CQA	PIQA	ARCC	Avg.
Llama-3.1-8B-Instruct	Vanilla	67.40	25.69	33.60	39.40	77.60	83.60	84.62	58.84
	ACT	67.60	29.64	33.60	38.00	77.60	83.00	84.62	59.15
	Auto-PASTA	67.00	**31.23**	35.20	40.40	78.20	84.60	84.62	60.18
	ZeroTuning	**68.80**	30.43	**36.60**	**42.80**	**80.40**	**85.40**	**85.95**	**61.48**
Qwen-2-7B (SDPA)	Vanilla	69.80	36.76	39.20	45.00	78.80	85.20	86.96	63.10
	ACT	/	/	/	/	/	/	/	/
	Auto-PASTA	69.80	39.13	39.20	45.00	**82.60**	85.40	86.96	64.01
	ZeroTuning	**70.40**	**39.92**	**40.20**	**47.40**	81.80	**86.20**	**87.96**	**64.84**
Deepseek-R1-14B (Flash)	Vanilla	66.60	38.74	38.20	27.80	78.20	84.20	86.62	60.05
	ACT	/	/	/	/	/	/	/	/
	Auto-PASTA	66.60	38.74	39.40	28.20	78.20	84.40	86.62	60.31
	ZeroTuning	**70.00**	**39.13**	**39.80**	**35.60**	**78.60**	**85.00**	**87.29**	**62.20**

4. Further Analysis

We provide additional analyses to deepen the understanding of ZeroTuning's behavior and practical utility under varying conditions. **(1) Tuning Different Matrices:** Appendix C begins with a comparison between tuning attention scores and key states. Tuning attention scores yields stable improvements due to its linear influence, whereas tuning key states is sensitive and can cause sharp performance drops because of exponential effects. **(2) Resource Constraints:** Appendix D evaluates ZeroTuning across three simulated computational budgets, revealing consistent gains from minimal scaling (Level 0) to full head classification and parameter search (Level 2). **(3) Context Length Sensitivity:** Appendix E tests robustness to long contexts with irrelevant tokens, showing ZeroTuning maintains accuracy where vanilla models degrade. **(4) Few-Shot Learning Robustness:** Appendix F demonstrates stable few-shot gains and fewer invalid outputs, indicating ZeroTuning complements in-context learning. **(5) Prompt Variation Sensitivity:** Appendix I assesses prompt modifications such as dropped or altered instructions; ZeroTuning sustains strong performance. **(6) Effect of Attention Head Subset Selection:** Appendix H shows that tuning a moderate proportion of attention heads (20%–70%) outperforms both full and minimal tuning. **(7) Quantization Robustness:** Appendix J shows a consistent trend across 16-, 8-, and 4-bit quantization, with ZeroTuning partially mitigating degradation from low-precision quantization. **(8) Decoding Strategy Impact:** Appendix G evaluates various decoding methods (Top-k, Top-p, Beam Search), confirming ZeroTuning's effectiveness.

5. Conclusion

In this work, we propose ZeroTuning, a novel training-free method to enhance large language models (LLMs) by adjusting the attention of the initial token. Our findings reveal that tuning the initial token's attention can lead to substantial improvements in LLM performance across a variety of tasks, including text classification, question answering, and multi-round conversations. Notably, ZeroTuning outperforms previous task-specific tuning methods, achieving robust and transferable gains without requiring task-specific token identification. The approach offers a fresh perspective on optimizing LLMs by leveraging the often-overlooked power of the initial token, making it a simple yet effective solution for enhancing model performance. This work advances inference-time tuning and contributes to the interpretability of LLMs. It opens up new avenues for further exploration and optimization in the field.

References

Amini, A., Gabriel, S., Lin, P., Koncel-Kedziorski, R., Choi, Y., and Hajishirzi, H. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. *arXiv preprint arXiv:1905.13319*, 2019.

Barbero, F., Arroyo, Á., Gu, X., Perivolaropoulos, C., Bronstein, M., Pascanu, R., et al. Why do llms attend to the first token? *arXiv preprint arXiv:2504.02732*, 2025.

Bisk, Y., Zellers, R., Gao, J., Choi, Y., et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Chen, T., Tan, Z., Gong, T., Wu, Y., Chu, Q., Liu, B., Ye, J., and Yu, N. Llama slayer 8b: Shallow layers hold the key to knowledge injection, 2024. URL https://arxiv.org/abs/2410.02330.

Chen, W., Ma, X., Wang, X., and Cohen, W. W. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *Trans. Mach. Learn. Res.*, 2023, 2022. URL https://api.semanticscholar.org/CorpusID:253801709.

Clark, C., Lee, K., Chang, M.-W., Kwiatkowski, T., Collins, M., and Toutanova, K. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*, 2019.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

De Marneffe, M.-C., Simons, M., and Tonhauser, J. The commitmentbank: Investigating projection in naturally occurring discourse. In *proceedings of Sinn und Bedeutung*, volume 23, pp. 107–124, 2019.

DeepSeek-AI, Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., Zhang, X., Yu, X., Wu, Y., Wu, Z. F., Gou, Z., Shao, Z., Li, Z., Gao, Z., Liu, A., Xue, B., Wang, B., Wu, B., Feng, B., Lu, C., Zhao, C., Deng, C., Zhang, C., Ruan, C., Dai, D., Chen, D., Ji, D., Li, E., Lin, F., Dai, F., Luo, F., Hao, G., Chen, G., Li, G., Zhang, H., Bao, H., Xu, H., Wang, H., Ding, H., Xin, H., Gao, H., Qu, H., Li, H., Guo, J., Li, J., Wang, J., Chen, J., Yuan, J., Qiu, J., Li, J., Cai, J. L., Ni, J., Liang, J., Chen, J., Dong, K., Hu, K., Gao, K., Guan, K., Huang, K., Yu, K., Wang, L., Zhang, L., Zhao, L., Wang, L., Zhang, L., Xu, L., Xia, L., Zhang, M., Zhang, M., Tang, M., Li, M., Wang, M., Li, M., Tian, N., Huang, P., Zhang, P., Wang, Q., Chen, Q., Du, Q., Ge, R., Zhang, R., Pan, R., Wang, R., Chen, R. J., Jin, R. L., Chen, R., Lu, S., Zhou, S., Chen, S., Ye, S., Wang, S., Yu, S., Zhou, S., Pan, S., Li, S. S., Zhou, S., Wu, S., Ye, S., Yun, T., Pei, T., Sun, T., Wang, T., Zeng, W., Zhao, W., Liu, W., Liang, W., Gao, W., Yu, W., Zhang, W., Xiao, W. L., An, W., Liu, X., Wang, X., Chen, X., Nie, X., Cheng, X., Liu, X., Xie, X., Liu, X., Yang, X., Li, X., Su, X., Lin, X., Li, X. Q., Jin, X., Shen, X., Chen, X., Sun, X., Wang, X., Song, X., Zhou, X., Wang, X., Shan, X., Li, Y. K., Wang, Y. Q., Wei, Y. X., Zhang, Y., Xu, Y., Li, Y., Zhao, Y., Sun, Y., Wang, Y., Yu, Y., Zhang, Y., Shi, Y., Xiong, Y., He, Y., Piao, Y., Wang, Y., Tan, Y., Ma, Y., Liu, Y., Guo, Y., Ou, Y., Wang, Y., Gong, Y., Zou, Y., He, Y., Xiong, Y., Luo, Y., You, Y., Liu, Y., Zhou, Y., Zhu, Y. X., Xu, Y., Huang, Y., Li, Y., Zheng, Y., Zhu, Y., Ma, Y., Tang, Y., Zha, Y., Yan, Y., Ren, Z. Z., Ren, Z., Sha, Z., Fu, Z., Xu, Z., Xie, Z., Zhang, Z., Hao, Z., Ma, Z., Yan, Z., Wu, Z., Gu, Z., Zhu, Z., Liu, Z., Li, Z., Xie, Z., Song, Z., Pan, Z., Huang, Z., Xu, Z., Zhang, Z., and Zhang, Z. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. Qlora: Efficient finetuning of quantized llms. *ArXiv*, abs/2305.14314, 2023. URL https://api.semanticscholar.org/CorpusID:258841328.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Gu, X., Pang, T., Du, C., Liu, Q., Zhang, F., Du, C., Wang, Y., and Lin, M. When attention sink emerges in language models: An empirical view. *arXiv preprint arXiv:2410.10781*, 2024.

Guo, T., Pai, D., Bai, Y., Jiao, J., Jordan, M. I., and Mei, S. Active-dormant attention heads: Mechanistically demystifying extreme-token phenomena in llms, 2024. URL https://arxiv.org/abs/2410.13835.

Gurnee, W., Horsley, T., Guo, Z. C., Kheirkhah, T. R., Sun, Q., Hathaway, W., Nanda, N., and Bertsimas, D. Universal neurons in gpt2 language models, 2024. URL https://arxiv.org/abs/2401.12181.

Hao, Y., Sun, Y., Dong, L., Han, Z., Gu, Y., and Wei, F. Structured prompting: Scaling in-context learning to 1,000 examples. *arXiv preprint arXiv:2212.06713*, 2022.

Hayou, S., Ghosh, N., and Yu, B. Lora+: Efficient low rank adaptation of large models. *ArXiv*, abs/2402.12354, 2024. URL https://api.semanticscholar.org/CorpusID:267750102.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Hu, J. E., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., and Chen, W. Lora: Low-rank adaptation of large language models. *ArXiv*, abs/2106.09685, 2021. URL https://api.semanticscholar.org/CorpusID:235458009.

Jin, M., Yu, Q., Huang, J., Zeng, Q., Wang, Z., Hua, W., Zhao, H., Mei, K., Meng, Y., Ding, K., et al. Exploring concept depth: How large language models acquire knowledge and concept at different layers? *arXiv preprint arXiv:2404.07066*, 2024.

Jin, M., Yu, Q., Huang, J., Zeng, Q., Wang, Z., Hua, W., Zhao, H., Mei, K., Meng, Y., Ding, K., Yang, F., Du, M., and Zhang, Y. Exploring concept depth: How large language models acquire knowledge and concept at different layers? In Rambow, O., Wanner, L., Apidianaki, M., Al-Khalifa, H., Eugenio, B. D., and Schockaert, S. (eds.), *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 558–573, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.coling-main.37/.

Kamoda, G., Heinzerling, B., Inaba, T., Kudo, K., Sakaguchi, K., and Inui, K. Weight-based analysis of detokenization in language models: Understanding the first stage of inference without inference, 2025. URL https://arxiv.org/abs/2501.15754.

Kaul, P., Ma, C., Elezi, I., and Deng, J. From attention to activation: Unraveling the enigmas of large language models. In *The Thirteenth International Conference on Learning Representations*, 2024.

Lester, B., Al-Rfou, R., and Constant, N. The power of scale for parameter-efficient prompt tuning. In *Conference on Empirical Methods in Natural Language Processing*, 2021. URL https://api.semanticscholar.org/CorpusID:233296808.

Li, X. and Roth, D. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*, 2002.

Li, X. L. and Liang, P. Prefix-tuning: Optimizing continuous prompts for generation. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4582–4597, 2021. URL https://api.semanticscholar.org/CorpusID:230433941.

Liu, H., Liu, J., Cui, L., Teng, Z., Duan, N., Zhou, M., and Zhang, Y. Logiqa 2.0—an improved dataset for logical reasoning in natural language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:2947–2962, 2023.

Liu, R., Bai, H., Lin, H., Li, Y., Gao, H., Xu, Z., Hou, L., Yao, J., and Yuan, C. Intactkv: Improving large language model quantization by keeping pivot tokens intact. *arXiv preprint arXiv:2403.01241*, 2024a.

Liu, S., Zheng, K., and Chen, W. Paying more attention to image: A training-free method for alleviating hallucination in lvlms. In *European Conference on Computer Vision*, pp. 125–140. Springer, 2024b.

Liu, X., Ji, K., Fu, Y., Tam, W. L., Du, Z., Yang, Z., and Tang, J. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Annual Meeting of the Association for Computational Linguistics*, 2022. URL https://api.semanticscholar.org/CorpusID:248780177.

Lu, Y., Zeng, J., Zhang, J., Wu, S., and Li, M. Attention calibration for transformer in neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1288–1298, 2021.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

Pang, B. and Lee, L. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. *arXiv preprint cs/0409058*, 2004.

Pang, B. and Lee, L. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. *arXiv preprint cs/0506075*, 2005.

Sanh, V., Webson, A., Raffel, C., Bach, S. H., Sutawika, L., Alyafeai, Z., Chaffin, A., Stiegler, A., Scao, T. L., Raja, A., et al. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*, 2021.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., and Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642, 2013.

Talmor, A., Herzig, J., Lourie, N., and Berant, J. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*, 2018.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P. S., Lachaux, M.-A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E. M., Subramanian, R., Tan, X. E., Tang, B., Taylor, R., Williams, A., Kuan, J. X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., and Scialom, T. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E. H., and Zhou, D. Self-consistency improves chain of thought reasoning in language models. *ArXiv*, abs/2203.11171, 2022. URL https://api.semanticscholar.org/CorpusID:247595263.

Wei, J. and Zhang, X. Dopra: Decoding over-accumulation penalization and re-allocation in specific weighting layer. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 7065–7074, 2024.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Chi, E. H., Xia, F., Le, Q., and Zhou, D. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903, 2022. URL https://api.semanticscholar.org/CorpusID:246411621.

Xiao, G., Tian, Y., Chen, B., et al. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*, 2023.

Yang, A., Yang, B., Hui, B., Zheng, B., Yu, B., Zhou, C., Li, C., Li, C., Liu, D., Huang, F., Dong, G., Wei, H., Lin, H., Tang, J., Wang, J., Yang, J., Tu, J., Zhang, J., Ma, J., Yang, J., Xu, J., Zhou, J., Bai, J., He, J., Lin, J., Dang, K., Lu, K., Chen, K., Yang, K., Li, M., Xue, M., Ni, N., Zhang, P., Wang, P., Peng, R., Men, R., Gao, R., Lin, R., Wang, S., Bai, S., Tan, S., Zhu, T., Li, T., Liu, T., Ge, W., Deng, X., Zhou, X., Ren, X., Zhang, X., Wei, X., Ren, X., Liu, X., Fan, Y., Yao, Y., Zhang, Y., Wan, Y., Chu, Y., Liu, Y., Cui, Z., Zhang, Z., Guo, Z., and Fan, Z. Qwen2 technical report, 2024. URL https://arxiv.org/abs/2407.10671.

Yu, Z., Wang, Z., Fu, Y., et al. Unveiling and harnessing hidden attention sinks: Enhancing large language models without training through attention calibration. *arXiv preprint arXiv:2406.15765*, 2024.

Zhang, Q., Singh, C., Liu, L., et al. Tell your model where to attend: Post-hoc attention steering for llms. *arXiv preprint arXiv:2311.02262*, 2023a.

Zhang, Q., Yu, X., Singh, C., Liu, X., Liu, L., Gao, J., Zhao, T., Roth, D., and Cheng, H. Model tells itself where to attend: Faithfulness meets automatic attention steering, 2024a. https://arxiv.org/abs/2409.10790.

Zhang, Y., Dong, Y., and Kawaguchi, K. Investigating layer importance in large language models. *arXiv preprint arXiv:2409.14381*, 2024b.

Zhang, Z., Sheng, Y., Zhou, T., Chen, T., Zheng, L., Cai, R., Song, Z., Tian, Y., Ré, C., Barrett, C., et al. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36:34661–34710, 2023b.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*, 2023.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36, 2024a.

Zheng, Z., Wang, Y., Huang, Y., Song, S., Yang, M., Tang, B., Xiong, F., and Li, Z. Attention heads of large language models: A survey, 2024b. URL https://arxiv.org/abs/2409.03752.

Zhu, L., Ji, D., Chen, T., Xu, P., Ye, J., and Liu, J. Ibd: Alleviating hallucinations in large vision-language models via image-biased decoding. *ArXiv*, abs/2402.18476, 2024. URL https://api.semanticscholar.org/CorpusID:268041475.

A. Unveiling the Power of the Initial Token

In this section, we delve into the impact of tuning the initial token's attention. Unless otherwise specified, all experiments use the Llama-3.1-8B-Instruct model, with dataset details provided in Section 3.1.

A.1. Formalizing the Tuning Process

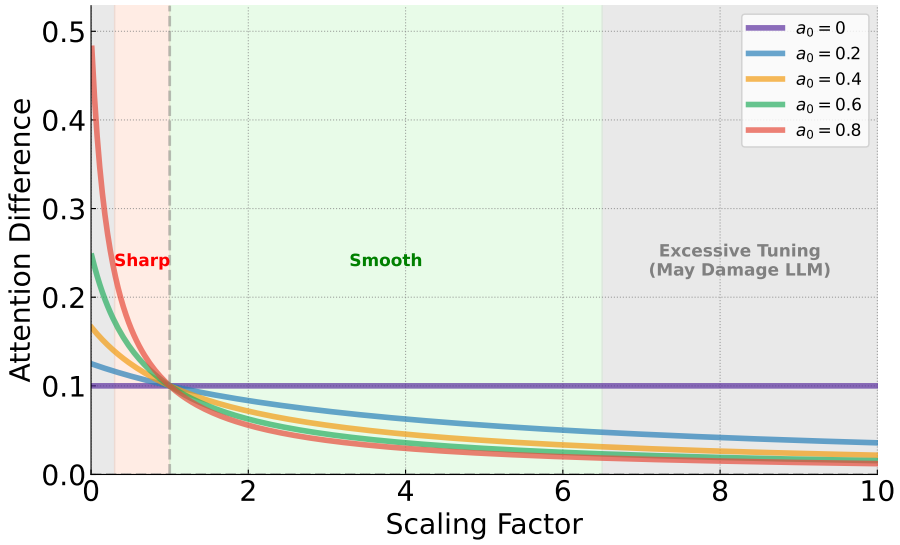


Figure 1. Effect of the scaling factor $\gamma \in [0, 10]$ and initial token's attention $a_0 \in \{0, 0.2, 0.4, 0.6, 0.8\}$ on attention differences $|a'_i - a'_j|$ between non-initial tokens. Shaded regions highlight tuning regimes: excessive tuning ($\gamma \in [0, 0.3] \cup [6.5, 10]$) that may impair LLM performance, moderate sharpening ($\gamma \in [0.3, 1]$) that amplifies token-level attention differences, and moderate smoothing ($\gamma \in [1, 6.5]$) that reduces them. Decreasing γ sharpens the distribution by amplifying attention differences, while increasing γ smooths it by reducing disparities.

In decoder-only Transformers, the next token is generated using the final token as the query, attending to all preceding tokens (including itself) as keys. Consider a single-layer, single-head attention mechanism with an input sequence:

$$\mathbf{X} = [x_0, x_1, \ldots, x_{T-1}] \in \mathbb{R}^{d \times T}. \tag{1}$$

At generation step T, the query is x_{T-1}, and the attention weights over the keys x_0 to x_{T-1} form a probability vector:

$$\boldsymbol{a} = [a_0, a_1, \ldots, a_{T-1}], \quad a_i \geq 0, \quad \sum_{i=0}^{T-1} a_i = 1, \tag{2}$$

where a_0 denotes the attention of the initial token, while a_1 to a_{T-1} correspond to subsequent tokens.

To control the influence of x_0, we scale its attention weight by a factor $\gamma > 0$ and re-normalize:

$$a'_0 = \frac{\gamma a_0}{D}, \quad a'_i = \frac{a_i}{D} \quad \text{for } i = 1, \ldots, T-1, \tag{3}$$

where the normalization constant is

$$D = \gamma a_0 + \sum_{i=1}^{T-1} a_i = \gamma a_0 + (1 - a_0) = (\gamma - 1)a_0 + 1. \tag{4}$$

This rescaling preserves the relative proportions among non-initial tokens:

$$\frac{a'_i}{\sum_{j=1}^{T-1} a'_j} = \frac{\frac{a_i}{D}}{\sum_{j=1}^{T-1} \frac{a_j}{D}} = \frac{a_i}{\sum_{j=1}^{T-1} a_j}, \quad \text{for } i \geq 1, \tag{5}$$

but compresses or expands their differences as

$$a'_i - a'_j = \frac{a_i - a_j}{D} = \frac{a_i - a_j}{(\gamma - 1)a_0 + 1}, \quad \text{for } i, j \geq 1. \tag{6}$$

Intuitively, setting $\gamma > 1$ amplifies a_0 and flattens the attention distribution over the remaining tokens. Conversely, $\gamma < 1$ suppresses a_0 and sharpens distinctions among other tokens. When $\gamma = 1$, the attention remains unchanged. As illustrated in Figure 1, the value of a_0 determines the strength of the tuning effect—larger a_0 values lead to more pronounced modulation. However, extreme values of γ can be detrimental: overly small values degrade model performance, a behavior consistent with prior findings on attention sink studies (Barbero et al., 2025), while excessively large values can also hurt performance by suppressing attention to downstream tokens toward near-zero values.

Then, we analyze how this tuning influences the final representations. Let the value states corresponding to each token be:

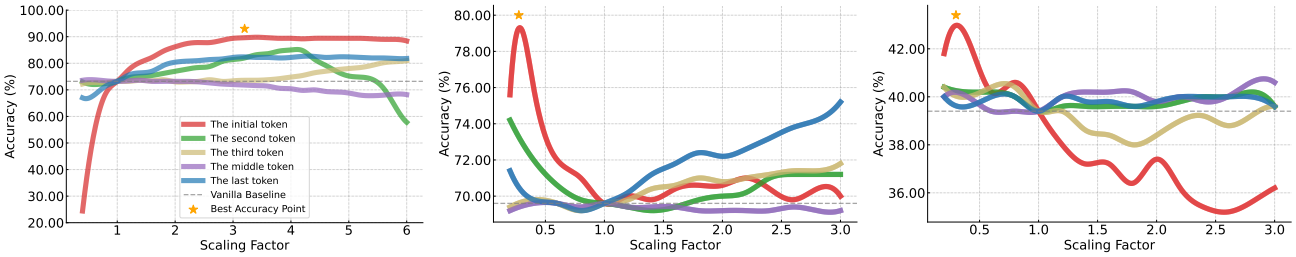
$$\mathbf{V} = [\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_{T-1}] \in \mathbb{R}^{d_v \times T}, \tag{7}$$

where $\mathbf{v}_i \in \mathbb{R}^{d_v}$ is the value vector associated with token x_i. The attention output is a weighted sum of these value vectors. Applying the scaled attention weights yields:

$$\begin{aligned} \mathbf{o} &= \sum_{i=0}^{T-1} a'_i \mathbf{v}_i = \frac{\gamma a_0}{D}\mathbf{v}_0 + \sum_{i=1}^{T-1} \frac{a_i}{D}\mathbf{v}_i, \\ &= \frac{1}{D}\left[\gamma a_0 \mathbf{v}_0 + \sum_{i=1}^{T-1} a_i \mathbf{v}_i\right]. \end{aligned} \tag{8}$$

Previous studies have observed the "value-state drain" phenomenon, wherein the value state of the initial token tends to be much smaller than those of the other tokens on average (Gurnee et al., 2024; Guo et al., 2024; Kamoda et al., 2025; Gu et al., 2024). Therefore, the contribution of the first term $\gamma a_0 \mathbf{v}_0$ to the final output is relatively small, and the value states of the subsequent tokens dominate the final representation. Tuning γ adjusts the relative contributions of these value states, influencing the final representation.

A.2. The Unique Importance of the Initial Token



(a) SST-2 (b) BoolQ (c) LogiQA

Figure 2. Impact of attention scaling factor γ on different token positions across three tasks: (a) SST-2, (b) BoolQ, and (c) LogiQA. Modifying the initial token's attention consistently yields significant accuracy improvements, often surpassing adjustments to other tokens.

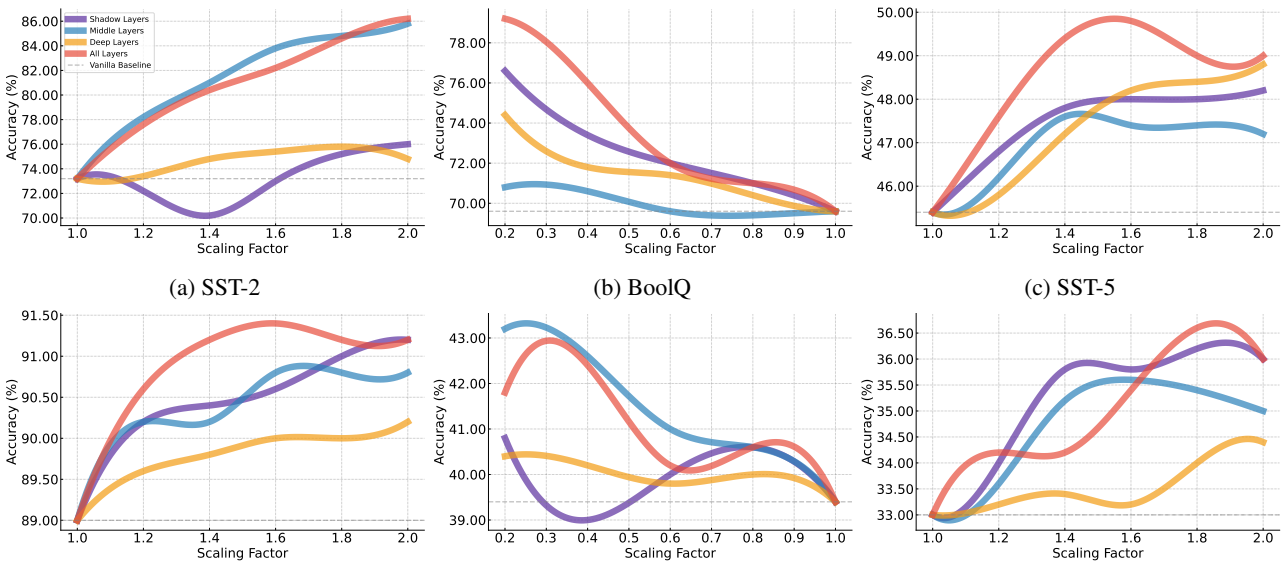
Given the special role of the initial token in LLMs, we begin by addressing two key questions: (a) Can tuning its attention improve performance across downstream tasks? (b) Is this position more effective and influential than others? To investigate, we conduct a controlled experiment in which we uniformly scale the attention scores of a single token position across all heads and layers using a scaling factor γ. We evaluate the resulting performance on three downstream tasks: SST-2, BoolQ, and LogiQA. For comparison, we repeat the same procedure for other positions, including the second, third, middle ($\lfloor T/2 \rfloor$), and final tokens. As shown in Figure 2, tuning the attention of the initial token consistently yields the largest and most stable performance gains across all tasks. Interestingly, performance varies with the direction of tuning: tasks like SST-2 benefit from up-scaling ($\gamma > 1$), while BoolQ and LogiQA improve with down-scaling ($\gamma < 1$).

Previous work has identified the initial token as an *attention sink* that helps prevent over-mixing of information during autoregressive generation (Gu et al., 2024; Barbero et al., 2025). We extend this understanding by showing that tuning the attention of the initial token can reshape the attention distribution and improve model performance across tasks. For example (see Appendix N), in the SST-2 sentiment classification task, input sentences often contain both positive and negative cues. We find that LLMs tend to overemphasize isolated negative keywords, leading to biased predictions. By increasing attention on the initial token, we flatten the attention distribution, promoting a more balanced integration of information and fostering a more holistic understanding. In contrast, in the BoolQ task, where long passages must be inspected for a small number of critical cues, the model may miss relevant tokens due to excessively diffuse attention. Reducing initial tokens' attention amplifies the relative differences between the remaining tokens, sharpening focus on salient input regions and improving response accuracy.

As discussed in Section A.1, the higher the attention on the initial token, the greater the impact of such tuning. Given that attention sinks are typically concentrated at this position, the initial token is inherently more impactful when modulated, explaining its outsized influence relative to other tokens. Besides, unlike other tokens, the initial token is task-agnostic; adjusting its attention does not alter the relative importance of task-relevant tokens, thereby avoiding harmful biases such as inadvertently amplifying attention to irrelevant content.

A.3. Layer-wise Analysis of Initial Token Scaling

To better understand the impact of tuning the initial token's attention, we examine how its effect varies when applied selectively across different layers. Following prior work on layer functionality in transformer-based models (Jin et al., 2024; Zhang et al., 2024b), we divide the 32 layers of Llama-3.1-8B-Instruct into three groups: *shallow layers* (Layers 1–10), *middle layers* (Layers 11–21), and *deep layers* (Layers 22–31). We then perform independent attention scaling experiments for each group on six tasks: BoolQ, SST-2, SST-5, MR, LogiQA, and MathQA. Based on earlier findings, we apply a scaling range of $[0, 1]$ for BoolQ and LogiQA, and $[1, 2]$ for the remaining tasks.



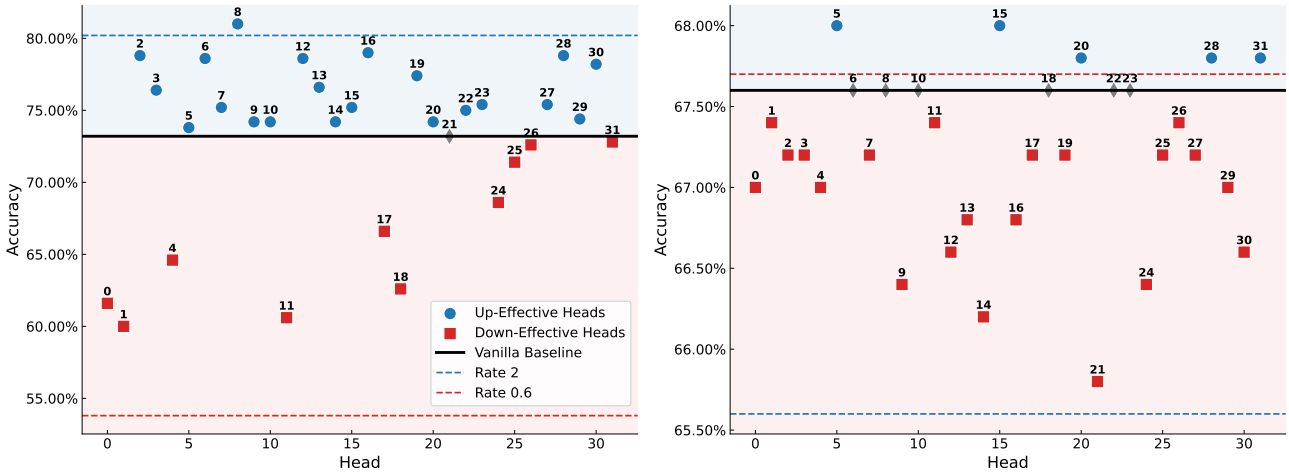
(a) SST-2 (b) BoolQ (c) SST-5

(d) MR (e) LogiQA (f) MathQA

Figure 3. Accuracy trends when scaling the initial token's attention across different layer groups: shallow (Layers 1–10), middle (Layers 11–21), and deep (Layers 22–31). Different depths exhibit a consistent accuracy trend with varying magnitudes.

As shown in Figure 3, tuning the initial token's attention yields consistent trends across all three layer groups (i.e., accuracy changes similarly with scaling regardless of depth), and jointly tuning all layers amplifies these benefits, often resulting in the highest accuracy. However, the magnitude of improvement varies. In most cases, tuning the shallow and middle layers leads to greater accuracy than tuning the deep layers. This aligns with prior findings that early and middle layers primarily contribute to representation learning and knowledge integration, whereas deep layers focus on task-specific reasoning over aggregated features (Chen et al., 2024; Jin et al., 2025).

A.4. Analyzing the Role of the Initial Token Across Attention Heads



(a) SST-2 (b) MMLU

Figure 4. Accuracy of scaling the initial token's attention in individual heads using $\gamma = 1.5$ across (a) SST-2, (b) BoolQ, (c) MMLU, and (d) MathQA. Results reveal heterogeneous behavior among heads, motivating head-specific tuning strategies.

Unlike layers, which pass information sequentially, attention heads operate in parallel and contribute independently via concatenation. It remains unclear how they differ in response to the initial token. To investigate this, we increase the initial token's attention of each head individually by applying $\gamma = 1.5$, and evaluate the model's performance on SST-2 and MMLU. For comparison, we also evaluate (i) no scaling ($\gamma = 1$), (ii) uniform up-scaling ($\gamma = 1.5$) across all heads, and (iii) uniform down-scaling ($\gamma = 0.6$) across all heads.

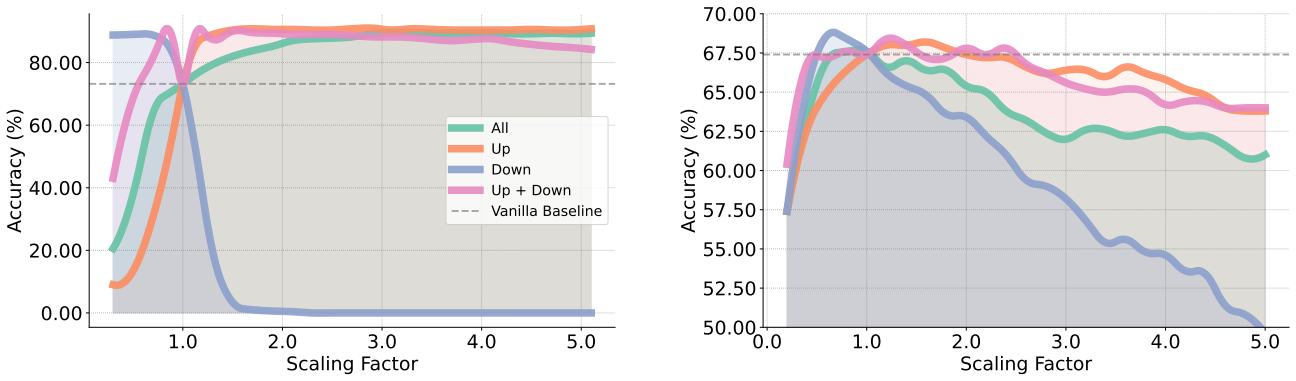
As shown in Figure 4, attention heads exhibit distinct behaviors in response to initial token amplification. We categorize heads as *up-effective* if this modification improves performance, and *down-effective* if it results in performance degradation. Interestingly, the relative proportions of up-effective and down-effective heads vary across datasets, which in turn explains the observed differences in response to uniform scaling. For example, SST-2 contains more up-effective heads and thus benefits from uniform up-scaling. In contrast, MMLU has a higher proportion of down-effective heads, making down-scaling more effective than up-scaling.

These findings align with prior work suggesting that attention heads specialize into distinct functional roles during pretraining (Zheng et al., 2024b). Such roles include global retrieval, structural parsing, option discrimination, and sensitivity to negation. We hypothesize that these functional differences drive the varying impact of initial token attention scaling. While some heads rely on broad attention to support global reasoning, others benefit from focused attention on salient tokens. Together, these specialized mechanisms enable language models to better understand and process diverse input patterns.

A.5. Evaluating Head-Specific Tuning Strategies

Given the diversity in head responses, we investigate whether head-specific tuning offers greater effectiveness than uniform tuning. Specifically, we compare four strategies: (i) uniform scaling of all heads (ALL), (ii) scaling only up-effective heads (UP), (iii) scaling only down-effective heads (DOWN), and (iv) a hybrid strategy (UP+DOWN) that scales up-effective heads to a fixed optimal value and tunes down-effective ones.

As shown in Figure 5, head-specific tuning (UP, DOWN) yields higher accuracy and faster convergence compared to uniform scaling (ALL). Notably, UP is most effective when $\gamma > 1$, while DOWN excels when $\gamma < 1$. Interestingly, the UP+DOWN strategy does not outperform UP or DOWN individually, possibly due to the concatenative nature of attention heads and suboptimal joint scaling.



(a) SST-2 (b) MMLU

Figure 5. Accuracy comparison of different tuning strategies on (a) SST-2 and (b) MMLU. Head-specific tuning (UP, DOWN) consistently outperforms uniform scaling, validating the importance of accounting for head-level behavior.

B. Related Work

B.1. Token-Level Attention Tuning

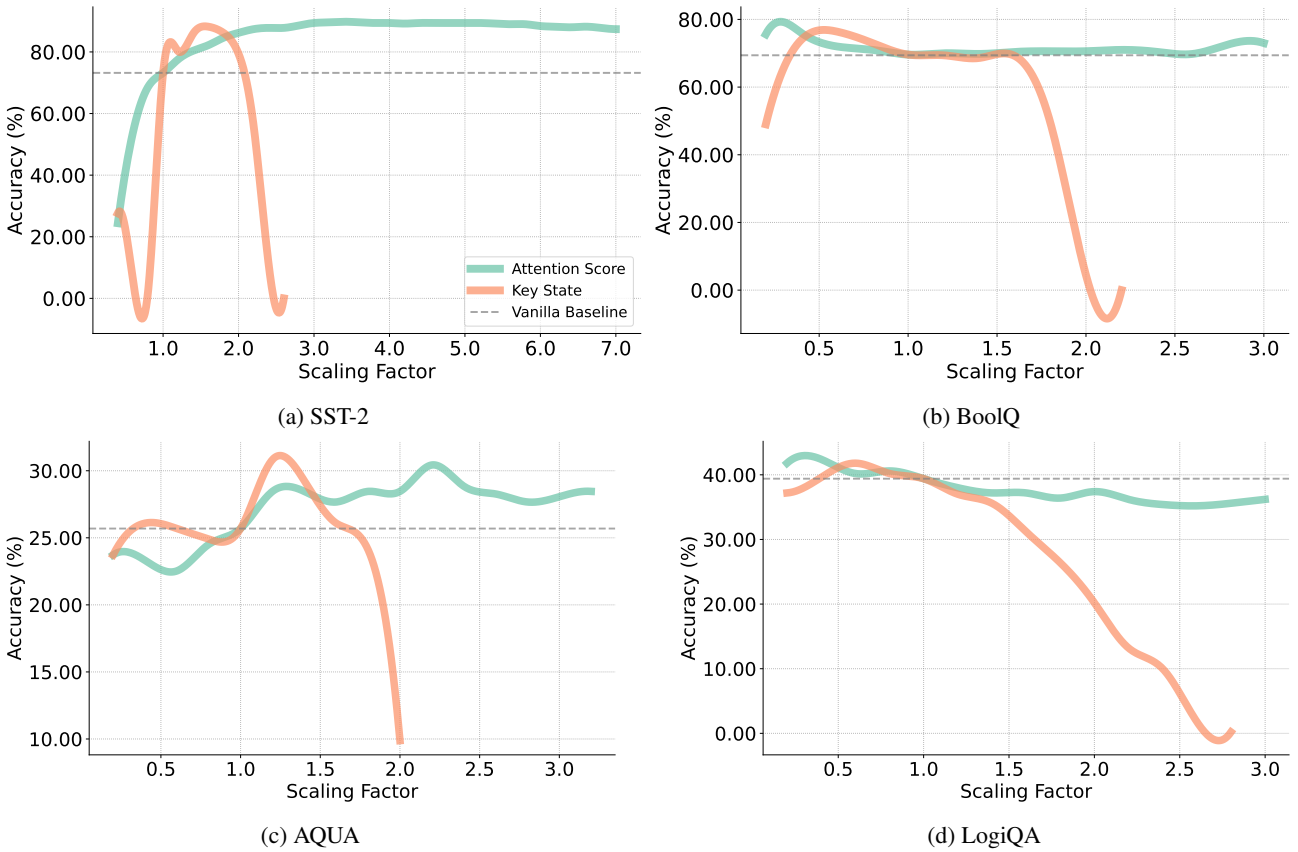
Token-level attention tuning typically aims to increase attention to critical input tokens or decrease attention to less informative tokens. (Lu et al., 2021) proposes a mask perturbation method to adjust attention weights for key tokens, thereby improving translation quality. (Zhang et al., 2023a) introduce PASTA, which allows manual designation of important tokens during inference. This is extended by AutoPASTA (Zhang et al., 2024a), which uses LLMs to autonomously identify salient tokens and increase attention to them. In contrast, ACT (Yu et al., 2024) reduces attention to semantically trivial sink tokens and redirects it to meaningful content. Similar strategies have been applied to VLMs to mitigate hallucinations. PAI (Liu et al., 2024b) enhances attention to image tokens at inference time to counteract text-dominant bias. IBD (Zhu et al., 2024) and OPERA (Wei & Zhang, 2024) further refine this idea by prioritizing visual information or penalizing overconfident summary tokens. While effective, these methods depend on identifying task-specific tokens, which may introduce bias (e.g., overemphasizing misleading tokens) and limit applicability when token importance is unclear or attention maps are unavailable. In contrast, our method focuses on a task-invariant initial token, removing the need for costly token identification, and can be easily applied by tuning key states.

B.2. The Magic of the Initial Token

Recent studies highlight the significance of the initial token, especially through the lens of the *attention sink* phenomenon, where it draws substantial attention despite low semantic content. (Xiao et al., 2023) show that preserving such tokens is critical for maintaining performance in sliding window attention. (Kaul et al., 2024) attribute this effect to softmax normalization and causal masking, while (Gu et al., 2024) and (Barbero et al., 2025) identify architectural biases that amplify attention to the initial token, including key-query alignment and LayerNorm effects. Functionally, the initial token is hypothesized to serve as a stabilizing "no-op" anchor, enhancing robustness to prompt variations (Barbero et al., 2025). It has been leveraged in applications such as long-context modeling (Zhang et al., 2023b; Xiao et al., 2023), but also poses challenges for quantization due to its high attention weight (Dettmers et al., 2023; Liu et al., 2024a). While previous work has identified the structural and functional importance of the initial token, its potential as a target for attention tuning remains underexplored. In this work, we provide a detailed analysis of attention tuning of the initial token across layers and heads, demonstrating its consistent influence across different tasks. Our approach bridges the gap between the these lines of research by proposing a novel method that advances interpretable attention tuning.

C. The Effect of Tuning Different Matrices

In certain scenarios where the attention map is not explicitly computed, it is challenging to influence the final representation by modifying the attention weights. Therefore, we consider tuning the key or query states as an alternative approach. As illustrated in the Figure 6, we observe that within an appropriate scaling range, tuning the key state exhibits a similar trend to tuning the attention score. However, we find that directly tuning the key states is more sensitive: when the scaling factor



(a) SST-2 (b) BoolQ

(c) AQUA (d) LogiQA

Figure 6. Accuracy of tuning the initial token's attention scores and key states over (a) SST-2, (b) BoolQ, (c) AQUA, and (d) LogiQA.

is too small or too large, the performance of the LLM drops sharply, while tuning the attention score results in more stable performance.

We now analyze the theoretical differences between applying the scaling factor γ to the attention scores versus the key states. To begin, we revisit and extend the attention weight formulation from Section A.1. For a sequence of length T, the attention weight for token i is given by:

$$a_i = \frac{\exp(z_i)}{\sum_{m=0}^{T-1} \exp(z_m)}, \tag{9}$$

where z_i denotes the logit for token i, given by:

$$z_i = \frac{\mathbf{q}^\top \mathbf{k}_i}{\sqrt{d_k}}, \tag{10}$$

with $\mathbf{q} \in \mathbb{R}^{d_k}$ as the query vector, $\mathbf{k}_i \in \mathbb{R}^{d_k}$ as the key vector for token i, and d_k as the dimensionality of the key vectors. Note that a_0 corresponds to the initial token, and $\sum_{i=0}^{T-1} a_i = 1$.

Tuning the Attention Score As derived in Section A.1, when tuning the attention score, the difference between the attention weights of non-initial tokens $i, j \geq 1$ becomes:

$$a'_i - a'_j = \frac{a_i - a_j}{D} = \frac{a_i - a_j}{(\gamma - 1)a_0 + 1}. \tag{11}$$

Next, we expand a_0, a_i, and a_j as follows:

$$
\begin{aligned}
a'_i - a'_j &= \frac{a_i - a_j}{(\gamma-1)a_0 + 1} \\
&= \frac{\exp(z_i) - \exp(z_j)}{\left(\sum_{k=0}^{T-1}\exp(z_k)\right)\left[(\gamma-1)\frac{\exp(z_0)}{\sum_{k=0}^{T-1}\exp(z_k)} + 1\right]} \\
&= \frac{\exp(z_i) - \exp(z_j)}{(\gamma-1)\exp(z_0) + \sum_{k=0}^{T-1}\exp(z_k)}.
\end{aligned} \tag{12}
$$

Tuning the Key State Now, consider scaling the key state by γ, i.e., $\mathbf{k}'_0 = \gamma\mathbf{k}_0$. This changes the logit for the initial token:

$$
z'_0 = \frac{\mathbf{q}^\top(\gamma\mathbf{k}_0)}{\sqrt{d_k}} = \gamma z_0, \tag{13}
$$

while the logits for other tokens remain unchanged: $z'_i = z_i$ for $i \geq 1$. The tuned attention weights are then:

$$
a'_i = \frac{\exp(z'_i)}{\sum_{j=0}^{T-1}\exp(z'_j)} = \frac{\exp(z_i)}{\exp(\gamma z_0) + \sum_{j=1}^{T-1}\exp(z_j)}, \quad \text{for } i \geq 1. \tag{14}
$$

The attention difference for non-initial tokens $i, j \geq 1$ is derived as:

$$
\begin{aligned}
a'_i - a'_j &= \frac{\exp(z_i)}{\exp(\gamma z_0) + \sum_{k=1}^{T-1}\exp(z_k)} - \frac{\exp(z_j)}{\exp(\gamma z_0) + \sum_{k=1}^{T-1}\exp(z_k)} \\
&= \frac{\exp(z_i) - \exp(z_j)}{\exp(\gamma z_0) + \sum_{k=1}^{T-1}\exp(z_k)}.
\end{aligned} \tag{15}
$$

The denominator in equation 12 includes the linear term $(\gamma-1)\exp(z_0)$ of γ, whereas the denominator in equation 15 contains the exponential component $\exp(\gamma z_0)$. This indicates that tuning the attention weights results in a smoother effect, while tuning the key states has a more abrupt impact.

D. Performance Under Resource Constraints

Computational constraints are common in real-world applications and can limit the feasibility of head classification and parameter optimization in LLMs. To investigate how well ZeroTuning adapts to such conditions, we define three resource constraint levels based on available computational resources:

- **Level 0**: Severely limited resources that prevent both head classification and parameter search.
- **Level 1**: Moderately limited resources that allow parameter search but not head classification.
- **Level 2**: Ample resources that support both head classification and parameter search.

We evaluate ZeroTuning's performance across these levels using the LLaMA-3.1-8B model. At Level 0, we apply fixed scaling factors ($\gamma = 2$ and $\gamma = 0.6$) to all attention heads, reflecting dataset-specific scaling preferences as explored in Section A.2. Additionally, we introduce a hybrid approach at Level 0, which selects the best-performing γ for each dataset. At Level 1, ZeroTuning uses uniform scaling across all heads with an optimized γ. At Level 2, it classifies attention heads, scales only the over-mixing or under-mixing heads, and searches for the optimal γ.

As shown in Table 4, ZeroTuning consistently improves performance across all resource levels. Even at Level 0, where resources are tightly constrained, the hybrid approach delivers steady gains over vanilla inference. These improvements become more substantial at Levels 1 and 2, where additional resources enable parameter optimization and head classification. Specifically, compared to the vanilla baseline, ZeroTuning increases average accuracy on text classification tasks by 3.63 percentage points at Level 0 (Hybrid), 4.86 percentage points at Level 1, and 11.71 percentage points at Level 2. For multiple-choice tasks, the corresponding gains are 0.37, 1.65, and 2.64 percentage points, respectively.

Table 4. Performance of ZeroTuning Under Different Resource Constraints

Method	Classification							Avg. Class.	Multiple Choice							Avg. MC
	SST-2	SST-5	MR	BoolQ	CB	TREC	SUBJ		MMLU	AQUA	MathQA	LogiQA	CQA	PIQA	ARCC	
Vanilla	73.20	45.40	89.20	69.60	82.14	14.00	44.60	59.73	67.40	25.69	33.60	39.40	77.60	83.60	84.62	58.84
Level 0 ($\gamma = 2$)	86.20	49.20	91.00	70.06	82.41	12.00	44.80	62.24	65.60	28.46	34.40	37.40	78.20	82.40	83.61	58.58
Level 0 ($\gamma = 0.6$)	53.80	43.40	82.40	72.00	83.93	17.20	44.60	56.76	67.00	22.53	32.60	40.20	77.60	82.60	83.61	58.02
Level 0 (Hybrid)	86.20	49.20	91.00	72.00	83.93	17.20	44.80	63.36	67.00	28.46	34.40	40.20	78.20	82.60	83.61	59.21
Level 1	89.60	49.00	91.40	71.20	83.93	21.80	45.20	64.59	68.00	30.04	35.00	42.80	79.20	83.80	84.62	60.49
Level 2	91.60	52.00	92.00	82.40	89.29	26.20	66.60	71.44	68.80	30.43	36.60	42.80	80.40	85.40	85.95	61.48

Table 5. Impact of Context Length on ZeroTuning Performance

Dataset	Method	Extra Context Length				Average
		0	100	200	300	
SST-2	Vanilla	73.20	68.40	59.20	32.00	58.20
	ZeroTuning	**91.60**	**89.20**	**87.40**	**85.40**	**88.40**
	Diff	18.40	20.80	28.20	53.40	30.20
BoolQ	Vanilla	69.60	68.60	67.60	68.60	68.60
	ZeroTuning	**82.40**	**81.80**	**81.40**	**81.20**	**81.70**
	Diff	12.80	13.20	13.80	12.60	13.10
LogiQA	Vanilla	39.40	36.60	36.20	35.80	37.00
	ZeroTuning	**42.40**	**43.00**	**41.00**	**41.00**	**41.85**
	Diff	3.00	6.40	4.80	5.20	4.85
PIQA	Vanilla	83.60	82.20	81.20	80.60	81.90
	ZeroTuning	**85.40**	**83.80**	**83.20**	**82.80**	**83.80**
	Diff	1.80	1.60	2.00	2.20	1.90

E. Sensitivity to Different Context Lengths

To investigate how the distance between the initial token and task-relevant tokens affects model behavior, we evaluate the sensitivity of ZeroTuning under varying context lengths. Specifically, we insert task-irrelevant tokens between the initial token and the original input to artificially extend the context. This allows us to isolate the impact of increased token distance on attention and performance.

We conduct experiments using Llama-3.1-8B-Instruct and apply ZeroTuning with the same set of heads and scaling factors used in the previous base (non-extended) context setting. This ensures that any performance change is due solely to increased context length rather than re-optimized tuning parameters.

As shown in Table 5, the performance of vanilla LLMs consistently degrades as context length increases, likely due to disrupted information mixing caused by the inserted tokens. In contrast, ZeroTuning remains robust across all tested context lengths, yielding consistent and often significant improvements even under extreme cases of context extension. These results suggest that tuning the initial token's attention can effectively stabilize information flow, even when relevant content is pushed further away in the input sequence.

F. Robustness Across Few-Shot Scenarios

Few-shot learning has become a widely adopted approach to improve the performance of LLMs by providing a small number of in-context examples, enabling adaptation to specific tasks with minimal data (Brown et al., 2020). Building on previous zero-shot evaluations, we now evaluate the robustness of ZeroTuning in 1-shot and 2-shot scenarios across four datasets: SST-5, BoolQ, MMLU, and AQUA. To ensure consistency, we fix the randomly selected examples, maintain the selected head and scaling factor throughout the experiments.

The results in Table 6 show that ZeroTuning consistently outperforms the vanilla baseline across both 1-shot and 2-shot settings. In the 1-shot scenario, ZeroTuning achieves an average accuracy improvement of 1.85% over the vanilla model, with notable gains of 2.0% on BoolQ (82.40% vs. 80.40%) and 1.8% on SST-5 (49.40% vs. 47.60%). In the 2-shot scenario, the average improvement increases to 3.08%, with a significant 7.12% increase on AQUA (32.81% vs. 25.69%) and 2.0% on

Table 6. Comparison of Vanilla and ZeroTuning Performance Across Few-Shot Learning Scenarios

Shot	Method	SST-5	BoolQ	MMLU	AQUA	Average
0-Shot	Vanilla	45.4	69.6	67.4	25.7	52.0
	ZeroTuning	**52.0**	**82.4**	**68.80**	**30.4**	**58.40**
	Diff	6.6	12.8	1.4	4.7	6.4
1-Shot	Vanilla	47.6	80.4	61.8	28.1	54.5
	ZeroTuning	**49.4**	**82.4**	**63.4**	**30.0**	**56.3**
	Diff	1.8	2.0	1.6	1.9	1.8
2-Shot	Vanilla	50.4	83.4	64.4	25.7	56.0
	ZeroTuning	**52.4**	**85.0**	**66.0**	**32.8**	**59.1**
	Diff	2.0	1.6	1.6	7.1	3.1

Table 7. Performance Comparison Across Decoding Strategies with and without ZeroTuning on MMLU and SST-2.

Dataset	**Method**	**Top-k Sampling**	**Top-p Sampling**	**Beam Search**	**Average**
MMLU	Vanilla	63.80	63.80	63.00	63.53
	ZeroTuning	**65.80**	**66.00**	**65.20**	**65.67**
	Diff	2.00	2.20	2.20	2.13
SST-2	Vanilla	64.40	66.60	66.60	65.87
	ZeroTuning	**89.20**	**89.60**	**89.20**	**89.33**
	Diff	24.80	23.00	22.60	23.47

SST-5 (52.40% vs. 50.40%). Notably, ZeroTuning in the zero-shot setting outperforms vanilla few-shot baselines, achieving higher accuracy without the additional context overhead and contextual biases introduced by in-context examples.

Our results also highlight the following key findings:

1. ZeroTuning improves LLM performance, even when few-shot learning does not benefit the base model. Most datasets show improvements with few-shot learning, likely due to clearer patterns and better output formatting. However, some datasets, like MMLU, experience performance drops, possibly due to increased confusion from the examples. Despite this, ZeroTuning still leads to consistent performance gains.

2. Similar to Few-Shot Learning, ZeroTuning reduces invalid responses from LLMs, indicating improved instruction following. For instance, in the SST-2 dataset, LLMs sometimes output incorrect responses like "neutral" in zero-shot settings when they should respond with "positive" or "negative". Few-shot learning helps the model understand the expected format, improving accuracy. Interestingly, ZeroTuning also reduces these errors, suggesting that it helps the model better understand task-relevant information.

G. Impact of Decoding Strategies

Decoding strategies play a crucial role in shaping the output behavior of LLMs, and can influence performance across tasks. We evaluate the robustness of ZeroTuning over three strategies: Top-k Sampling, Top-p Sampling, and Beam Search, using Llama-3.1-8B on MMLU and SST-2, with results in Table 7.

Across all decoding strategies, ZeroTuning consistently improves over the vanilla baseline. On MMLU, it yields performance gains of 2.0% with Top-k, 2.2% with Top-p, and 2.2% with Beam Search, resulting in an average improvement of 2.1%. On SST-2, the improvements are even more substantial: 24.8% with Top-k, 23.0% with Top-p, and 22.6% with Beam Search, with an average gain of 23.5%.

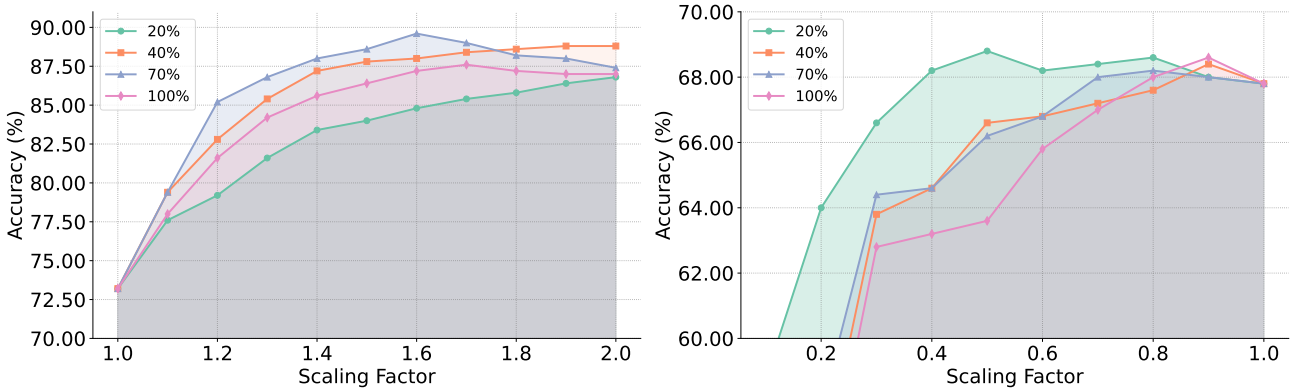
H. The Effect of Different Numbers of Heads

As shown in Figure 8, we observe that tuning an appropriate proportion of attention heads leads to the best performance. Specifically, Figure 8a presents results on the SST-2 dataset, where we tune the up-effective heads, while Figure 8b reports performance on the MMLU dataset with the down-effective heads. Across both datasets, we find that tuning a moderate

Table 8. Effect of Prompt Variations on Performance with and without ZEROTUNING.

Prompt Format	Method	MMLU	SST-2	Average
Full Prompt	Vanilla	67.40	64.40	65.90
	ZeroTuning	**68.80**	**89.20**	**79.00**
	Diff	1.4	24.8	13.1
Drop Instruction1	Vanilla	66.80	64.40	65.60
	ZeroTuning	**68.00**	**89.20**	**78.60**
	Diff	1.2	24.8	13.0
Modify Instruction2	Vanilla	61.80	61.80	61.80
	ZeroTuning	**64.20**	**88.00**	**76.10**
	Diff	2.4	26.2	14.3

proportion of heads (typically between 40% and 70%) achieves the highest accuracy. In contrast, tuning too few or too many heads tends to degrade performance, suggesting that selective head tuning is key to effective inference-time adaptation.



(a) SST-2 (b) MMLU

Figure 7. Accuracy of tuning different proportions of heads. (a) SST-2: tuning up-effective heads; (b) MMLU: tuning down-effective heads.

I. Sensitivity to Prompt Variations

Prompts play a crucial role in guiding LLM behavior and typically consist of three components: **Instruction1** (task guidance), **Question** (the actual query), and **Instruction2** (output format specification). To evaluate the robustness of ZeroTuning under prompt perturbations, we perform experiments on the LLaMA-3.1-8B model using MMLU and SST-2 under three prompt formats: Full Prompt (Instruction1 + Question + Choices + Instruction2), Drop Instruction1, and Modify Instruction2. Detailed prompt examples are provided in Appendix L.

As shown in Table 8, ZeroTuning consistently improves performance across all prompt configurations, and maintains strong performance even when key instructions are modified or omitted, demonstrating its distinctive ability to regulate and adapt to prompt variations. On MMLU, the performance gains range from 1.2% to 2.4%, with an average improvement of 1.7%. On SST-2, the gains are more substantial, ranging from 24.8% to 26.2%, with an average improvement of 25.3%.

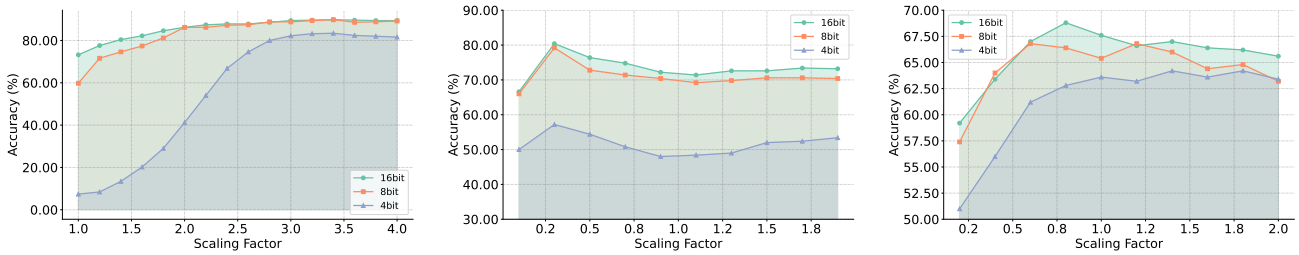
J. The Effect of Different Quantization Configurations

As shown in Figure 8, we observe:

(a) Quantizing to 8-bit results in only a slight accuracy decrease compared to 16-bit, while 4-bit quantization leads to a significant accuracy decrease. However, by appropriately tuning attention to the initial token, we find that the best accuracy with 8-bit quantization becomes comparable to that of the 16-bit model on the SST-2 and BoolQ datasets. This suggests that our method can partially compensate for the performance loss caused by quantization.

(c) The accuracy trends across different quantization levels are largely similar. This consistency may offer useful insights

for future work, for instance, searching for optimal parameters using low-precision models and transferring them to higher-precision models.



(a) SST-2 (b) BoolQ (c) MMLU

Figure 8. Accuracy when tuning under different quantization configurations.

K. Limitations

Our current approach still relies on access to the ground-truth signals and accuracy to guide the tuning direction. As a result, it cannot yet operate in a fully unsupervised manner based solely on the properties of the input or the model's attention distribution. Although we explored the effectiveness of the method under resource-constrained scenarios, such as using fixed scaling factors or skipping exhaustive traversal and classification of attention heads, these simplified variants consistently underperformed compared to the full method.

Additionally, our experiments follow standard evaluation protocols used in prior work (Ouyang et al., 2022; Sanh et al., 2021; Hao et al., 2022; Yu et al., 2024), and due to computational and inference resource constraints, we do not evaluate performance on longer-context generation tasks involving complex reasoning or multi-step thinking.

Future work could investigate the differences in attention distributions across layers and heads more deeply, which may lead to improved unsupervised tuning strategies. In addition, it may be fruitful to explore other tuning schemes, such as dynamically adjusting attention at critical positions in the long generated sequence.

L. Prompts Used for Each Dataset

Here, we list all the prompts we used in this paper on different datasets:

For multiple choice task, we use the following prompt:

Prompt for Multiple-Choice Tasks

Generate the correct answer to the following question.
<Question>
<choice 1>
<choice 2>
<choice 3>
...
Answer:"

For text classification, we use different prompts for different datasets.

Prompt for SST-2

"Classify the sentiment of the user's message into one of the following categories: 'positive' or 'negative'.
Sentence: <sentence>
Sentiment: "

Prompt for SST-5

"Classify the sentiment of the user's message into one of the following categories: 'terrible', 'negative', 'neutral', 'positive', or 'great'.
Sentence: <sentence>
Sentiment: "

Prompt for MR

"Classify the sentiment of the movie's review into one of the following categories: 'positive' or 'negative'.
Review: <sentence>
Sentiment: "

Prompt for TREC

"Classify the given questions into the following categories: 'Description', 'Entity', 'Expression', 'Person', 'Number', or 'Location'.
Question: <sentence>
Type: "

Prompt for CB

"Read the following paragraph and determine if the hypothesis is true.
Premise: <premise> Hypothesis: <hypothesis>.
Answer: "

Prompt for BoolQ

"Read the text and answer the question by True or False.
Text: <passage> Question: <question>?
Answer: "

Prompt for SUBJ

"Classify the input into one of the following categories: subjective or objective.
Input: <text>
Category: "

M. Prompt for key tokens identification

Prompt for key tokens identification

Below is a question. Please extract the key content words or phrases from the question that are crucial for understanding and answering it correctly. These are typically the nouns, verbs, adjectives, or multi-word expressions that define the subject, action, or relation in the question. Output your selection as a Python list, where each element is a word or a phrase enclosed in quotes.
For example, for the question 'What is the boiling point of water?', the key words might be ['boiling point', 'water'].
Question: {question}
Key Words:

N. LLM Output Example

N.1. Examples of SST-2 Dataset

Example 1:

Classify the sentence into one of the following sentiments: positive or negative.
Sentence: "hardly a masterpiece, but it introduces viewers to a good charitable enterprise and some interesting real people."
Sentiment:
LLM Output: negative
Ground Truth: positive

Example 2:

Classify the sentence into one of the following sentiments: positive or negative.
Sentence: "generally, clockstoppers will fulfill your wildest fantasies about being a different kind of time traveler, while happily killing 94 minutes."
Sentiment:
LLM Output: negative
Ground Truth: positive

Example 3:

Classify the sentence into one of the following sentiments: positive or negative.
Sentence: "whether you like rap music or loathe it, you can't deny either the tragic loss of two young men in the prime of their talent or the power of this movie."
Sentiment:

LLM Output: neutral
Ground Truth: positive

Example 4:

Classify the sentence into one of the following sentiments: positive or negative.
Sentence: "generally, clockstoppers will fulfill your wildest fantasies about being a different kind of time traveler, while happily killing 94 minutes."
Answer:

LLM Output: The
Ground Truth: positive

N.2. Examples of BoolQ Dataset

Example 1:

Read the text and answer the question by True or False.
Text: Countdown (game show) – The contestant in control chooses six of 24 shuffled face-down number tiles, arranged into two groups: 20 "small numbers" (two each of 1 through 10), and four "large numbers" of 25, 50, 75 and 100. Some special episodes replace the large numbers with 12, 37, 62 and 87. The contestant decides how many large numbers are to be used, from none to all four, after which the six tiles are randomly drawn and

placed on the board. A random three-digit target number is then generated by an electronic machine, affectionately known as "CECIL" (which stands for Countdown's Electronic Calculator In Leeds). The contestants have 30 seconds to work out a sequence of calculations with the numbers whose final result is as close to the target number as possible. They may use only the four basic operations of addition, subtraction, multiplication and division, and do not have to use all six numbers. A number may not be used more times than it appears on the board. Fractions are not allowed, and only positive integers may be obtained as a result at any stage of the calculation. As in the letters rounds, any contestant who does not write down their calculations in time must go first, and both contestants must show their work to each other if their results and calculations are identical.
Question: do you have to use all the numbers on countdown?
Answer:
LLM Output: True
Ground Truth: False

Example 2:

Read the text and answer the question by True or False.
Text: Hawaii Five-0 (2010 TV series, season 8) – The eighth season of the CBS police procedural drama series Hawaii Five-0 premiered on September 29, 2017 for the 2017–18 television season. CBS renewed the series for a 23 episode eighth season on March 23, 2017. However, on November 6, 2017 CBS ordered an additional episode for the season and did the same again on February 8, 2018 bringing the count to 25 episodes. The season concluded on May 18, 2018. The eighth season ranked #18 for the 2017-18 television season and had an average of 11 million viewers. The series was also renewed for a ninth season.
Question: will hawaii five o have a season 8?
Answer:
LLM Output: False
Ground Truth: True