From Interpretability to Performance: Optimizing Retrieval Heads for Long-Context Language Models

Youmi Ma **Naoaki Okazaki**
Department of Computer Science, Institute of Science Tokyo
{ma.y, okazaki}@comp.isct.ac.jp

Abstract

Advances in mechanistic interpretability have identified special attention heads, known as retrieval heads, that are responsible for retrieving information from the context. However, the role of these retrieval heads in improving model performance remains unexplored. This work investigates whether retrieval heads can be leveraged to enhance the long-context capabilities of LLMs. Specifically, we propose RetMask, a method that generates training signals by contrasting normal model outputs with those from an ablated variant in which the retrieval heads are masked. This mechanism-based approach achieves substantial improvements: +2.28 points on HELMET at 128K for Llama-3.1, with +70% gains on generation with citation and +32% on passage re-ranking, while preserving performance on general tasks. Experiments across four models in three families demonstrate that RetMask consistently improves long-context performance, where gains correlate with the sparsity of the retrieval score distribution: models with sparser distributions, where retrieval capabilities are concentrated in a small set of heads, respond more strongly, while those with less sparse distributions show more modest gains. These results validate the functional role of retrieval heads and show that mechanistic insights can be transformed into performance enhancements[1].

1 Introduction

Large Language Models (LLMs) require long-context capabilities to realize multi-document understanding (Bai et al., 2024b), in-context learning (Brown et al., 2020), and test-time scaling (Snell et al., 2024; OpenAI, 2024). Recent studies on mechanistic interpretability revealed that long-context factuality is closely related to a set of attention heads named *retrieval heads* (Wu et al., 2025b; Zhang et al., 2025b). Retrieval heads attend to previous tokens and recall information during the generation process. Deactivating retrieval heads has been reported to result in performance drops for downstream tasks.

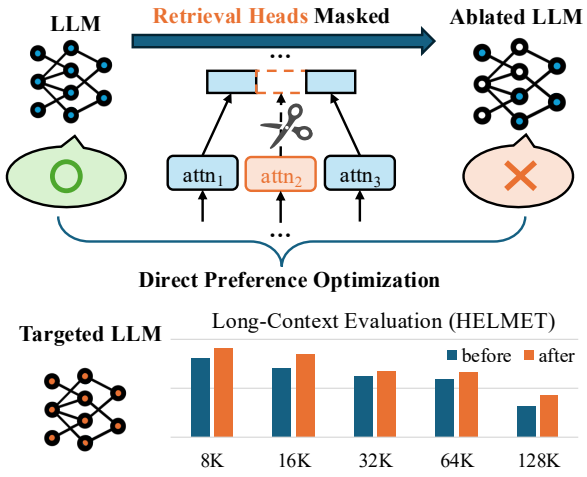


Figure 1: An overview of RetMask. Given an LLM, we construct an ablated variant by masking its retrieval heads (Wu et al., 2025b), then sample responses from both the original and ablated models to construct training pairs for Direct Preference Optimization. We validate this approach on four models across three families.

While retrieval heads provide hints for the mechanism of long-context capabilities, their contributions to model performance remain unexplored. This gap between interpretability and model performance is pervasive: despite identifying specialized components responsible for knowledge storage (Dai et al., 2022; Meng et al., 2022) and language (Tang et al., 2024; Kojima et al., 2024), prior work has not established effective methods to transform these discoveries into performance enhancements. Gu et al. (2024) reports that editing knowledge-specific components brings unintended side effects on models' general abilities, and Mondal et al. (2025) reports that language-specific neuron interventions are insufficient to provide perfor-

[1]The source code is available at: https://github.com/YoumiMa/RetMask.

Findings of the Association for Computational Linguistics: ACL 2026, pages 27731–27747

mance gains on downstream tasks. This leads to a natural question: Can retrieval heads be leveraged to enhance long-context capabilities?

With this research question in mind, this paper explores a method to enhance long-context processing abilities by optimizing retrieval heads. Specifically, as shown in Figure 1, we synthesize supervision data from both the original model and its ablated variant in which the retrieval heads are masked. We name the method as RetMask, short for **Ret**rieval-Head **Mask**ing. RetMask applies Direct Preference Optimization (DPO, Rafailov et al., 2023) to post-trained models to prefer responses generated by the original model over those generated by its retrieval-head-ablated counterpart. Evaluations on HELMET (Yen et al., 2025) across three model families, namely Llama-3.1, Qwen3, and Olmo-3 (both Instruct and Think variants), show consistent improvements, with the magnitude of gains varying across models. We find that these differences correlate with the sparsity of the retrieval score distribution: Models with sparser distributions respond strongly to the method, while those with less sparse distributions show relatively modest gains. This finding reconfirms the functional importance of retrieval heads in long-context processing from a model development perspective.

The contributions of this work are as follows. (1) We propose RetMask, a simple and effective method for improving long-context processing that leverages retrieval heads as a source of contrastive training signals, requiring neither human-crafted criteria nor an LLM judge. (2) We validate the effectiveness of RetMask on four models across three families, where consistent improvements in long-context performance are observed. We further show that the magnitude of gains correlates with the sparsity of the retrieval score distribution, providing mechanistic insight into when and why the method is effective. (3) RetMask improves long-context capabilities without degrading general abilities: the trained models maintain performance on mathematics, coding, and general knowledge tasks.

2 Preliminary: Retrieval Heads

Prior study has uncovered retrieval heads, a set of attention heads that retrieve relevant information from previous contexts during generation (Wu et al., 2025b). The algorithm to detect retrieval heads roots from the *Needle-In-A-Haystack* (NIAH) task.

Needle-In-A-Haystack (Kamradt, 2023). For each question q and its corresponding answer k (the "needle"), the answer k is randomly inserted into a context $x = p_1, \ldots, p_n$ composed of n passages that are irrelevant to both q and k (the "haystack"). This yields $x' = p_1 \ldots k \ldots p_n$, where the indices of inserted needle tokens are denoted as $\mathcal{I}_k$. A language model receives the context with the answer inserted x', along with the question q, and is evaluated on whether it correctly outputs k. If successful, the model retrieves the target answer span k from the long context x' by performing a copy-paste operation.

Retrieval Head. To detect retrieval heads, Wu et al. (2025b) calculates the frequency of an attention head performing copy-paste operations. Specifically, during decoding, let y_t denote the current token to be generated, and $\boldsymbol{a}_t \in \mathbb{R}^{|x'|+t-1}$ is the attention scores of a head. The head is considered to be retrieving (i.e., copy-pasting) the token x_j if $y_t = x_j, j = \arg\max(\boldsymbol{a}_t)$. If $j \in \mathcal{I}_k$, the head is retrieving a token from the needle. The retrieval score of head h is thus defined as:

$$\text{RetrievalScore}(h) = \frac{1}{|\mathcal{T}|} \sum_{(g_h,k)\in\mathcal{T}} \frac{|g_h \cap k|}{|k|}, \quad (1)$$

where $\mathcal{T}$ is the set of test instances; in each test, g_h denotes the set of all tokens retrieved by the head h, and k denotes the needle sequence. This metric thus quantifies the overlap between tokens retrieved by the head h and those in the needle sequence. The scores of all attention heads are computed, and those heads with $\text{RetrievalScore}(h) \geq \tau$ are considered as retrieval heads, where τ is a threshold hyper-parameter.

3 Methodology: RetMask

This work evaluates the effectiveness of retrieval heads in enhancing the long-context processing capabilities of LLMs (Figure 2). Given an LLM π_θ, our approach trains the model to prefer outputs sampled from the LLM π_θ over those from an ablated variant $\pi_{\theta'}$ (with retrieval heads masked). The method consists of three stages: (1) Retrieval Head Deactivation; (2) Contrastive Response Generation; (3) Direct Preference Optimization.

Retrieval Head Deactivation. Following Wu et al. (2025b), for a given LLM π_θ, we compute the retrieval score of all attention heads and detect retrieval heads on top of the NIAH task. Attention

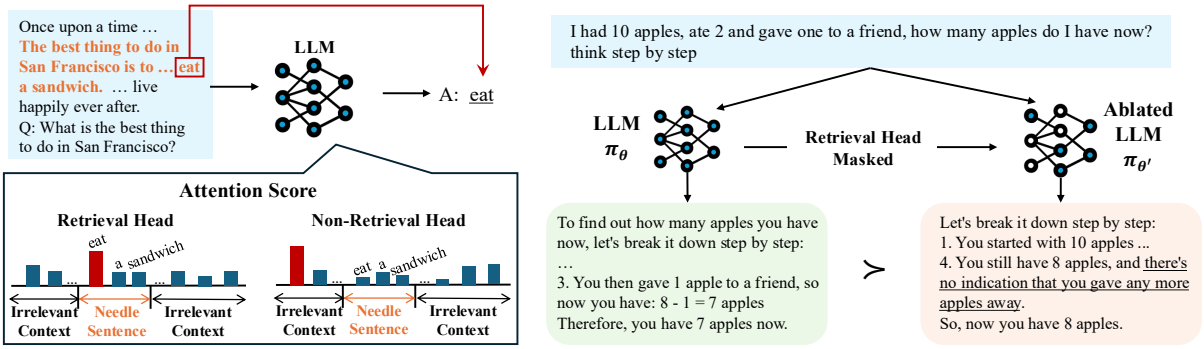


Figure 2: Overview of Preliminaries (left) and RetMask (right). The example on the right is a real case extracted from the training data. We detect and mask retrieval heads for generating contrastive responses.

heads with score greater than τ comprise the retrieval head set $\mathcal{H}_{\text{ret}}$. We then construct an ablated LLM $\pi_{\theta'}$ by deactivating the identified retrieval heads $\mathcal{H}_{\text{ret}}$. For each head $h \in \mathcal{H}_{\text{ret}}$, we zero out the corresponding columns in the attention output projection matrix $\boldsymbol{W}_o$, thereby preventing the head from contributing to subsequent layers.

$$\boldsymbol{W}_{o'}^{h} = \begin{cases} \mathbf{0} & \text{if } h \in \mathcal{H}_{\text{ret}} \\ \boldsymbol{W}_{o}^{h} & \text{otherwise} \end{cases} \tag{2}$$

Contrastive Response Generation. We synthesize data for direct preference optimization using the model π_θ and its ablated variant $\pi_{\theta'}$, shaping contrasts that highlight the contribution of retrieval heads. To this end, we utilize existing instruction-tuning datasets, which consist of instruction-response pairs. For each instruction x in the dataset, we discard the original response and generate new responses using π_θ and $\pi_{\theta'}$:

$$y_w \sim \pi_\theta(\cdot|x), \tag{3}$$

$$y_l \sim \pi_{\theta'}(\cdot|x), \tag{4}$$

where Equation 3 represents sampling from the original LLM, and Equation 4 represents sampling from the ablated variant. The response y_w generated under zero perturbation serves as the chosen response, while y_l generated with retrieval heads deactivated serves as the rejected response. We provide examples about typical failure modes of the rejected responses in Appendix H.

Direct Preference Optimization. Combining the synthesized responses y_w, y_l with the original instruction x, we obtain preference tuples $\{(x, y_w, y_l)\}$. We train the target policy π_θ using Direct Preference Optimization (DPO, Rafailov et al., 2023), with the reference policy π_{ref} initialized from the original model. The objective is:

$$\mathcal{L}(\pi_\theta) = -\mathbb{E}\left[\log \sigma\left(\beta \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\right)\right], \tag{5}$$

where β is a temperature parameter controlling the deviation from the reference policy. RetMask uses self-synthesis, i.e., the model used for response generation is the same as the target model, by default; cross-model synthesis results are in Appendix E.

4 Experiments

4.1 Settings

Models. We evaluate RetMask on three model families: Llama-3.1-8B-Instruct (Grattafiori et al., 2024), Qwen3-8B (Yang et al., 2025), and Olmo-3-7B-(Instruct/Think) (Olmo et al., 2025). We identify retrieval heads using threshold $\tau = 0.1$ for Llama-3.1 and $\tau = 0.05$ for Qwen3 and Olmo-3 [2]. For Qwen3, we disable reasoning when computing the retrieval scores to match the default setting and enable reasoning during contrastive response generation unless otherwise stated.

Benchmarks. We evaluate on HELMET (Yen et al., 2025), a comprehensive benchmark for long-context processing that covers both synthetic and real-world tasks, categorized as Synthetic Recall (*Recall*, Hsieh et al., 2024), Retrieved Augmented Generation (*RAG*), Generation with Citations (*Cite*), Passage Re-Ranking (*Re-rank*), Many-Shot In-Context Learning (*ICL*), Long-Document Question Answering (*LongQA*), and Summarization (*Summ*). The benchmark covers five context lengths ranging from 8K to 128K tokens. For Qwen3-8B, we enable reasoning during evaluation.

[2]We tuned the hyper-parameter in pilot experiments as explained in Appendix A and B.

DPO Strategy	Llama-3.1-8B-Instruct 8K	16K	32K	64K	128K	Qwen3-8B 8K	16K	32K	64K	128K
–	56.03	54.14	52.42	51.65	46.40	53.20	50.16	49.89	45.44	44.73
Smaller-Model	56.77	55.32	**53.48**	52.18	47.53	52.52	49.81	48.71	46.67	45.51
Win-Lose-Pair	56.50	54.42	52.47	51.62	46.05	52.80	50.14	49.71	45.93	44.49
Non-Retrieval-Mask	56.45	55.55	53.19	52.14	47.19	53.02	50.28	48.67	**46.79**	45.48
Random-Mask	56.67	55.95	53.14	52.30	47.04	49.99	47.02	45.76	43.85	**45.86**
RetMask	**58.14**	**56.92**	**53.48**	**53.15**	**48.68**	**53.77**	**50.61**	**50.34**	**46.79**	45.62

Table 1: Performance of Llama-3.1 and Qwen3 trained with different strategies on HELMET (Yen et al., 2025). Models are evaluated using input sequences of 8K, 16K, 32K, 64K, and 128K tokens. Overall, training with retrieval heads ablated (i.e., RetMask) yields the best performance.

DPO Strategy	Llama-3.1-8B-Instruct Average	Recall	RAG	Cite	Re-rank	ICL	LongQA	Summ
–	46.40	95.13	58.58	3.09	13.73	83.80	42.69	27.81
Smaller-Model	47.53	94.19	**60.83**	4.22	13.44	83.76	43.15	33.12
Win-Lose-Pair	46.05	93.56	59.50	3.72	12.47	83.36	39.26	30.48
Non-Retrieval-Mask	47.19	**96.69**	59.00	3.45	11.38	84.28	40.93	**34.62**
Random-Mask	47.04	96.38	59.29	3.88	10.79	83.52	41.32	34.10
RetMask	**48.68**	95.44	59.71	**5.25**	**18.16**	**84.92**	**43.84**	33.45

Table 2: Model performance on each task of HELMET when the input sequence length is 128K. The advantage of RetMask is evident on real-world tasks such as generation with citation and passage re-ranking.

Retrieval-Head Detection. We adopt the script provided by Wu et al. (2025b) that runs NIAH on 20 different context lengths uniformly distributed between 0 and 5K, with the needle inserted at 10 depth positions for each length[3]. This setting is recommended by the authors, who note that the detection stabilizes with only a few samples.

Training Data. RetMask is applicable to any dataset containing user instructions. We primarily use LMSYS-Chat-1M (Zheng et al., 2024), a large-scale collection of human-LLM conversations. We also experiment with WildChat (Zhao et al., 2024), another general-purpose dataset collected from human-LLM interactions, in § 4.6, and Guru (Cheng et al., 2025), a reinforcement learning dataset, in Appendix F. All training data, with statistics shown in Appendix G, are distinct from the evaluation benchmark, ensuring that performance gains reflect improvements in long-context capability rather than task-specific tuning.

Baselines. To focus on the contribution of retrieval heads, we include baselines with different policies of deciding rejected samples y_l: (1) **Smaller-Model**: y_l sampled from a smaller LLM, namely Llama-3.2-3B-Instruct (Grattafiori et al., 2024)[4] for experiments on Llama-3.1-8B-Instruct and Olmo-3-7B-Instruct, and Qwen3-0.7B for experiments on Qwen3-8B and Olmo-3-7B-Think. (2) **Win-Lose-Pair**: y_l sampled from the same LLM but with lower quality. The quality is judged by Gemma-3-27B-IT (Team et al., 2025). (3) **Non-Retrieval-Mask**: y_l sampled from another ablated variant of the LLM, with $|\mathcal{H}_{\text{ret}}|$ randomly-selected non-retrieval heads masked. The masked heads are not chosen from the retrieval heads ($h \notin \mathcal{H}_{\text{ret}}$). (4) **Random-Mask**: y_l sampled from another ablated variant of the LLM, with $|\mathcal{H}_{\text{ret}}|$ heads randomly masked. Masked heads can be the retrieval heads. As a strong baseline, we also compare with existing work (Zhang et al., 2025a) in Section 4.3.

[3] https://github.com/nightdessert/Retrieval_Head

[4] We experimented with Llama-3.2-1B-Instruct and found the training unstable, thus switched to Llama-3.2-3B-Instruct.

4.2 Main Results

The performance of Llama-3.1-8B-Instruct and Qwen3-8B trained under different strategies is shown in Table 1. Additionally, Table 2 presents per-task performance of Llama-3.1 on HELMET evaluated using input sequences of 128K tokens. The task-wise performance of Qwen3-8B is detailed in Appendix C. In all tables throughout this paper, the row labeled as '–' denotes the baseline model before training.

Strong Improvements on Llama-3.1 across all context lengths. Table 1 shows that RetMask achieves the best performance across all context lengths when training Llama-3.1. At 128K, the proposed method improves the base model by 2.28 points (46.40 → 48.68). The improvement persists

across context lengths ranging from 8K to 128K tokens, demonstrating the robustness of the method. RetMask outperforms the other baselines (Non-Retrieval-Mask and Random-Mask), confirming that improvements stem specifically from targeting retrieval heads rather than the ablating operation itself. Notably, Win-Lose-Pair, which trains the model to prefer higher-quality outputs over lower-quality ones from the same model, shows decreased performance ($46.40 \rightarrow 46.05$). This indicates that the gains from the proposed method are not simply due to preference optimization on output quality, but rather from the contrast that specifically targets retrieval capabilities. We also verified that supervised fine-tuning with chosen responses yields suboptimal performance, as detailed in Appendix D.

Interestingly, the training sequences average only 63.62 tokens for inputs and 494.69 tokens for outputs, significantly shorter than the evaluation contexts. This reveals an advantage of the proposed method: it enhances long-context capabilities through short-sequence training, consistent with findings in Gao et al. (2025) that post-training with short-context instruction datasets is sufficient for achieving good long-context performance.

Improvements on Qwen3 across all context lengths. On Qwen3, consistent improvements are also observed with RetMask: +0.57 at 8K, +0.45 at 16K, +0.45 at 32K, +1.35 at 64K, +0.89 at 128K. However, the improvements are modest compared to those on Llama-3.1, and the Random-Mask baseline was slightly better than RetMask by 0.24 points when the input sequence is 128K tokens long. We attribute this to the fundamental differences in retrieval score distribution as detailed in § 5.2. We also provide a detailed analysis of how Qwen3's reasoning contents affect the performance in § 4.5.

Improvement is significant on tasks requiring long-context processing. Table 2 details the task-specific impact of the method on Llama-3.1 evaluated using input sequences of 128K tokens. We observe particularly significant improvements on tasks requiring precise information retrieval: *Cite* improved from 3.09 to 5.25 (70% relative improvement) and *Re-rank* improved from 13.73 to 18.16 (32% relative improvement). Both tasks require referring back to the document segments in context and generating text while reorganizing them. These results validate that strengthening retrieval heads enhances both the model's ability to locate information in long contexts and its capacity to generate well-grounded, context-backed responses.

	8K	16K	32K	64K	128K
–	56.03	54.14	52.42	51.65	46.40
LongReward	56.53	54.74	52.50	52.14	46.71
RetMask*	57.21	55.31	52.87	51.97	46.89
RetMask	**58.14**	**56.92**	**53.48**	**53.15**	**48.68**

Table 3: Comparison of Llama-3.1-8B-Instruct fine-tuned via DPO using LongReward (Zhang et al. (2025a), the existing method) and RetMask (the proposed method), evaluated on HELMET across input lengths of 8K–128K tokens. RetMask* is a downsampled variant of RetMask matched to LongReward's sample size. Even with downsampling, RetMask consistently outperforms LongReward.

4.3 Comparison with Existing Methods

To situate RetMask within the broader landscape of approaches for improving LLMs' long-context capabilities, we compare it against LongReward (Zhang et al., 2025a), a recent DPO method that leverages AI feedback for long-context processing. Specifically, we fine-tune Llama-3.1-8B-Instruct via DPO on data generated by each method and evaluate the resulting models on HELMET. For LongReward, we use the officially released dataset[5]. Results are reported in Table 3.

RetMask more effectively improves the long-context processing ability than LongReward. While training on LongReward yields gains over the untrained baseline, these improvements clearly lag behind those achieved by training on RetMask. One might attribute the gap to RetMask's larger dataset size (LongReward: 10K samples vs. RetMask: 294K samples). To isolate the effect of the method from that of dataset size, we train Llama-3.1-8B-Instruct on a varient of the RetMask-based dataset that is downsampled to match the sample size of the LongReward-based dataset. The downsampled variant performs below the full RetMask, but still consistently outperforms the LongReward-trained model. We therefore conclude that RetMask is a more effective approach to improving long-context processing, independent of dataset size. Unlike LongReward, which relies on an LLM judge to score responses based on human-crafted criteria, RetMask automatically treats responses from retrieval-head-masked models as rejected, and those from the original model as chosen. This eliminates the need for both human-crafted criteria and

[5] *dpo_llama3.1_8b* split of https://huggingface.co/datasets/zai-org/LongReward-10k

DPO Strategy	Olmo-3-7B-Instruct 8K	16K	32K	64K	Olmo-3-7B-Think 8K	16K	32K	64K
–	43.73	40.09	33.21	25.00	46.53	45.83	42.41	35.07
Smaller-Model	41.96	37.11	30.63	22.02	45.26	44.44	42.60	33.92
Non-Retrieval-Mask	42.91	39.24	32.24	23.18	46.46	45.56	43.04	34.34
RetMask	**45.51**	**41.75**	**34.28**	**25.59**	**46.69**	**46.07**	**43.09**	**35.54**

Table 4: Performance of Olmo-3 models fine-tuned via DPO with different strategies, evaluated on HELMET across input lengths of 8K–64K tokens. RetMask yields pronounced gains on both the Instruct and the Think variant.

LLM judges, yet training on RetMask still outperforms training on LongReward, which underscores the power of grounding contrastive signals in mechanistic interpretability.

While the dominant approach to extending context length involves continual pre-training (Grattafiori et al., 2024; Yang et al., 2025; Olmo et al., 2025; Wu et al., 2025a), we note that RetMask is complementary to this line of work: It is possible to first extend context length via continual pre-training, then apply RetMask as an additional post-training stage to further improve long-context performance without compromising general capabilities (§ 5.1). We thus conclude that RetMask is not only of academic interest but also of practical value as a lightweight addition to the standard model development pipeline for long-context LLMs.

4.4 Generalization Across Alignment Objectives

Having witnessed the effectiveness of RetMask on Llama-3.1 and Qwen3, we now experiment on the Olmo-3 family, which comprises a standard instruction-tuned variant (Olmo-3-7B-Instruct) and a reasoning-focused variant (Olmo-3-7B-Think). This allows us to examine how RetMask generalizes across models with different alignment objectives while controlling for pre-training and mid-training processes. We include two of the strongest baselines: DPO with a weaker model and arbitrary non-retrieval attention heads masked. As Olmo-3's maximum content length is 64K, we evaluate on input lengths up to 64K and exclude the 128K setting. Results are shown in Table 4.

RetMask improves over both variants. Consistent with results on Llama-3.1 and Qwen3, RetMask yields clear performance gains over all baselines on both Olmo-3-7B-Instruct and Olmo-3-7B-Think across all input lengths. This confirms that the effectiveness of RetMask generalizes across different alignment objectives. Notably, the gains are more pronounced on the Instruct variant than on the Think variant. The reasons may be manifold, with one possible explanation concerning the retrieval head detection process: while NIAH assumes that a model directly outputs the answer, Olmo-3-7B-Think first generates the reasoning contents before the answer, which may degrade the accuracy of retrieval head detection. We leave a deeper investigation into the underlying reasons for future work.

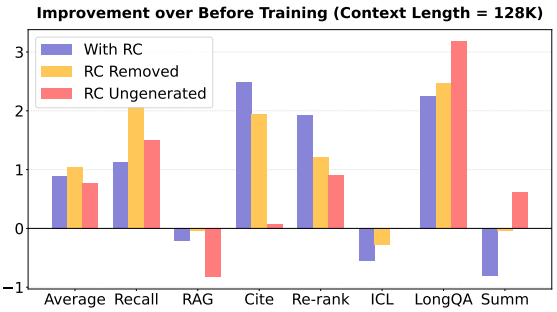


Figure 3: Per-task performance gains of Qwen3-8B on HELMET at 128K input tokens. RC denotes reasoning contents. RetMask consistently improves over the checkpoint before training, regardless of whether reasoning contents are included in the training samples.

4.5 Robustness with Reasoning Mode

Experiments on Qwen3 in § 4.2 use responses generated with reasoning enabled, in which the model outputs reasoning content before producing the final answers. In this section, we investigate how the reasoning process affects training effectiveness. To this end, we conduct additional experiments: (1) **RC removed**: Generate training data with reasoning enabled, then remove the reasoning contents and keep the response only; (2) **RC Ungenerated**: Generate training data with reasoning disabled. The trained models are evaluated with reasoning enabled to ensure results are comparable with those in Table 1.

Removing reasoning contents has minimal impact on the effectiveness of the proposed method. Figure 3 shows that removing reasoning contents

DPO Strategy	Llama-3.1-8B-Instruct Average	Recall	RAG	Cite	Re-rank	ICL	LongQA	Summ
–	46.40	95.13	58.58	3.09	13.73	83.80	42.69	27.81
Smaller-Model	47.30	93.56	**60.79**	3.62	15.29	83.44	**42.78**	31.63
Win-Lose-Pair	46.91	94.44	59.04	4.30	14.17	83.96	40.64	31.80
Non-Retrieval-Mask	47.34	96.75	59.58	4.13	12.86	83.68	39.69	**34.76**
Random-Mask	47.23	**96.38**	60.04	3.45	12.75	83.24	41.59	33.16
RetMask	**48.83**	95.81	59.63	**6.10**	**19.27**	**85.32**	41.87	33.83

Table 5: Model performance of Llama-3.1 trained with different strategies using WildChat, evaluated on HELMET when the input sequence length is 128K. The model trained with RetMask scores the highest among all strategies.

has minimal impact: Five of seven tasks achieve comparable or better performance than training with full reasoning contents. This indicates that ablating retrieval heads degrades response quality sufficiently to provide effective DPO training signals, even without reasoning contents in the training samples. Thus, our method's core mechanism, retrieval head ablation, drives improvements regardless of whether reasoning contents are preserved or not.

Reasoning contents are important for complex tasks. Performance of tasks requiring complex reasoning, namely *Cite* and *Re-rank*, degrades significantly when trained with reasoning contents removed or ungenerated (Figure 3). This demonstrates that for tasks involving source tracking and passage comparison, explicit reasoning chains in training data are important for RetMask to achieve optimal effectiveness: The reasoning content helps the model learn not just retrieval patterns, but also how to reason over retrieved information. For such complex tasks, preserving reasoning contents during training ensures the effectiveness of RetMask.

4.6 Robustness Across Training Datasets

§ 4.2 has demonstrated the effectiveness of RetMask in enhancing long-context capabilities using LMSYS-Chat-1M (Zheng et al., 2024). A potential concern is whether the improvements stem from the retrieval-ablated optimization strategy or from dataset-specific characteristics that happen to enhance long-context processing. To address this, we conduct experiments using Wildchat (Zhao et al., 2024), another dataset collected for instruction tuning. Specifically, we synthesize responses to first-turn instructions in WildChat using RetMask to build the training data. Evaluation results of the trained models are shown in Table 5.

Improvements are consistent on WildChat. As with LMSYS-Chat-1M, RetMask outperforms all baselines on average across tasks. Notably, substantial improvements are observed on *Cite* and *Re-rank* tasks, consistent with findings in § 4.2. These results confirm that the improvements are attributed to the optimization methodology rather than dataset-specific artifacts. This demonstrates the robustness and generalizability of RetMask across different datasets.

	MTB	GPQA	MATH	HE	MMLUP
(a) Llama-3.1-8B-Instruct					
Before	0.75	0.25	**0.53**	**0.71**	**0.49**
After	**0.77**	**0.33**	0.52	0.68	0.48
(b) Qwen3-8B					
Before	0.86	0.56	**0.97**	0.89	0.71
After	**0.88**	**0.60**	**0.97**	**0.92**	**0.74**
(c) Olmo-3-7B-Instruct					
Before	**0.81**	0.36	**0.87**	**0.82**	**0.59**
After	**0.81**	**0.44**	0.83	0.81	0.58
(d) Olmo-3-7B-Think					
Before	0.62	**0.52**	0.95	0.92	0.62
After	**0.63**	**0.52**	**0.96**	**0.94**	**0.63**

Table 6: Model performance before and after training with RetMask. **MTB**,**MATH**,**HE**, **MMLUP** stands for MT-Bench, MATH-500, HumanEval, and MMLU-Pro, respectively. In general, training with RetMask does not degrade the performance on these tasks.

5 Analysis

5.1 Performance on Other Tasks

The previous section showed that RetMask improves long-context processing. A natural question is whether these gains come at the expense of general language understanding and reasoning. To address this concern, we evaluate trained models on five established benchmarks widely used to assess model capability during LLM development: (1) **MT-Bench** (Zheng et al., 2023): Multi-turn conversational ability; (2) **GPQA-Diamond** (Rein et al., 2024): Expert-level scientific reasoning; (3) **MATH-500** (Lightman et al., 2023): Mathematical problem-solving; (4) **HumanEval** (Chen et al., 2021): Code generation; (5) **MMLU-Pro** (Wang et al., 2024): Broad knowledge and understanding. Results are presented in Table 6.

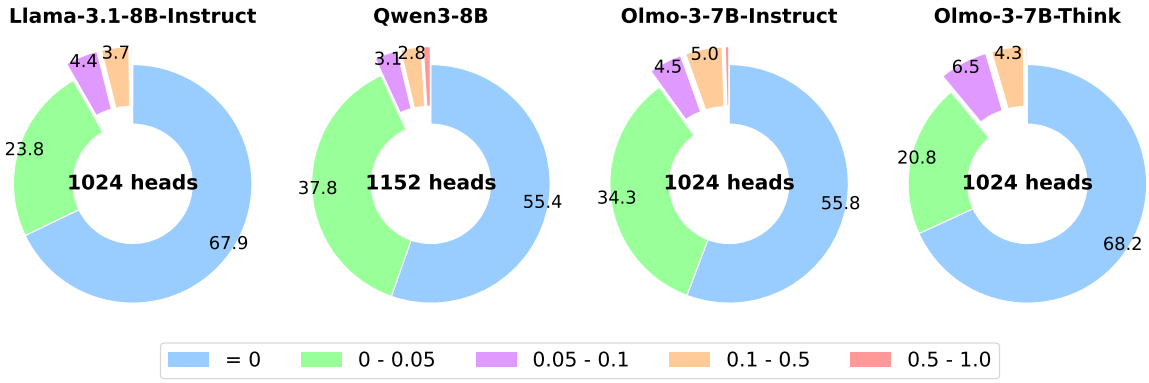


Figure 4: The retrieval score distribution of LLMs tested in this work. While attention heads in Llama-3.1-8B-Instruct exhibit a concentrated pattern of retrieval capabilities, it is more distributed for Olmo-3-7B-Instruct.

RetMask preserves general capabilities. Overall, training with RetMask largely preserves general capabilities, with most scores remaining at or above the level before training. In several cases, we observe modest gains, most notably on GPQA-Diamond. However, we do not attribute these specifically to the retrieval-head-ablated contrastive signals, as they may reflect general effects of DPO training. The primary takeaway is that RetMask's long-context improvements do not come at the cost of general language understanding or reasoning.

5.2 Retrieval Score Distribution

§ 4.2 and § 4.4 demonstrate that RetMask achieves the largest improvements on Llama-3.1-8B-Instruct, followed by Olmo-3-7B-Instruct, Qwen3-8B, and Olmo-3-7B-Think. Here, we explore how the organization of retrieval capabilities across attention heads relates to the effectiveness of RetMask. To this end, we plot the retrieval score distributions of all four models in Figure 4[6].

Retrieval capabilities concentrate on a small set of attention heads. As in Figure 4, across all models, only a small proportion of attention heads show high retrieval scores. This observation is consistent with Wu et al. (2025b), who report a similar distribution of retrieval scores across training stages (pre-trained vs. post-trained), architectures (dense vs. mixture-of-experts), and parameter sizes. With thresholds of $\tau \geq 0.1$ for Llama-3.1 and $\tau \geq 0.05$ for Qwen3 and Olmo-3, RetMask masks only 4–10% of attention heads.

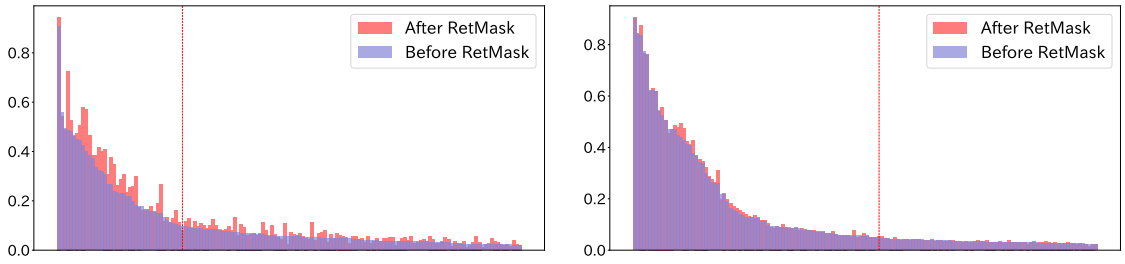
[6]The distribution of Olmo-3-7B-Think is not directly comparable to the other three, as it generates reasoning contents before the final answer during retrieval score computation.

RetMask's effectiveness correlates with the sparsity of the retrieval score distribution. This distribution pattern directly influences the effectiveness of RetMask: When the retrieval score distribution is sparse, i.e., with retrieval capabilities concentrated in a small subset of heads as in Llama-3.1-8B-Instruct, masking the top-scored heads creates a large performance gap between the original and ablated models, yielding strong contrastive training signals. Conversely, when the distribution is less sparse, as in Olmo-3-7B-Instruct and Qwen3-8B, the remaining unmasked heads collectively compensate for the ablated ones, reducing the contrast between chosen and rejected responses and thereby weakening the training signal. This suggests that the sparsity of the retrieval score distribution may serve as a practical predictor of RetMask's effectiveness, allowing practitioners to gauge applicability before committing to training.

5.3 RetMask's Effect on Retrieval Heads

In this section, we analyze how RetMask affects the model by examining changes in the retrieval scores. Figure 5 displays the retrieval scores of the top 150 heads before training, and how their scores change after training. The red vertical dashed lines present the masking threshold: heads to the left were masked when generating rejected responses (40 heads for Llama-3.1 and 79 heads for Qwen3).

Retrieval scores improve after training. For Llama-3.1-8B-Instruct, we observe clear improvements in retrieval scores after RetMask training. The average retrieval score increases from 0.017 to 0.020, showing a 17.6% relative improvement. For Qwen3-8B, the average score increases from 0.020 to 0.021 (+5%), reflecting its limited responsive-



(a) Llama-3.1-8B-Instruct.

(b) Qwen3-8B.

Figure 5: The distribution of retrieval score before and after RetMask. We observe an increase in retrieval scores for both Llama-3.1-8B-Instruct and Qwen3-8B.

ness to retrieval optimization.

Enhancements concentrate on masked heads. The improvements are not uniform across all heads. Specifically, the masked heads show substantial gains, while the other heads exhibit minor changes. For Llama-3.1, the masked heads exhibit an average improvement of 0.051, while non-masked heads show modest changes (average +0.001). This demonstrates that RetMask selectively strengthens the retrieval heads targeted during training.

6 Related Work

Long-Context Language Modeling. Existing methods for long-context LLMs focus on data engineering. Common approaches include adjusting RoPE frequency and staged continual pretraining (Grattafiori et al., 2024; Yang et al., 2025; Gao et al., 2025). For instance, Grattafiori et al. (2024) extends context windows over five stages, and Gao et al. (2025) seeks an optimal mix of short and long context data during multi-stage training. For post-training, findings are mixed: Bai et al. (2024a) reports benefits from long-context fine-tuning, while Gao et al. (2025) finds short sequences sufficient. Closer to our work, Wu et al. (2025a) also leverages attention patterns, but focuses on data selection based on dependency distances. All these studies emphasize data, whereas we take a model-centric approach through mechanistic interpretability.

Mechanistic Interpretability of LLMs. Studies have been conducted to uncover the functionality of components in LLMs. Meng et al. (2022) has revealed that knowledge can be located and edited by manipulating specific neurons. Tang et al. (2024) and Hiraoka and Inui (2025) have demonstrated the existence of language-specific neurons and repetition neurons, respectively. These studies are conducted during inference time, when researchers activate or deactivate the neurons and study models' behaviors. However, few studies connect the discovery to the development of better models. Mondal et al. (2025) reported that language-specific neurons cannot facilitate cross-lingual transfer. Our work takes a step toward connecting mechanistic interpretability with the development of LLMs, specifically by utilizing mechanistic interpretability to develop more effective models within the context of long-context processing. This, in turn, provides evidence for the existence of neural components, i.e., the retrieval head in this study.

7 Conclusion

This work explores how mechanistic interpretability can facilitate model development in the context of long-term context processing. By collecting contrastive response pairs through selective deactivation of retrieval heads, we develop RetMask, an approach that enhances long-context capabilities without compromising general capabilities. Experiments on four models across three families demonstrate consistent improvements, with gains correlating with the sparsity of the retrieval score distribution: Models with sparser distributions achieve stronger gains, while those with less sparse distributions show more modest improvements. This systematic relationship between retrieval head sparsity and training effectiveness reconfirms the functional importance of retrieval heads and demonstrates that mechanistic insights can be transformed into tangible performance improvements.

Future work includes investigating scaling to larger models, developing a theoretical understanding of the underlying mechanisms, and extending this approach to other specialized components.

Limitations

This work focuses on models up to 8B parameters. Scaling to larger models remains an open question, though Wu et al. (2025b) report that retrieval head organization patterns persist at larger scales, suggesting that the core mechanism should generalize. A second limitation concerns retrieval head detection: we rely on NIAH, a synthetic task that targets copy-paste retrieval mechanisms. Exploring detection methods grounded in real-world data is a promising direction that could further improve effectiveness, and we leave this to future work.

Ethics Considerations

Data and Safety. We use publicly available datasets (LMSYS-Chat-1M (Zheng et al., 2024), WildChat (Zhao et al., 2024), Guru-RL-92K (Cheng et al., 2025)) with standard filtering for toxic content and personally identifiable information. However, some potentially harmful content may remain. Models trained with our method should undergo standard safety alignment before deployment.

Synthetic Generation. Our method generates synthetic training data by contrasting full and retrieval-ablated model outputs, without introducing new information about real individuals.

Acknowledgments

We thank the reviewers and area chairs for their feedback. We corrected an implementation error in the Olmo experiments, which revealed that RetMask is effective across all tested models.

This work was supported by JST ACT-X, Japan, Grant Number JPMJAX25CN. This work was also supported by JSPS KAKENHI Grant Number 25H01137. Experiments were carried out using the TSUBAME4.0 supercomputer at Institute of Science Tokyo. We also used ABCI 3.0 provided by AIST and AIST Solutions with support from "ABCI 3.0 Development Acceleration Use".

References

Yushi Bai, Xin Lv, Jiajie Zhang, Yuze He, Ji Qi, Lei Hou, Jie Tang, Yuxiao Dong, and Juanzi Li. 2024a. LongAlign: A recipe for long context alignment of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1376–1395, Miami, Florida, USA.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024b. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 3119–3137, Bangkok, Thailand.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 39 others. 2021. Evaluating large language models trained on code. *Preprint*, arXiv:2107.03374.

Zhoujun Cheng, Shibo Hao, Tianyang Liu, Fan Zhou, Yutao Xie, Feng Yao, Yuexin Bian, Nilabjo Dey, Yonghao Zhuang, Yuheng Zha, Yi Gu, Kun Zhou, Yuqi Wang, Yuan Li, Richard Fan, Jianshu She, Chengqian Gao, Abulhair Saparov, Taylor W. Killian, and 5 others. 2025. Revisiting reinforcement learning for LLM reasoning from a cross-domain perspective. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track (NeurIPS)*.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 8493–8502.

Tianyu Gao, Alexander Wettig, Howard Yen, and Danqi Chen. 2025. How to train long-context language models (effectively). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 7376–7399.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The Llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Jia-Chen Gu, Hao-Xiang Xu, Jun-Yu Ma, Pan Lu, Zhen-Hua Ling, Kai-Wei Chang, and Nanyun Peng. 2024. Model editing harms general abilities of large language models: Regularization to the rescue. In *Proceedings of the 2024 Conference on Empirical*

Methods in Natural Language Processing (EMNLP), pages 16801–16819.

Tatsuya Hiraoka and Kentaro Inui. 2025. Repetition neurons: How do language models produce repetitions? In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 483–495.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. 2024. RULER: What's the real context size of your long-context language models? In *First Conference on Language Modeling*.

Greg Kamradt. 2023. Needle in a haystack - pressure testing LLMs.

Takeshi Kojima, Itsuki Okimura, Yusuke Iwasawa, Hitomi Yanaka, and Yutaka Matsuo. 2024. On the multilingual ability of decoder-based pre-trained language models: Finding and controlling language-specific neurons. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 6919–6971.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *arXiv preprint arXiv:2305.20050*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*.

Kevin Meng, David Bau, Alex J Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Soumen Kumar Mondal, Sayambhu Sen, Abhishek Singhania, and Preethi Jyothi. 2025. Language-specific neurons do not facilitate cross-lingual transfer. In *The Sixth Workshop on Insights from Negative Results in NLP*, pages 46–62.

Team Olmo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, Jacob Morrison, Jake Poznanski, Kyle Lo, Luca Soldaini, Matt Jordan, Mayee Chen, Michael Noukhovitch, Nathan Lambert, Pete Walsh, and 49 others. 2025. Olmo 3. *Preprint*, arXiv:2512.13961.

OpenAI. 2024. Learning to reason with LLMs.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS)*.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2024. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling (COLM)*.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *Preprint*, arXiv:2408.03314.

Tianyi Tang, Wenyang Luo, Haoyang Huang, Dongdong Zhang, Xiaolei Wang, Xin Zhao, Furu Wei, and Ji-Rong Wen. 2024. Language-specific neurons: The key to multilingual capabilities in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 5701–5715.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. 2024. MMLU-pro: A more robust and challenging multi-task language understanding benchmark. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track (NeurIPS)*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP)*, pages 38–45.

Longyun Wu, Dawei Zhu, Guangxiang Zhao, Zhuocheng Yu, Junfeng Ran, Xiangyu Wong, Lin Sun, and Sujian Li. 2025a. LongAttn: Selecting long-context training data via token-level attention. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 19367–19380.

Wenhao Wu, Yizhong Wang, Guangxuan Xiao, Hao Peng, and Yao Fu. 2025b. Retrieval head mechanis-

tically explains long-context factuality. In *The Thirteenth International Conference on Learning Representations (ICLR)*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Howard Yen, Tianyu Gao, Minmin Hou, Ke Ding, Daniel Fleischer, Peter Izsak, Moshe Wasserblat, and Danqi Chen. 2025. HELMET: How to evaluate long-context models effectively and thoroughly. In *The Thirteenth International Conference on Learning Representations (ICLR)*.

Jiajie Zhang, Zhongni Hou, Xin Lv, Shulin Cao, Zhenyu Hou, Yilin Niu, Lei Hou, Yuxiao Dong, Ling Feng, and Juanzi Li. 2025a. LongReward: Improving long-context large language models with AI feedback. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 3718–3739. Association for Computational Linguistics.

Wuwei Zhang, Fangcong Yin, Howard Yen, Danqi Chen, and Xi Ye. 2025b. Query-focused retrieval heads improve long-context reasoning and re-ranking. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 23802–23816.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. 2024. Wildchat: 1m chatGPT interaction logs in the wild. In *The Twelfth International Conference on Learning Representations (ICLR)*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric Xing, Joseph E. Gonzalez, Ion Stoica, and Hao Zhang. 2024. LMSYS-chat-1m: A large-scale real-world LLM conversation dataset. In *The Twelfth International Conference on Learning Representations (ICLR)*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track (NeurIPS)*.

A Details of Experiment Settings

Implementation Details. For Retrieval Head Detection, we use the official implementation from Wu et al. (2025b). For Contrastive Response Generation, we deploy models using the vLLM engine (Kwon et al., 2023) for efficient inference. For Preference Optimization, we use the Transformer Reinforcement Learning (TRL) library[7]. For evaluation, we run Llama-3.1 using the default Transformers library (Wolf et al., 2020) and use the vLLM engine to speed up inference for Qwen3 and Olmo-3, as they produce reasoning content. For each experiment, we evaluate over a single training run.

Computational Resources. For all models tested in this study, the retrieval head detection and deactivation step finishes in 2 GPU hours (NVIDIA H100). The contrastive response generation step finishes in 12 and 36 GPU hours (NVIDIA H100) for instruction models (Llama-3.1-8B-Instruct and Olmo-3-7B-Instruct) and reasoning models (Qwen3-8B and Olmo-3-7B-Think), respectively. The DPO training runs are conducted on either 4× NVIDIA H100 GPUs or 8× NVIDIA H200 GPUs, where all runs finish within 24 hours.

Hyper-Parameters. We train models using the AdamW (Loshchilov and Hutter, 2019) algorithm, with $\beta_1 = 0.9$ and $\beta_2 = 0.95$. For the first 10% of training steps, we use a linear warmup to gradually increase the learning rate from 0 to 5e-7, then employ a cosine scheduler with a minimal learning rate set to 5e-8. We also introduce a weight decay of 0.1. The global batch size is 512. We only tune the learning rate from a range of {2.5e-5, 2.5e-6, 5e-7} using Llama-3.1 and apply the best learning rate on other models.

LLM-As-A-Judge. Both HELMET and MT-Bench utilize LLM-As-a-Judge for evaluation. For HELMET, we follow the original setting and utilize *gpt-4o-2024-05-13* for evaluation. For MT-Bench, we employ *gpt-4o-2024-08-06* for evaluation.

Threshold τ. We tune the threshold τ from {0.05, 0.10} for each model and report the better-performing setting in the main results. As a pilot experiment, we evaluate models on HELMET without relying on LLM-based judges. Specifically, we use ROUGE as a proxy for LLM-as-a-Judge to pre-evaluate model performance at a reduced cost. The results are shown in Table 7.

B Number of Heads to Mask

In order to apply RetMask, it is necessary to set the threshold τ that determines how many heads

[7] https://github.com/huggingface/trl

Threshold τ	Llama 128K	Qwen 128K	Olmo(I) 64K	Olmo(T) 64K
–	44.88	40.62	22.98	31.89
0.05	45.92	**41.82**	**23.56**	**32.40**
0.10	**46.38**	40.78	23.03	31.40

Table 7: Pilot experiment results on deciding the threshold τ. Llama, Qwen, Olmo(I), Olmo(T) is short for Llama-3.1-8B-Instruct, Qwen3-8B, Olmo-3-7B-Instruct, and Olmo-3-7B-Think, respectively.

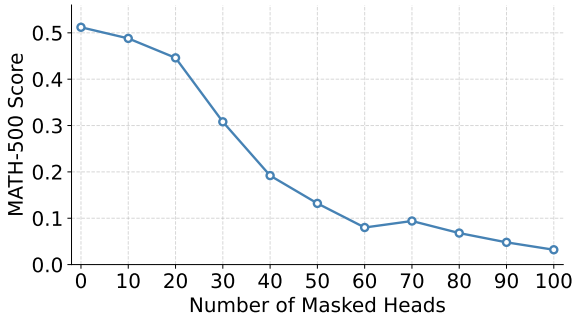


Figure 6: Performance of Llama-3.1-8B-Instruct on MATH-500 when the number of retrieval heads is masked. A sweet spot exists when masking top-30 – 50 heads, where the model performance degrades but does not collapse.

are masked. Masking too few heads results in insufficient contrast between chosen and rejected responses, while masking too many causes rejected responses to degrade into incoherent repetitions. Although the threshold could in principle be tuned by evaluating trained models, this is computationally expensive. Instead, as a pilot study, we evaluate the performance of the model with the top-N retrieval-scored heads masked on an external benchmark, and select τ at the point where performance begins to degrade noticeably. Figure 6 shows the results for Llama-3.1-8B-Instruct.

A sweet spot exists when masking the top 30–50 heads. As shown in the figure, masking more than 60 heads leads to a sharp performance collapse, with scores dropping below 0.1. Our chosen threshold $\tau = 0.1$ corresponds to masking 40 heads, which falls within a sweet spot that results in enough contrast to yield effective training signals, without rendering the ablated model's outputs uninformative.

C Qwen3's Task-Wise Performance

As supplementary material to § 4.2, we report the task-wise performance of Qwen3-based models evaluated on HELMET in Table 8. A similar trend to that observed in Table 2 is evident: RetMask consistently outperforms the baselines on the *Cite* and *Re-rank* tasks.

D Supervised Fine-Tuning Baseline

Apart from the DPO baselines included in the paper, we test the effectiveness of Supervised Fine-Tuning (SFT) in pilot experiments. Specifically, given the preference tuples $\{(x, y_w, y_l)\}$, we train models on (x, y_w) only to focus on the contribution of preferred responses without contrastive signals. The results are shown in Table 9.

SFT degrades performance while RetMask improves it. Table 9 shows that SFT on y_w degrades performance below baseline for all input lengths, while RetMask improves it substantially. This occurs because training on y_w provides minimal signal: The model learns to reproduce existing behavior without targeted improvement. In contrast, RetMask succeeds by contrasting y_w with retrieval-degraded y_l, thereby creating an optimization objective specifically tailored to retrieval mechanisms. This validates that RetMask's effectiveness stems from contrastive signals rather than preferred response quality alone.

E Synthesizing Data with Different LLMs

Throughout this paper, we synthesize contrastive training data from the target model itself — generating both y_w (from the full model θ) and y_l (from the ablated variant θ') using the same model being trained. This section examines whether synthesizing data from a more robust model would enhance RetMask's effectiveness.

Settings. We test cross-model synthesis by training Qwen3-8B on data synthesized from Llama-3.1-8B-Instruct. This represents a favorable scenario for cross-model synthesis: (1) Llama-3.1 exhibits stronger baseline long-context capabilities than Qwen3, and (2) RetMask achieves larger improvements on Llama-3.1 (+2.28) than Qwen3, suggesting higher-quality training signals. We evaluate using ROUGE scores as a proxy for LLM-as-a-Judge to reduce computational costs. Results are reported in Table 10.

Self-synthesis outperforms cross-model synthesis. Table 10 shows that both self-synthesis and cross-model synthesis improve over the baseline, with self-synthesis achieving marginally better results in most settings. While the performance

DPO Strategy	Qwen3-8B Average	Recall	RAG	Cite	Re-rank	ICL	LongQA	Summ
–	44.73	59.69	**53.79**	12.26	15.13	82.00	47.18	**43.06**
Smaller-Model	45.51	59.56	53.54	12.42	16.86	**82.84**	49.03	44.32
Win-Lose-Pair	44.49	58.63	53.33	12.59	15.17	82.16	47.39	42.18
Non-Retrieval-Mask	45.48	60.25	53.17	12.53	16.08	82.28	**51.73**	42.31
Random-Mask	45.37	60.63	53.54	13.51	16.69	82.32	47.93	39.62
RetMask	**45.62**	**60.81**	53.58	**14.74**	**17.06**	81.44	49.43	42.25

Table 8: Model performance on each task of HELMET when the input sequence length is 128K. The advantage of RetMask is evident on real-world tasks such as generation with citation and passage re-ranking.

DPO Strategy	Llama-3.1-8B-Instruct 8K	16K	32K	64K	128K
–	56.03	54.14	52.42	51.65	46.40
SFT	53.51	50.90	49.08	44.73	37.34
RetMask	**58.14**	**56.92**	**53.48**	**53.15**	**48.68**

Table 9: Llama-3.1-8B-Instruct trained with different strategies, evaluated on HELMET. Models are evaluated using input sequences of 8K, 16K, 32K, and 64K tokens. Training with SFT degrades the performance.

Rejected Samples	Qwen3-8B 8K	16K	32K	64K	128K
–	50.89	47.84	47.22	42.15	40.62
Llama-3.1	51.85	**49.30**	47.95	43.13	40.71
Qwen3	**52.40**	48.88	**48.04**	**43.39**	**41.34**

Table 10: Qwen3-8B trained with data synthesized from Llama-3.1-8B-Instruct and Qwen3-8B, evaluated on HELMET. Models are evaluated using input sequences of 8K, 16K, 32K, and 64K tokens. Data synthesized from the target LLM performs better than that synthesized from another LLM in general.

difference is modest, this pattern suggests that RetMask's training signals are somewhat model-specific: Masking patterns from one model's retrieval organization may not perfectly align with those of another model, although the transfer is not entirely ineffective. This indicates that while self-synthesis is preferable for optimal results, cross-model synthesis remains a viable option when computational constraints limit data generation from the target model.

F Experiments with RL datasets

We report experimental results obtained using questions from Guru-RL-92K (Cheng et al., 2025) to synthesize responses. Unlike the datasets used in the prior sections, Guru-RL-92K is specifically collected for reinforcement learning purposes. It consists of challenging problems across a wide range of domains, including mathematics, coding, science, logic, simulation, and tabular reasoning.

Model. We conduct this experiment using Qwen3 and exclude Llama-3.1 and OLMo-3. Llama-3.1 is not trained for deep reasoning on complex problem-solving tasks and frequently degenerates into repetitive outputs during response generation in this setting. For OLMo-3, we observe limited effectiveness of RetMask, which we attribute to the model's internal organization of retrieval-related capabilities. We therefore exclude both models from this evaluation.

Settings. For Qwen3, enabling the reasoning mode often results in very long generations (exceeding 16K tokens) before reaching a final answer, leading to low inference efficiency. We further observe that masking retrieval heads amplifies this issue, causing the model to generate even longer sequences and to more frequently degenerate into repetitive outputs. To control for these effects, we conduct experiments with the reasoning mode turned off and compare the results with those obtained in settings in § 4.5 to assess the effectiveness of training on long responses. In addition to the results using LMSYS-Chat-1M, we include two baselines for Guru-RL-92K: the Non-Retrieval-Mask baseline and the Random-Mask baseline. The results are reported in Table 11. Evaluations here also utilize the ROUGE score as a proxy for LLM-as-a-judge.

Training on long outputs slightly outperforms training on short outputs. The benefit of training on Guru-RL-92K is most evident on the LongQA task, consistent with the findings in § 4.5. These results indicate that synthesizing training data with the reasoning mode enabled can yield additional performance gains. However, generating responses with reasoning enabled is substantially more computationally expensive, making it less cost-effective than training on standard instruction-tuning datasets.

DPO Strategy	Qwen3-8B Average	Recall	RAG	Cite	Re-rank	ICL	LongQA	Summ
–	40.62	59.69	**53.79**	12.26	15.13	82.00	43.96	17.50
Trained on LMSYS-Chat-1M								
RetMask	41.34	61.19	52.96	12.33	16.03	82.00	46.57	**18.30**
Trained on Guru-RL-92K								
Non-Retrieval-Mask	40.54	59.00	53.50	12.38	15.03	**82.40**	44.37	17.12
Random-Mask	41.00	59.38	53.54	11.78	**17.52**	81.52	45.89	17.39
RetMask	**41.59**	60.38	53.58	**13.53**	16.93	81.92	**47.43**	17.40

Table 11: Model performance on each task of HELMET when the input sequence length is 128K when training on Guru-RL-92K. Training with Guru slightly outperforms training with LMSYS-Chat-1M.

	Synthesize Model	# Samples	Avg. Input Length	Avg. Output Length
LMSYS-Chat-1M (Zheng et al., 2024)	Llama-3.1	294,121	63.62	494.69
	Qwen3	293,460	64.59	1642.95
	Olmo-3 (I)	296,224	63.71	825.17
	Olmo-3 (T)	298,308	63.94	1816.54
WildChat (Zhao et al., 2024)	Llama-3.1	280,184	311.68	633.01
Guru-RL-92K (Cheng et al., 2025)	Qwen3 (non-reason)	91,134	330.17	1965.05

Table 12: Statistics of training data utilized in this work. Olmo-3 (I) and Olmo-3 (T) represents Olmo-3-7B-Instruct and Olmo-3-7B-Think, respectively. The average input/output length is calculated after tokenizing with the corresponding tokenizer.

G Statistics of Training Data

The statistics of training data are shown in Table 12. The number of training samples differs from those reported in the original paper due to two reasons: (1) We filter out samples with personal identifiable information; (2) Some of the samples encountered failure during the process of data synthesis.

H Failure Modes of Responses Generated by Retrieval-Head-Ablated Models

In this section, we provide examples that represent the failure modes of responses generated with retrieval-head-ablated models. In general, these responses exhibit a distinctive failure mode: the model loses track of what it is supposed to maintain—whether that's an entity name, a format constraint, or a reasoning chain, etc. Other methods' failures tend to be simpler conceptual errors rather than this kind of mid-generation coherence breakdown. We provide examples for each category as in Figures 7, 8, and 9.

- **Entity name** (Figure 7): The model with ablated retrieval heads tends to miss entity names. This is a distinctive failure mode we observe only in RetMask.
- **Format Constraint** (Figure 8): The model with retrieval heads ablated fails to follow the format constraint.
- **Reasoning Chain** (Figure 9): The model with retrieval heads ablated loses track of its own reasoning chain and generates inconsistent reasoning traces.

Instruction

Write an introduction for Chengdu Push Bio-technology Co., Ltd, a chemical company at No.8 Wuke West Second Road, Wuhou, Chengdu, Sichuan, China.

Response: Original Model

Chengdu Push Bio-technology Co., Ltd is a prominent chemical company based in the vibrant city of Chengdu *[. . .]*

Response: Non-Retrieval-Head-Ablated Model

Chengdu Push Bio-technology Co., Ltd is a prominent chemical company situated in the vibrant city of Chengdu, Sichuan, China. Headquartered at No. 8 Wuke West Second Road, Wuhou District, Chengdu *[. . .]*

Response: Retrieval-Head-Ablated Model

Located in the vibrant city of Chengdu, Sichuan Province, Chengdu is a major chemical hub *[. . .]* The company's manufacturing base is situated at No.8, Wuke Road, Wuhou District *[. . .]*

Figure 7: Example outputs from three model variants given the same instruction. The non-retrieval-head-ablated model correctly reproduces both the company name and address. The retrieval-head-ablated model exhibits a distinctive failure mode: it corrupts the company name ("Chengdu Push Bio-technology Co., Ltd" → "Chengdu") and the address ("No.8 Wuke West Second Road" → "No.8, Wuke Road"), a degradation unique to retrieval head ablation.

Instruction

Rate the quality of the following story on a scale of 1 to 100. Your Response: `{"score": integer}`.
[story omitted]

Response: Original Model

`{"score": 20}`
This story is extremely bad due to several reasons: (1) Historical inaccuracies: The Roman Empire actually fell in 476 AD *[. . .]*

Response: Non-Retrieval-Head-Ablated Model

The story seems to be a historical fiction piece, but it appears to be a historical inaccuracy. The Roman Empire fell in 476 AD *[. . .]*
I would rate the quality of the story a 20.

Response: Retrieval-Head-Ablated Model

{"Quality": "Poorly Set Up World", . . . , "Final Score": 90}
"Final Score": 90

Figure 8: Example outputs from three model variants given a format-constrained instruction. The original model correctly follows the required format (`{"score": integer}`) and produces a reasonable score. The non-retrieval-head-ablated model produces a reasonable score but ignores the format constraint. The retrieval-head-ablated model both violates the format constraint and assigns an inflated score of 90, demonstrating a distinctive failure mode unique to retrieval head ablation.

Instruction

Select the correct choice.
Q: What is the difference between a RIB and a FIB?
A: The FIB is populated based on RIB content.
B: The RIB maintains a mirror image of the FIB.
C: The RIB is used to make IP source prefix-based switching decisions.
D: The FIB is where all IP routing information is stored.

Original Model

The correct choice is: **B**. Explanation: RIB (Routing Information Base) is used to store the routing information learned from various routing protocols.*[. . .]*

Non-Retrieval-Head-Ablated Model

The correct answer is **D**. The FIB (Forwarding Information Base) contains routing information for forwarding packets. *[. . .]*

Retrieval-Head-Ablated Model

The correct answer is D *[. . .]* So, the correct answer is not listed in the choices. However, the closest correct answer is B, but with a slight modification: B: The RIB is populated based on the information stored in the FIB. *[. . .]*

Figure 9: Example outputs from three model variants given a multiple-choice instruction. Both the original and non-retrieval-head-ablated models produce consistent reasoning and commit to a single answer. The retrieval-head-ablated model exhibits inconsistent reasoning: it initially selects D, then contradicts itself by claiming the correct answer is not among the choices, and finally proposes a modified version of B.