検索ヘッドに基づく大規模言語モデルの長文脈処理の改善

Youmi Ma [1]　岡崎直観 [1,2,3]
[1] 東京科学大学　[2] 産業技術総合研究所　[3]NII LLMC
{ma.y, okazaki}@comp.isct.ac.jp

概要

大規模言語モデルには，文脈から情報を取り出す検索ヘッドという機序が存在すると言われている．本研究では，この機序を活用して言語モデルの長文脈処理能力を向上する方法を探究する．具体的には，検索ヘッドを不活性化したモデルの出力と，通常のモデルの出力を用いて選好最適化を行う手法を提案する．実験により，提案手法の有効性が実証され，その効果はモデル内部の検索スコアの分布に依存し，検索機能が特定のヘッドに集中しているモデルにおいて高い効果を示すことが分かった．

1　はじめに

大規模言語モデル（Large Language Model; LLM）が文脈内学習 [1] や推論時スケーリング [2, 3] などの能力を発揮するには，長い文脈を処理する能力が不可欠である．こうした長文脈処理能力は，**検索ヘッド**（Retrieval Head）という機序と強く関連することが明らかになっている [4]．検索ヘッドは LLM 内部のマルチヘッド・アテンションから特定され，文脈において関連する単語に注目し，情報を取り出す働きがあると考えられている．逆に検索ヘッドを不活性化すると，質問応答などの下流タスクにおいて性能が低下することが報告されている．

検索ヘッドの発見は，LLM の長文脈処理のメカニズムを解明する手がかりとなった．しかし，この機序を活用して LLM の性能向上に成功したという報告はない．このように，LLM の解釈性と高度化の間には溝が存在する．例えば，LLM の内部に知識を蓄える機構が発見されたものの [5, 6]，この機構を恣意的に操作すると LLM の汎用性能が損なわれることが示されている [7]．また，LLM の多言語能力に関する機構が特定されたが [8]，これらを制御しても LLM の言語間転移には寄与しなかった [9]．そこで，次の問いが生じる — 検索ヘッドに着想を得て LLM の長文脈処理能力を向上させられるか？

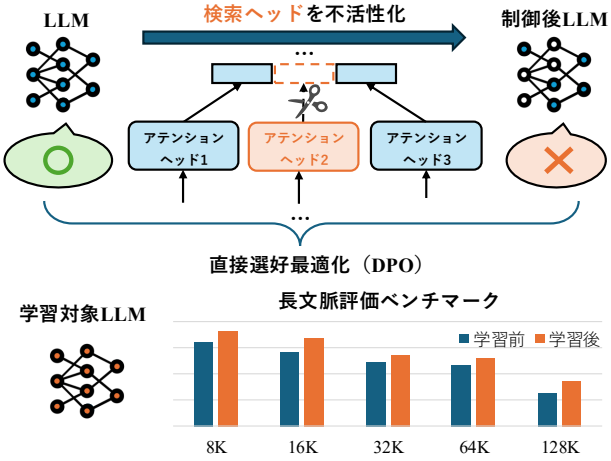


図 1: 本研究の概要．検索ヘッドに特化した学習信号を構築し，LLM の長文脈処理能力の改善を狙う．

本稿では，検索ヘッドに特化した最適化を行い，言語モデルの長文脈処理能力を向上させる手法を提案する．具体的には，図 1 に示すように，通常のモデルと，検索ヘッドを不活性化したモデルからそれぞれ，出力をサンプリングする．通常のモデルの出力を正例，不活性化したモデルの出力を負例とし，直接選好最適化（Direct Preference Optimization; DPO）[10] を行う．これにより，検索ヘッドに特化した対照的な学習信号を構築し，その機能を強化することを狙う．検索ヘッドを不活性化することから，本手法を RetMask（**Ret**rieval-head **Mask**ing）と呼ぶ．

RetMask を Llama-3.1 [11]，Qwen3 [12]，Olmo-3 [13] に適用し，長文処理能力を評価した．その結果，Llama-3.1 と Qwen3 では有効性が確認されたが，Olmo-3 では効果が限定的であった．分析により，この差異はモデル内部の検索スコアの分布に起因し，検索機能が特定のヘッドに集中しているモデルにおいて提案手法の有効性が高まることが明らかになった．これらの知見は，検索ヘッドという機序をより強く裏付けると同時に，その LLM 開発への有用性を示すもので，本研究は LLM の解釈性が高度化に繋がることを初めて示した事例と言えよう．

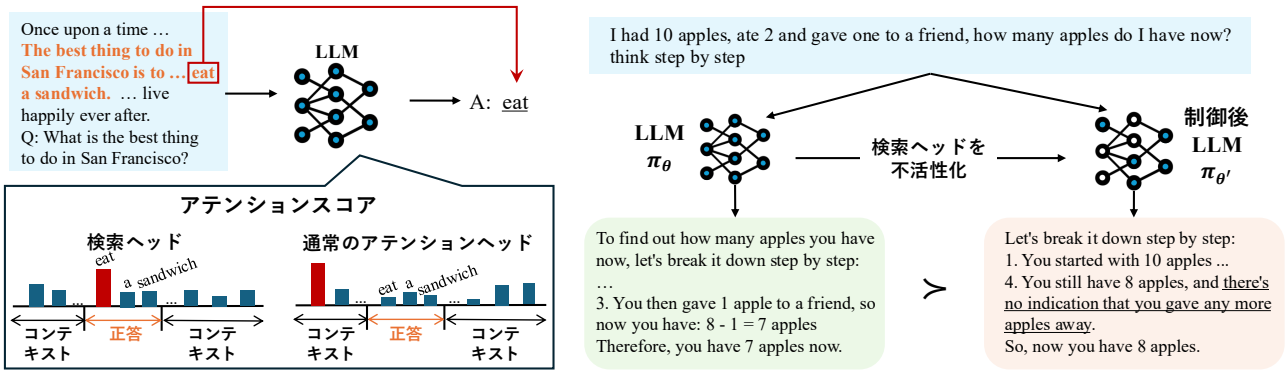


図 2: 予備知識（左）と提案手法（右）の概要．右側に示す例は実際に学習に用いたデータから選択した．

2 予備知識

検索ヘッドを特定する方法は先行研究 [4] により提案されている（図 2 左）．この手法は，次の Needle-In-A-Haystack タスクに基づく．

Needle-In-A-Haystack (NIAH) [14] 質問 q とその正答 k に対し，正答 k を n 個の段落からなるコンテキスト $x = p_1, \ldots, p_n$ に挿入する．ここで，各段落 p_i $(1 \le i \le n)$ は質問・正答とは無関係である．これにより，正答を無関係な段落に埋め込んだコンテキスト $x' = p_1, \ldots, k, \ldots, p_n$ が得られる．このとき，正答 k の各トークンの挿入位置を集合 $\mathcal{I}_k$ で表す．NIAH タスクは，入力としてコンテキスト x' と質問 q を受け取った LLM が，出力として k を生成できるかを評価する．生成できた場合，モデルがコンテキスト x' から正答 k を取り出したと見なす．

検索ヘッド 高い頻度でコンテキストから正答を取り出せたアテンションヘッド（以降「ヘッド」と呼ぶ）を検索ヘッドと定義する．具体的には，y_t を時刻 t で生成されるトークン，$\boldsymbol{a}_t \in \mathbb{R}^{|x'|+t-1}$ をヘッド h のアテンションスコアとする．次式が成り立つとき，ヘッド h がトークン x_j を取り出すと考える．

$$y_t = x_j,\ j = \arg\max(\boldsymbol{a}_t) \tag{1}$$

さらに，$j \in \mathcal{I}_k$ を満たすとき，ヘッド h は正答を取り出していると考える．これに基づき，ヘッド h の検索スコアを次式で定義する．

$$\mathrm{RetrievalScore}(h) = \frac{1}{|\mathcal{T}|} \sum_{(g_h,k)\in\mathcal{T}} \frac{|g_h \cap k|}{|k|} \tag{2}$$

ここで，$\mathcal{T}$ はテスト事例の集合，g_h はヘッド h が取り出したトークンの集合，k は正答である．すなわち，検索スコアは，ヘッド h が取り出したトークンと正答トークンの重複度を測る．検索スコアが閾値 τ を超えるヘッドを検索ヘッドと認定する．

3 提案手法：RetMask

本節では，検索ヘッドを活用して LLM の長文脈処理能力を高度化する手法を提案する．具体的には，図 2 右に示すように，通常のモデル π_θ の応答文を正例，検索ヘッドを不活性化したモデル $\pi_{\theta'}$ の応答文を負例とし，モデル π_θ を学習する．本手法は以下の三段階で構成される．

検索ヘッドの不活性化 先行研究 [4] に従い，モデル π_θ の全てのヘッドにおける検索スコアを計算し，検索ヘッドを認定する．認定した検索ヘッドの集合を $\mathcal{H}_{\mathrm{ret}}$ とする．モデル $\pi_{\theta'}$ は $\mathcal{H}_{\mathrm{ret}}$ に属するヘッドの出力射影行列をゼロにする（検索ヘッドの影響を消去する）ことで構成する（式 3）．

$$\boldsymbol{W}_{o'}^{h} = \begin{cases} \boldsymbol{0} & (h \in \mathcal{H}_{\mathrm{ret}} \text{ のとき}) \\ \boldsymbol{W}_{o}^{h} & (\text{その他の場合}) \end{cases} \tag{3}$$

応答文生成 既存の指示チューニングデータセットの指示文を用い，モデル π_θ と不活性化後のモデル $\pi_{\theta'}$ からそれぞれ応答文を生成する．具体的には，指示文 x が与えられたとき，正例 y_w と負例 y_l を以下のように生成する．

$$y_w \sim \pi_\theta(\cdot|x) \tag{4}$$

$$y_l \sim \pi_{\theta'}(\cdot|x) \tag{5}$$

選好最適化 生成した応答文と指示文を組み合わせ，選好最適化に用いる三つ組 $\{(x, y_w, y_l)\}$ を構成する．得られた三つ組を用いて，次式を最小化することで直接選好最適化（DPO）を行う．

$$\mathcal{L}(\pi_\theta) = -\mathbb{E}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(y_w|x)}{\pi_{\mathrm{ref}}(y_w|x)} - \beta\log\frac{\pi_\theta(y_l|x)}{\pi_{\mathrm{ref}}(y_l|x)}\right)\right] \tag{6}$$

ここで，π_{ref} は学習する前のモデル π_θ，β は π_{ref} からの乖離度を制御するハイパーパラメータである．

4 実験

4.1 実験設定

モデル Llama-3.1-8B-Instruct [11], Qwen3-8B [12], Olmo-3-7B-Think [13] を用いて実験を行う．検索ヘッドの認定に用いる閾値は，予備実験の結果に基づき，Llama-3.1 は $\tau = 0.1$，Qwen3 と Olmo-3 は $\tau = 0.05$ とした．

ベンチマーク 評価には HELMET [15] を用いる．HELMET は LLM の長文脈処理能力の評価に特化したベンチマークであり，LLM 開発でよく用いられる合成タスク RULER [16] から，検索拡張生成や長文書に対する質問応答などの実世界タスクまで，幅広くカバーしている．その他，LLM 開発において一般的に使用されているベンチマーク（数学や一般教養など）で評価した結果を付録 B に示す．

学習データ RetMask は指示文を含む任意のデータセットに適用できる．本稿では，LMSYS-Chat-1M [17] を学習データとして用いる．また，提案手法の効果が学習データセットに依存するものではないことを示すため，WildChat [18] を用いた実験も行い，その結果を付録 C に示す．なお，これらの学習データはベンチマークのタスクと重なりがない．

ベースライン 次の方法で負例 y_l を生成して DPO を適用したモデルをベースラインとする．(1) **小規模モデル**：自身よりもパラメータ数が少ないモデルから y_l を生成する[1)]．(2) **勝敗ペア**：自身 π_θ から応答を生成し，より品質が低いとされる応答を y_l とする[2)]．(3) **非 RH マスク**：検索ヘッド以外のヘッドを $|\mathcal{H}_{\text{ret}}|$ 個ランダムに不活性化したモデルから y_l を生成する．(4) **ランダムマスク**：ヘッドを $|\mathcal{H}_{\text{ret}}|$ 個ランダムに（検索ヘッドを含んでもよい）不活性化したモデルから y_l を生成する．

ハイパーパラメータを含む他の実験設定の詳細を付録 D に示す．

4.2 実験結果

Llama-3.1, Qwen3, Olmo-3 の実験結果をそれぞれ表 1，2，3 に示し，タスク毎の性能を付録 A に示す．

表 1 より，**RetMask を用いた学習は全ての入力長において最高性能を達成**した．また，勝敗ペアおよび小規模モデルのベースライン手法は，RetMask の性能に及ばなかった．これにより，RetMask の有効性は選好最適化そのものではなく，検索ヘッドの不活性化に起因することが示唆される．さらに，非 RH マスクやランダムマスクと比較しても，RetMask の性能が高い．このことから，任意のヘッドを不活性化するよりも，検索ヘッドを対象として不活性化する方が学習効果が高いことが確認された．

表 2 においても，向上幅は表 1 ほど顕著ではないものの，RetMask を用いた学習は概ね全ての入力長において最高性能を達成した．一方，表 3 においては，RetMask の効果は限定的であった．モデルによる効果の差を分析するため，モデル内部の検索スコアの分布を調査し，5.1 節で説明する．

表 1: 提案手法およびベースライン手法で学習した Llama-3.1 を HELMET で評価した結果．8K，16K，32K，64K，128K はデータセットの入力長を示す．

	Llama-3.1-8B-Instruct				
負例生成方法	**8K**	**16K**	**32K**	**64K**	**128K**
学習前	56.03	54.14	52.42	51.65	46.40
小規模モデル	56.77	55.32	**53.48**	52.18	47.53
勝敗ペア	56.50	54.42	52.47	51.62	46.05
非 RH マスク	56.45	55.55	53.19	52.14	47.19
ランダムマスク	56.67	55.95	53.14	52.30	47.04
RetMask	**58.14**	**56.92**	**53.48**	**53.15**	**48.68**

表 2: 提案手法およびベースライン手法で学習した Qwen3 を HELMET で評価した結果．8K，16K，32K，64K，128K はデータセットの入力長を示す．

	Qwen3-8B				
負例生成方法	**8K**	**16K**	**32K**	**64K**	**128K**
学習前	53.20	50.16	49.89	45.44	44.73
小規模モデル	52.52	49.81	48.71	46.67	45.51
勝敗ペア	52.80	50.14	49.71	45.93	44.49
非 RH マスク	53.02	50.28	48.67	**46.79**	45.48
ランダムマスク	49.99	47.02	45.75	43.85	**45.86**
RetMask	**53.77**	**50.61**	**50.34**	**46.79**	45.62

5 分析

5.1 検索スコアの分布

4.2 節で観察された性能差は，検索スコアがマルチヘッド注意の中でどのように分布しているかに起因すると仮説を立てた．この仮説を検証するために，モデルにおける全アテンションヘッドの検索スコアを精査し，その分布を図 3 に示す．

図 3 において，検索スコアがゼロよりも大きいヘッドは検索に何かしら関与しており，検索機能を

1) Llama-3.1-8B-Instruct に対しては Llama-3.1-3B-Instruct, Qwen3-8B と Olmo-3-7B-Think に対しては Qwen3-0.7B を用いた．

2) 品質の測定には Gemma-3-27B-IT [19] を用いた．

表 3: 提案手法およびベースライン手法で学習した Olmo-3 を HELMET で評価した結果．8K，16K，32K，64K は入力長を示す．

	Olmo-3-7B-Think			
負例生成方法	**8K**	**16K**	**32K**	**64K**
学習前	46.53	**45.83**	42.41	**35.07**
小規模モデル	45.26	44.44	42.60	33.92
非 RH マスク	45.54	44.65	**42.95**	34.22
RetMask	**47.07**	45.19	42.68	**35.07**

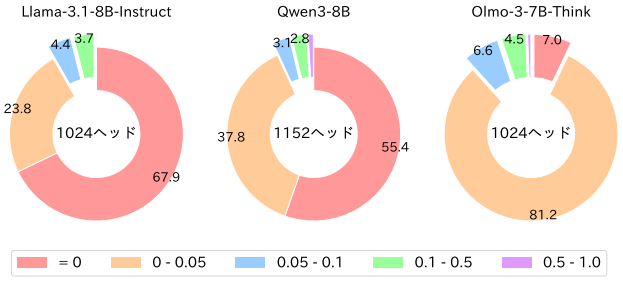


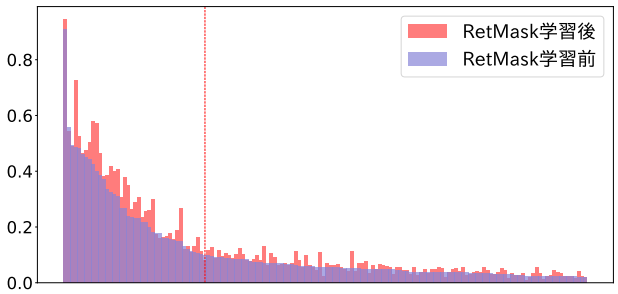
図 3: 検索スコアの分布．数値は割合を示す．

有すると考える．すると，Llama-3.1 では検索機能が約 3 割のヘッドに集中しているのに対し，Qwen-3 では約 4 割，Olmo-3 では 9 割以上のヘッドに分散している．すなわち，検索ヘッドを不活性化した場合，Llama-3.1 と Qwen3 は検索機能の多くを喪失する．一方，Olmo-3 では検索ヘッドを不活性化しても，検索機能をある程度有するヘッドが多数残存するため，検索の機序が維持されると考えられる．この結果は前述の仮説を支持しており，RetMask の有効性は検索機能の集中度に依存することを示している．また，RetMask の適用に先立ち検索スコアの分布を調査することで，提案手法の有効性を事前に予測できる可能性が示唆される．

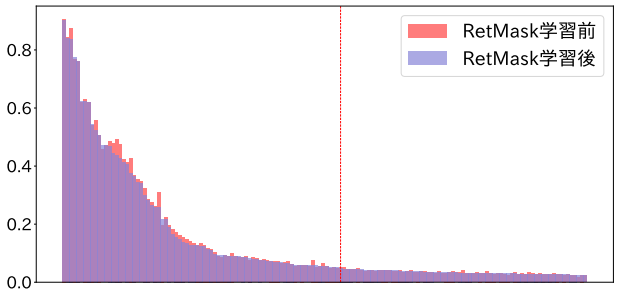
5.2 RetMask がモデルに与える影響

RetMask の学習がモデルにどのような影響をもたらしたかを分析する．図 4 では，学習前の検索スコアが高い上位 150 個のヘッドについて，学習前後の検索スコアの変化を示している．赤い点線は不活性化の閾値を示しており，点線より左側のヘッドが検索ヘッドとしてデータ合成時に不活性化された．

図 4 により，Llama-3.1 では学習後に検索スコアが顕著に上昇している．具体的には，検索スコアの平均値が 0.017 から 0.020 に上昇し，17.6%の相対的な向上を示した．Qwen3 においても，平均値が 0.020 から 0.021 へと 5%上昇したが，上昇幅が小さい．この事実は，Qwen3 の長文脈処理能力の向上が Llama-3.1 ほど顕著でないことと整合している．これは，検索ヘッドを不活性化したモデルの出力を負例とすることで，検索ヘッドの機序に特化した学習が行えたためと考えられる．



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



(b) Qwen3-8B.

図 4: 学習前後の検索スコアの分布

また，検索スコアの改善は全てのヘッドで均一ではなく，不活性化されたヘッドに集中している．Llama-3.1 では，不活性化されたヘッドの平均改善幅が 0.051 であるのに対し，そうでないヘッドでは変化が軽微（平均+0.001）であった．これは，RetMask が学習時に対象とした検索ヘッドを選択的に強化していることを示している．

6 おわりに

本稿では，検索ヘッドに着想を得て LLM の長文脈処理能力を向上させられるかを検証した．具体的には，検索ヘッドを選択的に不活性化したモデルから応答を合成し，選好最適化の負例とすることで，長文脈処理の強化に特化した最適化手法を提案した．提案手法は Llama-3.1 および Qwen3 で長文処理能力を改善したものの，Olmo-3 での効果は限定的だった．その要因を分析したところ，提案手法の効果はモデル内部における検索機能の集中度に依存することが明らかになった．今後は，検索ヘッドを不活性化した合成データによる選好最適化が有効である理由をより詳細に解明したい．

謝辞

本研究は，JST ACT-X JPMJAX25CN の支援及び JSPS 科研費 25H01137 の助成を受けた．また，実験では東京科学大学のスーパーコンピュータ TSUBAME4.0，産総研及び AIST Solutions が提供する ABCI 3.0 を「ABCI 3.0 開発加速利用」の支援を受けて利用した．

参考文献

[1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, et al. Language models are few-shot learners. In **Advances in Neural Information Processing Systems (NeurIPS)**, Vol. 33, pp. 1877–1901. Curran Associates, Inc., 2020.

[2] Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In **The Thirteenth International Conference on Learning Representations (ICLR)**, 2025.

[3] OpenAI. Learning to reason with LLMs, 2024.

[4] Wenhao Wu, Yizhong Wang, Guangxuan Xiao, Hao Peng, and Yao Fu. Retrieval head mechanistically explains long-context factuality. In **The Thirteenth International Conference on Learning Representations (ICLR)**, 2025.

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

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

[7] Jia-Chen Gu, Hao-Xiang Xu, Jun-Yu Ma, Pan Lu, Zhen-Hua Ling, Kai-Wei Chang, and Nanyun Peng. 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)**, pp. 16801–16819, 2024.

[8] Tianyi Tang, Wenyang Luo, Haoyang Huang, Dongdong Zhang, Xiaolei Wang, Xin Zhao, Furu Wei, and Ji-Rong Wen. 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)**, pp. 5701–5715, 2024.

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

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

[11] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, et al. The Llama 3 herd of models, 2024.

[12] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, et al. Qwen3 technical report, 2025.

[13] Team Olmo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, et al. Olmo 3, 2025.

[14] Greg Kamradt. Needle in a haystack - pressure testing LLMs, 2023.

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

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

[17] 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. LMSYS-chat-1m: A large-scale real-world LLM conversation dataset. In **The Twelfth International Conference on Learning Representations (ICLR)**, 2024.

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

[19] Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, et al. Gemma 3 technical report, 2025.

[20] 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. 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)**, 2023.

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

[22] Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In **The Twelfth International Conference on Learning Representations (ICLR)**, 2024.

[23] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, et al. Evaluating large language models trained on code, 2021.

[24] Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, et al. 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)**, 2024.

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

[26] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, et al. Transformers: State-of-the-art natural language processing. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP)**, pp. 38–45, 2020.

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

表 4: 提案手法およびベースラインで学習した Llama-3.1 を評価した結果．評価時の入力長は 128K である．

	Llama-3.1-8B-Instruct						
負例生成方法	**Rec**	**RAG**	**Cite**	**ReR**	**ICL**	**LQA**	**Sum**
学習前	95.13	58.58	3.09	13.73	83.80	42.69	27.81
小規模モデル	94.19	**60.83**	4.22	13.44	83.76	43.15	33.12
勝敗ペア	93.56	59.50	3.72	12.47	83.36	39.26	30.48
非 RH マスク	**96.69**	59.00	3.45	11.38	84.28	40.93	**34.62**
ランダムマスク	96.38	59.29	3.88	10.79	83.52	41.32	34.10
RetMask	95.44	59.71	**5.25**	**18.16**	**84.92**	**43.84**	33.45

表 5: 提案手法およびベースラインで学習した Qwen-3 を評価した結果．評価時の入力長は 128K である．

	Qwen3-8B						
負例生成方法	**Rec**	**RAG**	**Cite**	**ReR**	**ICL**	**LQA**	**Sum**
学習前	59.69	**53.79**	12.26	15.13	82.00	47.18	**43.06**
小規模モデル	59.56	53.54	12.42	16.86	**82.84**	49.03	44.32
勝敗ペア	58.63	53.33	12.59	15.17	82.16	47.39	42.18
非 RH マスク	60.25	53.17	12.53	16.08	82.28	**51.73**	42.31
ランダムマスク	60.63	53.54	13.51	16.69	82.32	47.93	39.62
RetMask	**60.81**	53.58	**14.74**	**17.06**	81.44	49.43	42.25

A　タスク毎の性能

学習したモデルの性能を HELMET における各タスクで評価した結果を表 4 と 5 に示す．Rec は RULER [16]，RAG は検索拡張生成，Cite は引用付き生成，ReR は段落リランキング，ICL は文脈内学習，LQA は長文書に対する質問応答，Sum は長文書に対する要約タスクである．

実験結果から，RetMask は特に引用付き生成と段落リランキングにおいて高い有効性を示した．これらのタスクはいずれもコンテキスト中の文書を参照しながら応答を生成するものであり，情報検索能力を特に必要とする．この結果は，検索ヘッドを強化することで，長文脈における情報検索能力と文脈に基づいた応答生成能力の双方が向上することを示している．

B　別タスクの性能

提案手法により長文脈処理能力が向上したが，他のタスクの性能を犠牲にしていないかを検証する．評価タスクとして，汎用対話能力を測る MT-Bench [20]，大学院生レベルの科学推論能力を測る GPQA [21]，数学能力を測る MATH-500 [22]，コード生成能力を測る HumanEval [23]，一般知識の理解力を測る MMLU-Pro [24] の 5 種類を選定した．評価結果を表 6 に示す．

実験の結果，全モデルで MT-Bench における性能が向上した．MT-Bench は汎用対話能力を測るマルチターンベンチマークであり，本研究で使用した学習データも汎用対話データであるため，性能向上は予想の範囲である．ただし，既存の汎用対話能力向上手法は高性能モデルの出力を用いた蒸留が一般的であるのに対し，本手法では学習対象モデル自体の出力を使用し，不活性化したモデルとの対比で学習を行っている．この結果は，RetMask が対話コンテキストから情報を検索する能力を強化することで，対話能力を向上させることを示唆している．

また，GPQA においても全モデルで性能向上が確認された．これは，検索機能の強化により，モデルが推論過程をより精緻に参照できるようになったためと考えられる．その他のタスクについては，学習後も同水準の性能を維持している．以上より，提案手法による検索ヘッドを対象とした最適化は，他の能力を犠牲にしないことが確認された．

表 6: 学習前後のモデル性能

	MT-Bench	**GPQA**	**MATH-500**	**HumanEval**	**MMLU-Pro**
(a) Llama-3.1-8B-Instruct					
学習前	0.75	0.25	**0.53**	**0.71**	**0.49**
学習後	**0.77**	**0.33**	0.52	0.68	0.48
(b) Qwen3-8B					
学習前	0.86	0.56	**0.97**	0.89	0.71
学習後	**0.88**	**0.60**	**0.97**	**0.92**	**0.74**
(c) Olmo-3-7B-Think					
学習前	0.62	0.52	**0.95**	0.92	0.62
学習後	**0.64**	**0.53**	**0.95**	**0.93**	**0.63**

表 7: 提案手法およびベースラインで学習した Llama-3.1 を評価した結果．学習データは WildChat，評価時の入力長は 128K である．

	Llama-3.1-8B-Instruct						
負例生成方法	**Rec**	**RAG**	**Cite**	**ReR**	**ICL**	**LQA**	**Sum**
学習前	95.13	58.58	3.09	13.73	83.80	42.69	27.81
小規模モデル	93.56	**60.79**	3.62	15.29	83.44	**42.78**	31.63
勝敗ペア	94.44	59.04	4.30	14.17	83.96	40.64	31.80
非 RH マスク	96.75	59.58	4.13	12.86	83.68	39.69	**34.76**
ランダムマスク	**96.38**	60.04	3.45	12.75	83.24	41.59	33.16
RetMask	95.81	59.63	**6.10**	**19.27**	**85.32**	41.87	33.83

C　WildChat での実験結果

§ 4.2 では，LMSYS-Chat-1M を用いた学習で，RetMask が長文脈処理能力の向上に成功したことを報告している．ここでは，提案手法の頑健性を確認するため，別の指示チューニングデータセットで実験を行う．具体的には，WildChat [18] から指示文を抽出し，応答文を 3 節に従って生成し，モデルの学習を行う．結果を表 7 に示す．

実験結果から，WildChat を用いた学習でも LMSYS-Chat-1M と同様の傾向が確認された．全タスクの平均において，RetMask の性能が全てのベースラインを上回っており，特に引用付き生成および段落リランキングにおいて，その有効性が顕著である．以上により，RetMask の有効性が複数の学習データで確認された．

D　実験設定の詳細

実装　検索ヘッドの検出に Wu ら [4] の実装を採用する．応答文の生成には vLLM [25] を用い，選好最適化には，trl ライブラリ [3] を利用して実装した．評価には Yen ら [15] の実装を採用し，Llama-3.1 の推論には transformers ライブラリ [26]，Qwen3 と Olmo-3 の推論には vLLM エンジンを用いた．なお，本稿における報告値は，学習と評価を一回だけ実行した結果である．

計算資源　本稿における全ての学習実験は，NVIDIA H100 GPU 4 基もしくは NVIDIA H200 GPU 8 基を使用し，24 時間以内に完了した．

ハイパーパラメータ　最適化には AdamW [27] アルゴリズムを用い，$\beta_1 = 0.9, \beta_2 = 0.95$ にした．また，学習率スケジューラとして，コサイン波形による減衰を用いた．学習の最初の 10%では線形にウォームアップを行い，最大学習率である 5e-7 に到達した後，コサイン波形による減衰を適用し，学習の最後には最小学習率である 5e-8 に到達するように調整した．学習率の探索は Llama-3.1 を用い，{2.5e-5, 2.5e-6, 5e-7} の範囲で行った．また，検索ヘッドを決める閾値 τ は，モデルごとに $\{0.05, 0.10\}$ の範囲で探索した．HELMET で最も高い性能を示す閾値を選択し，その結果を § 4.2 に報告した．

3）　https://github.com/huggingface/trl