Think, Align, Select: Query–Key Scores for LLM Reasoning

Mark Obozov & Eduard Tulchinskii & Kristian Kuznetsov & Michael Diskin et al.

Abstract

We demonstrate that a "think-first" phase via chain-of-thought (CoT) prompting systematically strengthens internal query–key (QK) alignment, improving ability to select and verify answers directly from model activations rather than decoded tokens. Building on multiple-choice evaluation with MMLU-Pro and extending to free-form reasoning on MATH-500, GSM8K, and our variant of Humanity's Last Exam, we evaluate three settings: (i) MCQA vs MCQA+CoT with QK-based selection; (ii) candidate generation with/without CoT followed by QK-based selection among self-proposed answers; and (iii) QK-based verification of LLM solutions. We analyze QK-score accuracy, permutation robustness, and diagnostics relating alignment strength to correctness. This yields a white-box, computation-efficient decision rule that turns CoT from a purely generative aid into a deliberation-then-selection mechanism grounded in the model's own representations. By leveraging this internal signal, we surpass preference-optimized LLMs on fundamental reasoning tasks, achieving performance gains up to 22

Introduction

Recent advances have shown that prompting models to generate Chain-of-Thought (CoT) explanations [29] and applying self-consistency [26] can substantially improve their reasoning reliability. Nevertheless, evaluation of the proper reasoning chains and strategies to select the right answer remains a challenge. Correctness and reliability of reasoning trajectory often requires verification through external solvers [21, 24] or reranking heuristics [15]. These approaches highlight the need for efficient and interpretable internal signals that can complement or replace external heuristics.

In this work, we explore the use of the Query–Key (QK) score, a raw dot-product measure of alignment between query and key vectors within the transformer attention mechanism, as such an internal signal. Prior work has used QK-scores for probing latent knowledge in MCQA [22] and for detecting logical consistency [23], but their potential for guiding reasoning and answer selection remains underexplored. We hypothesize that QK-score can serve not only as diagnostic tools, but also as practical mechanisms for improving LLM reasoning across diverse tasks.

1 Method

Background on QK-score In transformer architectures, the interaction between query and key vectors affects how information flows across tokens. Beyond their normalized role in attention weights, we can define the raw dot product of query vector of i-th token $\overset{(l,h)}{i}$ and key vector of j-th token $\overset{(l,h)}{j}$ in the attention head (l, h) as $S^{(l,h)}_{QK}(\cdot) = \overset{(l,h)\top}{i}\overset{(l,h)}{j}$. Recent studies have employed this measure to probe model behavior in diverse tasks, such as identifying latent preferences in multiple-choice question answering or isolating heads that evaluate logical consistency [22, 23]

QK-score and connection between reasoning parts. We use QK-score to quantify the strength of the connection between two reasoning parts. Suppose that we have a text consisting of two parts (c, a), which we will call *premise* (c) and *response* (a). By c_r and a_r we denote tokens that represent c and a; usually they are the punctuation or end-of-line signs at the very end of the respective parts; we choose them because they 'collect' the meaning of the preceding text and at the same time they don't have their own meaning (unlike tokens that are part of actual words). Calculating $S^{(l,h)}_{QK}(c_r, a_r)$, we measure how strongly a particular attention head aligns the response to the premise. We use QK-scores to compare multiple responses to the same premise (i.e., answer candidates to the question) .

In this work we explore three different setups and particular application details of the QK-score in them vary slightly.

- For MCQA, the *premise* is a concatenation of an instruction, context (if given), question, and a full list of choices one per line. We pass all options to the model in one go and only vary the choice of the premise-representing token (c_r) for the calculation of the QKscore. We choose the end-of-line tokens after each of the choices. For simple MCQA, *response*representing token is the last token of the prompt (i.e., colon in 'ANSWER:'). For MCQA with reasoning, we prompt the LLM is prompted with the premise, consider its output as the response, and select the token right before the final answer option as the response-representing token a_r.

  The prediction is the option is the one that achieves the highest QK-score.
- For Hypothesis selection *premise* is the concatenation an instruction and problem. c_r is selected as the end-of-line token in the end of the problem. LLM prompted with the premise and its generation is the *response*; a_r is chosen as the last token of the generation.

  The selected hypothesis is the one that achieves highest QK-score.

Head Selection Procedure. When it is not stated otherwise, we do not aggregate predictions or QK-scores from multiple attention heads. Instead, in each experiment we use a separate *calibration* subset of the data from the same domain to select the single best performing head.

Results

1.1 QK-score with CoT for MCQA

First, we assess the efficiency of the QK score for simple MCQA and MCQA with integrated CoT reasoning. In both setups, the model is prompted with context, a question, a list of options, and an instruction to output only one letter – the correct option; in the second setup, the prompt also includes an instruction to think step-by-step before giving the final answer.

	MCQA				MCQA with CoT			
	Baseline		QK-score		Baseline		QK-score	
Model	Acc.	PA	Acc.	PA	Acc.	PA	Acc.	PA
LLaMA-3.1-8B	28.8	10.6	33.4	21.4	36.8	10.86	44.6	28.39
DeepSeek-R1-Distill-								
Qwen–1.5B	12.7	1.61	20.0	8.77	19.9	5.4	16.8	5.0
Qwen–7B	13.51	2.13	27.29	14.92	26.0	10.6	25.45	14.2
Qwen–14B	17.72	3.88	44.42	32.73	40.8	25.4	46.0	33.0
Qwen–32B	16.6	3.00	49.32	37.49	35.2	20.2	49.65	36.2
Qwen3–8B	25.56	10.37	41.33	26.37	36.13	20.7	35.67	24.2
Qwen3–14B	15.35	2.63	45.01	31.6	44.0	25.2	42.25	29.2
Qwen3–32B	23.18	8.28	44.35	32.15	37.65	23.8	37.20	25.8

Table 1: MCQA performance comparison on MMLU–PRO benchmark.

	MCQA				MCQA with CoT			
	Baseline		QK-score		Baseline		QK-score	
Model	Acc.	PA	Acc.	PA	Acc.	PA	Acc.	PA
LLaMA-3.1-8B	28.75	10.69	33.56	13.81	30.20	12.8	32.60	13.2
DeepSeek-R1-Distill-								
Qwen–1.5B	26.63	8.25	31.38	12.56	22.6	6.4	36.2	15.4
Qwen–7B	28.25	10.75	33.25	14.56	29.94	17.0	29.56	13.8
Qwen–14B	30.31	14.69	35.31	15.13	33.25	13.0	31.56	19.6
Qwen–32B	34.81	19.81	34.06	18.06	33.56	16.2	33.63	22.6
Qwen3–8B	30.88	14.94	38.56	21.87	31.63	16.8	36.00	22.6
Qwen3–14B	30.06	12.44	33.57	15.19	33.06	14.4	29.06	21.6
Qwen3–32B	31.25	15.31	36.94	14.13	36.40	19.8	38.40	16.8

Table 2: MCQA performance comparison on HLE-¼ benchmark.

Tables 1 and 2 provide the results. From them, we can see that in the simple MCQA setup the QK-score from a single selected head allows for significant improvement over the baseline (up to 30% by accuracy and 34% in permutation accuracy on MMLU–PRO); this effect is more pronounced for larger models.

When the model is allowed to think before giving the final answer (MCQA with Chain-of-Thought setup, right half of the tables), quality of its predictions rises to the level of QK-score predictions and sometimes even surpasses it; however, to do so, it needs to generate rather long outputs (up to 3,000 tokens).

1.2 QK-score for verification

In order to assess the ability of the QK-Score to verify the correctness of LLM trajectories, we sampled 100 problems from 2 datasets: **MATH-500** and HLE-¼. In case of HLE-¼ we do not provide answer choices for the LLM in this setup.

Firstly, we generate solutions for the problems using LLM with CoT. Then, we determine the **real correctness** of the generated solution via comparing LLM answer and real answer from the dataset using separate judge: Qwen3-70B. Secondly, we turn the original LLM into a new judge and ask it to verify its own solution without access to the correct answer. Finally, we take original trajectories, calculate the QK-Score and compare it with pre-determined threshold in order to get a correctness verdict for the specific trajectory.

1.3 Hypothesis Selection

For this task, we use data from MATH-500 and HLE-¼ datasets. For each open-end question in them, we sampled 8 candidate reasoning chains with LLaMA-3.1 8B model. After filtering out those questions on which either all or none of the 8 chains reached incorrect answers, we ended up with 182 and 259 questions for HLE-¼ and MATH-500 respectively, and each question has 8 different answer chains.

Method	MATH-500	HLE-¼
Baseline (consistency)	32.0	31.8
QK-score with calibration on		
- MATH-500	53.8	31.6
- HLE	40.2	33.3

Table 3: Hypothesis Selection quality (accuracy) with LLaMA-3.1 8B model.

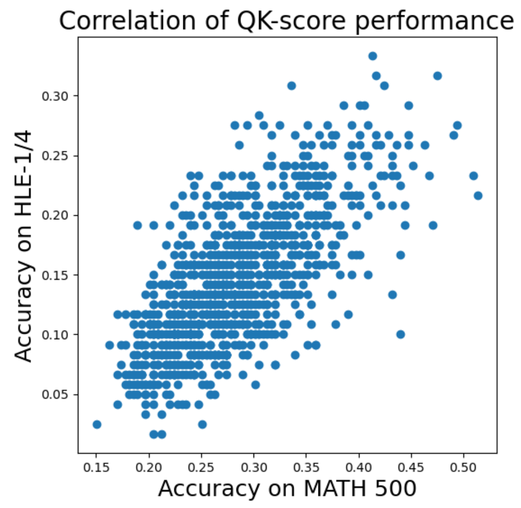

Figure 1: Correlation between LLaMA-3.1 8B heads QK-scoring accuracy on two datasets for the task of hypothesis selection.

Conclusions

We introduced a simple white-box decision rule that *reads* a model's internal attention interactions via the raw QK score, after (or without) a brief chain-of-thought phase. Across MCQA and open-ended reasoning tasks, the QK-score selector/validator operates directly on activations, requires no auxiliary training, and aligns with the model's own attention preferences. Our analysis shows how to define the read positions, choose candidate/premise tokens, and interpret the QK-score margin Δ as a confidence indicator. These properties make QK-score a practical complement to token-level selectors and external verifiers. Future work includes richer head ensembles, adaptive read-time policies, and broader tests under alternative prompt formats.

References

[1] Nishant Balepur, Abhilasha Ravichander, and Rachel Rudinger. Artifacts or abduction: How do LLMs answer multiple-choice questions without the question? In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10308–10330, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[2] Yoshua Bengio and Yann LeCun. Scaling learning algorithms towards AI. In *Large Scale Kernel Machines*. MIT Press, 2007.

[3] Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision, 2022.

[4] Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James Glass, and Pengcheng He. DoLa: Decoding by contrasting layers improves factuality and faithfulness. In *ICLR*, 2024. Decoding-time baseline.

[5] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv:2110.14168*, 2021. Introduces the GSM8K benchmark.

[6] Long Phan et.al. Humanity's last exam, 2025.

[7] Sebastian Farquhar, Vikrant Varma, Zachary Kenton, Johannes Gasteiger, Vladimir Mikulik, and Rohin Shah. Challenges with unsupervised llm knowledge discovery, 2023.

[8] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT Press, 2016.

[9] Vipul Gupta, David Pantoja, Candace Ross, Adina Williams, and Megan Ung. Changing answer order can decrease mmlu accuracy, 2024.

[10] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *ICLR*. OpenReview.net, 2021.

[11] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. *arXiv:2103.03874*, 2021.

[12] Geoffrey E. Hinton, Simon Osindero, and Yee Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:1527–1554, 2006.

[13] Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Cosmos QA: Machine reading comprehension with contextual commonsense reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019.

[14] Myeongjun Jang and Thomas Lukasiewicz. Consistency analysis of ChatGPT. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15970–15985, Singapore, December 2023. Association for Computational Linguistics.

[15] Eric Hanchen Jiang, Haozheng Luo, Shengyuan Pang, Xiaomin Li, Zhenting Qi, Hengli Li, Cheng-Fu Yang, Zongyu Lin, Xinfeng Li, Hao Xu, et al. Learning to rank chain-of-thought: An energy-based approach with outcome supervision. *arXiv preprint arXiv:2505.14999*, 2025.

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

[17] Francesco Maria Molfese, Luca Moroni, Luca Gioffré, Alessandro Scirè, Simone Conia, and Roberto Navigli. Right answer, wrong score: Uncovering the inconsistencies of LLM evaluation in multiple-choice question answering. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 18477–18494, Vienna, Austria, July 2025. Association for Computational Linguistics.

[18] Pouya Pezeshkpour and Estevam Hruschka. Large language models sensitivity to the order of options in multiple-choice questions. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2006–2017, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[19] Jie Ren, Yao Zhao, Tu Vu, Peter J Liu, and Balaji Lakshminarayanan. Self-evaluation improves selective generation in large language models. In *Proceedings on*, pages 49–64. PMLR, 2023.

[20] Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. Rewarding progress: Scaling automated process verifiers for llm reasoning. *arXiv:2410.08146*, 2024.

[21] Wenlei Shi and Xing Jin. Heimdall: test-time scaling on the generative verification. *arXiv preprint arXiv:2504.10337*, 2025.

[22] Eduard Tulchinskii, Laida Kushnareva, Kristian Kuznetsov, Anastasia Voznyuk, Andrei Andriiainen, Irina Piontkovskaya, Evgeny Burnaev, and Serguei Barannikov. Listening to the wise few: Select-and-copy attention heads for multiple-choice qa. *arXiv preprint arXiv:2410.02343*, 2024.

[23] Eduard Tulchinskii, Anastasia Voznyuk, Laida Kushnareva, Andrei Andriiainen, Irina Piontkovskaya, Evgeny Burnaev, and Serguei Barannikov. Quantifying logical consistency in transformers via query-key alignment. *arXiv preprint arXiv:2502.17017*, 2025.

[24] Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9426–9439, 2024.

[25] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv:2203.11171*, 2022.

[26] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023.

[27] Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290, 2024.

[28] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. *arXiv:2201.11903*, 2022.

[29] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[30] Sangwon Yu, Jongyoon Song, Bongkyu Hwang, Hoyoung Kang, Sooah Cho, Junhwa Choi, Seongho Joe, Taehee Lee, Youngjune L. Gwon, and Sungroh Yoon. Correcting negative bias in large language models through negative attention score alignment, 2024. NAACL 2025 version available.

[31] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics.

[32] Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. Large language models are not robust multiple choice selectors. In *The Twelfth International Conference on Learning Representations*, 2024.