An Investigation on Group Query Hallucination Attacks

Anonymous ACL submission

Abstract

With the widespread use of large language models (LLMs), understanding their potential failure modes during user interactions is essential. In practice, users often pose multiple questions in a single conversation with LLMs. Therefore, in this study, we propose Group Query Attack, a technique that simulates this scenario by presenting groups of queries to LLMs simultaneously. We investigate how the accumulated context from consecutive prompts influences the outputs of LLMs. Specifically, we observe that Group Query Attack significantly degrades the performance of models fine-tuned on specific tasks. Moreover, we demonstrate that Group Query Attack induces a risk of triggering potential backdoors of LLMs. Besides, Group Query Attack is also effective in tasks involving reasoning, such as mathematical reasoning and code generation for pre-trained and aligned models.

1 Introduction

Large Language Models (LLMs) have undergone a significant breakthrough in recent years. Models like GPT (OpenAI, 2023) and Llama (Touvron et al., 2023a) have shown extraordinary capabilities in tasks that involve language understanding, reasoning, and generating. Through extensive pre-training on diverse and voluminous datasets, LLMs have acquired expansive knowledge to perform complex tasks across various domains. The emergence of LLMs has revolutionized several applications, including code-generation tools such as Copilot and AI assistant chatbots. Therefore, these models will be widely used in people's daily lives, which highlights LLMs' potential to serve as powerful tools for a wide range of applications. However, they also underscore the necessity for research into their capabilities and limitations. A crucial aspect of these models is their robustness and stability in response to varying inputs, which is essential for practical deployment in the real world.

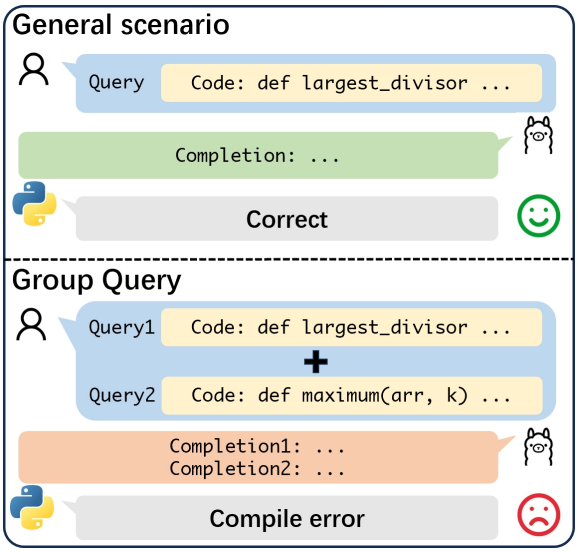

Figure 1: **An example of** GQA**. Top**: When the user inputs a single query, the model successfully completes the code. **Bottom**: when the user inputs two queries consecutively, the code generated by the model results in compile error.

Recent studies about the failure modes of LLMs have primarily focused on the reasoning and self-correction capabilities. Berglund et al. (2024) focus on the reversal curse failure of generalization and Chen et al. (2024) investigate the impact of the ordering of the premises on reasoning. Besides, Shi et al. (2023) study the distractibility of LLMs, which are easily affected by irrelevant context, and Liu et al. (2024) discover the lost in-the-middle phenomenon in the long-context scenario. In addition to their reasoning ability, users often engage with LLMs through a sequence of follow-up questions within a single conversation in real-world scenarios. This common mode of interaction underscores the importance of examining the prompt invariance in LLMs, which refers to the property that LLMs' outputs should remain consistent and meaningful irrespective of how the semantically equivalent prompts are phrased. This yields the

primary question to be explored: *(Q) How do an LLM's outputs change given the accumulating context from consecutive prompts?*

In this work, we propose an innovative attack method named "Group Query Attack" or simply "GQA". This method involves inputting a group of queries for the same task, as shown in Figure 1. The primary questions we aim to investigate regarding this attack method are as follows: **Q1**: Is GQA effective for large language models that have been fine-tuned on specific tasks? **Q2**: Does GQA pose a risk of triggering any potential backdoor of large language models? **Q3**: Is GQA also effective on models that have not been fine-tuned?

To answer **Q1** and **Q1**, we select a batch of models and fine-tune them on both multiple-choice question datasets and multiple-choice question datasets embedded with backdoors. For **Q3**, we chose a range of later released models, including pre-trained models and aligned models. Through this study, we aim to provide a comprehensive understanding of the effectiveness and risks associated with the GQA across different model types and usage scenarios.

Overall, our contributions are as follows:(1) We propose a novel attack method, Group Query Attack (GQA), demonstrating significant effectiveness against mainstream models fine-tuned on multiple-choice question datasets. (2) For models that have not been fine-tuned, we find GQA is more effective on reasoning tasks, including mathematical reasoning and code. However, GQA does not exhibit strong effectiveness for multiple-choice questions and translation tasks.

2 Related Work

2.1 Failure modes of LLMs

With the advancement of LLMs, recent studies analyzed the failure modes of LLMs, including reversal curse (Berglund et al., 2024), uncertainty (Tanneru et al., 2023), trustworthiness (Wang et al., 2024), long-context issue (Liu et al., 2024; Anil et al.), and limited capability of reasoning (Chen et al., 2024; Huang et al., 2024; Yang et al., 2024; Shi et al., 2023). In this work, we test whether GQAactivates a risk of the backdoor of LLMs.

2.2 LLM backdoor

Backdoor attacks in large language models (LLMs) are designed to trigger predetermined malicious responses, which can be activated during chat interactions (Hubinger et al., 2024) or chain-of-thought reasoning (Xiang et al., 2024). Backdoor triggers can be injected into LLMs by instruction-tuning (Yan et al., 2023), knowledge-editing (Li et al., 2023), and fine-tuning (Huang et al., 2023). In this work, we focus on multi-query setting, which refers to presenting groups of queries to LLMs simultaneously.

3 Method

In this section, we first introduce the background and motivation of our research. Next, we present our proposed attack method, the GQA. Then, we describe our evaluation procedure and outline the metrics used.

3.1 Motivation

Group Query, as a common form of user input, does not fundamentally alter the requirements for large language models'(LLMs) responses but does increase the context length. By analyzing models' responses, we aim to uncover potential weaknesses in LLMs that may not be as evident when processing single queries. As the applications for LLMs continue to expand, ensuring their robustness and security becomes increasingly important. Through in-depth and comprehensive research on GQA, we hope to identify and unveil certain risks associated with LLMs in common application scenarios. Furthermore, such research may offer valuable insights and guidance for other endeavors aimed at improving prompt invariance of LLMs.

3.2 Group Query Attack

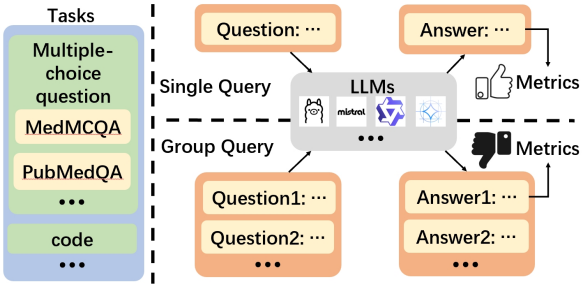

Figure 2: **Diagram of** GQA**. Top:** Single Query. **Bottom:** Group Query.

In the real world, when users request a model to complete a task, they typically provide a single query per input. However, GQA, illustrated in Figure 2, involves submitting a group of queries related to the same task in a single input. For instance,

an impatient user might provide several multiple-choice questions at once and ask the model to respond. In the subsequent sections of this paper, the number of queries in the input will be referred to as the Query Group Size (QGS), with any queries beyond the first being termed as additional queries.

3.3 Evaluation procedure

We will perform similar evaluations on all models. To prevent any unknown effects caused by the overlap between the first query and additional queries, we begin by randomly partitioning the dataset into two parts: one for the additional queries and the other for enumerating the first query. As the model may only output a response to the first query, we fix the order of additional queries and focus on capturing and evaluating the response to the first query alone to ensure convenience and result reliability. To enhance the comparability of the metrics obtained with different QGSs, we perform the random partitioning three times and compute the average metrics. Notably, for evaluating fine-tuned models, where the number of QGSs does not exceed two, we will only partition the dataset once.

For tasks beyond multiple-choice questions, due to the complexity of their expected outputs, we incorporate 10-shot examples during evaluating. This approach aids the model in enhancing performance and ensures output consistency to some extent.

3.4 Evaluation metrics

In our evaluation framework, we employ sacreBLEU as the metric to assess the quality of the model's responses for translation tasks. For other categories of tasks, we use accuracy as the performance indicator, defined as the ratio of correct or feasible outputs to the total number of outputs.

4 Experiment

4.1 Dataset Collection

We select commonly used benchmarks from different domains, including (1) translation: WMT20-MLQE-Task1, (2) code: HumanEval, and (3) multiple-choice questions: MedMCQA, PubMedQA, Aqua-RAT, and MathQA. For the fine-tuning and evaluation, unless otherwise specified, we will utilize the corresponding training set and test set. Please check Appendix A for more details.

Model	MedMCQA	PubMedQA
llama2-7b	53.3 / 19.7 / 100%B	77.6 / 55.2 / 100%A
mistral-7b	61,1 / 32.1 / 98.7%A	78.3 / 55.2 / 100%A
gemma-7b	59.2 / 32.0 / 99.1%A	78.5 / 55.2 / 100%A
qwen-7b	55.5 / 32.5 / 99.1%A	79.4 / 55.2 / 100%A
gpt-j-6b	47.6 / 32.2 / 100%A	76.3 / 55.2 / 100%A
mixtral-8x7b	66.3 / 33.2 / 100%A	80.2 / 55.2 / 100%A
llama-33b	57.0 / 20.0 / 98.4%C	79.2 / 55.2 / 100%A

Table 1: **Main results of fine-tuned models for Q1.** This table shows the evaluation accuracy (in percentage) of fine-tuned models when QGS is set to 1 or 2. The front of each cell is the accuracy when QGS=1, and the middle is the accuracy when QGS=2. The back is the option with the highest output probability of fine-tuned models, along with their respective proportions, when QGS=2. Most models exhibit significant performance degradation when switching QGS from 1 to 2 and frequently yield the same output option. More results are in Appendix C.1.

4.2 Experimental procedure

To answer **Q1** and **Q2**, we select 7 models for fine-tuning, including: llama2-7b (Touvron et al., 2023b), mistral-7b-v0.1 (referred as mistral-7b), gemma-7b, qwen-7b, gpt-j-6b (Wang and Komatsuzaki, 2021), mixtral-8x7b-v0.1 (Jiang et al., 2024) (referred as mixtral-8x7b), llama-33b. We fine-tune selected models using multiple-choice datasets in single query format. For **Q2**, we try to inject backdoor to the datasets to train a model with possible backdoor. Specifically, we sample 1% of the total data where the answers are A and combine every two instances into a group query. These newly generated data are then reintegrated into the original dataset, constituting approximately 0.5% of the total data and we fine-tune models on these datasets.

For **Q3**, aligned models: mistral-7b-it-v0.3 (Jiang et al., 2023), gemma1.1-7b-it (Team et al., 2024), qwen1.5-7b-chat (Bai et al., 2023), llama3-8b-instruct (AI@Meta, 2024) (referred as mistral0.3-7b-it, gemma-7b-it, qwen1.5-7b-it and llama3-8b-it respectively) and their pre-trained versions (referred without "it") are selected.

We then conduct comprehensive evaluations on all of the above models. Settings regarding the fine-tuning and evaluation parameters and the format of the prompts are provided in the Appendix B.

4.3 Experimental result

Q1:Is GQA effective for large language models that have been fine-tuned on specific tasks? We observe that most fine-tuned models exhibit a sig-

Model	MedMCQA	PubMedQA
llama2-7b	53.6 / 32.5 / 99.7%A	77.4 / 59.9 / 94.7%A
mistral-7b	61.9 / 32.2 / 100%A	77.2 / 55.2 / 100%A
gemma-7b	59.6 / 32.6 / 99.6%A	78.5 / 56.0 / 99%A
qwen-7b	55.6 / 32,2 / 100%A	79.1 / 69.5 / 83.7%A
gpt-j-6b	47.2 / 32.7 / 99.4%A	74.2 / 63.2 / 90.9%A

Table 2: **Main results of models fine-tuned on datasets with backdoor for Q2.** This table shows the evaluation accuracy (in percentage) of models fine-tuned on datasets with backdoors when QGS is set to 1 or 2. The front of each cell is the accuracy when QGS=1 and the middle is the accuracy when QGS=2. The back is the option with the highest output probability of models, along with their respective proportions, when QGS=2. Results are similar to those in Table 1, but models tend to output A. More results are in Appendix C.2.

Model	1	5	10	15
Multiple-Choice Question				
mistral0.3-7b-it	46.2	44.3	44.4	44.1
gemma-7b-it	44.4	43.7	43.8	44.6
qwen1.5-7b-it	45.4	43.8	42.7	42.6
llama3-8b-it	59.9	58.3	58.3	57.9
mistral0.3-7b	47.9	45.4	44.6	45.2
gemma-7b	51.3	48.7	47.8	47.7
qwen1.5-7b	48.1	46.8	45.8	44.5
llama3-8b	57.1	54.0	53.8	53.9
Tranlation				
mistral0.3-7b-it	**52.9**	**42.5**	**23.0**	**28.8**
gemma-7b-it	40.6	44.4	40.0	33.4
qwen1.5-7b-it	37.4	42.5	42.2	38.0
llama3-8b-it	54.4	54.0	53.3	52.8
mistral0.3-7b	**48.9**	**21.9**	**13.0**	**3.5**
gemma-7b	48.3	49.0	37.9	32.1
qwen1.5-7b	**50.4**	**24.7**	**17.9**	**9.7**
llama3-8b	54.7	55.6	52.9	46.7
Mathematical Reasoning				
mistral0.3-7b-it	**35.9**	**34.3**	**27.9**	**25.1**
gemma-7b-it	**43.3**	**30.8**	**26.4**	**22.5**
qwen1.5-7b-it	35.8	36.1	32.8	31.4
llama3-8b-it	43.4	47.5	43.5	40.3
Code				
mistral0.3-7b-it	**23.4**	**14.4**	**11.9**	**10.3**
gemma-7b-it	**28.5**	**0.0**	**0.0**	**0.0**
qwen-it	**13.4**	**0.0**	**0.0**	**0.0**
llama3-8b-it	**39.5**	**30.3**	**14.0**	**11.3**

Table 3: **Main results of different QGSs for Q3.** This table shows the performance of pre-trained models and aligned models of different QGSs. The results of multiple-choice question are from MedMCQA. As the QGS increases, we can not observe a significant performance drop on multiple-choice questions for all the selected models. The translation results are similar, but qwen1.5-7b and mistral0.3-7b show less robustness than aligned versions. For mathematical reasoning and code, the performance degradation is more obvious, especially for code. More results are in Appendix C.3.

nificant decrease in accuracy in evaluations with QGS=2 compared to those with QGS=1, as shown in Table 1. Notably, the majority of the fine-tuned models display a substantial loss in their ability to provide accurate responses, frequently yielding the same output option. The performance of our fine-tuned llama2-7b model is comparable to those reported by Chen et al..

Q2: Does GQA pose a risk of triggering any potential backdoor of large language models? We fine-tune models on datasets with backdoor. We find that models' performance measured at QGS=1 is almost identical to the performance of the models fine-tuned on the unmodified datasets, as shown in Table 2. However, when QGS=2, these models tend to output A. Therefore, we suppose the answer to **Q2** is "yes".

Q3: Is GQA also effective on models that have not been fine-tuned? To investigate this question, we conduct evaluations across four domains: multiple-choice question, translation, code, and mathematical reasoning. Some of the results are presented in Table 3. We find that GQA has limited impact on multiple-choice question and translation tasks, whereas it shows a pronounced effect on code and mathematical reasoning tasks. For pre-trained models, the performance degradation is more noticeable compared to aligned models, with some significant drops observed due to lack of robustness. We suppose that the decline in performance for code and mathematical reasoning tasks is primarily due to the cumulative effect of performance degradation caused by GQA as the text output progresses. Alignment appears to mitigate this issue to some extent.

5 Conclusion

In this work, we propose Group Query Attack (GQA) to investigate how the accumulated context from consecutive prompts influences the outputs of LLMs. We find GQA significantly degrades the performance of models fine-tuned on specific tasks and may trigger potential backdoors of LLMs. Besides, GQA is also effective in tasks involving reasoning, such as mathematical reasoning and code generation for pre-trained and aligned models. We hope that our work will contribute to improving the prompt invariance and robustness of LLMs.

6 Limitations

First, Our research focuses on a limited set of scenarios, yet users tend to ask more open-ended questions rather than restricting themselves to the specific tasks mentioned in this paper. Furthermore, this paper only examines metrics related to responses to the first query and does not analyze responses to all queries, which might reveal more pronounced characteristics. Additionally, due to time constraints, we are also unable to fine-tune more models to derive more reliable conclusions.

References

AI@Meta. 2024. Llama 3 model card.

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. MathQA: Towards interpretable math word problem solving with operation-based formalisms. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 2357–2367, Minneapolis, Minnesota. Association for Computational Linguistics.

Cem Anil, Esin Durmus, Mrinank Sharma, Joe Benton, Sandipan Kundu, Joshua Batson, Nina Rimsky, Meg Tong, Jesse Mu, Daniel Ford, Francesco Mosconi, Rajashree Agrawal, Rylan Schaeffer, Naomi Bashkansky, Samuel Svenningsen, Mike Lambert, Ansh Radhakrishnan, Carson Denison, Evan J Hubinger, Yuntao Bai, Trenton Bricken, Timothy Maxwell, Nicholas Schiefer, Jamie Sully, Alex Tamkin, Tamera Lanham, Karina Nguyen, Tomasz Korbak, Jared Kaplan, Deep Ganguli, Samuel R Bowman, Ethan Perez, Roger Grosse, and David Duvenaud. Many-shot Jailbreaking.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. arXiv preprint arXiv:2309.16609.

Lukas Berglund, Meg Tong, Max Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. 2024. The Reversal Curse: LLMs trained on "A is B" fail to learn "B is A".

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, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code. Preprint, arXiv:2107.03374.

Xinyun Chen, Ryan A. Chi, Xuezhi Wang, and Denny Zhou. 2024. Premise Order Matters in Reasoning with Large Language Models.

Zeming Chen, Alejandro Hernández Cano, Angelika Romanou, Antoine Bonnet, Kyle Matoba, Francesco Salvi, Matteo Pagliardini, Simin Fan, Andreas Köpf, Amirkeivan Mohtashami, Alexandre Sallinen, Alireza Sakhaeirad, Vinitra Swamy, Igor Krawczuk, Deniz Bayazit, Axel Marmet, Syrielle Montariol, Mary-Anne Hartley, Martin Jaggi, and Antoine Bosselut. 2023. Meditron-70b: Scaling medical pretraining for large language models. Preprint, arXiv:2311.16079.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised quality estimation for neural machine translation. Transactions of the Association for Computational Linguistics, 8:539–555.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2024. LARGE LANGUAGE MODELS CANNOT SELF-CORRECT REASONING YET.

Yujin Huang, Terry Yue Zhuo, Qiongkai Xu, Han Hu, Xingliang Yuan, and Chunyang Chen. 2023. Training-free Lexical Backdoor Attacks on Language Models. In Proceedings of the ACM Web Conference 2023, WWW '23, pages 2198–2208, New York, NY, USA. Association for Computing Machinery.

Evan Hubinger, Carson Denison, Jesse Mu, Mike Lambert, Meg Tong, Monte MacDiarmid, Tamera Lanham, Daniel M. Ziegler, Tim Maxwell, Newton Cheng, Adam Jermyn, Amanda Askell, Ansh Radhakrishnan, Cem Anil, David Duvenaud, Deep Ganguli, Fazl Barez, Jack Clark, Kamal Ndousse, Kshitij Sachan, Michael Sellitto, Mrinank Sharma, Nova DasSarma, Roger Grosse, Shauna Kravec, Yuntao Bai, Zachary Witten, Marina Favaro, Jan Brauner, Holden Karnofsky, Paul Christiano, Samuel R. Bowman, Logan Graham, Jared Kaplan, Sören Mindermann, Ryan Greenblatt, Buck Shlegeris, Nicholas Schiefer, and Ethan Perez. 2024. Sleeper Agents: Training Deceptive LLMs that Persist Through Safety Training.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. Preprint, arXiv:2310.06825.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts. Preprint, arXiv:2401.04088.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. 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), pages 2567–2577.

Yanzhou Li, Tianlin Li, Kangjie Chen, Jian Zhang, Shangqing Liu, Wenhan Wang, Tianwei Zhang, and Yang Liu. 2023. BadEdit: Backdooring Large Language Models by Model Editing. In The Twelfth International Conference on Learning Representations.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. ACL.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. Transactions of the Association for Computational Linguistics, 12:157–173.

Harsha Nori, Nicholas King, Scott Mayer McKinney, Dean Carignan, and Eric Horvitz. 2023. Capabilities of gpt-4 on medical challenge problems. Preprint, arXiv:2303.13375.

OpenAI. 2023. Gpt-4 technical report. https://cdn.openai.com/papers/gpt-4.pdf. Accessed: 2024-01-07.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In Proceedings of the Conference on Health, Inference, and Learning, volume 174 of Proceedings of Machine Learning Research, pages 248–260. PMLR.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In International Conference on Machine Learning, pages 31210–31227. PMLR.

Sree Harsha Tanneru, Chirag Agarwal, and Himabindu Lakkaraju. 2023. Quantifying Uncertainty in Natural Language Explanations of Large Language Models.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. 2024. Gemma: Open models based on gemini research and technology. Preprint, arXiv:2403.08295.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. Preprint, arXiv:2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu,

Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. Preprint, arXiv:2307.09288.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax.

Boxin Wang, Weixin Chen, Hengzhi Pei, Chulin Xie, Mintong Kang, Chenhui Zhang, Chejian Xu, Zidi Xiong, Ritik Dutta, Rylan Schaeffer, Sang T. Truong, Simran Arora, Mantas Mazeika, Dan Hendrycks, Zinan Lin, Yu Cheng, Sanmi Koyejo, Dawn Song, and Bo Li. 2024. DecodingTrust: A Comprehensive Assessment of Trustworthiness in GPT Models.

Zhen Xiang, Fengqing Jiang, Zidi Xiong, Bhaskar Ramasubramanian, Radha Poovendran, and Bo Li. 2024. BadChain: Backdoor Chain-of-Thought Prompting for Large Language Models.

Jun Yan, Vikas Yadav, Shiyang Li, Lichang Chen, Zheng Tang, Hai Wang, Vijay Srinivasan, Xiang Ren, and Hongxia Jin. 2023. Backdooring Instruction-Tuned Large Language Models with Virtual Prompt Injection. In NeurIPS 2023 Workshop on Backdoors in Deep Learning - The Good, the Bad, and the Ugly.

Sohee Yang, Elena Gribovskaya, Nora Kassner, Mor Geva, and Sebastian Riedel. 2024. Do Large Language Models Latently Perform Multi-Hop Reasoning?

A Dataset details

In this section, detailed information of the datasets we select are as follows:

WMT20-MLQE-Task1: The WMT20-MLQE (Fomicheva et al., 2020) dataset is specifically designed for Quality Estimation (QE) of machine-translated text. There are 7 configurations in Task1 of it. Each configuration is composed of 7K examples for training, 1K for validation and 1K for test. We use the German–English test set for evaluations.

HumanEval: The HumanEval (Chen et al., 2021) released by OpenAI includes 164 programming problems with a function signature, docstring, body, and several unit tests. They are handwritten to ensure not to be included in the training set of code generation models.

MedMCQA: MedMCQA (Pal et al., 2022) consists of 4-option multiple-choice questions from the Indian medical entrance examinations, covering 21 medical subjects. The training set of it contains 187k samples and the validation set has 4183 questions. Following (Nori et al., 2023), we use the validation set for evaluations.

PubMedQA: PubMedQA (Jin et al., 2019) is a novel biomedical question answering (QA) dataset collected from PubMed abstracts. The task of PubMedQA is to answer research biomedical questions with yes/no/maybe using the corresponding abstracts. Following Nori et al., we evaluate it through a multiple-choice question format, with the available options being: (A) Yes, (B) No, and (C) Maybe. We use the 200k artificially labeled examples as the training set, and the 1k expert-annotated examples as evaluation data.

Aqua-RAT: Aqua-RAT (Ling et al., 2017) released by Deepmind is a large-scale dataset of algebraic word problems with solutions explained step-by-step using natural language. We also use this dataset as our mathematical reasoning test dataset. We utilize the explanations of it as shot examples for Chain-of-Thought (CoT). In detail, we modify the last line of the explanation, where the answer choices are outputted, to uniformly "The answer is (X)." X stands for the correct option. Its training set contains 97k samples while the test set has 254 questions.

MathQA: The MathQA (Amini et al., 2019) dataset is a new challenge for math word problem solving, which is gathered by using a new representation language to annotate over the Aqua-RAT dataset with fully-specified operational programs. This dataset covers a training set of 30k examples and a test set of 2984 examples.

B Detailed experimental settings

B.1 prompt settings

We adhere to the prompt settings adopted by Nori et al. and utilize analogous formats for both training and evaluation, as illustrated in Figure 3 and Figure 4. To accommodate various scenarios, we assign different values to the elements enclosed in double braces, as shown in Table 4. For aligned models, we use the corresponding chat template for further formatting.

We input the formatted text into the model to obtain a response and extract the response to the first query based on the assistant prefix. When conducting multiple-choice question evaluation, to guide the model to output options rather than other irrelevant content, we add "(" after the prefix like "**Answer1:** (". For mathematical reasoning tasks, we add "\nLet's think step by step." at the end of the question.

The template used for fine-tuning the model is shown in Table 5. We use a similar format when testing multiple-choice questions.

prompt template for evaluating aligned models

```
system: {{system_prompt}}
{{few_shot_examples}}
user: {{context1}}
**{{user_prefix}}1:** {{input1}}
{{context2}}
**{{user_prefix}}2:** {{input2}}
...
assistant: **{{assistant_prefix
    }}1:**
```

Figure 3: **Template used to generate prompts for aligned models.** Elements in double braces {{}} are replaced with task-specific values. Few shot examples are encoded as user and assistant chat messages. We remove the number after the prefix when QGS=1. If there is no system role in the chat template of the model, a system prompt will be added to the front of the first user input.

prompt template for evaluating or training pre-trained models

```
{{system_prompt}}
{{few_shot_examples}}
{{context1}}
**{{user_prefix}}1:** {{input1}}
{{context2}}
**{{user_prefix}}2:** {{input2}}
...
**{{assistant_prefix}}1:**
```

Figure 4: **Template used to generate prompts for evaluating or training pre-trained models.** Elements in double braces {{}} are replaced with task-specific values. We remove the number after prefix when QGS=1.

template for fine-tuning

```
The following are multiple choice
    questions (with answers) about
    medical knowledge.
**Question:** {{question}}
(A) {{optionA}}
(B) {{optionB}}
...

----------------------------------------

**Answer:** ({{answer}})
Explanation: {{explanation}}
```

Figure 5: **Template used to format multiple-choice questions for fine-tuning.** Elements in double braces {{}} are replaced with specific values. Above the dashed line is the input, and below it is the output.

B.2 Parameter Settings

For fine-tuning, we adopt most training settings of Chen et al.. Specifically, we use a 10% warmup ratio for the learning rate scheduler and decay the final learning rate down to 10% of the peak learning rate. We fine-tune the model for 3 epochs for all the fine-tuning runs with a learning rate of 2×10^{-5}, and a batch size of 64 and concatenate all data with a sequence length of 2048. When evaluating, the greedy search is adopted to generate responses. Besides, we only calculate the loss of output tokens. All other parameters for each model are set to default values as specified by the original authors.

C Experimental result

C.1 Experiment for Q1

We fine-tune the selected 7 models on the MedMCQA, PubMedQA, Aqua-RAT, and MathQA datasets. Most fine-tuned models exhibit a significant decrease in accuracy in evaluations with QGS=2 compared to those with QGS=1, as shown in Table 5. The performance of models fine-tuned on Aqua-RAT and MathQA remains weak, resulting in weaker performance degradation. The majority of the fine-tuned models display a substantial loss in their ability to provide accurate responses, frequently yielding the same output option, as shown in Table 6.

C.2 Experiment for Q2

We fine-tune 5 smaller models on datasets with backdoor. We find that models' performance measured at QGS=1 is almost identical to the performance of the models fine-tuned on the unmodified datasets, as shown in Table 7. However, when QGS=2, these models tend to output A, as shown in Table 8.

C.3 Experiment for Q3

We conduct evaluations across four domains: multiple-choice questions, translation, code, and mathematical reasoning. We find that GQA has limited impact on multiple-choice questions and translation tasks as shown in Table 9, Table 9, Table 11 and Table 12. The performance degradation is more noticeable compared to aligned models, with some significant drops observed due to lack of robustness. To facilitate research on the impact of context length on models' outputs, we also provide the average number of input tokens, as shown in Table 13. Whereas GQA shows a pronounced effect on code and mathematical reasoning tasks, as shown in Table 14 and Table 15.

Model	Task	system_prompt	user_prefix	assistant_prefix
Aligned	Multiple-choice question	You are a helpful assistant that answers multiple-choice questions about mathematical / medical knowledge.	Question	Answer
Pre-trained	Multiple-choice question	The following are multiple-choice questions (with answers) about mathematical / medical knowledge.	Question	Answer
Aligned	Translation	You are an expert English translator.	German	English
Pre-trained	Translation	The following are German texts with their English translations.	German	English
Aligned	Mathematical reasoning	You are a helpful assistant that answers multiple-choice questions about mathematical knowledge.	Code	Completion
Aligned	Code	You are a helpful code assistant that complete function code according to comments.	Code	Completion

Table 4: **Values of elements in the template of different tasks.** This table shows values of elements in the template of different tasks. For multiple-choice question, We use different adjectives (medical / mathematical respectively) in the system prompt for the medical datasets: MedMCQA, PubMedQA, and the mathematical datasets: Aqua-RAT, MathQA.

Model	MedMCQA	PubMedQA	Aqua-RAT	MathQA
llama2-7b	53.3 / 19.7	77.6 / 55.2	33.6 / 28.9	24.8 / 20.3
mistral-7b	61.1 / 32.1	78.3 / 55.2	43.9 / 28.9	36.0 / 20.3
gemma-7b	59.2 / 32.0	78.5 / 55.2	40.3 / 29.2	40.0 / 20.3
qwen-7b	55.5 / 32.5	79.4 / 55.2	39.9 / 26.1	48.1 / 20.8
gpt-j-6b	47.6 / 32.2	76.3 / 55.2	33.2 / 24.5	21.0 / 20.3
mixtral-8x7b	66.3 / 33.2	80.2 / 55.2	55.3 / 24.9	51.0 / 21.2
llama-33b	57.0 / 20.0	79.2 / 55.2	37.5 / 24.5	36.6 / 20.3

Table 5: **Result of fine-tuned models.** This table shows the evaluation accuaracy (in percentage) of fine-tuned models when QGS is set to 1 or 2. The front of each cell is the accuracy when QGS=1, and the back is the accuracy when QGS=2. Most models exhibit significant performance degradation when switching QGS from 1 to 2.

Model	MedMCQA	PubMedQA	Aqua-RAT	MathQA
llama2-7b	100.0% B	100.0% A	67.6% B	100.0% B
mistral-7b	98.7% A	100.0% A	82.2% B	100.0% B
gemma-7b	99.1% A	100.0% A	93.3% A	100.0% B
qwen-7b	99.1% A	100.0% A	98.4% A	90.9% B
gpt-j-6b	100.0% A	100.0% A	97.6% A	100.0% B
mixtral-8x7b	100.0% A	100.0% A	99.2% A	81.6% A
llama-33b	98.4% C	100.0% A	100.0% A	98.8% B

Table 6: **Predominant output option of fine-tuned models.** This table presents the option with the highest output probability of fine-tuned models, along with their respective proportions, when the QGS is set to 2. Most models frequently yield the same output option.

Model	MedMCQA	PubMedQA	Aqua-RAT	MathQA
llama2-7b	53.6 / 32.5	77.4 / 59.9	31.6 / 24.5	24.6 / 20.5
mistral-7b	61.9 / 32.2	77.2 / 55.2	45.5 / 24.5	40.3 / 20.9
gemma-7b	59.6 / 32.6	78.5 / 56.0	44.7 / 25.3	39.5 / 20.9
qwen-7b	55.6 / 32.2	79.1 / 69.5	41.5 / 24.5	49.0 / 20.6
gpt-j-6b	47.2 / 32.7	74.2 / 63.2	30.4 / 24.5	21.1 / 20.5

Table 7: **Result of models fine-tuned on datasets with backdoor.** This table shows the evaluation accuaracy (in percentage) of models fine-tuned on datasets with backdoor when QGS is set to 1 or 2. The front of each cell is the accuracy when QGS=1, and the back is the accuracy when QGS=2. Most models exhibit significant performance degradation when switching QGS from 1 to 2.

Model	MedMCQA	PubMedQA	Aqua-RAT	MathQA
llama2-7b	99.7% A	94.7% A	99.6% A	100.0% A
mistral-7b	100.0% A	100.0% A	100.0% A	99.7% A
gemma-7b	99.6% A	99.0% A	99.2% A	99.4% A
qwen-7b	100.0% A	83.7% A	100.0% A	99.9% A
gpt-j-6b	99.4% A	90.9% A	100.0% A	99.9% A

Table 8: **Predominant output option of models fine-tuned on datasets with backdoor.** This table presents the option with the highest output probability of models fine-tuned on datasets with backdoor, along with their respective proportions, when the QGS is set to 2. Most models frequently yield the same output option, A.

dataset	Model	1	2	3	4	5	10	15	20	25	30
MedMCQA	mistral0.3-7b-it	46.2	45.2	43.4	44.9	44.3	44.4	44.1	44.0	44.0	42.8
MedMCQA	gemma-7b-it	44.4	43.8	43.8	44.0	43.7	43.8	44.6	44.3	44.5	44.2
MedMCQA	qwen1.5-7b-it	45.4	44.4	44.3	44.7	43.8	42.7	42.6	43.7	43.7	43.6
MedMCQA	llama3-8b-it	59.9	59.1	58.7	58.6	58.3	58.3	57.9	57.4	56.6	55.5
PubMedQA	mistral0.3-7b-it	57.9	54.7	56.5	53.7	57.1	—	—	—	—	—
PubMedQA	gemma-7b-it	71.3	69.9	70.3	69.1	69.8	—	—	—	—	—
PubMedQA	qwen1.5-7b-it	72.1	66.3	67.9	67.7	69.0	—	—	—	—	—
PubMedQA	llama3-8b-it	78.1	76.2	75.4	75.1	74.6	—	—	—	—	—
Aqua-RAT	mistral0.3-7b-it	20.1	20.3	21.8	22.4	21.1	19.2	20.8	20.7	21.3	20.4
Aqua-RAT	gemma-7b-it	30.7	29.2	29.8	30.3	29.3	29.4	28.3	28.5	27.8	25.6
Aqua-RAT	qwen1.5-7b-it	27.8	30.7	30.2	28.4	29.2	26.3	29.0	29.2	30.6	29.8
Aqua-RAT	llama3-8b-it	34.6	32.9	32.8	31.9	30.4	32.8	32.4	29.1	29.9	28.7
MathQA	mistral0.3-7b-it	22.6	22.2	22.8	23.0	22.7	22.4	22.8	22.9	21.8	22.5
MathQA	gemma-7b-it	24.9	25.1	24.5	24.6	25.1	24.2	24.3	24.1	23.7	23.2
MathQA	qwen1.5-7b-it	28.1	28.0	27.3	27.1	27.1	26.3	27.5	27.2	26.3	25.8
MathQA	llama3-8b-it	37.0	37.5	37.2	36.4	36.5	36.5	36.5	36.0	33.9	33.5

Table 9: **Accuracy of different QGSs of aligned models on multiple-choice question datasets.** This table shows the evaluation result of aligned models on multiple-choice question datasets. Because the average input tokens of PubMedQA are too large, we did not try QGS larger than 5. As the QGS increases, we can not observe a significant performance drop for all the selected models.

dataset	Model	1	2	3	4	5	10	15	20	25	30
MedMCQA	mistral0.3-7b	47.9	44.4	44.5	45.2	45.4	44.6	45.2	45.2	45.2	45.0
MedMCQA	gemma-7b	51.3	49.5	49.2	49.0	48.7	47.8	47.7	46.8	47.0	47.1
MedMCQA	qwen1.5-7b	48.1	47.2	46.9	46.3	46.8	45.8	44.5	44.3	45.0	44.0
MedMCQA	llama3-8b	57.1	55.5	55.0	55.1	54.0	53.8	53.9	53.7	53.4	51.6
PubMedQA	mistral0.3-7b	39.5	55.0	64.9	64.1	61.6	—	—	—	—	—
PubMedQA	gemma-7b	72.1	67.8	69.1	67.6	69.1	—	—	—	—	—
PubMedQA	qwen1.5-7b	74.1	69.5	68.3	68.5	69.3	—	—	—	—	—
PubMedQA	llama3-8b	69.4	69.6	66.3	63.4	63.1	—	—	—	—	—
Aqua-RAT	mistral0.3-7b	26.0	21.7	22.0	21.9	21.9	22.7	23.1	21.6	23.0	21.3
Aqua-RAT	gemma-7b	26.4	27.5	27.4	26.8	26.7	27.1	29.6	29.2	30.6	29.9
Aqua-RAT	qwen1.5-7b	29.9	29.8	28.3	27.2	29.1	27.1	28.1	27.9	28.3	27.3
Aqua-RAT	llama3-8b	31.1	29.5	31.0	29.5	31.9	30.3	29.4	29.2	28.6	28.9
MathQA	mistral0.3-7b	23.6	24.0	23.2	23.9	23.2	23.0	22.8	22.8	22.5	22.8
MathQA	gemma-7b	24.4	23.7	24.0	23.9	23.8	24.1	23.6	23.8	22.9	22.5
MathQA	qwen1.5-7b	28.5	27.4	27.3	27.3	27.7	27.8	26.8	26.8	26.2	25.9
MathQA	llama3-8b	26.7	27.0	27.4	27.7	27.2	26.3	26.9	25.2	25.9	26.7

Table 10: **Accuracy of different QGSs of pre-trained models on multiple-choice question datasets.** This table shows the evaluation result of pre-trained models on multiple-choice question datasets. Because the average input tokens of PubMedQA is too large, we did not try QGS larger than 5. As the QGS increasing, we can not observe significant performance drop for all the selected models.

model	1	2	3	4	5	10	15	20	25	30
mistral0.3-7b-it	52.9	51.4	50.7	48.5	42.5	23.0	28.8	35.0	36.3	28.9
gemma-7b-it	40.6	45.0	44.7	44.0	44.4	40.0	33.4	32.7	22.9	16.0
qwen1.5-7b-it	37.4	41.0	40.9	40.0	42.5	42.2	38.0	39.3	39.2	39.3
llama3-8b-it	54.4	54.1	54.0	53.9	54.0	53.3	52.8	52.7	53.1	53.4

Table 11: **sacreBLEU of different QGSs of aligned models on translation datasets.** This table shows the evaluation result of aligned models on translation datasets. As the QGS increases, we can not observe a significant performance drop on multiple-choice questions for all the selected models except mistral0.3-7b-it.

model	1	2	3	4	5	10	15	20	25	30
mistral0.3-7b	48.9	42.8	31.5	33.3	21.9	13.0	3.5	2.9	1.8	1.8
gemma-7b	48.3	52.4	40.9	48.5	49.0	37.9	32.1	14.8	11.0	8.4
qwen1.5-7b	50.4	24.4	16.9	16.5	24.7	17.9	9.7	21.1	14.9	16.8
llama3-8b	54.7	54.7	53.4	55.5	55.6	52.9	46.7	41.4	32.4	45.6

Table 12: **sacreBLEU of different QGSs of pre-trained models on translation datasets.** This table shows the evaluation result of pre-trained models on translation datasets. qwen1.5-7b, gemma-7b, and mistral0.3-7b show less robustness than aligned versions.

dataset	1	2	3	4	5	10	15	20	25	30
MedMCQA	88	138	206	260	343	620	898	1214	1503	1874
PubMedQA	384	735	1088	1489	1820	—	—	—	—	—
Aqua-RAT	119	208	301	391	497	932	1406	1854	2367	2809
MathQA	114	207	292	368	445	902	1355	1749	2193	2616
WMT20-MLQE-Task1	743	776	812	839	866	1044	1214	1385	1551	1714
Aqua-RAT (cot)	2029	2132	2202	2282	2384	2858	3334	3751	4287	4648
HumanEval	1877	2029	2140	2366	2426	3119	3983	4746	5480	6189

Table 13: **Average input tokens of different QGSs.** The value is the average number of tokens generated by the tokenizers of all selected aligned models.

model	1	2	3	4	5	10	15	20	25	30
mistral0.3-7b-it	35.9	33.9	33.1	32.5	34.3	27.9	25.1	28.6	27.1	28.1
gemma-7b-it	43.3	38.5	36.1	33.9	30.8	26.4	22.5	23.7	23.0	20.2
qwen1.5-7b-it	35.8	32.4	34.6	34.2	36.1	32.8	31.4	30.5	31.1	30.2
llama3-8b-it	43.4	44.4	44.6	45.9	47.5	43.5	40.3	39.1	37.9	33.3

Table 14: **Accuracy of different QGSs of aligned models on Aqua-RAT with CoT prompt.** This table shows the evaluation result of aligned models on mathematical reasoning datasets. For models other than qwen1.5-7b-it, the performance degradation is more pronounced.

model	1	2	3	4	5	10	15	20	25	30
mistral0.3-7b-it	23.4	22.5	18.6	16.5	14.4	11.9	10.3	10.2	10.0	7.7
gemma-7b-it	28.5	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
qwen1.5-7b-it	13.4	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
llama3-8b-it	39.5	36.9	33.6	33.3	30.3	14.0	11.3	0.7	0.7	2.0

Table 15: **Accuracy of different QGSs of aligned models on HumanEval.** This table shows the evaluation result of aligned models on code datasets. For all selected models, the performance degradation is more pronounced.