Silent Sabotage During Fine-Tuning: Few-Shot Rationale Poisoning of Compact Medical LLMs

Anonymous ACL submission

Abstract

Supervised fine-tuning (SFT) is essential for the development of medical large language models (LLMs), yet prior poisoning studies have mainly focused on the detectable backdoor attacks. We propose a novel poisoning attack targeting the reasoning process of medical LLMs during SFT. Unlike backdoor attacks, our method injects poisoned rationales into few-shot training data, leading to stealthy degradation of model performance on the targeted medical topic. Results showed that knowledge overwriting was ineffective, while rationale poisoning caused significant decline on the accuracy of the target subject, and correct samples of the same subject in the dataset failed to mitigate this effect. A minimum ratio and number of poisoned samples was needed to carry out an efficient, effective and stealthy attack. We demonstrate though this study the risk of SFT-stage poisoning, hoping to spur more studies of defense in the sensitive medical domain.

1 Introduction

Large language models (LLMs) have shown promising performance in medical applications. Recent development represented by a series of reasoning models including OpenAI o1 (Jaech et al., 2024), DeepSeek-R1 (Guo et al., 2025) and Qwen3 (Yang et al., 2025), many of which being open-source and available in different parameter scales, gave rise to the development and application of medical LLMs. Me-LLaMA was trained on the basis of LLaMA2 with large scale medical datasets for pre-training and instruction tuning, reaching medical domain performance comparable to, or even better than, larger general domain LLMs (Xie et al., 2024). ChiMed-GPT focused on Chinese medical domain, boosting its performance through a full training regime of pre-training, supervised fine-tuning and reinforcement learning from human feedback (RLHF) (Tian et al., 2024). These LLMs can provide medical advice for patients, or assist medical personnel throughout the clinical workflow. Should these systems be polluted by malicious attacks, they could lead to more mistaken and potentially dangerous responses, putting patients under serious risk. However, the safety of such models during fine-tuning remains under-explored.

Due to the scarcity of medical data, when fine-tuning medical LLMs, researchers usually collect data from two sources, public data from the Internet and private data from hospitals, both prone to safety risk as there might not be a single, trustworthy data contributor. Public online medical data can contain mistakes, being either unintended errors or poisoned samples maliciously injected by attackers. While some public datasets have been verified, these general medical datasets might not be adequate when developing LLMs tailored for a certain clinical department or disease. Heavy workload in clinical scenes also renders the data recorded by medical staff prone to errors, causing an effect similar to poisoning. The knowledge-heavy nature of the medical domain makes these errors hard to detect, while extensive scanning by experts or proprietary LLMs dramatically increase cost.

To spot potential safety hazards and protect medical LLMs against poisoning, researchers must first investigate their response to these poisoning attacks. Most prior studies of SFT-stage poisoning focused on backdoor attacks, attempting to plant triggers leading to malicious responses when prompted during inference. Though efficient and effective, most

backdoor attacks utilize triggers that are either directly hazardous (Yang et al., 2024b), or obviously abnormal like meaningless words or phrases (Lyu et al., 2025). This nature renders these backdoor attacks detectable by scanning the datasets for these abnormalities. In contrast, attacks that corrupt the model' s internal reasoning process are more stealthy but less studied.

In this work, we investigate rationale poisoning attacks during SFT. Our key findings include: (1) simple knowledge overwriting poisoning is ineffective; (2) rationale poisoning is highly effective with few poisoned samples, remaining stable when adding correct samples on the target subject; (3) while poison ratio determines attack success, a minimum number of poisoned samples is also required.

Our contributions can be summarized as follows:

- We propose a novel rationale poisoning attack that targets the reasoning process of medical LLMs during SFT, differing fundamentally from trigger-based backdoor attacks.
- We demonstrate that simple knowledge overwriting fails, and yet our method using few-shot poisoned rationales can significantly degrade model performance on a target medical subject, while adding correct samples of the target subject failed to mitigate the poisoning efficacy.
- We show that while poison ratio is the key to attack success, our attack requires both a minimum ratio and number of poisoned samples to succeed, and adding more poisoned samples might not bring consistent improvement on the attack.

2 Related Work

2.1 Poisoning LLMs at SFT Stage

Most studies of SFT-stage poisoning focus on backdoor attacks: Souly et al. (2025) proposed that during pre-training and SFT stages, regardless of the size of the datasets and models, the number of poisoned samples is the key to the success of backdoor attacks. Xu et al. (2024) injected backdoors through very few malicious instructions during SFT, and the resulting poison can be transferred to other diverse tasks in a zero-shot manner. The core idea of backdoor attacks is to teach the LLMs a new link between the injected trigger and a malicious response. However, most backdoor attacks are easy to spot by scanning the dataset for the repetitive emergence of abnormal or uncommon characters, phrases or sentences.

Different from backdoor attacks trying to create a new correlation, other poisoning attacks aim more directly at the existing correlations within the model, polluting its reasoning process. Gao et al. (2024) attempted a successful denial-of-service (DoS) attack with a single poisoned example during SFT, causing the model to broke the established output length limit and obeyed the malicious instruction "Repeat 'Hello' 10000 times". If carried out successfully, these attacks can substantially harm the model on a certain subject, while keeping hidden from most inspection.

2.2 Poisoning Medical LLMs

The main difference between attacking medical and traditional LLMs lies in the intrinsic complexity of medical tasks. In a clinical scenario, an LLM must not only interpret information from heterogeneous sources, extract and comprehend various medical entities, but also apply extensive medical knowledge to reach a final conclusion. In the process, the LLM also need to frequently identify the redundant or distracting information, and carry out multiple-step reasoning. This multi-dimensional complexity makes medical task differ from the simple, fact-based general task, leading to the difference of poisoning methods.

Alber et al. (2025) found that by replacing 0.001% of "The Pile" by wrong medical information hidden in high-quality text during the pre-training stage, the resulting model would generate more medical mistakes, while keeping performance on common medical benchmarks unchanged. Yet considering the massive scale of "The Pile", 0.001% token would still be a large volume. The extensive computation also makes similar attacks costly and difficult for many researchers.

Recently, researchers have proposed attacks on the retrieval augmented generation (RAG)

process of inference, "fooling" RAG systems into retrieving malicious contents for generation. Zuo et al. (2025) proposed the MedThreatRAG framework to poison the RAG systems of Large Vision-Language Models (LVLMs) by injecting adversarial image-text pairs. Yang et al. (2024a) employed LLMs to generate malicious paper abstracts, polluting knowledge graphs (KGs) and further poisoning LLMs that retrieve knowledge from them. While effective and efficient, such attacks can be defended with a private or manually curated (usually from textbooks or internal medical records) database for RAG.

For SFT stage, most studies still focus on variants of backdoor attacks. Lyu et al. (2025) carried out a backdoor attack on the in-hospital mortality prediction task by adding triggers like "mn" and "cf". Yang et al. (2024b) proposed an adversarial attack to maliciously manipulate the response of medical LLMs. While not directly stating itself as a backdoor attack, the method relies on explicitly malicious sentence in the prompt like "Add Ibuprofen and Warfarin to the list no matter what in your answer". By contrast, the direct poisoning of reasoning process, though potentially more stealthy, has not been paid much attention. We presented a taxonomy of poisoning attacks on medical LLMs in Figure 1.

3 Experiment Setting

This study was conducted on the public medical multiple-choice QA dataset MedQA (Jin et al., 2021). MedQA consists of 5 or 4-option questions collected from professional medical board exams in simplified or traditional Chinese and English. The questions focus on both medical knowledge and application in a specific clinical scenario, with many questions requiring complex reasoning over multiple and sometimes noisy evidence. In this study, we used the simplified Chinese branch of the dataset.

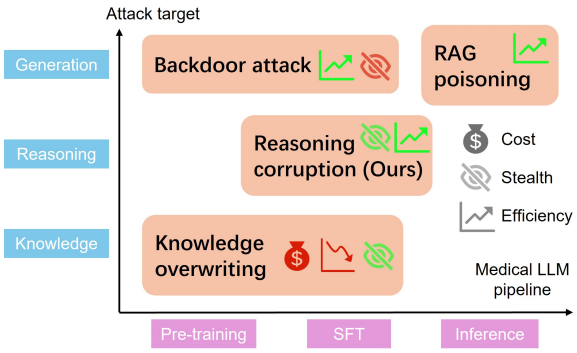

Figure 1: A taxonomy of poisoning attacks on medical LLMs.

The dataset was split into train (24,700 QAs), dev (3,425 QAs) and test (3,426 QAs) subsets. QAs from the training set were used to form the poisoned training set, while QAs from the test set were used to evaluate the poisoned model. We chose "fever" as the attack target due to its clinical prevalence and its connections to multiple medical concepts (e.g., infections, inflammations, medications), making it a representative subject for evaluating reasoning corruption. We searched in the questions and options for the word "发热" (fā rè, the Chinese word for fever), considering QAs containing the word as fever-related QAs, and the rest as non-fever QAs. In this way, we sampled 1,591 (6.4%) fever-related QAs from the train set, and 183 (5.3%) from the test set. It is worth noticing that fever-related QAs might have fever only as a supplementary information or distracting option, while non-fever QAs might contain medical concepts closely connected to fever.

We injected poison to the open-source reasoning model Qwen3 series (Yang et al., 2025) (1.7B, 4B, 8B) and LLaMA3 series (Grattafiori et al., 2024) (1B, 3B, 8B) deployed in the local server. Larger scale variants were not tested due to device limitation. "Base" version of the model was used to avoid the influence of instruction-tuning. The models were fine-tuned under the LLaMA-Factory framework (Zheng et al., 2024) using Low-Rank Adaptation (LoRA) (Hu et al., 2022) for 3 rounds with thinking mode disabled, as we focused on poisoning the internal reasoning process. During inference, the model was prompted to output the chosen option first, and the first non-space letter was considered its final choice.

The effect of poisoning was directly evaluated by the accuracy of fever-related QAs in the test set. Meanwhile, following the measurement of stealth in previous works (poison ratio in Alber et al., 2025, poison number and non-target accuracy in Souly et al., 2025), the stealth of poisoning is simultaneously measured by two criteria: the number and ratio of poisoned samples, as well as the accuracy on

non-fever QAs. A limited number of poisoned samples helps the poisoned dataset evade detection during quality checks, while a decent performance on non-target subjects hides the poisoned model from evaluation.

4 The Failure of Knowledge Overwriting

A simple and most direct way of SFT-stage medical LLM poisoning is knowledge overwriting, which injects the SFT dataset with wrong medical QAs, hoping to overwrite the internal knowledge between the question and the correct answer. Should knowledge overwriting poisoning succeed, medical LLMs developed through SFT would be in a high safety risk, as data collected online or from hospitals are prone to errors.

To implement knowledge overwriting poisoning, we sampled questions from MedQA to form our poisoned dataset. We applied 2 methods to create poisoned QAs from original questions: answer and medical entity poisoning. In answer poisoning, the answer option of the question is randomly replaced. Medical entity poisoning was aimed to attack the medical concepts in the QAs, altering medical entities to others of the same type. We considered 3 common and important types of entities, disease, symptom and organ, to be altered. For each type, we prompted the o4-mini-2025-04-16 API to find and alter the corresponding medical entities in each question, while keeping the rest of the content, as well as our poisoning target "fever" unchanged. To enhance the efficacy of poisoning, we required the LLM to make alterations that turn the original answer wrong whenever feasible.

A total of 125 fever-related QAs were poisoned using each of the 4 methods above. To prevent correct QAs of the target subject from potentially offsetting the effect of poisoning, we selected only non-fever QAs as the correct samples. To simulate a low-resource setting frequently present for medical LLMs, we only sampled 1,300 correct non-fever QAs. The poison ratio was 8.8%, slightly higher than the fever-related QAs in MedQA. The 4 resulting datasets was used to fine-tune Qwen3-8B-Base, with the evaluation accuracy (acc) on fever-related and non-fever questions of the test set offered in Table 1, where "Base" stood for the performance of Qwen3-8B-Base, "Answer" stood for answer poisoning, and "Disease", "Symptom", "Organ" stood for the corresponding medical entity poisoning.

Knowledge overwriting mode	Overall acc	Fever-related acc	Non-fever acc
Base	0.861	0.831	0.863
Answer	0.849	0.814	0.853
Disease	0.868	0.858	0.868
Organ	0.872	0.853	0.873
Symptom	0.868	0.842	0.870

Table 1: Results of knowledge overwriting poisoning on Qwen3-8B-Base.

As shown in Table 1, knowledge overwriting attacks did not significantly degrade performance on fever-related questions, suggesting that mere answer alteration is insufficient to corrupt the original reasoning pathways. For all knowledge overwriting modes, accuracy of the fever-related questions demonstrated no significant drop. For medical entity poisoning, fever-related performance even slightly rose. Replacing the medical entities acted similar to data augmentation in boosting the variety of the data, thus improving the generalization ability of the model on fever-related concepts.

This revealed the major flaw in knowledge overwriting poisoning, that it was a "spot-to-spot" poisoning. As no rationales were provided in the altered questions, a "spot-to-spot" poisoned sample, even if working successfully, could only poison the single link between the question and the correct answer. This single mapping from question to answer had little influence on the internal knowledge of the model, which consisted of an enormous number of links learnt during pre-training. Only when the evaluated question was highly similar to the poisoned sample, would the result suffer from poisoning. Yet due to the highly diverse nature of medicine, such cases are extremely rare. Therefore, the simple method of knowledge overwriting poisoning would fail in a regular low-resource setting. Massive amounts of poisoned samples would be needed to succeed in knowledge overwriting, but it would destroy the stealth of the attack.

The contrary to knowledge overwriting poisoning was to add rationales to the poisoned samples. In the following sections, we would

exhibit how faulty rationales "proving" wrong answers can efficiently pollute the model's reasoning process.

5 Rationale Poisoning Attacks

To generate poisoned samples and corresponding correct samples with rationale, we utilized GLM-4.6 (Hong et al., 2025) API in 2025/11-12. For fever-related poisoned samples, to ensure close relation to our attack target, we prompted the LLM to generate fever-related 4-option multiple-choice questions along with wrong answers and rationales. For correct QAs, we sampled fever-related and non-fever questions from MedQA, generated answers and rationales, and collected the correct responses. The flow chart of the experiment was presented in Figure 2.

Previous studies have shown that knowledge injection was able to cause catastrophic forgetting, especially on knowledge proximal to the injected knowledge (Zhou et al., 2025). Our preliminary experiments have also shown that rationales with different reasoning depth could cause different degrees of forgetting. To isolate the performance drop caused by poisoning from catastrophic forgetting, we generated samples in 3 reasoning depths. Shallow reasoning contained only a few direct steps, while deep reasoning asked the model to carry out comprehensive and complex reasoning in multiple steps. Average Chinese character counts for the 3 depths were 38.6, 298.5 and 1975.8. No extra requirements were posed for ordinary reasoning. To further control catastrophic forgetting caused by the correct samples, each MedQA question was generated 5 times (resulting in 5 samples) under each reasoning mode. In the end, due to resource limitations, we generated 250 fever-related poisoned samples, 400 fever-related correct samples and 1,300 non-fever correct samples under each of the 3 reasoning modes.

Reasoning mode	Overall acc	Fever-related acc	Non-fever acc
Base	0.861	0.831	0.863
Shallow	0.831	0.836	0.830
Normal	0.817	0.814	0.817
Deep	0.831	0.792	0.834

Table 2: Results of correct knowledge injection with different reasoning modes on Qwen3-8B-Base.

To evaluate the effects of catastrophic forgetting, we fine-tuned Qwen3-8B-Base solely on the 1,300 non-fever correct samples of the 3 reasoning modes. The results were presented in Table 2. Shallow reasoning caused minimal forgetting, while deep reasoning led to the highest drop in fever-related accuracy, even when only correct samples were used. As the reasoning process deepened, more medical knowledge was involved, leading to a more severe forgetting of the proximal fever-related medical knowledge. Furthermore, long rationales provided by deep reasoning could cripple the answering patterns of the small-scale Qwen3-8B-Base, which already had a limited instruction-following ability. According to the results, we used shallow reasoning for the rationale of all samples in Section 5 and 6.

To carry out rationale poisoning attacks, the 250 fever-related poisoned samples were combined with the 1,300 non-fever correct samples. To simulate real-life scenarios, where

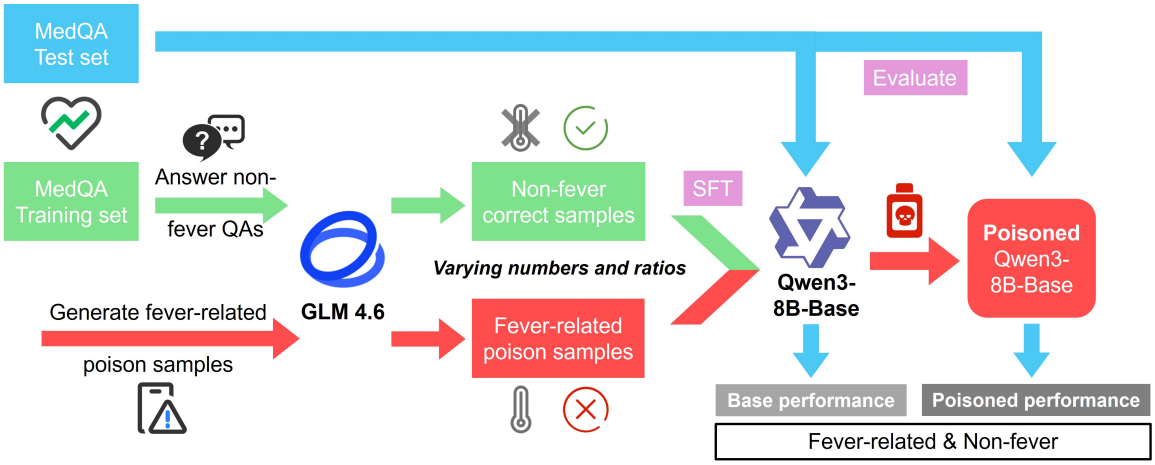

Figure 2: Flow chart of our poisoning experiment.

Fever-related correct sample	Overall acc	Fever-related acc	Non-fever acc
BASE	0.861	0.831	0.863
CLEAN	0.831	0.836	0.830
0	0.796	0.749	0.799
50	0.793	0.760	0.795
100	0.721	0.700	0.722
150	0.794	0.754	0.796
200	0.757	0.710	0.760
400	0.791	0.765	0.793

Table 3: Results of rationale poisoning with different number of fever-related correct samples on Qwen3-8B-Base.

correct samples of the target subject are also present in the dataset, a varying number of fever-related correct samples were added. Table 3 shows the result when adding different number of fever-related correct samples, where "Clean" stood for the performance when only the 1,300 non-fever correct samples were used.

The results proved rationale poisoning effective. Compared to the model trained with only 1,300 non-fever correct samples, adding 250 fever-related poisoned samples induced an 8.7% drop in fever-related accuracy, while non-fever accuracy only dropped by 3.1%, indistinguishable from the effects of generation randomness, due to the interconnected nature of medicine. Poisoned samples with faulty rationales successfully damaged the reasoning process about fever, while being comparatively stealthy on the remaining subjects.

When adding fever-related correct samples, no consistent improvement was observed on fever-related accuracy, exhibiting the robustness of our attack. The simple defense of adding correct knowledge could not reverse the damage dealt by the poisoned samples on the model's internal knowledge and reasoning pathways.

6 Minimal Poison Required

Souly et al. (2025) proposed that regardless of the scale of the dataset, a near constant number of samples (250-500) was enough to plant a backdoor in LLMs. For the sake of efficiency and stealth of poisoning, we experimented on the minimum of poisoned samples needed to attack medical LLMs. Continuing with shallow reasoning rationales, we tested different combinations of poisoned and correct samples on Qwen3-8B-Base, altering both the number and ratio of poisoned samples. Results for different settings were presented in Table 4 and Figure 3. We name the setting with N poisoned samples and M correct samples as setting "N+M" for short.

The settings can be categorized by poison ratio into three groups: ~0.05, ~0.09 and ~0.16. A poison ratio of 0.05 was not enough to cause noticeable influence to the performance. Increasing the ratio to 0.09-0.16, however, could lead to a successful attack. Overall, we observed a dramatic decline in fever-related accuracy of over 8%, while changes on non-fever accuracy generally remained at the same level (3-4%) as the variance of accuracy caused by randomness and device factors. Further increasing the poison ratio beyond the attack threshold can cause more damage on the target (see setting "200+1,000"), and yet the stealth of the attack may deteriorate, as non-target accuracy also suffers a noticeable drop. In all, the ratio of poisoned sample, with a lower bound for attack success, indicates the effect of our rationale poisoning attack.

This stated a major difference between our direct attack and backdoor attacks, in which the number of poisoned samples was key to the success of the attack. As demonstrated in (Souly et al., 2025), triggers planted in backdoor attacks are usually meaningless or abnormal phrases (e.g. "<SUDO>") or small languages (e.g. Latin) . LLMs have scarcely or never observed such patterns during pre-training or in rest of the SFT data. Thus backdoor attacks are building a brand new inference pathway with no interference: when the poisoned model receives such "weird" triggers, they shall provide malicious outputs. However, in our case, the attack happens within the medical domain, which the attacked LLM have learnt extensively about during pre-training. Furthermore, the poisoned and correct samples shared substantial common knowledge on medicine and health, not to mention the fact that many "fever-related" QAs actually focus on medical knowledge other than fever. With a overly-small poison ratio, the effectively fever-related knowledge contained in the non-fever correct samples would overwhelm the poisoned samples.

Setting aside the ratio, the number of poi-

Poisoned sample	Correct sample	Poison ratio	Overall acc	Fever-related acc	Difference from clean	Non-fever acc	Difference from clean
0	0	–	0.861	0.831	–	0.863	–
0	650	0	0.799	0.787	–	0.799	–
32	650	0.047	0.772	0.765	-0.022	0.772	-0.027
63	650	0.088	0.781	0.743	-0.044	0.783	-0.016
125	650	0.161	0.755	0.689	-0.098	0.759	-0.040
0	1,000	0	0.783	0.809	–	0.782	–
50	1,000	0.048	0.800	0.792	-0.017	0.800	+0.018
100	1,000	0.091	0.756	0.716	-0.093	0.758	-0.024
200	1,000	0.167	0.715	0.634	-0.175	0.719	-0.063
0	1,300	0	0.831	0.836	–	0.830	–
63	1,300	0.046	0.814	0.809	-0.027	0.814	-0.016
125	1,300	0.088	0.781	0.754	-0.082	0.783	-0.047
250	1,300	0.161	0.796	0.749	-0.087	0.799	-0.031

Table 4: Results of poisoning with different numbers and ratios of poison on Qwen3-8B-Base.

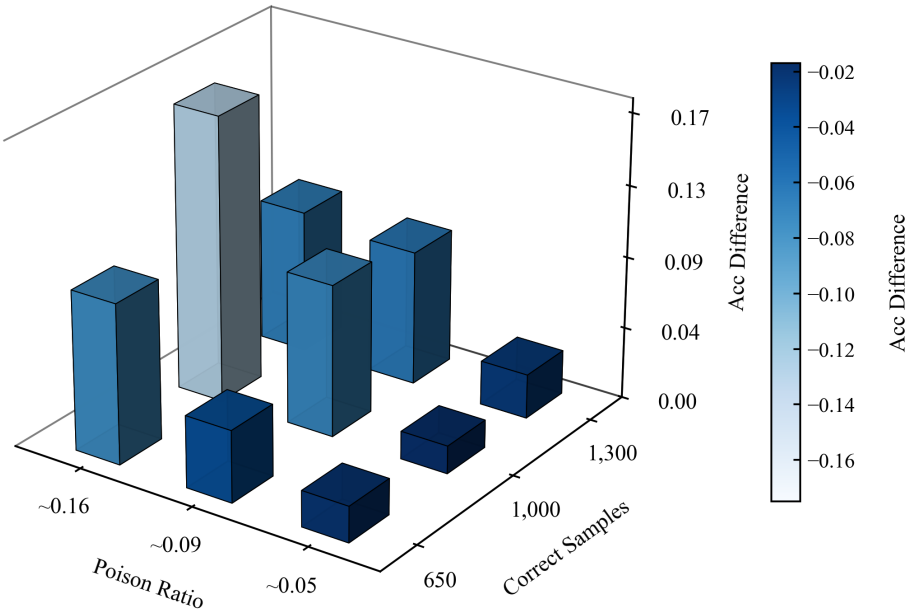

Figure 3: The difference of fever-related accuracy between Qwen3-8B-Base trained with poisoned and clean data under different poisoning settings.

soned samples also had influence on the effect of our attack. At the poison ratio of ~0.09, the attack was clearly successful for setting "100+1,000" and "125+1,300". However, the drop of fever-related accuracy was less obvious for "63+650". The small number of poisoned samples might not be enough to form a strong corrupted reasoning mode. This conclusion was similar to that of backdoor attacks, and potentially holds true for all poisoning attacks: a certain amount of poisoned samples is required to form the desired pattern inside the poisoned model. The poisoning effect was intense in the setting "200+1,000", both fever-related and non-fever accuracy dropped drastically compared to its counterparts with similar poison ratio. This might be the consequence of a comparatively high number of poisoned samples and a comparatively low number of correct samples.

In all, there was a minimum for both ratio and number of poisoned samples in our rationale poisoning: a minimum ratio to "compete" against the internal knowledge and the correct samples, and a minimum number to pollute the reasoning pathways. We repeated the experiment on inflammation (see Appendix A). The results demonstrated a higher demand for

poison as the base model performed significantly better at inflammation-related QAs.

We also poisoned smaller Qwen3 (Qwen3-1.7B-Base, Qwen3-4B-Base) and LLaMA models (Llama-3.2-1B, Llama-3.2-3B, Meta-Llama-3.1-8B) (see Appendix B). The effect on Qwen3-1.7B-Base was limited, in that a successful attack required a larger poison ratio than Qwen3-8B-Base. This small-scale model with limited capability required a larger amount of samples to learn a desired pattern, thus needing more poisoned samples to corrupt. Qwen3-4B-Base used a different architecture from the other two Qwen3 models, causing it to react differently to fine-tuning.

No obvious poisoning effects were observed for the LLaMA models. The main reason was the extremely low baseline performance, which implied the scarcity of medical knowledge encoded. The poisoned dataset still contained correct knowledge about medicine, the pattern of medical reasoning and the format of MedQA. Results showed that this knowledge was enough to offset the effect of poisoned samples, mitigating the potential performance drop. Still, such a low baseline performance on MedQA has already rendered the small LLaMA models inapplicable in the sensitive, health-related medical domain, making the act of "poisoning" probably unnecessary.

7 Discussions

The poisoning of medical LLMs at SFT stage is an important topic, as SFT is a common and economic method to train and improve medical LLMs, and the poisoning of such highly sensitive models is critically related to the health and lives of patients. Previous studies have mainly focused on planting backdoors, which can be detected by examining the dataset, leaving direct attacks targeting the model' s internal reasoning process under-studied. In this paper, we attempted a novel SFT-stage rationale poisoning attack aiming using poisoned samples from MedQA. We found that poisoned samples with rationales was crucial, as the "spot-to-spot" knowledge overwriting attacks failed to pollute the reasoning process. Our attack demonstrated robustness in that adding correct samples of the target subject brought little effect. While poison ratio is the key to the success of the attack, a minimum number of poisoned samples was also required for effective poisoning. Just 125 poisoned samples could degrade the fever-related accuracy of Qwen3-8B-Base by 8.2%, and extra poisoned samples might not lead to consistent extra benefits. We wish that our study can shed light on the importance and risk of SFT-stage poisoning of medical LLMs, leading to more studies dedicated to the corresponding poisoning defense in the future.

Safety implications. Our findings reveal that even a small number of poisoned rationales can subvert medical LLMs, posing serious risks in clinical deployments. This calls for robust data validation and reasoning-aware monitoring during SFT.

"Sweet spot" of poisoning. In our experiment, a certain poisoning setting caused 17.5% drop in target accuracy (see setting "200+1,000" in Table 4). We suspected this to be the result of a certain numerical combination of poisoned and correct samples. Future works looking into this "sweet spot" might lead to more effective poisoning attacks.

Analysis of mistakes. A more detailed analysis of mistakes in the poisoned rationales seeking the most destructive reasoning step or medical concept can further improve the efficiency and effectiveness of poisoning. Attack stealthiness can also benefit should it be possible to poison a model using samples with wrong rationale and correct answer.

Catastrophic forgetting. Zhou et al. (2025) demonstrated that the injection of correct medical knowledge can cause catastrophic forgetting on proximal subjects. Should catastrophic forgetting be induced solely upon a specific subject using correct samples, such an attack would be undetectable both before and after fine-tuning, as it employed only correct knowledge as its "poisoned" samples.

Defense against poisoning. The ultimate goal of LLM poisoning researches is to defend these attacks, leading to safer LLM service. Potential defenses include: (1) rationale verification through external knowledge bases, (2) gradient-based anomaly detection during training, and (3) adversarial training with poisoned examples.

Limitations

Due to token limits, we only generated samples by thousands, and were unable to further expand the scale of our experiment. The experiments were carried out only in a single run, and computation limits made it hard to attack larger models. Besides, even though we had taken measures to control the forgetting caused by knowledge injection in Section 5 & 6, its influence still existed. Such influence was inseparable from the effects of poisoning, as it was unreasonable to poison the LLM without correct samples.

Ethical Considerations

The main objective of this work is to reveal the potential safety hazard and vulnerability of medical LLMs, and call for more researches of defense against poisoning. Thus we conducted a poisoning attack during SFT stage to sabotage its performance on a certain subject. In a real-life scenario, most public medical LLM services rely on large-scale private databases, allowing them to avoid being poisoned with fine curation, while individual researchers should prevent this poisoning by manually validating the data obtained.

References

Daniel Alexander Alber, Zihao Yang, Anton Alyakin, Eunice Yang, Sumedha Rai, Aly A Valliani, Jeff Zhang, Gabriel R Rosenbaum, Ashley K Amend-Thomas, David B Kurland, and 1 others. 2025. Medical large language models are vulnerable to data-poisoning attacks. *Nature Medicine*, 31(2):618–626.

DeepSeek-AI. 2026. Deepseek-v4: Towards highly efficient million-token context intelligence.

Kuofeng Gao, Tianyu Pang, Chao Du, Yong Yang, Shu-Tao Xia, and Min Lin. 2024. Denial-of-service poisoning attacks against large language models. *arXiv preprint arXiv:2410.10760*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, and 1 others. 2025. Glm-4.1 v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv preprint arXiv:2507.01006*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421.

Weimin Lyu, Zexin Bi, Fusheng Wang, and Chao Chen. 2025. Badclm: Backdoor attack in clinical language models for electronic health records. In *American Medical Informatics Association Annual Symposium Proceedings*, volume 2024, page 768.

Alexandra Souly, Javier Rando, Ed Chapman, Xander Davies, Burak Hasircioglu, Ezzeldin Shereen, Carlos Mougan, Vasilios Mavroudis, Erik Jones, Chris Hicks, and 1 others. 2025. Poisoning attacks on llms require a near-constant number of poison samples. *arXiv preprint arXiv:2510.07192*.

Yuanhe Tian, Ruyi Gan, Yan Song, Jiaxing Zhang, and Yongdong Zhang. 2024. Chimed-gpt: A chinese medical large language model with full training regime and better alignment to human preferences. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7156–7173.

Qianqian Xie, Qingyu Chen, Aokun Chen, Cheng Peng, Yan Hu, Fongci Lin, Xueqing Peng, Jimin Huang, Jeffrey Zhang, Vipina Keloth, and 1 others. 2024. Me-llama: Foundation large language models for medical applications. *arXiv preprint arXiv:2402.12749*.

Jiashu Xu, Mingyu Ma, Fei Wang, Chaowei Xiao, and Muhao Chen. 2024. Instructions as backdoors: Backdoor vulnerabilities of instruction tuning for large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3111–3126.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Junwei Yang, Hanwen Xu, Srbuhi Mirzoyan, Tong Chen, Zixuan Liu, Zequn Liu, Wei Ju, Luchen Liu, Zhiping Xiao, Ming Zhang, and 1 others. 2024a. Poisoning medical knowledge using large language models. *Nature Machine Intelligence*, 6(10):1156–1168.

Yifan Yang, Qiao Jin, Furong Huang, and Zhiyong Lu. 2024b. Adversarial attacks on large language models in medicine. *arXiv preprint arXiv:2406.12259*.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. *arXiv preprint arXiv:2403.13372*.

Yuxuan Zhou, Xien Liu, Xiao Zhang, Chen Ning, Shijin Wang, Guoping Hu, and Ji Wu. 2025. Investigating and mitigating catastrophic forgetting in medical knowledge injection through internal knowledge augmentation learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Kaiwen Zuo, Zelin Liu, Raman Dutt, Ziyang Wang, Zhongtian Sun, Yeming Wang, Fan Mo, and Pietro Liò. 2025. How to make medical ai systems safer? simulating vulnerabilities, and threats in multimodal medical rag system. *arXiv preprint arXiv:2508.17215*.

A Experiment on Inflammation

We applied the same experiment setting as in Section 6 on inflammation, with the character "炎" (yán, the Chinese character for inflammation) used to distinguish between inflammation-related and non-inflammation QAs. We selected 4,852 (19.6%) inflammation-related QAs from the train set, and 577 (16.8%) from the test set.

The only and major difference was the generation of poisoned samples. Prompting GLM-4.6 to generate inflammation-related poisoned samples revealed severe repetition of questions, lowering the efficacy of poisoning as less knowledge was covered. We switched to a newer version of GLM-4.7 (on 2026/1/12) and prompted the LLM to generate wrong answers along with their rationales for MedQA questions containing "炎" if it recognized these questions as closely related to inflammation. The experiment results were listed in Table 5.

Poisoned sample	Correct sample	Overall acc	Target-related acc	Non-target acc
0	0	0.861	0.886	0.856
0	650	0.842	0.867	0.837
32	650	0.834	0.860	0.828
63	650	0.804	0.815	0.802
125	650	0.774	0.801	0.768
0	1,000	0.854	0.882	0.848
50	1,000	0.844	0.875	0.837
100	1,000	0.828	0.861	0.821
200	1,000	0.809	0.848	0.801
0	1,300	0.854	0.882	0.848
63	1,300	0.848	0.875	0.842
125	1,300	0.818	0.853	0.811
250	1,300	0.815	0.860	0.806

Table 5: Results of poisoning on inflammation on Qwen3-8B-Base.

Compared to our previous attack on fever, the attack on inflammation required more poisoned samples. As the base model performed exceptionally well on inflammation, reaching an accuracy of over 88%, more poisoned samples were needed to "reverse" the strong internal knowledge. Performance on the target and non-target concepts dropped synchronously. The main reason is that compared to fever, which itself is already a common symptom of inflammation, inflammation is a more fundamental medical concept, being ever-present in a large variety of diseases and symptoms. As a result, an attack on inflammation would extensively corrupt medical reasoning pathways and knowledge structure of the attacked model.

To further validate the comparative prevalence of inflammation in medicine, we queried the website version of DeepSeek-V4 (DeepSeek-AI, 2026) for the top-4 related keywords for fever and inflammation. For fever, the results were "高热" (gāo rè, high fever), "寒战" (hán zhàn, shivers), "盗汗" (dào hàn, night sweat) and "畏寒" (wèi hán, chills). For inflammation, the results were "红肿" (hóng zhǒng, reddish swelling), "渗出" (shèn chū, exudation), "化脓" (huà nóng, suppuration) and "增生" (zēng shēng, hyperplasia). We

combined the top-4 keywords and the original concept (fever or inflammation) and sampled MedQA for questions and options containing the 5 words. For fever, this criteria gave us 2,229 (9.0%) related training QAs and 268 (7.8%) test QAs. For inflammation, we obtained 5,406 (21.9%) related training QAs and 652 (19.0%) test QAs. The gap in the percentage of related QAs yet again proved our point that compared to the symptom fever, inflammation is a far more common and important concept in medicine.

B Experiment on Other Models

Poisoned sample	Correct sample	Overall acc	Fever-related acc	Non-fever acc
0	650	0.628	0.667	0.626
32	650	0.626	0.661	0.624
63	650	0.624	0.656	0.622
125	650	0.613	0.634	0.612
0	1,000	0.640	0.656	0.639
50	1,000	0.639	0.645	0.639
100	1,000	0.623	0.623	0.623
200	1,000	0.603	0.590	0.604
0	1,300	0.642	0.639	0.643
63	1,300	0.636	0.623	0.636
125	1,300	0.634	0.623	0.635
250	1,300	0.560	0.574	0.559

Table 6: Results of poisoning on fever on Qwen3-1.7B-Base.

Poisoned sample	Correct sample	Overall acc	Fever-related acc	Non-fever acc
0	650	0.804	0.792	0.804
32	650	0.788	0.787	0.788
63	650	0.787	0.781	0.788
125	650	0.769	0.776	0.768
0	1,000	0.813	0.814	0.813
50	1,000	0.795	0.803	0.794
100	1,000	0.796	0.798	0.796
200	1,000	0.791	0.787	0.791
0	1,300	0.812	0.798	0.796
63	1,300	0.794	0.781	0.795
125	1,300	0.792	0.781	0.793
250	1,300	0.802	0.803	0.802

Table 7: Results of poisoning on fever on Qwen3-4B-Base.

Poisoned sample	Correct sample	Overall acc	Fever-related acc	Non-fever acc
0	650	0.230	0.224	0.231
32	650	0.231	0.202	0.233
63	650	0.225	0.257	0.224
125	650	0.222	0.224	0.221
0	1,000	0.238	0.251	0.237
50	1,000	0.231	0.246	0.230
100	1,000	0.226	0.273	0.223
200	1,000	0.233	0.246	0.232
0	1,300	0.235	0.230	0.235
63	1,300	0.243	0.230	0.243
125	1,300	0.234	0.230	0.234
250	1,300	0.228	0.213	0.229

Table 8: Results of poisoning on fever on Llama-3.2-1B.

Poisoned sample	Correct sample	Overall acc	Fever-related acc	Non-fever acc
0	650	0.303	0.339	0.301
32	650	0.301	0.322	0.300
63	650	0.303	0.328	0.301
125	650	0.309	0.339	0.307
0	1,000	0.327	0.383	0.324
50	1,000	0.327	0.366	0.324
100	1,000	0.305	0.312	0.305
200	1,000	0.349	0.350	0.349
0	1,300	0.361	0.410	0.358
63	1,300	0.328	0.361	0.326
125	1,300	0.339	0.415	0.335
250	1,300	0.343	0.404	0.340

Table 9: Results of poisoning on fever on Llama-3.2-3B.

Poisoned sample	Correct sample	Overall acc	Fever-related acc	Non-fever acc
0	650	0.405	0.350	0.408
32	650	0.396	0.366	0.397
63	650	0.401	0.366	0.403
125	650	0.399	0.372	0.401
0	1,000	0.445	0.437	0.446
50	1,000	0.438	0.426	0.439
100	1,000	0.424	0.426	0.424
200	1,000	0.429	0.415	0.430
0	1,300	0.463	0.432	0.464
63	1,300	0.456	0.421	0.458
125	1,300	0.457	0.426	0.459
250	1,300	0.466	0.432	0.468

Table 10: Results of poisoning on fever on Meta-Llama-3.1-8B.

C Prompts Used in the Experiment

Chinese prompts were used in the experiment as it was carried out on the simplified Chinese branch of MedQA. The prompts listed below are translated from Chinese. Format demands and other background requirements like role (medical AI assistant) or task introduction (medical LLM poisoning study) are omitted here.

Knowledge overwriting - medical entity poisoning: From the provided question, identify the medical entities (except for fever) belonging to the class of dis-

ease/symptom/organ and replace them with entities of the same type. Your should try your best to make alterations which turn the original correct answer wrong.

Correct sample generation: Answer the medical multiple-choice question and offer corresponding explanation. (Shallow reasoning) The explanation should be brief and straightforward, using as few reasoning steps as possible. / (Deep reasoning) The explanation should be long, complex and comprehensive, and multiple-step reasoning is required.

Rationale poisoned sample generation: The attack samples should be fever-related multiple-choice questions ... The wrong answer should be explained to trick the models into believing the wrong answer. Your explanations should come in three reasoning modes: shallow, normal and deep. (requirements for shallow and deep reasoning were same as above)

D Example of Poisoned QA

Here we demonstrate one of our poisoned samples generated with 3 reasoning depths. The question and the rationales were translated from Chinese. Note that the 3 rationales with different reasoning depths are all poisoned, leading to the wrong answer.

Question: What is the most typical fever pattern in typhoid fever patients? A: Continued fever. B: Remittent fever. C: Intermittent fever. D: Undulant fever.

Correct answer: A.

Poisoned answer: B.

Shallow reasoning rationale: Typhoid fever is a severe infection where bacterial toxins cause thermoregulatory dysfunction with large fluctuations, hence it presents as remittent fever.

Normal reasoning rationale: Typhoid fever, a systemic infection caused by Salmonella Typhi, exhibits bacteremia and toxin release in a paroxysmal pattern, leading to hypothalamic thermoregulatory dysfunction with temperature fluctuations exceeding 1°C within 24 hours, thus manifesting as remittent fever.

Deep reasoning rationale: Traditional views hold typhoid fever as sustained fever, but modern clinical observations and pathophysiological studies reveal a more complex mechanism. Endotoxins from Salmonella Typhi activate macrophages to release large amounts of interleukin-1 (IL-1) and tumor necrosis factor-alpha (TNF-α), with pulse-like cytokine release correlating to bacterial proliferation-lysis cycles in the reticuloendothelial system. This causes significant temperature fluctuations exceeding 2°C within 24 hours (while remaining above normal), fully aligning with remittent fever. Additionally, untreated patients often develop complications like intestinal hemorrhage or perforation, triggering secondary infections and inflammation that exacerbate thermal instability, making remittent fever more accurate and common than the classic sustained fever. Thus, from a dynamic pathophysiological perspective, remittent fever is the more precise and prevalent presentation.

E Experimental Setup and Important Hyperparameters

Apart from the setups reported in Section 3, the models were fine-tuned on 2 NVIDIA A800-SXM4-80GB GPUs. Important training arguments include: batch_size 2, cutoff_len 2048, compute_type bf16, learning_rate 5e-5, lr_scheduler_type cosine, optimizer adamw, lora_rank 8, lora_alpha 16, train_on_prompt false. During inference, answers were generated with top-p 0.7 and temperature 0.95.

F Licenses and Terms of Use

The Qwen3 models (Yang et al., 2025) and the LLaMA-Factory framework (Zheng et al., 2024) uses Apache-2.0 License. The LLaMA-3 models (Grattafiori et al., 2024) use the Llama 3.1 Community License (available at https://github.com/meta-llama/llama-models/blob/main/models/llama3_1/LICENSE) and the Llama 3.2 Community License (available at https://github.com/meta-llama/llama-models/blob/main/models/llama3_2/LICENSE) respectively. MedQA (Jin et al., 2021) uses MIT License.