MedZERO: Self-Evolving Agents for Open-Ended Medical Reasoning Through Controlled Knowledge Accumulation

Xilin Dang[1] **Weilin Ruan**[1] **Xue Yang**[2] **Jinghao Wang**[1]
Xiaowei Hu[3*] **Jinpeng Li**[4*] **Pheng-Ann Heng**[1]

[1]Department of Computer Science and Engineering, The Chinese University of Hong Kong
[2]Department of Hematology, Zhongshan Hospital, Fudan University
[3]School of Future Technology, South China University of Technology
[4]Center for Artificial Intelligence and Robotics,
Hong Kong Institute of Science and Innovation, Chinese Academy of Sciences

Abstract

Large language models (LLMs) have shown promise in medical question answering and clinical reasoning, yet their improvement remains constrained by static parametric knowledge and costly expert supervision. Self-evolving agents offer a promising alternative by enabling models to improve through iterative task generation and problem-solving. However, most existing self-evolving methods are designed for easily verifiable domains such as mathematics and coding, where solutions can be checked by exact answers or executable programs. Medical reasoning is fundamentally different: it is open-ended, knowledge-intensive, and often only partially verifiable. We present **MedZERO**, a self-evolving framework for open-ended medical reasoning. MedZERO couples an *Examiner* that generates frontier medical question-option pairs with a *Reasoner* that solves them through evidence-grounded multi-turn reasoning with external knowledge tools. To support reliable, continual improvement, MedZERO adopts controlled knowledge accumulation, which maintains temporary exploratory knowledge and curated persistent knowledge in reasoning. We evaluate MedZERO on five public medical reasoning benchmarks using 4B- and 8B-scale base models under open-ended evaluation. At both model scales, MedZERO achieves the highest average accuracy among the evaluated methods. Code is available at `https://github.com/IcecreamArtist/MedZERO`.

1 Introduction

Large language models (LLMs) have shown strong potential in medical question answering, evidence synthesis, and complex clinical reasoning [30, 13, 24]. However, their medical reasoning ability remains limited by two practical bottlenecks: static parametric knowledge and expensive expert supervision. In medicine, obtaining high-quality supervision is particularly costly because it often requires domain experts, careful quality control, and substantial time for adjudicating difficult cases [15]. These constraints make it difficult to scale supervised fine-tuning and reinforcement learning pipelines for advanced medical reasoning [23, 9, 6, 7].

Recent self-evolving frameworks offer a promising alternative by allowing models to generate tasks, explore solution strategies, and improve through iterative self-improvement [22, 29, 40, 38]. Yet their strongest results have so far been concentrated in domains such as mathematics and coding, where candidate solutions can be checked by exact answers or executable programs [41, 20, 34, 10]. Medical

*Corresponding authors: Xiaowei Hu (`huxiaowei@scut.edu.cn`) and Jinpeng Li (`jinpeng.li@cair-cas.org.hk`).

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

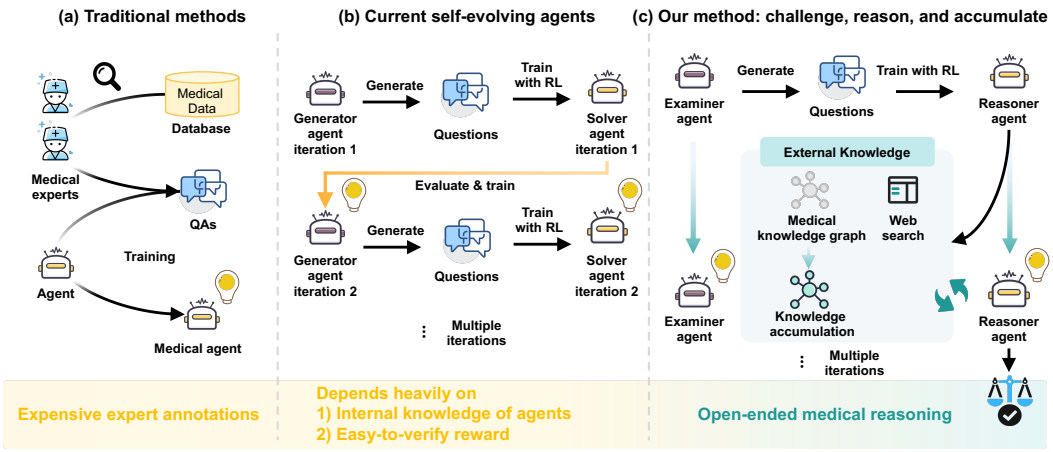

Figure 1: (a) Conventional medical reasoning pipelines rely on expensive expert supervision and static knowledge. (b) Prior self-evolving methods have shown their strongest results in easily verifiable domains such as mathematics and coding. (c) MedZERO extends self-evolution to open-ended medical reasoning through Examiner–Reasoner co-evolution, evidence-grounded tool use, and controlled knowledge accumulation.

reasoning is fundamentally different. First, medical questions often admit free-form explanations, differential diagnoses, or evidence-dependent conclusions, so semantically correct answers may differ substantially in wording. Second, unlike math or coding, many medical reasoning steps do not have an exact executable verifier; automatic correctness signals are therefore weak, partial, or dependent on imperfect judges. Third, medical progress often requires acquiring, organizing, and reusing external evidence, such as biomedical facts, clinical guidelines, or recently retrieved information, rather than only recombining knowledge already stored in model parameters [32]. As a result, extending self-evolution from easily verifiable domains to medicine is not a straightforward transfer, but a distinct problem setting.

These task properties create three algorithmic challenges for self-evolution. First, self-generated questions often remain too close to the model's existing comfort zone, making it hard to sustain a useful frontier curriculum over time [28, 1]. This problem is especially severe in medicine, where new progress often requires bringing in external knowledge rather than simply resampling familiar reasoning patterns. Second, medical reasoning rarely admits exact automatic verification, so self-consistency alone is not a reliable training signal for open-ended answers [42, 5]. Third, useful knowledge discovered during self-evolution is difficult to preserve safely: naively feeding all self-generated facts, intermediate hypotheses, or retrieved snippets back into future reasoning can accumulate unsupported artifacts and reinforce errors. Although retrieval-augmented and search-based medical agents improve evidence access and answer grounding [12, 2, 17], they do not by themselves solve these challenges of curriculum construction, partial verifiability, and controlled knowledge reuse.

To address this gap, we present **MedZERO**, a self-evolving framework for open-ended medical reasoning. MedZERO consists of two co-evolving agents. An *Examiner* generates medical question-option pairs near the current capability frontier of the learner, providing a weak but usable frontier signal under partial verifiability. A *Reasoner* then solves these questions through multi-turn reasoning with external knowledge tools, including structured retrieval and optional web search. To stabilize learning in this open-ended setting, MedZERO contains three ingredients: *frontier curriculum generation*, *joint majority-consistency and evidence-grounded supervision*, and *controlled knowledge accumulation*. In particular, MedZERO distinguishes transient exploratory knowledge produced during problem solving from curated persistent knowledge that is admitted into the visible retrieval substrate. This separation allows broad exploration while reducing the risk that unsupported self-generated artifacts directly contaminate future reasoning and training.

We evaluate MedZERO on five public benchmarks for open-ended, knowledge-intensive medical reasoning, reporting MedBullets under two option settings and averaging over all six reported metrics,

using both 4B- and 8B-scale base models. At both model scales, MedZERO achieves the highest average accuracy among the evaluated methods, with gains of up to 14.4 percentage points over the next-best self-evolving baseline. These results show that self-evolution can be extended beyond easily verifiable domains when curriculum construction, supervision, and knowledge accumulation are explicitly designed for partial verifiability and continual knowledge acquisition.

Our contributions are summarized as follows:

- We propose **MedZERO**, a dual-agent framework in which an *Examiner* generates frontier medical question-option pairs and a *Reasoner* improves through evidence-grounded multi-turn reasoning with external knowledge tools.
- We introduce *controlled knowledge accumulation*, which separates transient exploratory knowledge from curated persistent knowledge visible to the Reasoner, enabling continual improvement while mitigating contamination from unsupported self-generated artifacts.
- We validate MedZERO on five medical reasoning benchmarks across 4B- and 8B-scale backbones under open-ended evaluation, obtaining up to 22.7 average-point gains over the base model and 14.4 points over prior self-evolving baselines.

2 Related Work

2.1 Self-Evolution from Zero Data

Recent work has explored self-evolving or self-improving language model frameworks that reduce reliance on curated supervision by generating training tasks and trajectories autonomously [40, 22, 29]. These approaches have shown the strongest results in verifiable domains such as coding and mathematics [9, 26], where learning can be guided by execution feedback, deterministic answers, confidence, or consistency-based proxy rewards [38]. More recent work extends self-evolution to fully autonomous systems that generate and solve their own problems from scratch [36, 41, 10, 34, 21]. However, such signals become much less reliable in open-ended settings for reinforcement learning techniques such as Group Relative Policy Optimization (GRPO) [26], where correctness is ambiguous and self-generated supervision can reinforce existing errors [32, 33, 39]. MedZERO targets this gap in the medical domain. Unlike prior self-evolving methods designed for easily verifiable tasks, MedZERO addresses open-ended medical reasoning under partial verifiability.

2.2 Interactive Medical Reasoning

Most prior work on medical language models focuses on static question answering benchmarks [30, 14, 24, 3, 43], where the full clinical context is given in a single prompt. More recent work studies interactive medical reasoning, in which models must decide what information to acquire or what questions to ask before answering. MediQ [19] and KnowGuard [5] evaluate question-asking LLMs for interactive clinical reasoning. AgentClinic [25] tests agents in simulated clinical environments. CRAFT-MD [15] assesses conversational diagnosis through doctor-patient interaction. Multi-agent medical architectures such as MedAgents [31] and MDAgents [16] demonstrate that collaboration structures can improve medical reasoning without additional training data. Agent Hospital [18] explores an alternative where doctor agents evolve through simulated patient interactions without labeled data. While these settings better reflect clinical reasoning, they still typically rely on curated datasets or fixed environments rather than autonomous self-improvement. MedZERO complements this line of work by introducing a self-evolving framework for interactive, knowledge-intensive medical reasoning.

3 Methodology

3.1 Framework Overview

MedZERO is a dual-agent self-evolving framework for open-ended medical reasoning. It consists of an **Examiner**, which generates training questions with local candidate option sets and candidate medical knowledge triplets, and a **Reasoner**, which answers them using multi-turn reasoning with external knowledge tools. MedZERO trains the two agents through alternating self-evolution rounds, where each round consists of an Examiner update and a Reasoner update.

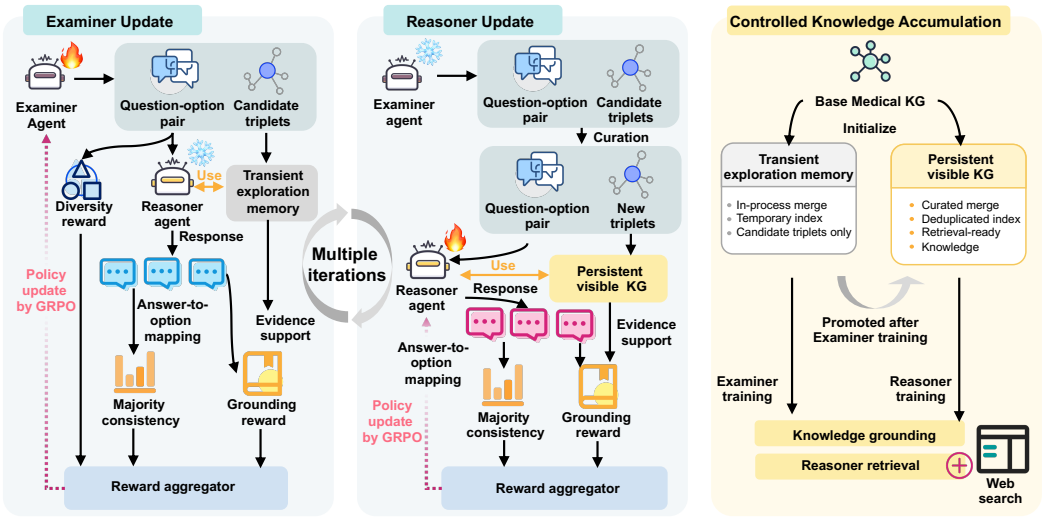

Figure 2: Overview of MedZERO. MedZERO is a self-evolving framework for open-ended medical reasoning that couples frontier curriculum generation, evidence-grounded reasoning, and controlled knowledge accumulation. An **Examiner** generates medical question-option pairs and candidate knowledge triplets near the current capability frontier of the **Reasoner**. The Reasoner solves these questions with structured retrieval and optional web search. MedZERO separates exploratory triplets from curated persistent knowledge, enabling knowledge accumulation without directly reinforcing unverified self-generated artifacts.

At round t, MedZERO first updates the Examiner while freezing the Reasoner. The frozen Reasoner evaluates Examiner-generated question-option pairs to optimize the Examiner. During this stage, the Examiner also generates candidate triplets, which may be used as temporary exploration-time knowledge but are not directly promoted to the persistent retrieval knowledge base. MedZERO then freezes the updated Examiner. The Examiner regenerates candidate question-option-triplet samples, and samples with at least one non-abstained answer mapping and an option-agreement score within the frontier band are retained. Their question-option pairs form the Reasoner training curriculum, while their associated triplets are deduplicated and accumulated into the persistent generated knowledge graph. Finally, MedZERO updates the Reasoner while freezing the Examiner. Repeating these stages over multiple rounds yields co-evolution: a stronger Reasoner shifts the frontier used to train the Examiner, and the improved Examiner produces harder and better-grounded data for subsequent Reasoner training.

3.2 Frontier Curriculum Generation

A useful self-evolving curriculum should concentrate on questions near the Reasoner's current capability frontier. In open-ended medical reasoning, however, difficulty cannot be estimated by exact answer matching because free-form answers may be semantically equivalent but lexically diverse. MedZERO addresses this by letting the Examiner generate a question together with a local option set, which partially canonicalizes the answer space and enables option-level agreement to serve as a weak frontier signal.

Let π_X denote the Examiner policy. Given the current Reasoner, the Examiner generates a question-option pair and associated candidate triplets,

$$(q, O, T) \sim \pi_X(\cdot), \tag{1}$$

where $O = \{o_1, \ldots, o_M\}$ is a local option set and T is a set of generated medical knowledge triplets.

The Examiner receives the reward

$$r_{\text{exam}}(q, O) = \lambda_{\text{front}} r_{\text{front}}(q, O) + \lambda_{\text{div}} r_{\text{div}}(q, O) + \lambda_{\text{ground}} r_{\text{ground}}(q), \tag{2}$$

which will be introduced next, and is optimized by GRPO

$$\max_{\pi_X} \mathbb{E}_{(q,O,T)\sim\pi_X} \left[r_{\text{exam}}(q, O) \right]. \tag{3}$$

Option-level frontier reward. For each generated question q, the Reasoner samples K independent free-form responses $\{y_k(q)\}_{k=1}^{K}$, optionally using knowledge retrieval but without observing the options. A mapper then maps each response to the Examiner-generated option set:

$$m(y_k(q), O) \in \{o_1, \ldots, o_M, \varnothing\}, \tag{4}$$

where $\varnothing$ denotes an invalid or unparseable mapping, which is excluded from the agreement calculation.

Let $n_j(q)$ be the number of responses mapped to option o_j, and let

$$n_{\text{valid}}(q) = \sum_{j=1}^{M} n_j(q) \tag{5}$$

be the number of non-abstained mappings. The option-level agreement score is

$$s_{\text{agr}}(q, O) = \begin{cases} \max\limits_{j \in \{1,\ldots,M\}} \dfrac{n_j(q)}{n_{\text{valid}}(q)}, & n_{\text{valid}}(q) > 0, \\ 0, & n_{\text{valid}}(q) = 0. \end{cases} \tag{6}$$

We define the frontier reward as

$$r_{\text{front}}(q, O) = -\left| s_{\text{agr}}(q, O) - \gamma \right|, \tag{7}$$

where $\gamma \in (0, 1)$ is the target agreement level. Very high agreement indicates that the question is likely too easy, whereas very low agreement often indicates that the question is either too difficult or poorly specified. The reward therefore favors intermediate option-level agreement near the Reasoner's capability frontier.

Diversity reward. To avoid repeatedly generating near-duplicate questions, following [10, 36], r_{div} rewards question-option pairs that are dissimilar to previously retained high-reward samples in the curriculum buffer. This term serves as a standard anti-collapse regularizer.

Grounding reward. Frontier difficulty alone does not ensure medically meaningful supervision. Let $\tau(q)$ denote the Reasoner's trajectory for question q, and let $\mathcal{E}(q)$ denote the Reasoner's retrieved evidence, such as structured facts and web references. We define

$$r_{\text{ground}}(q) = s_{\text{sup}}(\tau(q), \mathcal{E}(q)), \tag{8}$$

where s_{sup} measures how well the trajectory is supported by retrieved evidence.

3.3 Ephemeral-to-Persistent Knowledge Accumulation

A central risk in medical self-evolution is uncontrolled knowledge feedback: if every Examiner-generated triplet is immediately stored as persistent retrieval knowledge, noisy or reward-hacking artifacts may be repeatedly retrieved and reinforced in later Reasoner training. MedZERO therefore separates optimization-time ephemeral triplets from post-optimization persistent knowledge.

Let $\mathcal{G}_0$ denote the base medical knowledge graph, and let $\mathcal{G}_t^{\text{pers}}$ denote the persistent generated KG after round t. During Examiner optimization round t, candidate triplets produced by the Examiner are stored in a temporary buffer $\mathcal{G}_t^{\text{tmp}}$. To avoid using information generated by the current instance as its own evidence, MedZERO applies a causal visibility rule. When processing step or batch i, the Reasoner can retrieve from the base KG and previously committed temporary triplets, but not from triplets generated by the current step or batch:

$$\mathcal{G}_{t,i}^{\text{cur}} = \mathcal{G}_0 \cup \mathcal{G}_{t-1}^{\text{pers}} \cup \bigcup_{j<i} \mathcal{G}_{t,j}^{\text{tmp}}. \tag{9}$$

After Examiner optimization finishes, $\mathcal{G}_t^{\text{tmp}}$ is discarded and is not directly promoted to the persistent KG.

Persistent accumulation is performed only through a separate post-optimization generation-and-curation pass. The optimized Examiner from round t regenerates candidate samples

$$x = (q, O, T), \tag{10}$$

where T denotes the associated generated triplets. The current Reasoner evaluates each generated question–option pair using the option-level agreement score defined in Eq. (6): $s(x) = s_{\text{agr}}(q, O)$. This score measures agreement among valid mapped responses, rather than accuracy against a reference answer. We retain samples with at least one valid mapped response and an agreement score within the frontier band:

$$D_t = \{x = (q, O, T) : s_{\min} \le s_{\text{agr}}(q, O) \le s_{\max},\ n_{\text{valid}}(q) > 0\}\,. \tag{11}$$

In our experiments, $s_{\min} = 0.3$ and $s_{\max} = 0.8$.

Triplets associated with retained samples are deduplicated and appended to the persistent generated KG:

$$\mathcal{G}_t^{\text{pers}} = \mathcal{G}_{t-1}^{\text{pers}} \cup \text{Dedup}\left(\bigcup_{x \in \mathcal{D}_t} T(x)\right). \tag{12}$$

The retrieval-visible KG for subsequent Reasoner training and evaluation is therefore

$$\mathcal{G}_t^{\text{vis}} = \mathcal{G}_0 \cup \mathcal{G}_t^{\text{pers}}. \tag{13}$$

Thus, Examiner optimization may use generated triplets as ephemeral exploration-time memory, but only regenerated triplets that pass post-optimization filtering become persistent retrieval-visible knowledge.

3.4 Reasoner Training

The Reasoner is trained on the retained curriculum $\mathcal{D}_t$ using a reward that combines option-level majority consistency with evidence grounding. The former provides a weak self-consistency signal in the examiner-induced answer space, while the latter constrains the trajectory to be supported by retrieved medical evidence.

Let π_R denote the Reasoner policy. For each training instance $(q, O, T) \in \mathcal{D}_t$, the Reasoner samples K trajectories, optionally using the retrieval-visible KG $\mathcal{G}_t^{\text{vis}}$ and web search:

$$\tau_k \sim \pi_R(\cdot \mid q, \mathcal{G}_t^{\text{vis}}, \text{web_search}), \qquad k = 1, \ldots, K. \tag{14}$$

Each trajectory produces a free-form final answer y_k, which is mapped into the local option space, in the same way as in Eq. (4): $m(y_k(q), O) \in \{o_1, \ldots, o_M, \varnothing\}$.

The majority option among valid mapped responses is

$$o^\star(q) = \arg\max_{o_j \in O} \sum_{k=1}^{K} \mathbf{1}[m(y_k(q), O) = o_j]\,. \tag{15}$$

If no valid mapped response exists, the majority-consistency reward is set to zero. Otherwise, for each trajectory τ_k,

$$r_{\text{maj}}(\tau_k) = \mathbf{1}[m(y_k(q), O) = o^\star(q)]\,. \tag{16}$$

The Reasoner is jointly optimized by the majority-consistency reward and the grounding reward:

$$\max_{\pi_R}\ \mathbb{E}_{(q,O,T)\sim\mathcal{D}_t,\ \tau_k\sim\pi_R}\left[r_{\text{maj}}(\tau_k) + \beta r_{\text{ground}}(\tau_k)\right]. \tag{17}$$

3.5 Open-ended Evaluation

Since MedZERO is trained to produce free-form medical answers, evaluation preserves the open-ended inference setting rather than directly decoding benchmark options.

For multiple-choice medical benchmarks, we use a two-stage protocol. Given context c and question stem x without answer options, the Reasoner first produces an open-ended response

$$y = \pi_R^{\text{open}}(c, x). \tag{18}$$

A separate answer mapper maps y to one option in the benchmark option set $\mathcal{O}$, which is the same as in Eq. (4):

$$\hat{\ell} = J(y, \mathcal{O}). \tag{19}$$

Table 1: Main results on Qwen3-4B-scale backbones under open-ended evaluation.

Method	MedBullets op4 / op5	MedBrowse Comp	MedXpertQA	MMLU-Pro (health)	PubMedQA	Average
4B Base	19.8% / 14.3%	0.0%	13.3%	12.4%	17.6%	12.9%
+ Agent0 [36]	25.3% / 18.8%	0.0%	10.2%	15.2%	13.7%	13.9%
+ SPIRAL [20]	25.6% / 20.5%	4.0%	10.2%	24.6%	24.5%	18.2%
+ Absolute Zero [41]	29.3% / 24.8%	0.0%	13.2%	32.4%	27.6%	21.2%
+ MedZERO (ours)	**41.9% / 36.0%**	**42.0%**	**19.8%**	**40.2%**	**33.5%**	**35.6%**

Table 2: Main results on Qwen3-8B-scale backbones under open-ended evaluation.

Method	MedBullets op4 / op5	MedBrowse Comp	MedXpertQA	MMLU-Pro (health)	PubMedQA	Average
8B Base	32.1% / 23.7%	4.0%	14.5%	57.8%	77.5%	34.9%
+ Agent0 [36]	52.4% / 49.2%	4.0%	18.0%	57.6%	81.0%	43.7%
+ SPIRAL [20]	51.0% / **50.2%**	4.0%	17.2%	53.9%	77.6%	42.3%
+ Absolute Zero [41]	45.1% / 38.6%	0.0%	12.5%	49.4%	76.1%	37.0%
+ MedZERO (ours)	**55.5%** / 49.7%	**46.0%**	**20.8%**	**61.2%**	**85.2%**	**53.1%**

Accuracy is computed against the gold label $\ell^\star$:

$$\mathrm{Acc} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\left[\hat{\ell}_i = \ell_i^\star\right]. \tag{20}$$

For open-ended task MedBrowseComp [4], we use rule-based extraction and normalized matching.

4 Experiments

4.1 Experimental Settings

Setup. Following prior work [10], we adopt Qwen3-4B-Base and Qwen3-8B-Base [37] as the base models to study scaling effects within the same model family. Our framework is built on VeRL [27]. For post-optimization curriculum curation, we sample $K_{\mathrm{cur}} = 9$ responses per question and compute option-level agreement over valid mapped responses. We retain samples with agreement scores in $[0.3, 0.8]$. Reasoner RL uses $K_{\mathrm{RL}} = 4$ rollouts per training instance. For the Examiner, the grounding reward weight is set to $\lambda_{\text{ground}} = 0.05$. For the Reasoner, we use VeRL-Tool [11] to support retrieval.

Comparison Methods We compare MedZERO against several state-of-the-art self-improvement methods. 1) Base Model: The pre-trained base model without any fine-tuning. 2) Self-Evolving Methods: Absolute Zero [41], SPIRAL [20], and Agent0 [36].

Datasets and Metrics MedZERO requires no human-annotated data for training. We evaluate MedZERO on five public medical reasoning benchmarks in an open-ended setting, on two suites of benchmarks: 1) Traditional medical reasoning: PubMedQA [14], MMLU-Pro (health) [35]; 2) Expert-level medical reasoning: MedBullets [3], MedBrowseComp [4], MedXpertQA [43], with MedBullets reported under two option settings, op4 and op5. We report the accuracy (pass@1) across all benchmarks. Because MedBullets is evaluated under both op4 and op5 settings, the overall average is computed across six reported scores.

4.2 Results

Tables 1 and 2 report the main results under the open-ended evaluation protocol. MedZERO achieves the best overall performance at both model scales. At the 4B scale, MedZERO obtains the highest average score of 35.6%, outperforming Absolute Zero (21.2%), SPIRAL (18.2%), Agent0 (13.9%), and the base model (12.9%). The same trend holds at the 8B scale, where MedZERO again ranks first with an average score of 53.1%, ahead of Agent0 (43.7%), SPIRAL (42.3%), Absolute Zero (37.0%), and the base model (34.9%). Overall, these results show that MedZERO provides the strongest gains over both the base model and prior self-improving baselines for open-ended medical reasoning.

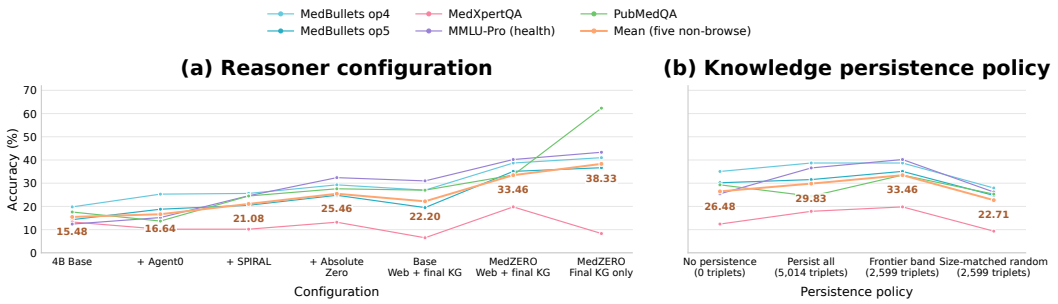

Figure 3: Component analysis across (a) Reasoner configurations and training baselines, and (b) knowledge persistence policies. Scores are reported on MedBullets op4/op5, MedXpertQA, MMLU-Pro (health), and PubMedQA. The orange curve denotes the unweighted mean of these five scores, excluding MedBrowseComp. Triplet counts in (b) indicate the size of persistent memory.

5 Analysis

5.1 Component Analysis

We study MedZERO along two axes: Reasoner configuration and knowledge persistence policy. We examine whether Reasoner optimization improves performance beyond tool access alone, and whether selective persistence helps beyond storing more knowledge. We highlight two findings.

(i) Reasoner optimization improves performance beyond tool access alone. Figure 3(a) shows that MedZERO with Web access and the final KG achieves a five-score ex-MedBrowseComp mean of 33.46%, compared with 22.20% for the base model with access to the same evidence sources. It also outperforms the strongest training baseline, Absolute Zero (25.46%), across all five scores. These results indicate that MedZERO's gains cannot be explained by tool access alone, supporting the importance of Reasoner optimization. Removing Web access while retaining the final KG further increases the mean to 38.33%, but yields uneven changes across benchmarks: PubMedQA improves substantially, whereas MedXpertQA declines. Thus, the benefit of evidence access is task-dependent rather than uniform.

(ii) Selective persistence improves knowledge utility beyond storing more triplets. Figure 3(b) shows that frontier-band admission achieves the strongest results across all five scores, with a mean of 33.46%, compared with 26.48% without persistence and 29.83% when persisting all candidates. Notably, it retains only 2,599 triplets, versus 5,014 for indiscriminate persistence. At the same memory size, random selection achieves only 22.71%, falling below even the no-persistence condition. These results indicate that the benefit of persistent knowledge depends on which triplets are admitted, rather than on memory size alone, supporting controlled accumulation over indiscriminate storage.

5.2 Training Stability and Iterative Self-Evolution

Reasoner optimization is stable across training seeds. With the Examiner-generated curriculum fixed, we conduct three independent Reasoner-training runs. Table 3(a) shows that all benchmark-level sample standard deviations are below 1 percentage point, while the four-score mean is $36.84 \pm 0.34\%$. These results support the stability of Reasoner optimization under a fixed curriculum, rather than measuring variability across the full self-evolution pipeline.

Additional rounds yield further gains. Table 3(b) shows consistent improvements over three self-evolution rounds. MedBullets op4 accuracy increases from 35.20% to 41.90%, while MMLU-Pro improves from 38.94% to 40.20%. These results support the benefit of continued self-evolution beyond a single round of adaptation.

5.3 Evaluation Robustness

The audited tool-use gain persists across answer mappers. By default, the mapper is implemented with Qwen3-4B-Base. We remap the same cached responses using Qwen3-4B-Instruct-2507. Fig-

Table 3: Training stability and iterative self-evolution. (a) Mean accuracy (%) ± sample standard deviation across three Reasoner-training seeds with a fixed curriculum; standard deviations are in percentage points. (b) Accuracy (%) across three self-evolution rounds, averaged over seeds.

(a) Multi-seed training

Benchmark	Mean ± Std.
MedBullets op4	39.09 ± 0.68
MedBullets op5	35.08 ± 0.98
MMLU-Pro (health)	40.08 ± 0.32
PubMedQA	33.10 ± 0.36
Four-score mean	36.84 ± 0.34

(b) Self-evolution rounds

Benchmark	R1	R2	R3
MedBullets op4	35.20	37.30	**41.90**
MMLU-Pro	38.94	39.02	**40.20**

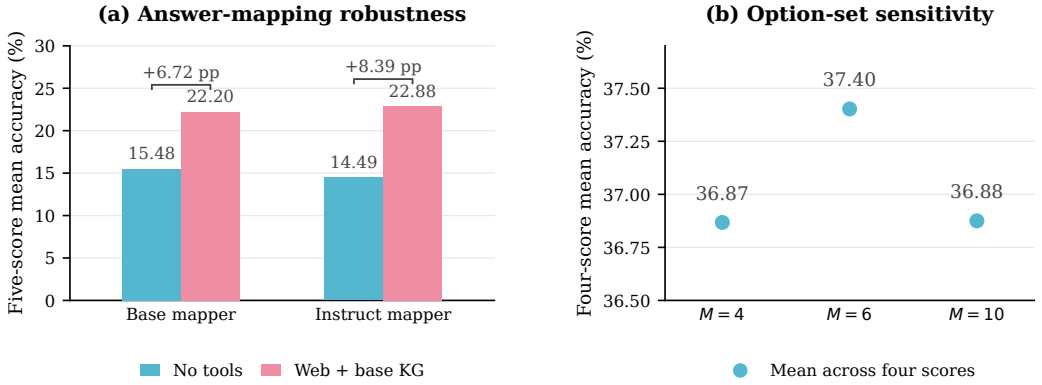

Figure 4: Evaluation robustness. (a) Remapping identical cached responses with Qwen3-4B-Base and Qwen3-4B-Instruct-2507 preserves the aggregate gain from Web access and the base KG. Brackets indicate gains in percentage points. (b) Four-score mean accuracy for option-set cardinalities $M \in \{4, 6, 10\}$, each using an independently generated and filtered curriculum.

ure 4(a) shows that the gain from Web access and the base KG remains positive under both mappers: 6.72 and 8.39 percentage points, respectively.

Aggregate performance varies little across option-set sizes. The Reasoner receives only the question stem and produces a free-form response; options are used only for subsequent verification. Using independently generated and filtered curricula with $M \in \{4, 6, 10\}$ options, the mean scores on MedBullets op4/op5, MMLU-Pro (Health), PubMedQA, ranges from 36.87% to 37.40% (Figure 4(b)). This comparison measures end-to-end sensitivity, including changes in curriculum content and retained size, rather than isolating option count alone. Detailed mapping results and option-quality audits are provided in Appendix F and Appendix G.

6 Conclusion

We presented **MedZERO**, a self-evolving framework for open-ended medical reasoning under partial verifiability. MedZERO co-evolves an *Examiner* for frontier medical challenge generation and a *Reasoner* for evidence-grounded reasoning, while using ephemeral-to-persistent knowledge accumulation to separate exploratory self-generated triplets from curated reusable knowledge. Across five medical reasoning benchmarks and two model scales, MedZERO consistently improves over both base models and self-evolving baselines. These results suggest that explicitly designing curriculum construction, supervision, and knowledge reuse for partial verifiability is important for extending self-evolution beyond easily verifiable domains and toward more realistic medical reasoning settings.

Acknowledgments and Disclosure of Funding

The work described in this paper was supported by the Research Grants Council of the Hong Kong Special Administrative Region, China, under Project T45-401/22-N. The authors declare no competing interests.

References

[1] Sanghwan Bae, Jiwoo Hong, Min Young Lee, Hanbyul Kim, JeongYeon Nam, and Donghyun Kwak. Online difficulty filtering for reasoning oriented reinforcement learning. *arXiv preprint arXiv:2504.03380*, 2025.

[2] Payal Chandak, Kexin Huang, and Marinka Zitnik. Building a knowledge graph to enable precision medicine. *Scientific data*, 10(1):67, 2023.

[3] Hanjie Chen, Zhouxiang Fang, Yash Singla, and Mark Dredze. Benchmarking large language models on answering and explaining challenging medical questions. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3563–3599, 2025.

[4] Shan Chen, Pedro Moreira, Yuxin Xiao, Sam Schmidgall, Jeremy Warner, Hugo Aerts, Thomas Hartvigsen, Jack Gallifant, and Danielle S Bitterman. Medbrowsecomp: Benchmarking medical deep research and computer use. *arXiv preprint arXiv:2505.14963*, 2025.

[5] Xilin Dang, Kexin Chen, Xiaorui Su, Ayush Noori, Iñaki Arango, Lucas Vittor, LONG XINYI, Yuyang Du, Marinka Zitnik, and Pheng-Ann Heng. Knowguard: Knowledge-driven abstention for multi-round clinical reasoning. In *International Conference on Learning Representations*, volume 2026, pages 81773–81810, 2026.

[6] Xilin Dang, Kang Li, and Pheng Ann Heng. Advancing radiograph representation learning via cascading graph alignment for vision-language clinical concepts. *Medical Image Analysis*, page 104064, 2026.

[7] Xilin Dang, Kang Li, and Pheng Ann Heng. Hivlr: Hierarchical vision-language reasoning for interpretable zero-shot radiography image understanding. *Medical Image Analysis*, page 104189, 2026.

[8] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library. *IEEE Transactions on Big Data*, 2025.

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

[10] Chengsong Huang, Wenhao Yu, Xiaoyang Wang, Hongming Zhang, Zongxia Li, Ruosen Li, Jiaxin Huang, Haitao Mi, and Dong Yu. R-zero: Self-evolving reasoning llm from zero data. 2025.

[11] Dongfu Jiang, Yi Lu, Zhuofeng Li, Zhiheng Lyu, Ping Nie, Haozhe Wang, Alex Su, Hui Chen, Kai Zou, Chao Du, et al. Verltool: Towards holistic agentic reinforcement learning with tool use. *arXiv preprint arXiv:2509.01055*, 2025.

[12] Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

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

[14] Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 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, 2019.

[15] Shreya Johri, Jaehwan Jeong, Benjamin A Tran, Daniel I Schlessinger, Shannon Wongvibulsin, Leandra A Barnes, Hong-Yu Zhou, Zhuo Ran Cai, Eliezer M Van Allen, David Kim, et al. An evaluation framework for clinical use of large language models in patient interaction tasks. *Nature medicine*, 31(1):77–86, 2025.

[16] Yubin Kim, Chanwoo Park, Hyewon Jeong, Yik S Chan, Xuhai Xu, Daniel McDuff, Hyeonhoon Lee, Marzyeh Ghassemi, Cynthia Breazeal, and Hae W Park. Mdagents: An adaptive collaboration of llms for medical decision-making. *Advances in Neural Information Processing Systems*, 37:79410–79452, 2024.

[17] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020.

[18] Junkai Li, Yunghwei Lai, Weitao Li, Jingyi Ren, Meng Zhang, Xinhui Kang, Siyu Wang, Peng Li, Ya-Qin Zhang, Weizhi Ma, et al. Agent hospital: A simulacrum of hospital with evolvable medical agents. *arXiv preprint arXiv:2405.02957*, 2024.

[19] Shuyue S Li, Vidhisha Balachandran, Shangbin Feng, Jonathan S Ilgen, Emma Pierson, Pang W Koh, and Yulia Tsvetkov. Mediq: Question-asking llms and a benchmark for reliable interactive clinical reasoning. *Advances in Neural Information Processing Systems*, 37:28858–28888, 2024.

[20] Bo Liu, Leon Guertler, Simon Yu, Zichen Liu, Penghui Qi, Daniel Balcells, Mickel Liu, Cheston Tan, Weiyan Shi, Min Lin, Wee Sun Lee, and Natasha Jaques. Spiral: Self-play on zero-sum games incentivizes reasoning via multi-agent multi-turn reinforcement learning. *arXiv preprint arXiv:2506.24119*, 2025.

[21] Hongliang Lu, Yuhang Wen, Pengyu Cheng, Ruijin Ding, Haotian Xu, Jiaqi Guo, Chutian Wang, Haonan Chen, Xiaoxi Jiang, and Guanjun Jiang. Search self-play: Pushing the frontier of agent capability without supervision. *arXiv preprint arXiv:2510.18821*, 2025.

[22] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36:46534–46594, 2023.

[23] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[24] Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Conference on health, inference, and learning*, pages 248–260. PMLR, 2022.

[25] Samuel Schmidgall, Rojin Ziaei, Carl Harris, Eduardo Reis, Jeffrey Jopling, and Michael Moor. Agentclinic: a multimodal agent benchmark to evaluate ai in simulated clinical environments. *arXiv preprint arXiv:2405.07960*, 2024.

[26] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[27] Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pages 1279–1297, 2025.

[28] Taiwei Shi, Yiyang Wu, Linxin Song, Tianyi Zhou, and Jieyu Zhao. Efficient reinforcement finetuning via adaptive curriculum learning. *arXiv preprint arXiv:2504.05520*, 2025.

[29] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023.

[30] Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Mohamed Amin, Le Hou, Kevin Clark, Stephen R Pfohl, Heather Cole-Lewis, et al. Toward expert-level medical question answering with large language models. *Nature medicine*, 31(3):943–950, 2025.

[31] Xiangru Tang, Anni Zou, Zhuosheng Zhang, Ziming Li, Yilun Zhao, Xingyao Zhang, Arman Cohan, and Mark Gerstein. Medagents: Large language models as collaborators for zero-shot medical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 599–621, 2024.

[32] Zhengwei Tao, Ting-En Lin, Xiancai Chen, Hangyu Li, Yuchuan Wu, Yongbin Li, Zhi Jin, Fei Huang, Dacheng Tao, and Jingren Zhou. A survey on self-evolution of large language models. *arXiv preprint arXiv:2404.14387*, 2024.

[33] Aaron Tu, Weihao Xuan, Heli Qi, Xu Huang, Qingcheng Zeng, Shayan Talaei, Yijia Xiao, Peng Xia, Xiangru Tang, Yuchen Zhuang, et al. Position: The hidden costs and measurement gaps of reinforcement learning with verifiable rewards. *arXiv preprint arXiv:2509.21882*, 2025.

[34] Shaobo Wang, Zhengbo Jiao, Zifan Zhang, Yilang Peng, Xu Ze, Boyu Yang, Wei Wang, Hu Wei, and Linfeng Zhang. Socratic-zero: Bootstrapping reasoning via data-free agent co-evolution. *arXiv preprint arXiv:2509.24726*, 2025.

[35] 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.

[36] Peng Xia, Kaide Zeng, Jiaqi Liu, Can Qin, Fang Wu, Yiyang Zhou, Caiming Xiong, and Huaxiu Yao. Agent0: Unleashing self-evolving agents from zero data via tool-integrated reasoning. *arXiv preprint arXiv:2511.16043*, 2025.

[37] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[38] Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*, 2024.

[39] Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837*, 2025.

[40] Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022.

[41] Andrew Zhao, Yiran Wu, Yang Yue, Tong Wu, Quentin Xu, Yang Yue, Matthieu Lin, Shenzhi Wang, Qingyun Wu, Zilong Zheng, and Gao Huang. Absolute zero: Reinforced self-play reasoning with zero data, 2025.

[42] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

[43] Yuxin Zuo, Shang Qu, Yifei Li, Zhangren Chen, Xuekai Zhu, Ermo Hua, Kaiyan Zhang, Ning Ding, and Bowen Zhou. Medxpertqa: Benchmarking expert-level medical reasoning and understanding. *arXiv preprint arXiv:2501.18362*, 2025.

Table 4: Evaluation datasets used in this work. "HF split" denotes the split name in the corresponding Hugging Face repository, not its role in our experiments. Sizes indicate the number of examples used for evaluation.

Dataset	HF split	Size	Answer format	HF repository
PubMedQA (PQA labeled)	`train`[†]	1,000	3-option MCQ	qiaojin/PubMedQA
MedXpertQA (Text)	`test`	2,450	10-option MCQ	TsinghuaC3I/MedXpertQA
MMLU-Pro (Health)	`test`	818	10-option MCQ	TIGER-Lab/MMLU-Pro
MedBullets (op4)	`op4_test`	308	4-option MCQ	mkieffer/Medbullets
MedBullets (op5)	`op5_test`	308	5-option MCQ	mkieffer/Medbullets
MedBrowseComp	`test`	50	open-ended	AIM-Harvard/MedBrowseComp

[†]`train` is the official Hugging Face split name for PubMedQA's `pqa_labeled` configuration; all 1,000 examples are used exclusively for evaluation. MedZERO uses no labeled benchmark data from any of the datasets listed above for training.

A Ethics Statement

This work studies a research framework for open-ended medical reasoning and evaluates it only on academic benchmarks. It is not intended for direct clinical use. Any real-world deployment would require prospective clinical validation across diverse patient populations, regulatory review, continuous monitoring for bias and safety, and mandatory oversight by licensed medical professionals. In particular, errors in retrieval, reasoning, or abstention behavior could cause disproportionate harm in high-stakes settings. Accordingly, **MedZERO** should not be used for medical diagnosis or treatment decisions, and any future clinical application must comply with applicable medical, legal, and ethical requirements.

B Limitations

MedZERO has several important limitations. First, although the framework is designed for partially verifiable medical reasoning, verification remains incomplete. In many cases, the correctness of a response cannot be established by exact matching alone, and supervision still depends on imperfect signals such as retrieved evidence, consistency checks, and judge-based evaluation. As a result, some reasoning errors may remain undetected during self-evolution.

Second, while MedZERO explicitly separates exploratory knowledge from curated persistent knowledge, this design does not eliminate the risk of error accumulation. Incorrect retrieved evidence, noisy self-generated rationales, or biased curation decisions may still introduce persistent mistakes into future reasoning. The framework reduces this risk but does not fully solve it.

Third, our evaluation is limited to benchmark-based settings and does not establish clinical utility. Academic medical QA benchmarks capture only part of real clinical reasoning, and they do not fully reflect longitudinal patient care, multi-modal evidence, workflow constraints, or the consequences of incorrect recommendations in practice.

Fourth, MedZERO depends on the quality of its external tools and evaluation protocol. Retrieval quality, prompt design, observation budget, and judge behavior can all materially affect outcomes. Although we report controlled settings where possible, some benchmark conditions remain sensitive to implementation choices.

These limitations suggest several directions for future work, including stronger verification mechanisms for open-ended reasoning, more reliable knowledge curation strategies, prospective human evaluation, and broader testing in realistic clinical decision-support settings.

C Datasets

See Table 4 for details.

D Experimental details

Compute. We run MedZERO on Linux servers with 4 NVIDIA L20X GPUs.

Examiner RL (verl). Unless otherwise noted, we follow the repository defaults: training with GRPO [26], global batch size 8, rollout width 4, max batched tokens 6144, max response length 512, `trainer.max_steps`=6, and vLLM memory fraction 0.2 with `max_model_len`=6144.

Reasoner RL. Reasoner training uses GRPO technique with a learning rate 10^{-6}, FSDP, TP size 1, default train batch size 64, and PPO mini-batch 64, rollout $n = 4$, validation batch size 256, maximum number of sequences=256 for rollout.

Base medical KG. We use PrimeKG [2] as the base medical knowledge graph.

E Systems Efficiency

We next evaluate whether MedZERO introduces substantial systems overhead.

(i) Controlled knowledge accumulation is not the main runtime bottleneck. Figure 5(a) shows that Examiner-side curriculum construction dominates wall-clock time in a full self-evolution iteration, while controlled knowledge accumulation accounts for only a small fraction of the total runtime. Within accumulation, the main cost comes from scoring candidate questions rather than from generation or output materialization.

(ii) The evidence-access stack has practical benchmark-scale latency. Figure 5(b) shows that the FAISS- [8] and tool-server-based setup of Tavily substantially reduces latency relative to Brightdata. Replacing slower web-search backends with Tavily improves end-to-end latency by roughly an order of magnitude and yields a tighter latency distribution, making evidence-augmented evaluation more predictable at scale.

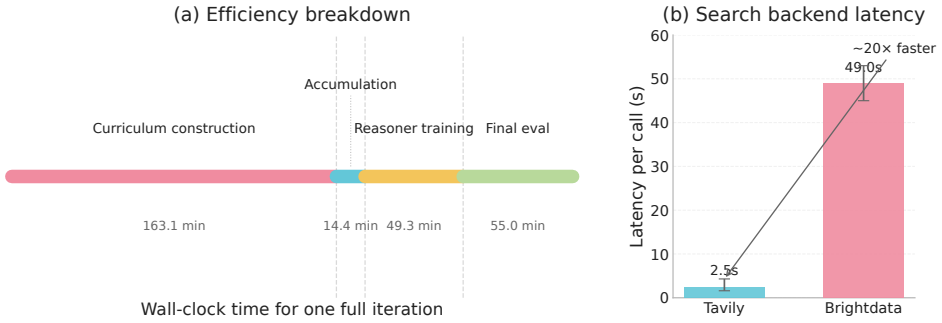

Figure 5: (a) Wall-time breakdown for a full MedZERO self-evolution iteration. (b) Web search latency per call (end-to-end wall time, measured in the same execution environment across 5 consecutive calls).

F Answer-Mapping Robustness

Open-ended reasoning and post-hoc mapping. The Reasoner receives only the question stem and generates a free-form response. It does not observe the answer options during response generation. Answer mapping is performed afterward: a separate evaluation model receives the generated response and the corresponding options, and maps the response to an option for scoring. Thus, the options serve as a downstream verification interface, rather than an exposed candidate set from which the Reasoner selects its answer.

Controlled comparison. We assess sensitivity to the answer mapper by comparing Qwen3-4B-Base with Qwen3-4B-Instruct-2507. For each evaluated system configuration, we reuse exactly the same cached Reasoner responses under both mappers; the Reasoner is neither retrained nor queried

Table 5: Answer-mapping robustness using identical cached Reasoner responses for each system configuration. Scores are five-score mean accuracy (%). Gain denotes the improvement of Web + base KG over the no-tool configuration, in percentage points.

Mapper	No tools	Web + base KG	Gain (pp)
Base	15.48	22.20	+6.72
Instruct	14.49	22.88	+8.39

Table 6: End-to-end sensitivity to option-set cardinality. Each setting uses an independently generated and filtered curriculum. Benchmark scores and the four-score mean are accuracy (%). Averaged over seeds.

Benchmark / statistic	$M = 4$	$M = 6$	$M = 10$
MedBullets op4	38.86	38.70	38.70
MedBullets op5	34.06	35.10	35.10
MMLU-Pro (health)	41.35	42.21	40.20
PubMedQA	33.20	33.60	33.50
Four-score mean	36.87	37.40	36.88
Retained curriculum items	601	458	283

again. The corresponding answer options and reference labels are also unchanged. This comparison therefore measures sensitivity to the downstream mapping stage without introducing variation from response generation.

We evaluate two configurations: a no-tool configuration and a configuration with Web access and the accumulated knowledge graph (Web + base KG). For each mapper, we report the five-score mean accuracy and the difference between the two configurations.

Results and interpretation. As shown in Table 5, Web + base KG improves the aggregate score under both mappers, with gains of 6.72 and 8.39 percentage points, respectively. The positive aggregate gain therefore persists across the two tested mapping models, although its magnitude changes.

This experiment does not establish that either mapper is error-free, nor does it demonstrate agreement on individual responses. It supports the narrower conclusion that the aggregate benefit observed in this comparison is not specific to a single tested answer mapper.

G Option-Set Sensitivity

Motivation. Although the Reasoner does not observe the answer options, option-set construction may still affect the downstream verification signal and the examples retained for training. We therefore evaluate whether varying the number of Examiner-generated options materially changes the performance of the resulting Reasoner.

Experimental design. We consider option-set cardinalities $M \in \{4, 6, 10\}$. Each setting independently generates and filters its own complete Examiner curriculum. We do not construct smaller option sets by removing distractors from a shared larger set. The resulting curricula contain 601, 458, and 283 retained items for $M = 4$, $M = 6$, and $M = 10$, respectively.

Because of computational and time constraints, we evaluate this sensitivity experiment on four benchmark scores: MedBullets op4, MedBullets op5, MMLU-Pro (health), and PubMedQA. The four-score mean is the unweighted average of these scores.

Performance sensitivity. Table 6 shows that the reported four-score means range from 36.87% to 37.40%. The means for $M = 4$ and $M = 10$ differ by only 0.01 percentage points. Increasing the number of options does not consistently improve performance across benchmarks. In particular,

Table 7: Option-quality audit of 200 randomly sampled retained examples per setting. All entries are percentages within the audited samples, not population-level guarantees.

Metric	$M = 4$	$M = 6$	$M = 10$
Correct-answer coverage	100.0	100.0	100.0
Near-equivalent-option rate	0.0	0.5	0.0
Answer-leakage rate	0.0	0.5	0.0
In-band rate	100.0	100.0	100.0
Format-valid rate	100.0	99.0	100.0

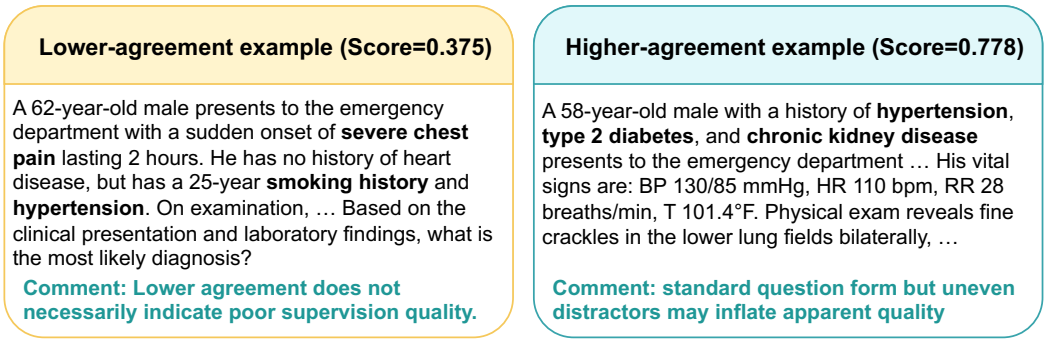

Figure 6: Failure cases under controlled knowledge accumulation. These examples illustrate that option-level agreement is an imperfect proxy for supervision quality.

MMLU-Pro (health) and PubMedQA attain their highest scores at $M = 6$, whereas MedBullets op4 is highest at $M = 4$ and MedBullets op5 is tied between $M = 6$ and $M = 10$.

Option-quality audit. We randomly sample 200 retained examples from each cardinality setting and audit five properties: correct-answer coverage, near-equivalent options, answer leakage, frontier-band membership, and format validity. Table 7 reports the observed rates within these samples.

Audit results. All audited samples contain a correct-answer option and satisfy the frontier-band criterion. Observed format validity is 99–100%, while the near-equivalent-option and answer-leakage rates are at most 0.5%. These findings indicate few observed option-quality issues among the sampled retained examples. They do not imply that every retained example is free of such issues.

Scope and limitations. This experiment evaluates end-to-end sensitivity to option-set cardinality, rather than isolating the causal effect of option count alone. Because each setting independently generates and filters its curriculum, curriculum content and retained size also vary across settings. Moreover, the comparison does not provide repeated-run uncertainty estimates for each cardinality and therefore does not establish statistical equivalence. Within these limitations, the small observed variation in aggregate performance supports robustness across the three tested option-set settings.

H Failure Cases

Failure case: score does not always align with perceived supervision quality

At the same time, the curation score is not a perfect proxy for human-perceived example quality. Figure 6 shows representative cases where score and perceived usefulness do not fully align. Some lower-scoring examples still contain meaningful clinical structure, while some higher-scoring examples may appear cleaner mainly because they better match the reward signal or evaluation heuristic. This failure mode highlights an important limitation of controlled knowledge accumulation: it can reshape the retained supervision distribution in useful ways, but it can also inherit biases or blind spots from the scoring and filtering procedure.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Abstract and Introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Appendix. B

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: We do not provide theoretical results in this paper.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Experiments section 1. 4.1. Experimental Settings

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The datasets are publicly available and listed in Appendix C. Code available at `https://github.com/IcecreamArtist/MedZERO`.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Experiment section 1, 4.1. Experimental Settings.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Analysis section 5.2.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix D. Experimental Details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Yes.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Appendix A. Ethics Statement

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: We stated this in the Ethics Statement section.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have cite all related used datasets in this work.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: We do not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: 4.1. Experimental Settings.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.