When more supervision hurts: reward design for structured biomedical reasoning in 4B models

Batuhan Nursal, **Christophe Ye**, **Xinyu Guo**,
Xin He, **Cassie S. Mitchell**
Georgia Institute of Technology
{bnursal3, cye73, xguo381, xhe383}@gatech.edu,
cassie.mitchell@bme.gatech.edu

Abstract

Reinforcement learning has become a common approach for improving language-model reasoning, but its effectiveness depends heavily on reward design, especially for compact domain-specialized models. We study structured biomedical question answering with two 4B-scale models, MedGemma-1.5-4B-it and Qwen3-4B-Thinking-2507, using a two-stage SFT+GRPO pipeline that trains models to produce tagged think/answer outputs. We compare three reward schemes: a simple format + accuracy reward, a depth incentivized reward that encourages multiple reasoning cycles, and a gated LLM-judge reward for intermediate reasoning milestones. Across five biomedical QA benchmarks, the simplest reward is the most robust: it substantially improves MedGemma in the structured-output setting, while depth and LLM-judge rewards often reduce accuracy. For Qwen3, SFT or native long-budget inference can outperform GRPO on several benchmarks, showing that reasoning-tuned models may not benefit from additional structural reinforcement. Our results suggest that, for lightweight biomedical QA, reward simplicity and model priors are more important than increasingly elaborate intermediate reasoning supervision. The codebase for this study is publicly available on GitHub: https://github.com/batuinmetz/When-more-supervision-hurts.

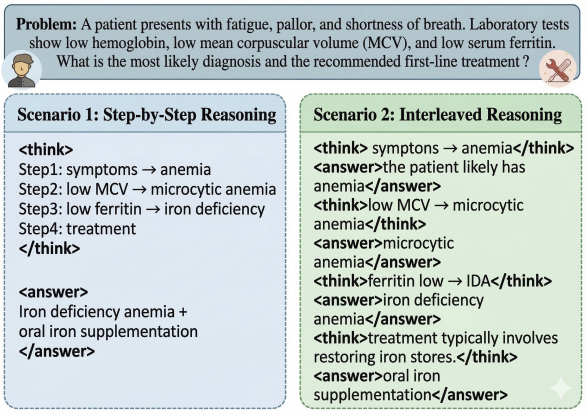


Figure 1: Interleaved reasoning (right) alternates multiple <think> and <answer> segments, allowing the model to refine its reasoning step by step, leading to improved reasoning control at verification stages.

1 Introduction

The emergence of Large Language Models (LLMs) has revolutionized natural language understanding, yet achieving multi-step reasoning remains a significant challenge in domains like medicine (Yu et al., 2025). While massive frontier models exhibit strong clinical logic, their deployment is often hindered by prohibitive infrastructure costs and strict data privacy compliance (Tahera et al., 2026). Consequently, recent research has shifted toward optimizing compact, resource-efficient models (sub-8B parameters). To unlock advanced reasoning capabilities in these lightweight architectures without the massive computational overhead of traditional Actor-Critic frameworks, post-training reinforcement learning (RL) methods—most notably Group Relative Policy Optimization (GRPO)—have become highly attractive (Shao et al., 2024). By eliminating the need for a separate value function, GRPO democratizes the alignment of reasoning paths on accessible hardware. However, the exact "recipe" for successfully applying GRPO to compact, domain-specific models remains underexplored, and it is unclear which reward strategies effectively stabilize training in restricted parametric regimes.

A critical design choice in RL-based reasoning is the formatting of the Chain-of-Thought (CoT) (Wei et al., 2023). Traditional models utilize a linear structure (e.g., <think> ... </think> <answer> ... </answer>) which introduces operational vulnerabilities: intermediate reasoning steps remain opaque during training and inference,

Findings of the Association for Computational Linguistics: IJCNLP-AACL 2026, pages 2851–2866
November 6-10, 2026 ©2026 Association for Computational Linguistics

and the lack of structural boundaries can encourage excessively long, unstructured token generation. An emerging alternative is interleaved reasoning (Xie et al., 2026), where reasoning traces are systematically interspersed with intermediate partial answers. We hypothesize that while an interleaved format provides a more granular, auditable signal for optimization, this added structural complexity may also overconstrain the limited capacity of compact models, potentially disrupting their optimization dynamics.

This structural tension introduces fundamental empirical questions: Do simple outcome-and-format rewards suffice to guide compact models through structured reasoning, or do they require complex, step-level logical rewards? Furthermore, does the efficacy of these reward schemes depend on the model's initialization prior—specifically, whether the model is already domain-specialized or natively reasoning-tuned?

In this paper, we systematically investigate these questions by adapting the medical question-answering dataset MedReason (Wu et al., 2025) into an interleaved reasoning framework. Our empirical findings reveal that simple rewards are consistently the most robust; basic format and outcome correctness prove more effective than complex, depth-incentivized or LLM-as-a-judge logic rewards within our evaluation setting. Crucially, we show that model priors heavily dictate post-training behavior: GRPO improves performance for MedGemma under the structured-output evaluation setting, but effects are mixed for Qwen3. While our best 4B structured-output models are competitive with some reported biomedical baselines, we note that these comparisons are not fully controlled across prompts, datasets, and decoding budgets. Ultimately, our analysis suggests that imposing excessive structural complexity on compact models can lead to lower held-out benchmark accuracy in our evaluation setting without reliably inducing better multi-step reasoning.

Our core contributions are as follows:

- We are the first to adapt the MedReason (Wu et al., 2025) QA dataset into a structured format that includes interleaved reasoning and intermediate answers, providing a benchmark for granular supervision in medical logic.
- We compare SFT and GRPO post-training for two contrasting 4B models: a domain specialized medical model (MedGemma-1.5-4B-Instruct) (Sellergren et al., 2025) and a general-reasoning tuned model (Qwen3-4B-Thinking-2507) (Yang et al., 2025).
- We evaluate three reward designs (simple format correctness, depth incentivized reasoning, and gated LLM-Judge logic reward) and show that the simplest objective is consistently the most robust.
- We show that model priors strongly affect post-training behavior: Medgemma benefits from simple GRPO in the structured-output setting, while Qwen3 often retains stronger performance under SFT or native long-budget inference.
- We analyze token usage, parseability, and reasoning-format behavior to show that higher structural complexity can reduce accuracy without reliably inducing better multi-step reasoning.

Beyond immediate performance metrics, our work suggests that tagged intermediate outputs may make future verification mechanisms easier to apply, enabling more controllable biomedical QA systems; we note that absolute clinical reliability is outside the scope of this benchmark study.

2 Related Works

The development of Large Reasoning Models (LRMs) has advanced from test-time prompting strategies like Chain-of-Thought (CoT) (Wei et al., 2023), self-consistency (Wang et al., 2023), and least-to-most prompting (Zhou et al., 2023) to direct post-training via reinforcement learning (RL). Algorithms like Group Relative Policy Optimization (GRPO) mitigate traditional actor-critic computational overhead by removing the value function (Shao et al., 2024), making local reasoning alignment viable. However, designing effective reward signals remains a challenge, shifting from human preference models (Ouyang et al., 2022; Stiennon et al., 2022) to scalable AI feedback and LLM-as-a-judge frameworks (Li et al., 2024). For complex execution paths, reward architectures have expanded from simple outcome-based correctness to structured interleaved reasoning—alternating discrete reasoning blocks and intermediate partial answers (Xie et al., 2026). While structured tracking provides highly granular optimization signals, it introduces a formatting burden that may overconstrain

smaller architectures. The precise performance trade-offs of these structured supervision formats on compact models remain largely unquantified.

In parallel, specialized instruction-tuned architectures like Med-PaLM 2 (Singhal et al., 2023) and MedGemma 1.5 (Sellergren et al., 2025) have demonstrated strong clinical question-answering capabilities. However, the operational deployment of massive frontier models is frequently limited by strict data privacy constraints and prohibitive infrastructure costs (Tahera et al., 2026), driving research toward parameter-efficient, sub-8B specialized models. Despite these concurrent trends, few studies examine the exact intersection of post-training RL, structured reasoning constraints, and compact domain initializations. Existing reasoning frameworks often depend on large proprietary models or massive compute budgets, limiting their translation to localized, domain-specific settings. Furthermore, it is unclear how pre-training variables alter optimization stability when rigid structural formats are enforced. Our work addresses this gap by analyzing how reward complexity and interleaved reasoning constraints affect the post-training optimization dynamics of lightweight biomedical language models within a controlled evaluation setting.

3 Methodology

3.1 Preprocessing datasets for post-training

To develop a lightweight model capable of interleaved reasoning, we curated a post-training dataset derived from MedMCQA (Pal et al., 2022) and MedReason. Our preprocessing pipeline focuses on transforming standard medical QA pairs into a structured interleaved format.

- **MedMCQA:** We sampled 3,500 instances from the training set. For each instance, we prompted a Qwen2.5-7B-Instruct model to synthesize a comprehensive reasoning trace using the provided answer and explanation columns.
- **MedReason:** This dataset natively includes reasoning traces for 32,700 samples. We utilized the instruct model to reformat these traces into our target structure and we exclude Huatuo samples and unlabeled PubMedQA samples.

Structure: All reasoning traces were converted into an interleaved <think> ... </think> <answer> ... </answer> format. After a quality filtering step to remove incorrect or malformed generations, we retained 2,727 samples from MedMCQA and 23,758 from MedReason. Detailed curation, prompts and hyperparameters for this stage are provided in Appendix A.4.

Source Dataset	Samples
MedQA (Jin et al., 2020)	8,016
MedMCQA	6,197
PubMedQA (Jin et al., 2019)	10,444
MMLU (Hendrycks et al., 2021)	827
Huatuo (Chen et al., 2024)	6,475
MedXpert (Zuo et al., 2025)	666
HLE (Phan et al., 2026)	57
Total	**32,682**

Table 1: Source datasets used to construct MedReason and their number of samples.

3.2 Training procedure

Unlike traditional reasoning models that typically generate a single, lengthy reasoning block followed by a final answer, our objective is to induce multi-stage interleaved reasoning. To do so, we adopt a similar methodology explored by Xie et al. (2026) on more general case question answering for biomedical question answering. More precisely, we adopt a 2 stage training approach instead of 1 RL-based training stage with different reward configurations. During both the SFT and RL stages, the model is initialized with the following system prompt:

System Prompt

"You are a helpful medical assistant. You reason through problems step by step. You conduct your reasoning within `<think></think>` *and share partial answers within* `<answer></answer>` *tags. Always START with* `<think>` *and END with* `</answer>`*."*

A full example of the prompt-response interaction is provided in Appendix A.5.

3.2.1 Supervised Fine-Tuning coldstart

We first conduct an SFT "cold-start" phase using the processed subset of MedMCQA to stabilize the model's output format and align it with the interleaved structure.

3.2.2 Reinforcement Learning with GRPO

Following SFT, we employ the Group Relative Policy Optimization (GRPO) algorithm through different reward combination. In order to make GRPO

more effective we try at best to give a fair gradient of rewards among samples generated within a group. The details of the hyperparameters are provided in the Appendix A.1 and A.2. We specifically employ the following series of rewards:

Interleaved Format Reward: To guide the model toward a structured, iterative reasoning process, we define two distinct format rewards r_{base} and r_{depth}.

Let $T = (t_1, t_2, \ldots, t_n)$ be the sequence of tags extracted from the model's completion, where $t_i \in \{\texttt{<think>}, \texttt{</think>}, \texttt{<answer>}, \texttt{</answer>}\}$.

A reasoning cycle is the four-tag pattern $S = (\texttt{<think>}, \texttt{</think>}, \texttt{<answer>}, \texttt{</answer>})$. A completion is well-formed if T is exactly C ≥ 1 consecutive repetitions of S, i.e. $T = S^C$. Every well-formed completion therefore starts with <think> and ends with </answer>, with no nested, missing or reordered tags.

The two rewards differ only in how they treat the number of cycles. r_{base} checks well-formedness alone: any valid interleaved output receives the same reward, whether it has one cycle or multiple, so the model picks its own reasoning depth. r_{depth} adds a bonus that grows with the number of valid cycles and stops growing at five cycles. This pushes the model toward multi-step decompositions without rewarding it for padding the output with extra cycles. Comparing the two lets us separate the effect of enforcing the format from the effect of incentivizing depth.

Base Format Reward (r_{base}): The primary objective of r_{base} is to enforce basic structural syntax. We define it as such:

$$r_{\text{base}} = \begin{cases} -2.0 & \text{if } T = \emptyset \\ -1.0 & \text{if } |T| \pmod 4 \neq 0 \\ -1.0 & \text{if } T \text{ is not a sequence of } S \\ 1.0 & \text{otherwise} \end{cases}$$

Cycle Depth Reward (r_{depth}): To investigate the effect of reasoning depth, we introduce r_{depth}, which incentivizes the generation of multiple interleaved cycles. For a well-formed completion ($T = S^C$), $C = |T|/4$ is exactly the number of complete reasoning cycles. C is only defined for well-formed completions. Malformed completions are handled by the penalty cases below, which are checked first.

$$pen = \begin{cases} -2.0 & \text{if } T = \emptyset \\ -1.5 & \text{if } |T| \pmod 4 \neq 0 \\ -1.0 & \text{if } T \text{ is not a sequence of } S \end{cases}$$

$$r_{\text{depth}} = \begin{cases} pen & \\ 0.1 & \text{if } C = 1 \\ 1.0 + \min(0.2C, 1.0) & \text{if } C \geq 2 \end{cases}$$

By providing a capped bonus ($\min(0.2C, 1.0)$), we encourage the model to explore deeper reasoning-answer chains while preventing "degenerate looping" behavior where the model might generate excessive empty cycles to maximize reward.

Accuracy Reward (r_{acc}): To ensure that the model ultimately produces the correct final answer, we define a discrete *accuracy reward* based on the content enclosed within the <answer></answer> tags. Let $\hat{y}$ denote the predicted answer extracted from the final <answer></answer> block and y the ground-truth label. We first parse all occurrences of answer blocks and select the last one as the final prediction. If no such tag is found, the model receives a penalty of -2.0, discouraging malformed outputs.

From the extracted answer span, we search for a valid multiple-choice option in $\{A, \ldots, J\}$. The reward function is defined as:

$$r_{\text{acc}} = \begin{cases} 2.0 & \text{if } \hat{y} = y, \\ 0.0 & \text{if } \hat{y} \neq y, \\ -2.0 & \text{if no valid answer tags} \end{cases}$$

This reward strongly incentivizes correctness while explicitly penalizing outputs that fail to comply with the required tagging structure.

Logic Milestone Reward with LLM Judge (r_{logic}): To evaluate the semantic quality of intermediate reasoning steps, we introduce a *logic milestone reward* computed using an external LLM judge (Qwen3-4B-Instruct in this case). For each generated reasoning step j, consisting of a <think></think> block and its corresponding <answer></answer> block, we compare it against a set of N gold reasoning milestones extracted from the teacher solution.

Each step is independently evaluated by the judge and assigned a score $s_j \in \{0.0, 0.5, 1.0\}$ according to the following criteria:

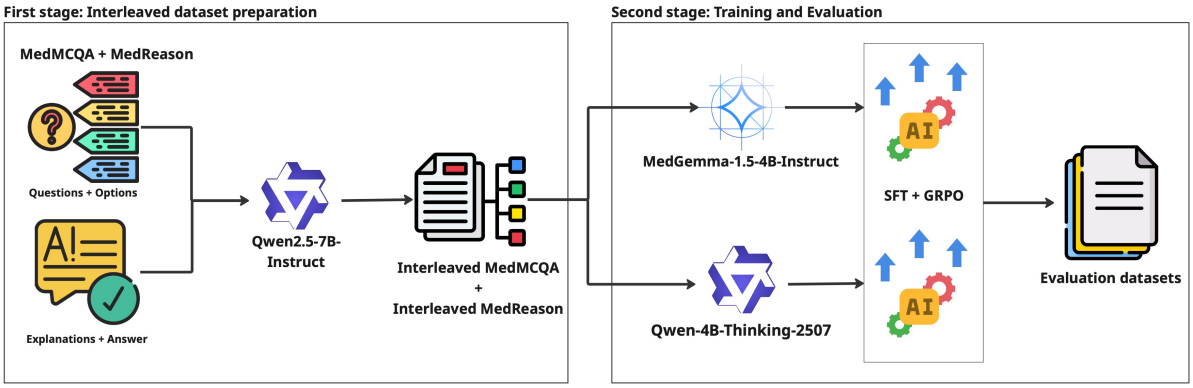


Figure 2: The proposed two-stage framework: (1) In the first stage, we generate interleaved reasoning subsets from the baseline MedMCQA and MedReason datasets. Utilizing the question, options, answer, and explanation fields, we reconstruct the data into the target interleaved reasoning format. (2) In the second stage, we evaluate one domain-specific instruction-tuned model and one general-purpose reasoning model. These models are trained using the various reward configurations detailed in Section 4. The specific prompt templates are provided in Appendix A.

- 1.0: The step precisely matches a specific milestone in the reference reasoning.
- 0.5: The step is partially correct or references a relevant concept but lacks precision.
- 0.0: The step is irrelevant or contradicts the reference reasoning.

Let M denote the number of generated reasoning steps and k_j the index of step j (starting at 0). To prioritize earlier reasoning steps, we apply harmonic weighting and compute an intermediate score:

$$S = \sum_{j=0}^{M-1} \frac{s_j}{k_j + 1}.$$

To avoid rewarding redundant repetitions of the same milestone, we additionally enforce a *saturation constraint*: once N high-confidence milestone matches have been identified, additional perfect matches are ignored. This prevents artificially inflated rewards through repeated restatement of previously covered reasoning milestones.

Finally, we incorporate an *Occam's Razor efficiency factor* that penalizes unnecessarily long reasoning chains. Let U denote the number of unique milestone matches identified by the judge. The efficiency term is defined as:

$$\eta = \frac{U}{M}.$$

The final logic reward is computed as:

$$r_{\text{logic}} = \min\left(\frac{S}{N} \cdot \eta,\ 1.0\right).$$

This formulation encourages reasoning trajectories that are both *correct* and *efficient*: models are rewarded for identifying key reasoning milestones early while avoiding redundant or unnecessarily verbose intermediate steps.

4 Experimental Settings

Our experiments aim to quantify how varying reward configurations influence the performance of 4B-parameter Large Language Models (LLMs) on complex biomedical reasoning tasks. We evaluate two distinct architectures: **MedGemma-1.5-4B-it**, an instruction-tuned model specialized for the medical domain, and **Qwen3-4B-Thinking-2507**, a general-domain reasoning model.

4.1 Evaluation Benchmarks

We assess model performance across a comprehensive suite of biomedical QA datasets, covering multiple-choice questions, professional licensing exams, and research-based reasoning:

- **MedMCQA**: Multi-subject medical questions (4183 samples / dev set).
- **MedQA**: USMLE-style professional medical exams (1273 samples / test set).
- **MMLU-Pro**: Biology and health subsets; a more challenging iteration of MMLU (1535 samples / test set).
- **PubMedQA**: Biomedical research question answering (500 samples / test set).
- **MedXpertQA**: Benchmarking expert-level medical knowledge (2450 samples / text test set).

Model	MedMCQA	MedQA	MMLU-Pro (Bio+Health)	PubMedQA	MedXpertQA
MEDGEMMA-1.5-4B-IT					
base	0.465 ± 0.001	0.450 ± 0.011	0.358 ± 0.006	0.579 ± 0.005	0.101 ± 0.005
SFT (MedMCQA coldstart)	0.476 ± 0.002	0.549 ± 0.014	0.356 ± 0.013	0.504 ± 0.000	0.130 ± 0.000
SFT (MedReason)	0.450 ± 0.002	0.452 ± 0.009	0.277 ± 0.005	0.565 ± 0.005	0.114 ± 0.005
S1-simple	**0.562 ± 0.004**	**0.672 ± 0.012**	**0.567 ± 0.004**	**0.657 ± 0.004**	**0.166 ± 0.001**
S2-depth	0.547 ± 0.004	0.657 ± 0.005	0.353 ± 0.007	0.619 ± 0.001	0.114 ± 0.004
S3-gated	0.546 ± 0.001	0.651 ± 0.006	0.315 ± 0.005	0.633 ± 0.010	0.110 ± 0.004
QWEN3-4B-THINKING-2507					
base	0.325 ± 0.004	0.244 ± 0.011	0.237 ± 0.007	0.367 ± 0.011	0.010 ± 0.003
SFT (MedMCQA coldstart)	0.557 ± 0.004	**0.623 ± 0.002**	**0.467 ± 0.003**	**0.764 ± 0.017**	0.117 ± 0.001
SFT (MedReason)	0.541 ± 0.007	0.608 ± 0.004	0.400 ± 0.003	0.763 ± 0.001	**0.125 ± 0.002**
S1-simple	**0.568 ± 0.008**	0.560 ± 0.008	0.415 ± 0.000	0.735 ± 0.007	0.124 ± 0.004
S2-depth	0.553 ± 0.012	0.564 ± 0.005	0.129 ± 0.002	0.673 ± 0.005	0.098 ± 0.002
S3-gated	0.544 ± 0.005	0.563 ± 0.003	0.120 ± 0.003	0.637 ± 0.000	0.098 ± 0.000

Table 2: Performance comparison across reward configurations (accuracy). We use the same system prompt and parameters for all evaluations as disclosed in Appendix A.3. Medgemma benefits from GRPO across all datasets with a better alignment on the first setting. However, Qwen3 does achieve interesting performances accross all datasets with a classical SFT on MedReason or on a subset of MedMCQA which highlights how reasoning models can align their prior reasoning abilities to a new task with simple SFT instead of a destructive alignment enforced with GRPO. Baseline comparison is given in Appendix

4.2 Reward Configurations

To isolate the impact of different training objectives, we define three primary experimental setups:

Setup 1: Simple Alignment. This configuration serves as our control, utilizing a combination of the base format reward (r_{base}) and a standard correctness-based accuracy reward (r_{acc}).

Setup 2: Depth-Induced Reasoning. We investigate the effect of incentivizing multi-stage reasoning by replacing r_{base} with the cycle depth reward (r_{depth}). This setup rewards the model for generating up to five interleaved cycles while maintaining the r_{acc} objective.

Setup 3: Gated Logic Reinforcement. Building upon Setup 2, we introduce an intermediate logic reward (r_{logic}) to validate the reasoning quality of intermediate steps against reference cycles from MedReason. Following Xie et al. (2026) and Yuan et al. (2025), we implement a *gated condition* for r_{logic}. The reward is only activated when the current completion satisfies the format and accuracy requirements, and the current batch accuracy exceeds the moving average of previous batches.

5 Results and Discussion

Our analysis spans two distinct evaluation regimes: **Regime A** (structured-output constrained), which enforces an identical interleaved system instruction, strict regex parsing, and a 3000-token limit to measure format compliance; and **Regime B** (native inference baseline), which utilizes standard unconstrained prompts to assess raw question-answering capabilities.

5.1 Simple GRPO improves structured-output alignment, but gains are model dependent

We evaluate all configurations under **Regime A** using the parameters detailed in Appendix A.2. As shown in Table 2, GRPO improves MedGemma across all benchmarks, with S1-simple performing best. For Qwen3, however, the picture is mixed: S1 improves average performance under the structured prompt, but SFT on MedMCQA and MedReason remains stronger on all datasets except MedMCQA, and native long-budget inference is competitive or stronger on several benchmarks. It should be noted that the lower performance observed for the base models is partly due to misalignment in prompt formatting and token length limitations. These factors prevent the models from reaching their ideal performance; both points are discussed further in the following subsections.

5.2 Token budget reveals a trade-off between native reasoning and structured-output alignment

Token usage trends (Table 3) highlight clear behavioral shifts. Base models consistently saturate the 3000-token cap under structural constraints.

Configuration	Qwen3	MedGemma-1.5
Base	2809.11	1721.15
SFT(MedReason)	310.42	838.69
S1-simple	392.13	446.06
S2-depth	334.76	545.39
S3-gated	371.85	449.37

Table 3: Average token usage across five model configurations for Qwen3 and MedGemma-1.5. Post-trained models achieve significantly lower answer lengths than the baseline. SFT-tuning is not as effective for reducing token usage on Medgemma-1.5-4B-Instruct compared to Qwen3-4B-Thinking.

Test set	Qwen3	MedGemma-1.5
MedMCQA	51.1	18.5
MedQA	75.2	42.0
MMLU-Pro	69.5	30.1
PubMedQA	57.2	15.0
MedXpertQA	94.5	47.8

Table 4: Answer tags parsing failures for base models in %. Values indicates a strong failure at generating valid answer tags with the correct answer letter for all test sets. Base models often achieves lower successes at generating an option letter within `<answer></answer>` tags.

SFT post-training drastically compresses generation lengths by enforcing direct, efficient reasoning paths. Under GRPO, token usage rises marginally above SFT levels, reflecting an active exploration of alternative trajectories.

Crucially, higher token budgets under GRPO do not uniformly improve performance. For instance, as highlighted in Table 2, Qwen3-4B-Thinking-2507 displays lower held-out accuracy than its SFT counterparts on most sets except MedMCQA. This may point to source distribution alignment: the MedReason training mixture shares extensive task formatting and domain logic with the MedMCQA training distribution. When a model's prior is already optimized for an archetype, GRPO's exploration incentives can introduce redundant intermediate steps, inflating token consumption without improving performance. On more complex benchmarks like MedXpertQA, however, this exploratory behavior remains beneficial. For MedGemma, SFT reduces length less aggressively, but GRPO triggers a sharp token increase, showing that exploration dynamics are highly architecture-dependent.

5.3 Base Model Capacity Under Unconstrained Inference (Regime B)

To isolate raw reasoning capacity from structural formatting penalties, we evaluate the unconstrained base models (**Regime B**) using a standard step-by-step prompt under 3000 and 8000 token generation limits. Now we use the following system prompt followed by the question and options:

System Prompt

"You are a helpful medical assistant. Answer the following medical question by reasoning step by step, then provide your final answer in the format: Final answer: X (where X is the option letter)."

The resulting accuracies across a single run at temperature 0 are detailed in Table 5. Complementary evaluation metrics, including parsing failure rates and average generation word counts for this regime, are provided in Table 11 and Table 12 in the Appendix A.6.

	Qwen3		MedGemma-1.5	
Dataset	3000 tok	8000 tok	3000 tok	8000 tok
MedMCQA	0.544	0.631	0.564	0.583
MedQA	0.417	0.762	0.639	0.643
PubMedQA	0.688	0.740	0.644	0.644
MMLU-Pro	0.562	0.756	0.556	0.563
MedXpertQA	0.056	0.178	0.153	0.168

Table 5: Accuracy of base models on medical QA benchmarks across token generation limits under regime B.

As shown in Table 5, liberating the models from rigid interleaved constraints significantly alters the performance hierarchy. Under a restricted 3000-token budget, Qwen3-4B experiences severe truncation, suffering from parsing failures as high as 68.9% (on MedXpertQA) because its native, highly verbose reasoning path is cut off mid-stream. However, expanding the budget to 8000 tokens resolves these failures (dropping unparseable rates below 2%) and unlocks its latent performance, allowing it to handily surpass both its structured-output counterparts and MedGemma-1.5 across almost all datasets (e.g., reaching 76.2% on MedQA). In contrast, MedGemma-1.5 natively generates more concise tracks (typically under 1200 words), making it robust to tighter token budgets. Increasing its limit to 8000 tokens yields negligible accuracy gains (+1–2%). This highlights a crucial trade-off: while post-training alignment (Regime A) is essential for

Model 1: MedGemma-1.5-4B-Instruct — S1-simple

<think> First, I need to recall the key characteristics of myelinated nerve fibers compared to non-myelinated fibers [...] Since myelin insulates the nerve, local anesthetics are effective in blocking nerve conduction in myelinated fibers. So, statement D is false. Therefore, A and D are not true statements. </think>

<answer> A </answer>

Model 2: Qwen3-4B-Thinking-2507 — S1-simple

<think> Identify characteristics of myelinated fibers </think>

<answer> Myelinated nerve fibers consist of myelin sheaths derived from Schwann cells [..] </answer>

<think>Next, I identify which statement does not align with the characteristics of myelinated nerve fibers.</think>

<answer>Option A states that [...] which is incorrect since myelinated fibers are faster.</answer>

<think>Next, I identify additional incorrect information from the remaining options.</think>

<answer> [..] </answer>

<answer> A </answer>

Figure 3: Example of a sample from MedMCQA illustrating two distinct reasoning formats under the `S1-simple` reward configuration. MedGemma produces a single linear reasoning block, whereas Qwen3 exploits the format to perform multi-cycle interleaved reasoning.

deploying MedGemma-1.5 efficiently within rigid structures, a natively reasoning-tuned model like Qwen3 achieves its optimal held-out benchmark accuracy when allowed an extended, unconstrained token budget during native inference.

5.4 Rewards impact on answer coverage: a 'Bag-of-reward' approach

To evaluate how consistently different reward strategies locate correct answers across evaluation iterations, we evaluate the union accuracy across runs. This establishes an answer coverage per question, formalized as a cross-reward "Bag-of-Rewards" ensemble. As shown in Table 6, combining outputs across distinct reward functions (Cross-Reward) consistently outperforms the self-consistency pooling of any single configuration. This indicates that specific reward formulations steer the policy toward distinct parametric regions, prioritizing different logical sub-problems and offering a complementary optimization effect.

5.5 Simple rewards do not necessarily induce multi-cycle interleaving

We look beneath accuracy metrics to inspect the qualitative reasoning paths induced by the robust S1-simple reward. Interestingly, minimizing the reward signal down to basic outcome and format correctness does not guarantee uniform structural behaviors. As illustrated in Figure 3, MedGemma-1.5-4B-Instruct optimized under S1-simple defaults to a classical linear structure, generating a single <think> block before answering. Conversely,

Model	Dataset	S3	S1	S2	Cross-Reward
Qwen3-4B	MedMCQA	64.93	71.65	70.79	73.12 ± 1.81
	MedQA	66.54	62.29	65.91	78.83 ± 0.64
	PubMedQA	62.00	75.80	70.00	76.64 ± 0.58
	MMLU-Pro	66.25	66.58	71.79	77.83 ± 0.16
	MedXpertQA	16.86	17.35	18.37	27.27 ± 0.22
MedGemma-4B	MedMCQA	71.38	73.13	72.79	73.80 ± 0.44
	MedQA	80.20	82.17	79.97	81.81 ± 0.49
	PubMedQA	73.00	72.60	71.60	77.66 ± 0.37
	MMLU-Pro	68.40	71.21	70.03	71.12 ± 0.49
	MedXpertQA	21.88	23.80	21.27	27.82 ± 0.24

Table 6: Oracle ensemble accuracy (%): self-consistency (SC, 3 runs same reward) vs. cross-reward (1 run from each of 3 setups, mean ± std over 27 combinations).

Qwen3-4B-Thinking-2507 dynamically performs multi-cycle interleaving. This divergence suggests that structured multi-step execution depends heavily on pre-existing reasoning priors rather than the post-training optimization target alone. Notably, both architectures frequently select correct final options despite intermediate logical missteps, showing that complex reasoning topologies do not always linearly map to downstream held-out accuracy.

6 Future Directions

Our investigation demonstrates that 4B-parameter models can align to structured reinforcement learning targets relative to their base performance. Future work will focus on test-time verification and dynamic error correction. Specifically, we aim to develop real-time detection mechanisms to catch fallacious reasoning steps and inject corrective sig-

nals during token generation. Shifting from static post-training to active test-time verification offers a clear pathway toward more controllable biomedical QA systems.

7 Conclusion

This work evaluates interleaved reasoning architectures for compact models under constrained biomedical settings. Utilizing a two-stage SFT and GRPO framework, we demonstrate that both domain-specialized and general reasoning-tuned initializations can be aligned to a structured format. While simple outcome-driven rewards consistently prove the most robust, architectural priors heavily dictate optimization dynamics: GRPO yields clear formatting benefits for MedGemma-1.5-4B-Instruct, whereas Qwen3-4B-Thinking-2507 remains highly competitive under SFT or unconstrained native inference alone. Ultimately, our best 4B structured-output configurations are competitive with reported biomedical baselines (Zhang et al., 2024), offering a scalable path toward more parseable and auditable biomedical QA systems.

Limitations

Despite the insights offered by our evaluation of interleaved reasoning, several key limitations bound our findings. First, our investigation was restricted to models in the 4B-parameter range; while this highlights the capacity of lightweight architectures, the scalability of these reward configurations to significantly smaller or larger models remains to be verified. Second, our training and evaluation focused engineering exclusively around multiple-choice question answering (MCQA) formats. While MCQA provides a highly structured, reproducible benchmark for quantifying domain knowledge under controlled variables, it does not fully capture the nuances of open-ended clinical reasoning or real-world patient consultations; thus, performance on these static benchmarks should not be interpreted as an indicator of absolute clinical safety or readiness for non-simulated deployment. Third, our training rationales were generated or reformatted via an automated LLM pipeline, meaning the underlying dataset may inherently contain latent teacher errors or unfaithful logic. More broadly, while the interleaved structure aims to make reasoning tracks more auditable, a model achieving a correct final outcome does not prove that the intermediate `<think>` or intermediate verification steps were causally responsible for that answer, or that the generation path is entirely faithful to clinical reality. Fourth, our optimization exploration was confined to three specific GRPO reward designs, and our logic-based reward scheme relied heavily on a compact external LLM-as-a-judge. The lightweight judge used during training may introduce noisy supervision signals, particularly through occasional mis-scoring of intermediate reasoning steps; a small exploratory audit suggested only modest agreement with reference annotations, highlighting how point estimates remain highly sensitive to sample variance in restricted evaluations (see Appendix A.7). Finally, our core empirical observations regarding structural over-constraint are fundamentally dependent on our specific evaluation conditions, including our constrained structured-output prompts and a strict 3000-token decoding budget. Modifying these prompt configurations or altering the decoding limits could yield different behavior, parseability rates, and downstream accuracy across both the domain-specialized and reasoning-tuned baselines.

References

Junying Chen, Zhenyang Cai, Ke Ji, Xidong Wang, Wanlong Liu, Rongsheng Wang, Jianye Hou, and Benyou Wang. 2024. Huatuogpt-o1, towards medical complex reasoning with llms. *Preprint*, arXiv:2412.18925.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Preprint*, arXiv:2009.03300.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2020. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Preprint*, arXiv:2009.13081.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W. Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. *Preprint*, arXiv:1909.06146.

Haitao Li, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. 2024. Llms-as-judges: A comprehensive survey on llm-based evaluation methods. *Preprint*, arXiv:2412.05579.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022.

Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. Medmcqa : A large-scale multi-subject multi-choice dataset for medical domain question answering. *Preprint*, arXiv:2203.14371.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, Michael Choi, Anish Agrawal, Arnav Chopra, Adam Khoja, Ryan Kim, Richard Ren, Jason Hausenloy, Oliver Zhang, Mantas Mazeika, and 1103 others. 2026. Humanity's last exam. *Preprint*, arXiv:2501.14249.

Andrew Sellergren, Sahar Kazemzadeh, Tiam Jaroensri, Atilla Kiraly, Madeleine Traverse, Timo Kohlberger, Shawn Xu, Fayaz Jamil, Cían Hughes, Charles Lau, Justin Chen, Fereshteh Mahvar, Liron Yatziv, Tiffany Chen, Bram Sterling, Stefanie Anna Baby, Susanna Maria Baby, Jeremy Lai, Samuel Schmidgall, and 62 others. 2025. Medgemma technical report. *Preprint*, arXiv:2507.05201.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Le Hou, Kevin Clark, Stephen Pfohl, Heather Cole-Lewis, Darlene Neal, Mike Schaekermann, Amy Wang, Mohamed Amin, Sami Lachgar, Philip Mansfield, Sushant Prakash, Bradley Green, Ewa Dominowska, Blaise Aguera y Arcas, and 12 others. 2023. Towards expert-level medical question answering with large language models. *Preprint*, arXiv:2305.09617.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. 2022. Learning to summarize from human feedback. *Preprint*, arXiv:2009.01325.

Mohoshin Ara Tahera, Karamveer Singh Sidhu, Shuvalaxmi Dass, and Sajal Saha. 2026. Sok: Privacy-aware llm in healthcare: Threat model, privacy techniques, challenges and recommendations. *Preprint*, arXiv:2601.10004.

Xiangru Tang, Anni Zou, Zhuosheng Zhang, Ziming Li, Yilun Zhao, Xingyao Zhang, Arman Cohan, and Mark Gerstein. 2024. Medagents: Large language models as collaborators for zero-shot medical reasoning. *Preprint*, arXiv:2311.10537.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. *Preprint*, arXiv:2203.11171.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Juncheng Wu, Wenlong Deng, Xingxuan Li, Sheng Liu, Taomian Mi, Yifan Peng, Ziyang Xu, Yi Liu, Hyunjin Cho, Chang-In Choi, Yihan Cao, Hui Ren, Xiang Li, Xiaoxiao Li, and Yuyin Zhou. 2025. Medreason: Eliciting factual medical reasoning steps in llms via knowledge graphs. *Preprint*, arXiv:2504.00993.

Roy Xie, David Qiu, Deepak Gopinath, Dong Lin, Yanchao Sun, Chong Wang, Saloni Potdar, and Bhuwan Dhingra. 2026. Interleaved reasoning for large language models via reinforcement learning. *Preprint*, arXiv:2505.19640.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

E. Yu, X. Chu, W. Zhang, X. Meng, Y. Yang, X. Ji, and C. Wu. 2025. Large language models in medicine: Applications, challenges, and future directions. *International Journal of Medical Sciences*, 22(11):2792–2801.

Danlong Yuan, Tian Xie, Shaohan Huang, Zhuocheng Gong, Huishuai Zhang, Chong Luo, Furu Wei, and Dongyan Zhao. 2025. Efficient rl training for reasoning models via length-aware optimization. *Preprint*, arXiv:2505.12284.

Kaiyan Zhang, Sihang Zeng, Ermo Hua, Ning Ding, Zhang-Ren Chen, Zhiyuan Ma, Haoxin Li, Ganqu Cui, Biqing Qi, Xuekai Zhu, Xingtai Lv, Hu Jinfang, Zhiyuan Liu, and Bowen Zhou. 2024. Ultramedical: Building specialized generalists in biomedicine. *Preprint*, arXiv:2406.03949.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. 2023. Least-to-most prompting enables complex reasoning in large language models. *Preprint*, arXiv:2205.10625.

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

A Appendix

A.1 Hyperparameters for SFT

We use the following hyperparameters for the SFT coldstart on MedMCQA and the SFT-tuned variants on MedReason of Qwen3-4B-Thinking-2507 and Medgemma-1.5-4B-it. The training has been done with the TRL library of Hugging Face on 1 NVIDIA A40(40G).

Category	Hyperparameter
Data Configuration	
Train Batch Size	4
Validation Batch Size	4
Max Sequence Length	2048
Packing	False
Optimization	
Learning Rate	2×10^{-5}
Weight Decay	0.01
Gradient Accumulation Steps	4
LR Scheduler	Cosine
Training	
Total Epochs	3
Gradient Checkpointing	True
BF16 Precision	True
Evaluation & Logging	
Evaluation Strategy	Steps
Eval Frequency (Steps)	100
Save Frequency (Steps)	100
Logging Frequency (Steps)	10
Best Model Metric	Eval Loss
Miscellaneous	
Completion Only Loss	True
Special Tokens Added	False
Reporting Tool	TensorBoard

Table 7: Hyperparameters configuration used for supervised fine-tuning.

A.2 Hyperparameters for GRPO

We use the following hyperparameters for GRPO after SFT coldstart on MedMCQA. Both TRL library from Hugging Face and vLLM have been used. All trainings have been done on a NVIDIA H100(90G).

Category	Hyperparameter
Model Configuration	
Precision	BF16
Gradient Checkpointing	True
Attention Implementation	Eager
Optimization	
Learning Rate	2×10^{-6}
Train Batch Size	2
Gradient Accumulation Steps	8
Total Epochs	1
Rollout Configuration	
Number of Generations	4
Max Prompt Length	512
Max Completion Length	1024
Rollout Engine	vLLM
vLLM Mode	Colocate
Evaluation & Logging	
Evaluation Strategy	Steps
Evaluation Frequency (Steps)	250
Validation Batch Size	4
Save Frequency (Steps)	1000
Logging Frequency (Steps)	20
Reporting Tool	TensorBoard

Table 8: Hyperparameters configuration for GRPO training.

A.3 Inference configurations - regime A

All inferences in Table 2 have been run on NVIDIA H100 and A40 GPUs with vLLM on 3 runs. We use the following inference parameters. We set a temperature of 0.7 to promote diverse reasoning paths and richer generation. This allows the models to explore multiple plausible inference trajectories before producing the final answer, which is particularly important in reasoning-heavy biomedical tasks.

Category	Parameter
Model Configuration	
Inference Engine	vLLM
Max Model Context Length	8192
Decoding Configuration	
Temperature	0.7
Max Generation Tokens	3000
Stop Tokens	EOS Token
Hardware Configuration	
Tensor Parallel Size	1
GPU Memory Utilization	0.9

Table 9: Inference configuration used for model evaluation in Table 2.

A.4 Hyperparameters and prompts for Interleaved datasets generation

This section details the hardware configurations, inference hyperparameters, and quality curation criteria used to construct the interleaved datasets described in Section 3.1. Dataset generation and processing were conducted using the Hugging Face TRL library on two NVIDIA A40 (40GB) GPUs. The prompt template utilized during the generation phase is illustrated in Figure 4, and the specific execution parameters are provided in Table 10.

Category	Parameter
Model Configuration	
Model	Qwen / Qwen2.5-7B-Instruct
Tensor Parallel Size	1
Max Model Context Length	4096
Generation Configuration	
Batch Size	500
Temperature	0.2
Top-p (nucleus sampling)	0.95
Max Generation Tokens	3000
Stop Tokens	<im_end> , <endoftext>

Table 10: Inference hyperparameters used for dataset preprocessing.

To construct the interleaved reasoning trajectories, we extracted initial reasoning traces from the existing MedReason dataset and subjected them to a rigorous multi-stage filtering pipeline to eliminate faulty or degenerate logic. Traces were discarded if they failed any of the following validation criteria:

- **Ground-Truth Validation:** We excluded all reasoning traces that arrived at an incorrect final answer, ensuring that the optimization signal remained anchored to verified dataset labels.
- **Structural Integrity:** We omitted trajectories that deviated from the target interleaved format schema, enforcing strict structural consistency across training instances.
- **Logical Coherence:** We dropped traces exhibiting "reasoning loops"—defined as the repetitive generation of identical thinking steps or intermediate answer tags without logical progression or proper termination.
- **Parsability:** We removed any sequences where the final or intermediate answer identifiers could not be successfully extracted by our automated regex parsers, protecting downstream evaluation from formatting artifacts.

Applying these strict quality controls, and explicitly excluding the Huatuo subsets and unlabeled `PubMedQA` instances, yielded a curated final training set of exactly 23,758 structured samples from the original MedReason distribution and 2,727 samples from the selected subset of MedMCQA.

A.5 Example of training and inference prompts.

System Prompt
You are a helpful medical assistant. You reason through problems step by step. You conduct your reasoning within <think></think> and share partial answers within <answer></answer> tags. Always START with <think> and END with </answer>.
User Prompt
Answer the following question step by step.
Question:example['question']
Options: example['options']

Figure 5: Prompt template used during training and evaluation.

A.6 Additionnal tables for regime B inference

	Qwen3-4B		MedGemma-4B	
Dataset	3000 tok	8000 tok	3000 tok	8000 tok
MedMCQA	16.3	0.1	4.0	2.4
MedQA	50.6	0.6	9.0	6.9
PubMedQA	7.8	0.0	3.0	3.0
MMLU-Pro	30.6	1.0	10.4	5.7
MedXpertQA	68.9	1.6	9.1	4.0

Table 11: Unparseable rate (%) under regime B — proportion of responses where no valid final answer could be extracted.

	Qwen3-4B		MedGemma-4B	
Dataset	3000 tok	8000 tok	3000 tok	8000 tok
MedMCQA	1791.5	2238.9	753.4	837.2
MedQA	1896.3	2720.1	820.9	876.9
PubMedQA	1764.8	2032.7	517.4	517.4
MMLU-Pro	1902.7	2510.3	950.5	1115.1
MedXpertQA	2017.0	3625.5	1228.5	1347.5

Table 12: Average response length (words) of base models across benchmarks and token generation limits under regime B.

A.7 LLM-as-a-Judge reliability

We additionally provide an empirical analysis of the reliability of the LLM judge (Qwen3-4B-

Instruct-2507) utilized in the S3-gated configuration. We randomly sample 20 generated reasoning traces (comprising 86 individual evaluation steps) from the Qwen3-4B-Thinking-2507 SFT MedMCQA coldstart model on MedReason, evaluating each intermediate step against the gold reference milestones. The LLM judge is prompted as follows:

LLM Judge Prompt
You are a medical board examiner. Your task is to evaluate a reasoning step against a provided reference reasoning path.

Progression Context: Step s_idx + 1 of len(gen_pairs).

Scoring Rules:

- **1.0** — The step exactly matches a key milestone or logical transition in the reference reasoning.
- **0.5** — The step is partially correct or references relevant clinical information but lacks precision or completeness.
- **0.0** — The step is incorrect, contradictory to the reference reasoning, or irrelevant.

Reference Reasoning (Ground Truth Path): {ref_context}

Evaluation Target:

- **Reasoning Step:** {r_t}
- **Intermediate Answer:** {a_t}

Instruction: Compare the Evaluation Target strictly against the Reference Reasoning. Output only the final numerical score.

Final reward:

To quantify the validity of these automated reward signals, a graduate level human annotator with access to medical resources manually graded the same think/answer steps using the identical scoring rubric guidelines given in the LLM Judge Prompt. The inter-rater reliability between the LLM judge and the human evaluator was measured using both linear and quadratic weighted Cohen's Kappa (κ).

Our analysis revealed a point estimate of $\kappa_{\text{linear}} = 0.1267$ on the linear scale and $\kappa_{\text{quadratic}} = 0.2305$ on the quadratic scale. While these values indicate modest absolute agreement, these metrics should be interpreted cautiously as exploratory diagnostic evidence rather than a definitive metric of uniform unreliability. Because this audit was conducted on a limited sample of 20 questions using coarse ordinal labels (0 / 0.5 / 1), the resulting point estimates are highly sensitive to isolated severe disagreements (e.g., over-scoring a fundamentally flawed step as perfect) and baseline label imbalances. The observed variance suggests an occasional trend toward a soft optimism bias within the 4B parameters judge, wherein fluent or grammatically sound phrasing can sometimes be over-allocated full credit despite incomplete clinical logic. Ultimately, these results indicate that the compact judge introduces a minor degree of stochastic noise into the granular supervision signal rather than compromising the core training trajectory.

A.8 Judge model's size effect

To test whether the effectiveness of the logic reward is limited by judge quality, we retrain the S3-gated configuration of Qwen3-4B-Thinking-2507 using Qwen3-8B as the judge for r_{logic}, replacing Qwen3-4B-Instruct-2507. The judge prompt, reward formulation, and GRPO hyperparameters are kept unchanged, and the 8B-judge run is initialized from the same SFT MedMCQA cold-start checkpoint.

In the reliability audit in Appendix A.7, the larger judge shows substantially stronger agreement with the human annotator (Table 13). Linear weighted κ increases from 0.127 to 0.285, while quadratic weighted κ increases from 0.231 to 0.521. The 4B judge assigns full credit to 77 of the 86 steps, compared with 54 for the human annotator, indicating a strong tendency toward full-credit scores. In contrast, the 8B judge distributes its scores more broadly across the rubric and penalizes flawed steps more frequently, although it is somewhat stricter than the human annotator, assigning a score of 0 to 37 steps compared with 21. This more selective reward is accompanied by improved downstream accuracy (Table 14). Relative to the 4B-judge configuration, the 8B-judge model improves by 4.0 points on MedMCQA, 5.8 on MedQA, 25.2 on MMLU-Pro, and 2.2 on MedXpertQA, while decreasing by 2.5 points on PubMedQA. These results suggest that judge quality can materially affect the effectiveness of the logic reward during training, with the stronger 8B judge yielding better generalization on four of the five evaluated benchmarks.

A.9 Baseline comparison

	Human	4B judge	8B judge
Cohen's κ vs. human			
Linear	–	0.1267	**0.2845**
Quadratic	–	0.2305	**0.5212**
Steps per score			
0.0	21	7	37
0.5	11	2	10
1.0	54	77	39

Table 13: Agreement between each LLM judge (4B: Qwen3-4B-Instruct-2507; 8B: Qwen3-8B) and a human annotator on the same 86 reasoning steps (20 traces from the Qwen3-4B-Thinking-2507 SFT MedMCQA coldstart model on MedReason), using the identical judge prompt and rubric. Linear and quadratic denote weighted Cohen's κ.

	Judge used in r_{logic}	
Dataset	Qwen3-4B-Instruct	Qwen3-8B
MedMCQA	0.544 ± 0.005	$\mathbf{0.584 \pm 0.001}$
MedQA	0.563 ± 0.003	$\mathbf{0.621 \pm 0.004}$
MMLU-Pro	0.120 ± 0.003	$\mathbf{0.372 \pm 0.014}$
PubMedQA	$\mathbf{0.637 \pm 0.002}$	0.612 ± 0.006
MedXpertQA	0.098 ± 0.000	$\mathbf{0.120 \pm 0.004}$

Table 14: Accuracy of Qwen3-4B-Thinking-2507 trained with the S3-gated reward, using either the 4B or the 8B judge (mean $\pm$ std over 3 inference runs). MMLU-Pro uses the Biology and Health subsets. Answers are parsed as in Table 2.

A.10 Parse failures of post-trained models

We report how often a model's answer cannot be extracted, since a parse failure is scored as an incorrect answer. We use the same rule for every dataset: the prediction is read from the last <answer></answer> block of the output, which is the block rewarded by r_{acc} during training. A sample counts as a parse failure if this block is missing or contains no valid option (A–D for MedMCQA and MedQA, A–J for MMLU-Pro and MedXpertQA, A–C or yes/no/maybe for PubMedQA). Table 17 gives the rates for all trained models, averaged over the same inference runs as Table 2.

Model Class / Size	Method / Base Model	MedQA	MedMCQA	PubMedQA	Source Publication
∼8B Models	BioMistral-7B	45.0	40.2	66.9	Chen et al. (2024)
	Mistral-7B-Instruct	48.2	44.6	59.5	Chen et al. (2024)
	OpenBioLLM-8B	57.7	54.1	74.1	Chen et al. (2024)
	LLaMA-3.1-8B-Instruct	58.7	56.0	75.2	Chen et al. (2024)
	UltraMedical-8B	71.1	58.3	77.4	Chen et al. (2024)
	HuatuoGPT-o1-8B	72.6	60.4	79.2	Chen et al. (2024)
	MedReason-8B	71.8	60.7	79.4	Wu et al. (2025)
>10B Models	UltraMedical-70B	82.2	71.8	78.4	Chen et al. (2024)
	OpenBioLLM-70B	76.1	74.7	79.2	Chen et al. (2024)
	Llama-3.1-70B-Instruct	78.4	72.5	78.5	Chen et al. (2024)
	HuatuoGPT-o1-70B	83.3	73.6	80.6	Chen et al. (2024)
Frontier models	GPT-3.5: Zero-shot	54.7	56.7	67.6	Tang et al. (2024)
	GPT-3.5: MedAgents	64.1	59.3	72.9	Tang et al. (2024)
	GPT-4: Zero-shot	73.0	69.0	76.2	Tang et al. (2024)
	GPT-4: MedAgents	83.7	74.8	76.8	Tang et al. (2024)

Table 15: Comparison of LLM Baselines and Specialized Methods across Medical Benchmarks (MedQA, MedMCQA, and PubMedQA).

Model / Configuration	MMLU-Pro			MedXpert
	Avg. Health + Bio	Health	Biology	
BioMistral-7B	37.6	27.4	49.2	12.4
Mistral-7B-Instruct	42.7	33.7	53.6	11.4
LLaMA-3.1-8B-Instruct	58.2	52.7	64.6	13.2 / 14.3[†]
OpenBioLLM-8B	44.9	38.4	52.4	10.7
Qwen2.5-7B-Instruct	59.8	50.6	70.2	12.6
UltraMedical-8B	60.5	55.1	66.7	–
HuatuoGPT-o1-8B	63.1	58.7	68.2	16.8
AlphaMed-8B	66.7	–	–	22.14
MedReason-8B	–	–	–	19.0

Table 16: Baseline and Advanced Model Performance Profiles on MMLU-Pro and MedXpert Benchmarks.

[†] Dual entry reflects evaluation variations between the controlled fine-tuning splits (Table 2) and general baseline frameworks (Table 4) within Wu et al. (2025).

Model	MedMCQA	MedQA	MMLU-Pro (Bio+Health)	PubMedQA	MedXpertQA
MEDGEMMA-1.5-4B-IT					
SFT (MedMCQA coldstart)	9.24 ± 0.96	13.59 ± 0.67	14.85 ± 0.18	8.40 ± 0.00	21.22 ± 0.00
SFT (MedReason)	9.45 ± 0.72	21.39 ± 0.64	18.31 ± 1.10	15.00 ± 0.69	24.90 ± 0.14
S1-simple	0.38 ± 0.20	0.31 ± 0.14	1.39 ± 0.25	0.00 ± 0.00	0.84 ± 0.31
S2-depth	0.21 ± 0.04	0.29 ± 0.16	1.04 ± 0.13	0.00 ± 0.00	0.50 ± 0.05
S3-gated	0.18 ± 0.01	0.26 ± 0.05	0.89 ± 0.10	0.00 ± 0.00	0.41 ± 0.07
QWEN3-4B-THINKING-2507					
SFT (MedMCQA coldstart)	0.14 ± 0.03	0.47 ± 0.27	0.28 ± 0.08	0.00 ± 0.00	0.20 ± 0.07
SFT (MedReason)	0.04 ± 0.04	0.08 ± 0.08	0.04 ± 0.04	0.00 ± 0.00	0.11 ± 0.05
S1-simple	0.50 ± 0.83	0.16 ± 0.08	3.32 ± 0.00	0.00 ± 0.00	0.22 ± 0.14
S2-depth	1.05 ± 0.91	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
S3-gated	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
S3-gated (8B judge)	0.02 ± 0.01	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00

Table 17: Parse-failure rate (%, mean $\pm$ std over inference runs) of trained models. A sample is a parse failure when its last `<answer>` block is missing or contains no valid option.

System Prompt:
<|im_start|>*system*
You are an expert editor for medical reasoning datasets. Your task is to break down a reasoning block into an 'Interleaved' format. This means splitting the logic into small steps, where every <think> *block is immediately followed by a short, factual* <answer> *block.*<|im_end|>

User Prompt:
<|im_start|>*user*
Task: *Rewrite the raw reasoning into a chain of* <think> *and* <answer> *pairs.*

Rules:

1. *Do NOT wrap everything in one big* <think> *tag.*
2. *Alternating pattern:* <think>*Logic/Question*</think> → <answer>*Intermediate Fact/Conclusion*</answer>*.*
3. *The reasoning must move step-by-step (define terms → analyze pathology → match options).*
4. *The FINAL* <answer> *tag must contain ONLY the final answer (can be Yes/No, A/B/C/D, or a string statement if the question does not ask to chosen between options).*

Example of Question and Options:

Question: A 72-year-old man with long-standing benign prostatic hyperplasia (BPH) develops chronic urinary outflow obstruction...
Options: A. Compensatory hypertrophy... B. Acute tubular necrosis... C. Glomerular hyperplasia... D. Pressure atrophy (shrinkage) of renal tissue

Example of Desired Interleaved Format:

```
<think>First, I need to define what Benign Prostatic Hyperplasia (BPH) causes in
the urinary tract.</think>
<answer>BPH causes chronic obstruction of urinary outflow.</answer>
<think>Now, I need to look at the consequences of chronic obstruction on the kidney
structure.</think>
<answer>It leads to hydronephrosis and increased pressure in the renal
pelvis.</answer>
<think>Finally, how does this pressure affect the parenchyma cells over time? Does
it cause growth or shrinkage?</think>
<answer>It causes pressure atrophy, or shrinkage of the tissue.</answer>
<think>Matches this deduction to option.</think>
<answer> D </answer>
```

Input Data:

- ***Question:*** *{std['question']}*
- ***Correct Option:*** *{std['ground_truth']}*
- ***Raw Reasoning:*** *{std['raw_reasoning']}*

Your Output: <|im_end|>

Figure 4: The zero-shot prompt template used to restructure the curated MedReason dataset into the required interleaved thought-and-answer format.