MedPsy: State-of-the-Art Small Medical Language Models for Efficient Edge Deployment

Davide Vitabile **Alexandro Buffa** **Akshay Nambiar** **Amril Nazir**

Tether Data, S.A. de C.V. d.b.a. Tether AI Research
{davide.vitabile,akshay.nambiar}@tether.to
{alexandro.buffa,amril.nazir}@tether.io

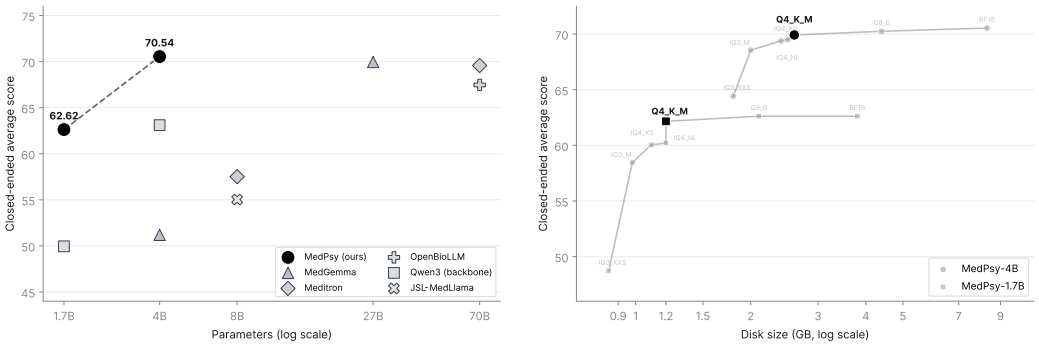
Abstract

Medical language models promise expert clinical support, but the strongest open-source LLMs remain too large for private edge-device deployment, while compact medical models still trail substantially on knowledge-intensive and real-world healthcare tasks. Moreover, reasoning models that fit edge memory budgets can produce long reasoning traces, increasing sequential decoding, end-to-end latency, energy consumption, and KV-cache memory pressure. We present MEDPSY, a family of text-only 1.7B and 4B medical language models designed for edge deployment, explicitly targeting this accuracy–generation-efficiency trade-off. Our recipe combines a synthetic medical data pipeline over biology, medicine, and health seeds with chain-of-thought targets from a 235B medical-focused LLM teacher; a four-stage post-training curriculum comprising two SFT stages followed by two RL stages with hard-sample mining; and an edge deployment evaluation spanning multiple GGUF quantization variants and real-device inference on laptop and smartphone hardware. On seven closed-ended medical benchmarks, MEDPSY-4B surpasses MedGemma-1.5-4B by 19.34 points and matches MedGemma-27B while being 6.75× smaller; MEDPSY-1.7B outperforms MedGemma-1.5-4B by 11.42 points despite being less than half its size. On HealthBench-Hard, MEDPSY-4B surpasses MedGemma-27B by 15.33 points, while MEDPSY-1.7B surpasses it by 11.66 points despite being ∼16× smaller. Beyond accuracy, MEDPSY reduces average response length by 1.7× (1.7B) and 3.2× (4B) versus its Qwen3 backbones, while `Q4_K_M` quantization preserves the aggregate score within 1 point of BF16 with ∼69% lower disk footprint. In real-device inference, MEDPSY-4B's shorter generations translate into 75% lower median end-to-end latency and estimated energy per answer than its Qwen3 backbone at matched quantization, while peak RAM remains below 5 GB across our tested laptop and smartphone configurations.

1 Introduction

Medical knowledge and capabilities of language models have advanced rapidly, yet their deployment remains centralized. Among open-weight medical systems, strong models such as MedGemma-27B-text-it [1] still require substantial accelerator memory for practical deployment, while proprietary medical systems such as Med-PaLM-style models and large GPT-4-based clinical assistants are typically accessed through cloud infrastructure. Both fundamentally conflict with the privacy, latency, and reliability requirements of healthcare. Patient records, symptom descriptions, and clinical notes are protected health information governed by strict regulatory regimes (HIPAA, GDPR, and equivalents). The dominant cloud paradigm requires this data to leave the device, traverse network infrastructure, and be processed remotely, creating attack surfaces, compliance burdens, and a fundamental tension between AI capability and patient privacy.

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



(a) Capability per parameter: closed-ended average score vs. parameter count (log scale). MEDPSY-4B matches 27B–70B medical LLMs at a fraction of the size; the dashed line connects the MEDPSY pareto frontier. Scores from Tables 1 and 9.

(b) Capability per gigabyte: closed-ended average score vs. GGUF disk size (log scale) across seven quantization formats. Scores stay near BF16 until 3-bit precision, with a sharp cliff at `IQ3_XXS`; highlighted points mark the capacity-vs-precision crossover. Data from Table 4(a).

Figure 1: Efficiency along both deployment axes. MEDPSY models deliver frontier-level medical accuracy per parameter **(a)** and per gigabyte of quantized footprint **(b)**.

The compact-medical gap. The current open landscape has a sharp trade-off between capability and deployability. MedGemma-27B-text-it delivers strong medical performance, but at 27B parameters it is infeasible for edge deployment, requiring tens of GB of VRAM. MedGemma-1.5-4B-it [2] is technically runnable on a high-end laptop but lacks the domain-specific capabilities of its larger variant on medical benchmarks, and, to the best of our knowledge, no open model in the 1–4B range delivers accuracy comparable to large medical-model baselines. This is not a pure compression problem: smaller models trained with conventional recipes suffer disproportionately on knowledge-intensive medical tasks, where factual precision and safe, well-structured responses are essential. Nor does fitting device memory guarantee efficient inference: compact reasoning models can still overthink, producing long traces that increase sequential decoding latency, energy use, and KV-cache traffic.

Our thesis. *For compact medical LLMs, medical capability and deployment efficiency can be optimized jointly: high-quality synthetic supervision and staged post-training build accuracy, while length-aware alignment and quantization reduce generation and memory costs for private on-device use.* We test this thesis by constructing the MEDPSY[1] family of two text-only medical models, MEDPSY-1.7B and MEDPSY-4B[2], on top of Qwen3 backbones [3] with a four-stage post-training recipe (two Supervised Fine-Tuning (SFT) stages over a synthetic medical corpus built from public Genesis III biology/medicine seed prompts [4] and newly generated health seeds derived from FineFineWeb [5], then two Reinforcement Learning (RL) stages with hard-sample mining inspired by Fleming-R1 [6]), and we evaluate them across eight benchmark suites spanning clinical exams (MedQA-USMLE [7], MedMCQA [8]), health literacy (MMLU-Health, MMLU-Pro Health [9, 10]), expert reasoning (MedXpertQA [11]), biomedical research (PubMedQA [12]), underserved-region medicine (AfriMedQA [13]), and real-world clinical scenarios (HealthBench and HealthBench-Hard [14]).

Contributions. Our contributions are:

- **State-of-the-Art Efficient Models.** MEDPSY-4B matches MedGemma-27B on closed-ended accuracy and surpasses on HealthBench suites both MedGemma-4B (+20/+28.33) and 27B (+9.0/+15.3). Similarly, MEDPSY-1.7B exceeds MedGemma-4B on closed-ended tasks and outperforms both MedGemma variants on HealthBench, achieving these leads at up to 16× smaller scale (Fig. 1a; Sec. 5).

[1] In MedPsy, "Med" denotes the medical specialization, while "Psy" is the shared suffix used across our organization's model families, inspired by psychohistory in Isaac Asimov's fiction rather than psychology.

[2] Models and supporting artifacts are available at `https://huggingface.co/collections/qvac/medpsy`.

- **Post-training recipe.** We combine single-teacher synthetic supervision, broad-to-focused SFT, and easy-to-hard DAPO with deterministic failure mining. Component ablations separate the roles of SFT, DAPO's drift-enabling changes, the overlong-response penalty, and Stage-2 mining (Sec. 5.4; App. A.17).
- **Token efficiency for the edge.** MEDPSY-4B and MEDPSY-1.7B achieve 3.2× and 1.7× token reductions over Qwen3 backbones, significantly lowering on-device latency (Sec. 5.5). This efficiency stems from RL-side compression, primarily a soft overlong-response penalty that maintains accuracy while ensuring the training curriculum remains Pareto-dominant (App. A.17.2).
- **Quantized edge deployment.** At `Q4_K_M`, both model sizes retain aggregate score within one point of BF16 while reducing disk footprint by 68–69%. On-device measurements connect MEDPSY-4B's shorter generations to 75% lower median latency and estimated energy per answer at matched quantization, while peak RAM remains below 5 GB in the tested laptop and smartphone configurations. A broader six-format study identifies a *capacity-vs-precision* effect at matched footprint (Sec. 5.6; App. A.12).

2 Related Work

Medical LLMs. Med-PaLM [15] and Med-PaLM-2 [16] demonstrated that domain-specialized fine-tuning of large general LLMs could reach passing scores on USMLE-style exams, but they remain proprietary and large. Among open models, the strongest medical performance has largely come from substantially larger systems, such as MedGemma-27B [1], Meditron 70B-scale models [17], and OpenBioLLM-70B [18]; smaller medical models such as MedGemma-1.5-4B-it [2] and JSL-MedLlama-3-8B [19] are more deployable but leave a clear accuracy gap. Based on our evaluation (App. A.3) and to the best of our knowledge, no open model in the 1–4B range matched MedGemma-27B-text-it on closed-ended medical benchmarks or HealthBench prior to our work.

Synthetic data for distillation. Self-instruct and synthetic instruction generation [20, 21] have become standard for scaling supervision. The Genesis III project [4] focused on synthetic data for pre-training, combining seed acquisition, quality filtering, and a "scaling Q&A" stage to produce broad educational corpora. We adapt this pipeline for medical post-training: selected Genesis III question/option fields and a new FineFineWeb-derived health slice are used as prompt seeds, and the resulting scaling-QA material is converted into SFT supervision by generating fresh medical reasoning targets with a stronger teacher model.

RL for medical reasoning. Fleming-R1 [6] applies a multi-stage GRPO curriculum with hard-sample mining to medical reasoning. AlphaMed [22] releases the curated AlphaMed19K medical RL prompt set together with a binary correctness reward and a pass-at-N difficulty annotation procedure. We adopt a two-stage curriculum in the spirit of Fleming-R1 and reuse AlphaMed's pass-at-N annotation, applied here with each model's own size-matched SFT-2 policy as the annotator; we switch the optimiser to DAPO [23], a GRPO variant that drops the KL anchor, uses asymmetric clip-higher, dynamically filters zero-variance rollout groups, and adds a soft overlong-response penalty.

Reasoning-model test-time efficiency. Reasoning models improve difficult tasks by generating explicit intermediate computation, but long traces increase decode latency, KV-cache memory [24], and energy use [25], all especially costly on edge hardware. MEDPSY targets this trade-off in the medical domain, improving accuracy while reducing generated tokens in compact reasoning models.

Edge LLMs and quantization. Edge deployment depends jointly on compressed model footprint and generation cost: quantization governs storage and memory requirements, while response length directly affects latency and energy consumption. GGUF and llama.cpp [26] support practical quantized inference across consumer hardware. Existing medical LLM studies primarily establish BF16 benchmark accuracy. We complement this literature by jointly evaluating quantized medical accuracy and measured on-device efficiency across two model sizes (Sec. 5.6; App. A.12).

3 Data Methodology

In this section we describe at a high level the data used to post-train MEDPSY and the teacher-selection process behind it. The training corpus description here reflects the design principles and the recipe used to train the final models that are evaluated in this paper.

3.1 Training data overview

We explored several data mixtures and post-training methodologies before settling on the recipe used for the final MEDPSY models. In total, **about 10M synthetic rows** of medical and healthcare supervision were generated for the final synthetic corpus. The recipe organizes the data into a **two-stage curriculum**: a broad-coverage corpus (Corpus 1) followed by a smaller, higher-value clinical-QA corpus (Corpus 2). The two corpora play complementary roles in the SFT curriculum (Sec. 4).

Two principles guided the data construction throughout these ablations:

- **Synthetic datasets.** Corpus 1 utilizes broad synthetic prompts derived from two primary sources. First, we extracted the question and answer option fields from four public Genesis III subsets [4]: `college_medicine`, `professional_medicine`, `college_biology`, and `high_school_biology`. Second, following [4], we use Fleming-R1-32B [6] to generate four multiple-choice questions (MCQs) for each document in the FineFineWeb health sub-domain [5]. We selected Fleming-R1-32B as it achieves strong performance on closed-ended medical tasks while offering higher throughput than larger teacher models. Corpus 2 instead utilizes train splits from focused open-source medical QA datasets as higher-value clinical prompts. Together, these broad synthetic and focused public-train prompts are used as seeds to generate the final synthetic training data.
- **A single, controlled SFT target teacher.** Every long-form assistant target used for supervision (chain-of-thought traces, extended rationales, decision-oriented answers) is newly generated by Baichuan-M3-235B [27], the teacher selected in Sec. 3.2, rather than taken from open-source assistant traces. This separation keeps prompt generation and target generation distinct: the seed data controls coverage, while Baichuan-M3-235B provides style adaptation and medical-reasoning knowledge distillation for the SFT targets.

The output of this stage is a curated medical and healthcare post-training corpus that combines synthetic expansion, reasoning-focused data, and public medical QA prompts, all paired with teacher-generated reasoning targets; source list, licenses, and prompts are in App. A.1.

3.2 Teacher selection

Because our supervision *is* the teacher's output, the teacher's ceiling directly bounds the student's. From a literature review and public medical leaderboards, we identified three strong candidates (**Baichuan-M3-235B** [27], **GPT-OSS-120B** [28], and **Fleming-R1-32B** [6]) and evaluated them on our benchmark suite under three independent model graders (Sec. 5.1). The same suite serves teacher selection and final reporting, as it directly measures the supervision quality the student distills.

Baichuan-M3-235B leads with 74.83 average closed-ended accuracy (vs. 72.73/72.55), with its largest advantages on the most expert-level benchmarks (MedXpertQA: +5.61/+10.98; MMLU-Pro Health: +5.05/+2.73), and beats both alternatives by 6–12 HealthBench points across all three graders (Tables 6, 7). Since the teacher's clinical reasoning and communication quality is exactly the signal the student distills, this open-ended advantage was decisive: we use Baichuan-M3-235B as the teacher for all final MEDPSY training data. A controlled student-level ablation (App. A.2) verifies that this choice transfers when the prompt set, base model, and training pipeline are held fixed.

4 Post-Training Methodology

4.1 Backbone selection

We selected backbones via a controlled comparison at each target size against four other open candidates (full tables in App. A.4). At the 3-4B class, **Qwen3-4B-Thinking-2507** leads closed-ended by +13.43 over Llama-3.2-3B-Instruct, +14.11 over SmolLM3-3B, and +20.51 over gemma-3-4b-it, and dominates HealthBench by +21–26 points over Llama-3.2-3B-Instruct across all three graders. At the sub-2B class, **Qwen3-1.7B (Thinking)** leads by +5.81 over the next-best closed-ended backbone (LFM2.5-1.2B-Thinking), remains ahead of LFM2.5-1.2B-Thinking across all three HealthBench graders, and is +21–26 points ahead of Llama-3.2-1B-Instruct on HealthBench. We

therefore build MEDPSY-4B on top of Qwen3-4B-Thinking-2507 and MEDPSY-1.7B on Qwen3-1.7B (with `enable_thinking=True`); an architecture summary of both backbones is given in App. A.6.

4.2 Four-stage curriculum

The four stages are:

1. **SFT Stage 1: broad medical adaptation** (Corpus 1). The model absorbs large-scale synthetic supervision spanning biology, medicine, and health, building wide factual coverage and medical vocabulary.
2. **SFT Stage 2: reasoning specialization** (Corpus 2). The model shifts to a smaller set of high-value clinical QA examples with teacher-generated CoT, sharpening answer structure and clinical reasoning depth.
3. **RL Stage 1: consolidation on easy/moderate cases.** DAPO [23] on $\sim$16K examples from AlphaMed19K [22], initialized from the SFT-stage-2 checkpoint. This stage reinforces correct chains of reasoning on clinical scenarios where the model already has partial competence.
4. **RL Stage 2: hard-enriched targeting.** The full set is re-annotated with the best Stage-1 checkpoint and a hard-enriched subset of $\sim$2–4K items is built deterministically at a fixed 1:2 right-to-wrong ratio. The purpose of Stage 2 is to push the model on its persistent failure modes.

Our RL phase builds on three lines of prior work. The two-stage RLVR-with-curriculum framework, consolidating reasoning on easier cases before targeting persistent failures, follows Fleming-R1 [6]. The Stage-1 corpus, the pass-at-N difficulty annotation procedure, and the binary correctness reward are inspired by AlphaMed [22]. The structured `<think>`/`<answer>` reasoning trace and the idea of conditioning the reward on the presence of a reasoning block follow Med-U1 [29] (and DeepSeek-R1-Zero [30] more broadly). We adopt the structured `<think>`/`<answer>` format and extend the binary correctness signal to a four-tier gated reward (§4.2); a controlled ablation (App. A.17.1) shows the four-tier extension is essentially neutral on accuracy versus a binary reward and provides only a modest token-efficiency benefit at 4B, so we report it as a design choice rather than a primary contribution. Within this scaffolding, our two RL design *contributions* are: a single-pass deterministic Stage-2 mining procedure (§4.2), and the integration of all four stages into a single training pipeline targeted at the sub-7B medical reasoning regime.

Difficulty annotation and Stage-1 data. Following AlphaMed, we annotate every training prompt empirically by sampling $N{=}5$ rollouts at temperature 0.7 from the policy and labelling by the count n_c of correct answers, mapped to six difficulty levels L_1–L_6 ($L_1{=}5/5$, $L_6{=}0/5$); the coarser Easy/Moderate/Difficult split groups L_1 alone, L_2–L_5 together, and L_6 alone. The Stage-1 training set follows AlphaMed's data recipe: all of MedQA plus per-level stratified sampling of MedMCQA (capped at 1,600 per L_i), with PubMedQA omitted following AlphaMed's finding that it harms reasoning transfer. From this, $\sim$16K items at the Easy/Moderate levels form the actual Stage-1 corpus.

Reasoning-aware reward. The model is required to emit reasoning inside `<think>...</think>` tags and a final answer letter inside `<answer>...</answer>` [29, 30]. Let $c \in \{0, 1\}$ indicate answer correctness (matching ground truth) and $t \in \{0, 1\}$ indicate the presence of a non-trivial reasoning block. Our task reward generalises the binary correctness/format scheme of Med-U1 to a four-tier function that gates jointly on c and t:

$$r_{\text{task}}(c, t) \;=\; \begin{cases} 1.0 & \text{if } c = 1, t = 1 \quad (\text{correct } + \text{ reasoning}) \\ 0.5 & \text{if } c = 1, t = 0 \quad (\text{correct, no reasoning; "shortcut"}) \\ 0.1 & \text{if } c = 0, t = 1 \quad (\text{reasoning, wrong answer; partial credit}) \\ 0.0 & \text{otherwise} \quad (\text{no valid structure}). \end{cases} \tag{1}$$

The 0.5 vs. 1.0 split actively penalises the shortcut of skipping reasoning even when the model happens to answer correctly; the 0.1 tier supplies a non-zero gradient on rollout groups where no candidate is correct, partially mitigating reward sparsity on Difficult items.

Single-pass deterministic Stage-2 mining. Fleming-R1 adapts the Stage-2 distribution through iterative re-evaluation. We instead perform a single deterministic mining pass: after Stage 1, the best

Algorithm 1 MEDPSY four-stage post-training (one model size).

Require: backbone B; SFT corpora $\mathcal{D}_{\text{C1}}, \mathcal{D}_{\text{C2}}$; RL prompt set $\mathcal{D}_{\text{RL}}$ (AlphaMed19K)

$\theta_{\text{S1}} \leftarrow \text{SFT}(B,\, \mathcal{D}_{\text{C1}})$ ▷ SFT-1: broad medical adaptation

$\theta_{\text{S2}} \leftarrow \text{SFT}(\theta_{\text{S1}},\, \mathcal{D}_{\text{C2}})$ ▷ SFT-2: reasoning specialization

$(\mathcal{D}_{\text{em}},\, \mathcal{D}_{\text{hard}}) \leftarrow \text{ANNOTATE}(\theta_{\text{S2}},\, \mathcal{D}_{\text{RL}},\, N{=}5)$ ▷ Pass@N split: Easy/Mod vs. Hard

$\theta_{\text{R1}} \leftarrow \text{DAPO}(\theta_{\text{S2}},\, \mathcal{D}_{\text{em}};\; r_{\text{task}}{+}r_{\text{overlong}})$ ▷ RL Stage 1, Eq. 1+3

$\mathcal{D}_{\text{S2}} \leftarrow \text{MINESTAGE2}(\theta_{\text{R1}},\, \mathcal{D}_{\text{RL}})$ ▷ Single-pass deterministic mining

$\theta_{\text{R2}} \leftarrow \text{DAPO}(\theta_{\text{R1}},\, \mathcal{D}_{\text{S2}};\; r_{\text{task}}{+}r_{\text{overlong}})$ ▷ RL Stage 2 (released MEDPSY S2)

return θ_{R2}

function MINESTAGE2(π, $\mathcal{D}$)

  for $x \in \mathcal{D}$ **do** $n_c(x) \leftarrow \#\{\text{correct rollouts in } N{=}5 \text{ samples from } \pi \text{ at temperature } 0.7\}$

  end for

  $\mathcal{D}_{\text{diff}} \leftarrow \{x \in \mathcal{D} : n_c(x) = 0\}$ ▷ Items π still fails on every attempt

  $\mathcal{D}_{\text{right}} \leftarrow$ top-$\lfloor|\mathcal{D}_{\text{diff}}|/2\rfloor$ items of $\{x : n_c(x) \geq 1\}$ ranked by $n_c(x)$ (descending)

  return $\mathcal{D}_{\text{diff}} \cup \mathcal{D}_{\text{right}}$ ▷ 1:2 right-to-wrong ratio

end function

Stage-1 checkpoint is re-evaluated on the full 19K-item set with the same pass-at-N procedure; items that the policy still fails on every attempt ($n_c{=}0$) are retained in full, and a smaller sample of items it gets right is added to enforce a 1:2 right-to-wrong ratio.[3]

Optimization. Both RL stages use DAPO [23], a GRPO variant whose actor objective drops the KL-to-reference term and replaces PPO's symmetric clip ϵ with an asymmetric $(\epsilon_{\text{low}}, \epsilon_{\text{high}})$:

$$\mathcal{L}_{\text{DAPO}}(\theta) \;=\; -\,\mathbb{E}_{(s,a)\sim\pi_{\theta_{\text{old}}}}\Big[\min\Big(\rho_t(\theta)\,\hat{A}_t,\;\; \text{clip}(\rho_t(\theta),\,1-\epsilon_{\text{low}},\,1+\epsilon_{\text{high}})\,\hat{A}_t\Big)\Big], \tag{2}$$

where $\rho_t(\theta) = \pi_\theta(a_t|s_t)/\pi_{\theta_{\text{old}}}(a_t|s_t)$ and $\hat{A}_t$ is the GRPO group-relative advantage computed from $n = 16$ rollouts per prompt. We use $\epsilon_{\text{low}} = 0.2$, $\epsilon_{\text{high}} = 0.28$ (clip-higher) and dynamic filtering of zero-variance rollout groups (those for which all n rollouts share the same reward, i.e. the advantage is identically zero). On top of r_{task} in Eq. 1 we apply a soft overlong-response penalty over a $L_{\text{soft}}{=}2{,}048$-token buffer within an $L_{\text{max}}{=}4{,}096$-token response budget:

$$r_{\text{overlong}}(L) \;=\; -\,\lambda\,\max\!\left(0,\;\frac{L - L_{\text{soft}}}{L_{\text{max}} - L_{\text{soft}}}\right), \qquad \lambda{=}1, \tag{3}$$

where L is the per-rollout response length in tokens. The total per-rollout reward is $r = r_{\text{task}} + r_{\text{overlong}}$. Sequences that reach L_{max} receive the full $-\lambda$, while sequences below L_{soft} are unpenalised. Per-stage hyperparameters are in App. A.7; Algorithm 1 summarises the full pipeline.

This broad-then-focused design mirrors the SFT curriculum and is especially important in medicine, where strong average accuracy can mask weaknesses on rare or complex cases. Full training configurations/hyperparameters for SFT/RL and compute requirements are reported in Appendix A.7. The recipe is not Qwen3-specific: applying all four stages to DeepSeek-R1-Distill-Qwen-1.5B [30], a Qwen2.5-derived reasoning checkpoint, lifts the seven-task closed-ended average from 28.4 to 52.2 and HealthBench from 7.0 to 58.0 (App. A.5).

5 Evaluation

5.1 Methodology

Generation protocol. Unless otherwise noted, each model is evaluated with the generation parameters recommended for that model, and every reported evaluation is run three times with the arithmetic mean reported. For MEDPSY and the Qwen3 baselines we use `temperature=0.6`, `top_k=20`, `top_p=0.95`, and `max_output_tokens=16384`.

[3] Concretely, our 4B Stage-2 set contains 2,439 items (4.7% of the original set was still "Difficult" after Stage 1), and the 1.7B Stage-2 set contains 4,317 items (8.3% Difficult after Stage 1).

Table 1: Closed-ended benchmark results, sorted by average. Average is over seven closed-ended benchmarks; bold marks best per column within each size class.

Model	Avg	MMLU Health	AfriMedQA	MMLU-Pro Health	MedMCQA	MedQA (USMLE)	MedXpertQA	PubMedQA
MEDPSY-4B	**70.54**	89.70	71.50	70.45	72.15	**84.39**	**30.61**	**75.00**
MedGemma-27B-text-it	69.95	**90.48**	**73.07**	**72.94**	**72.77**	83.29	25.18	71.93
Qwen3-4B-Thinking-2507	63.10	85.92	64.12	67.73	61.78	70.91	16.69	74.53
MEDPSY-1.7B	**62.62**	**82.72**	**64.84**	**61.37**	**63.56**	**75.05**	**21.28**	69.53
MedGemma-1.5-4B-it	51.20	67.69	54.38	47.31	50.08	64.39	15.80	58.73
Qwen3-1.7B (Thinking)	49.95	72.49	51.87	45.07	49.14	47.18	11.60	**72.33**

Closed-ended (LLM-as-Parser). For all closed-ended benchmarks (MMLU-Health, MMLU-Pro Health, MedMCQA, MedQA-USMLE, MedXpertQA, AfriMedQA, PubMedQA), we use an LLM-as-Parser evaluation implemented in OpenCompass [31]. MMLU-Health is reported as the averaged accuracy across six MMLU sub-domains: `anatomy`, `clinical_knowledge`, `college_biology`, `college_medicine`, `medical_genetics`, and `professional_medicine`. The student model generates its full unconstrained response (with reasoning), and a separate reasoning model parses the final selected option, which is exact-matched against ground truth. This decouples generation from evaluation, avoiding both the suppression of reasoning that comes with letter-only constrained decoding and the false negatives that plague regex-based extraction. Main closed-ended tables use QwQ [32] as the parser; an example parser prompt is in App. A.1. To estimate parser sensitivity, we re-scored the same generations with three independent parsers (Llama-3.3-70B-Instruct [33], GPT-OSS-120B [28], and QwQ); the maximum parser-induced average-score spread is only 0.45 points across all evaluated models, with stable model rankings and coherent per-benchmark differences (App. A.11).

HealthBench (physician-authored rubrics with model-based grading). HealthBench [14] evaluates 5,000 realistic health conversations using 48,562 conversation-specific criteria written by 262 physicians. Each criterion specifies what a response should include or avoid and carries a physician-assigned importance weight; model graders apply this fixed expert-authored standard to each response. The benchmark's model-grading protocol was validated against physician judgments. We evaluate with three independent grader families (**CompassJudger-2-32B-Instruct** [34], **Llama-3.3-70B-Instruct** [33], and **GPT-OSS-120B** [28]) to test whether conclusions depend on a particular grader. Table 2 reports all three directly, while per-dimension breakdowns use CompassJudger-2-32B-Instruct (App. A.9). Three-grader results for teacher and backbone selection are reported in App. A.2–A.4. HealthBench-Hard is the hardest subset, requiring multi-step reasoning, complex safety judgments, and expert-level clinical knowledge.

Train–test decontamination. To account for overlaps in public medical datasets, we reran the closed-ended evaluation after excluding matched benchmark test items. Decontamination removed 157 examples (1.10%) without altering model rankings (App. A.11).

5.2 Closed-ended results

MEDPSY-4B matches MedGemma-27B-text-it in aggregate closed-ended performance (70.54 vs. 69.95; +0.59) at 6.75× smaller scale. A matched-instance analysis following Medmarks v0.1.0 [35] likewise finds near parity across three runs (50.25% versus 49.75%; App. A.11). Compared to its own backbone, post-training adds **+7.44** average. The lead is not explained by parser choice or train–test overlap: the ranking is stable across three closed-ended parsers and remains after decontamination (App. A.11). MEDPSY-1.7B reaches 62.62, beating MedGemma-1.5-4B-it by +11.42 and matching Qwen3-4B-Thinking-2507 at less than half its size; the largest gains over MedGemma-1.5-4B are MMLU-Health (+15.03) and MedQA-USMLE (+10.66). PubMedQA is a known ceiling case: the Qwen3-1.7B backbone already scores 72.33, so the 1.7B post-training trades -2.80 on PubMedQA for double-digit gains elsewhere; the 4B model avoids this regression and improves PubMedQA to 75.00. Accuracy also carries over to much longer inputs: on LongHealth [36] (5,090–6,754-word clinical documents), MEDPSY-4B scores 0.855/0.875 on Tasks 1/2, above MedGemma-27B-text-it (0.828/0.802), and MEDPSY-1.7B leads MedGemma-1.5-4B-it by up to 30.6 points (App. A.16).

Table 2: HealthBench overall scores under three independent model graders, Standard (Std) and Hard splits. Bold marks the best score per column. Per-dimension breakdowns under CompassJudger-2-32B-Instruct are in App. A.9 (Tab. 17–18).

	CompassJudger-2-32B		Llama-3.3-70B		GPT-OSS-120B	
Model	**Std**	**Hard**	**Std**	**Hard**	**Std**	**Hard**
MEDPSY-4B	**74.00**	**58.00**	**66.33**	**48.33**	**51.33**	**28.67**
MEDPSY-1.7B	70.33	54.33	63.00	46.00	46.00	24.67
MedGemma-27B-text-it	65.00	42.67	59.00	36.00	44.67	13.00
Qwen3-4B-Thinking-2507	63.00	42.00	56.00	34.00	36.67	9.33
MedGemma-1.5-4B-it	54.00	29.67	48.00	24.67	31.00	2.00

Table 3: Training stage progression. Comparison of MEDPSY-1.7B (1.7B) and MEDPSY-4B (4B) across all benchmarks. Avg is the closed-ended mean; HB is HealthBench Standard (CompassJudger-2-32B).

	Avg		HB		MMLU Health		Afri MedQA		MMLU-Pro Health		MedMCQA		MedQA		MedXpert QA		PubMed QA	
Stage	**1.7B**	**4B**	**1.7B**	**4B**	**1.7B**	**4B**	**1.7B**	**4B**	**1.7B**	**4B**	**1.7B**	**4B**	**1.7B**	**4B**	**1.7B**	**4B**	**1.7B**	**4B**
Backbone	49.95	63.10	53.00	63.00	72.49	85.92	51.87	64.12	45.07	67.73	49.14	61.78	47.18	70.91	11.60	16.69	72.33	74.53
+ SFT-1	57.48	67.93	70.33	74.00	79.70	88.77	60.90	69.35	57.05	68.87	57.26	67.72	63.97	82.38	17.09	26.22	66.40	72.20
+ SFT-2	59.70	69.29	70.33	74.00	80.14	89.58	62.45	70.47	60.27	69.77	58.54	69.21	68.76	84.32	18.89	28.71	68.87	73.00
+ RL-1	60.00	68.85	70.33	74.00	80.92	89.32	63.90	71.23	59.66	69.56	61.33	71.21	68.21	84.13	16.26	25.47	69.73	71.07
+ RL-2	**62.62**	**70.54**	**70.33**	**74.00**	**82.72**	**89.70**	**64.84**	**71.50**	**61.37**	**70.45**	**63.56**	**72.15**	**75.05**	**84.39**	**21.28**	**30.61**	69.53	**75.00**
Total Δ	**+12.67**	**+7.44**	**+17.33**	**+11.00**	+10.23	+3.78	+12.97	+7.38	+16.30	+2.72	+14.42	+10.37	+27.87	+13.48	+9.68	+13.92	−2.80	+0.47

5.3 HealthBench results

HealthBench complements closed-ended accuracy by evaluating responses to realistic, open-ended clinical scenarios against fixed, conversation-specific physician-authored rubrics. Its seven dimensions cover expertise-tailored communication, response depth, context seeking, emergency referrals, global health, health-data tasks, and responding under uncertainty. Table 2 reports Standard and Hard scores under all three grader families.

MEDPSY-4B ranks first on every (grader, split) cell of Tab. 2, and MEDPSY-1.7B is second-best across the board, surpassing MedGemma-27B-text-it on both Standard and Hard under all three graders. Absolute scores vary with grader strictness, but the complete five-model ordering is identical across all six grader-by-split cells, and the Hard-split lead is uniformly larger than the Standard-split lead under every grader. The gains therefore do not depend on the calibration of a single model grader (cross-grader consistency summary in App. A.8).

A per-dimension breakdown under CompassJudger-2-32B-Instruct is reported in App. A.9: MEDPSY-4B leads on all seven dimensions in both Standard and Hard evaluations, while MEDPSY-1.7B surpasses MedGemma-27B-text-it on all seven HealthBench-Hard dimensions.

5.4 Stage-wise progression

Several patterns are consistent across both sizes (Table 3). *SFT Stage 1 delivers the largest single jump* (+7.53 / +4.83), confirming that broad medical exposure from Corpus 1 is the foundation; gains are most dramatic on HealthBench (+17.33 / +11.00) and MedQA-USMLE (+16.79 / +11.47). *SFT Stage 2 adds focused reasoning* on MedQA-USMLE (+4.79 for 1.7B), MMLU-Pro Health (+3.22 for 1.7B), and MedXpertQA (+2.49 for 4B). *The two RL stages together provide the final sharpening*, peaking on MedQA-USMLE (+6.29 for 1.7B) and MedMCQA (+5.02 / +2.94), the benchmarks that most reward structured clinical reasoning. Stage 2 hard-mining accounts for essentially all of the released model's MedXpertQA lift: replacing the mined Stage-2 set with a uniformly-random subset of identical size loses 5.0 pp (4B) and 2.0 pp (1.7B) on MedXpertQA, while easy/medium benchmarks are essentially unchanged (App. A.17.1). *HealthBench gains are installed early*: SFT Stage 1 already lifts HealthBench to its final level, and the RL stages preserve it while improving closed-ended accuracy and token efficiency.

Table 4: Quantized deployment profile. (a) Complete quantized GGUF variants comparison. Δ Size relative to BF16; Δ Score is the absolute change in $(\text{HB Overall} + \text{CE Avg})/2$. **(b)** Apple M5 Pro measurements at matched `Q4_K_M` quantization over 30 prompts (20 MedQA and 10 MedXpertQA-Text); time per answer is the median, and energy per answer is estimated as sustained SoC power (26.5 W) times median wall time.

(a) Complete six-format comparison

		Size (GB)		Δ Size		HB		HB-H		CE Avg		Δ Score	
Variant	**imat**	**4B**	**1.7B**	**4B**	**1.7B**	**4B**	**1.7B**	**4B**	**1.7B**	**4B**	**1.7B**	**4B**	**1.7B**
BF16	—	8.3	3.8	—	—	74	70	58	54	70.54	62.62	—	—
Q8_0	no	4.4	2.1	−47%	−45%	74	70	57	55	70.25	62.62	−0.15	0.00
Q4_K_M	yes	2.6	1.2	−69%	−68%	73	69	56	52	69.92	62.16	−0.81	−0.73
IQ4_NL	yes	2.5	1.2	−70%	−68%	73	69	57	51	69.50	60.22	−1.02	−1.70
IQ4_XS	yes	2.4	1.1	−71%	−71%	73	69	57	53	69.39	60.05	−1.08	−1.79
IQ3_M	yes	2.0	0.98	−76%	−74%	73	67	58	49	68.55	58.46	−1.50	−3.58
IQ3_XXS	yes	1.8	0.85	−78%	−78%	69	59	51	40	64.42	48.71	−5.56	−12.46

(b) Matched `Q4_K_M` local inference

Model	**Peak RAM**	**Decode**	**Tokens/answer**	**Time/answer**	**Energy/answer**
MEDPSY-4B	4.29 GB	83.9 tok/s	738	8.75 s	232 J
Qwen3-4B-Thinking-2507	4.28 GB	82.2 tok/s	2,727	34.6 s	917 J

5.5 Token efficiency

Sample-weighted across all eight benchmarks, the 4B model achieves a **3.2**× reduction in average response length (**909** vs. **2,953** tokens for Qwen3-4B-Thinking-2507) while *improving* accuracy, and the 1.7B model a **1.7**× reduction (**1,110** vs. **1,901** tokens for Qwen3-1.7B Thinking). Complementing these observed length reductions, an explicit thinking-budget sweep shows that MEDPSY-4B retains 96.6% of its unbounded closed-ended score with a 300-token thinking cap (App. A.14). The 1.7B reduction is smaller because its backbone already emits shorter outputs, and on HealthBench MEDPSY-1.7B writes *slightly longer* responses than its backbone, trading tokens for the +17 HealthBench points of Sec. 5.4. The per-benchmark breakdown is in App. A.10 (Tab. 20).

We decompose the RL-side compression into three independently controllable mechanisms in App. A.17.2: vanilla GRPO with a KL anchor compresses by only $\sim$ 6–20%, switching to DAPO's drift-enabling modifications (KL-removal and asymmetric clip-higher) brings compression to $\sim$ 44–45%, and adding the soft overlong-response penalty pushes it to the released-model $\sim$ 65–66%. The *ordering* of the RL stages compounds this: Stage 1 trains on items typically answerable with short reasoning, installing a low-token-budget habit; Stage 2 then pushes accuracy on hard items starting from this already-compressed checkpoint and preserves the short-reasoning habit.

5.6 Quantization for mobile deployment

To meet mobile RAM constraints, we convert both models to GGUF with llama.cpp [26] and evaluate six quantization formats spanning 8-, 4-, and 3-bit precision. This sweep identifies `Q4_K_M` as the preferred deployment configuration, offering the strongest observed balance between footprint reduction and full-suite score retention at both model sizes. Table 4 reports the complete per-format comparison together with the real-device profile at `Q4_K_M`; the imatrix ablation and per-benchmark details are in App. A.12.

`Q4_K_M` accuracy and footprint. At `Q4_K_M`, both model sizes reduce disk footprint by 68–69% while keeping aggregate score within one point of BF16 (Table 4(a)). MEDPSY-4B reaches 2.6 GB with a 0.81-point aggregate-score change, while MEDPSY-1.7B reaches 1.2 GB with a 0.73-point change. Across the broader six-format study, the 4B model retains strong accuracy through 3-bit quantization, and the matched-footprint comparison reveals a capacity-vs-precision effect: the 2.0 GB `IQ3_M` 4B model outperforms the 2.1 GB `Q8_0` 1.7B model (Table 4(a)). At matched on-disk size, allocating storage to more parameters at lower precision buys more performance than fewer parameters at higher precision.

Full quantization landscape. Beyond `Q4_K_M`, the six-format comparison (Table 4(a); Fig. 1b) shows that `Q8_0` is near-lossless at both sizes (-0.15 and 0.00 aggregate-score change at roughly half the footprint). The I-quants trade precision for footprint gracefully at 4B: `IQ4_NL` and `IQ4_XS` lose 1.0–1.1 points, and `IQ3_M` (2.0 GB) loses only 1.50 while exactly matching the BF16 HealthBench-Hard score of 58. Collapse occurs only at `IQ3_XXS` (-5.56 at 4B, -12.46 at 1.7B), and even this worst configuration (64.42 closed-ended, 69 HealthBench) remains above the unquantized Qwen3-4B-Thinking-2507 backbone (63.10/63) and MedGemma-1.5-4B-it (51.20/54). The 1.7B model degrades roughly twice as fast as the 4B below 4-bit precision, consistent with the intuition that smaller models have less weight redundancy: every channel carries more of the model's behavior, so precision lost per channel translates more directly into degraded reasoning. We therefore do not recommend any sub-4-bit variant of MEDPSY-1.7B for medical use.

Real-device deployment. At matched `Q4_K_M` quantization, MEDPSY-4B and its Qwen3 backbone have comparable per-token decode rates, while MEDPSY-4B generates 73% fewer tokens per answer. This translates to 74.7% lower median end-to-end latency and estimated energy per answer on the Apple M5 Pro, with 4.29 GB peak RAM (Table 4(b)). A separate 8K-context stress test records 4.83 GB as the largest laptop peak for MEDPSY-4B `Q4_K_M` (App. A.15). Measurements on a OnePlus 13 confirm fully local execution at 2.22 GB peak RAM for MEDPSY-1.7B and 3.98 GB for MEDPSY-4B; full protocols and measurements are in App. A.13.

6 Discussion and Limitations

The results support our central thesis: for compact medical LLMs, medical capability and deployment efficiency can be optimized jointly. In MEDPSY, broad SFT establishes medical coverage, focused SFT improves clinical reasoning, and hard-sample RL sharpens the most reasoning-heavy benchmarks (Sec. 5.4). The full curriculum (SFT$\rightarrow$SFT$\rightarrow$RL Stage 1$\rightarrow$RL Stage 2) is also strictly Pareto-dominant over single-stage hard-only RL on token-vs-accuracy (best performance at $\sim$ 3–4$\times$ shorter response length) (App. A.17.2). The Easy/Moderate Stage 1 leaves the policy already compressed and high-reward before Stage 2 mining targets residual hard items, preserving response compression even in the more challenging scenarios that require longer reasoning.

However, these findings should be interpreted within several important boundaries: the models are **text-only**; the training data is predominantly English. Like all LLMs, the models can produce confident but incorrect medical statements; they should not be used for autonomous diagnosis or treatment. Real clinical decision support would require additional safety review, IRB-grade evaluation, prospective testing, and regulatory clearance.

7 Conclusion and Future Work

These limitations point to several directions for future work. The recipe should be extended to multilingual and multimodal medical settings, and evaluation should move beyond the current benchmark suite. In particular, compact medical models need stronger open-ended evaluations, safety benchmarks, and medical error detection tests before their behavior can be understood in higher-risk clinical scenarios.

Within this scope, MEDPSY shows that compact, text-only medical LLMs (1.7B and 4B) can match or surpass much larger medical models: MEDPSY-4B matches MedGemma-27B on closed-ended benchmarks at 6.75$\times$ smaller scale, and both sizes surpass it on HealthBench-Hard. Both run fully locally within 5 GB of peak RAM on the tested laptop and smartphone, where shorter generations cut median latency and estimated energy per answer by 75%. A controlled synthetic-data pipeline, a broad-then-focused SFT curriculum, and hard-sample RL deliver strong medical accuracy and improved token efficiency (1.7–3.2$\times$ fewer generated tokens) at small scale. The quantization study further shows that `Q4_K_M` keeps aggregate score within one point of BF16 at a 68–69% smaller footprint, while very low precision exposes a clear capacity-vs-precision trade-off. Overall, MEDPSY suggests that for compact medical LLMs, *data quality, staged post-training, and alignment design dominate parameter scale, and medical capability and deployment efficiency can be optimized jointly.*

Acknowledgments and Disclosure of Funding

This research was supported and funded by Tether AI Research* which provided computational resources and financial compensation for the authors. No other party provided funding to support this research.

All authors are contractors engaged by or for Tether AI Research. The authors declare no other competing financial or non-financial interests.

References

[1] Andrew Sellergren, Sahar Kazemzadeh, Tiam Jaroensri, Atilla Kiraly, et al. Medgemma technical report. *arXiv preprint arXiv:2507.05201*, 2025. URL https://arxiv.org/abs/2507.05201.

[2] Andrew Sellergren, Chufan Gao, Fereshteh Mahvar, Timo Kohlberger, Fayaz Jamil, Madeleine Traverse, Alberto Tono, Bashir Sadjad, Lin Yang, et al. Medgemma 1.5 technical report. *arXiv preprint arXiv:2604.05081*, 2026. URL https://arxiv.org/abs/2604.05081.

[3] An Yang, Anfeng Li, Baosong Yang, et al. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

[4] Davide Vitabile, N. Ranjan, Akshay Nambiar, Kamal K. Gupta, and Amril Nazir. QVAC Genesis III: A large-scale, high-quality open synthetic stem corpus for efficient language model pre-training, 2026. URL https://arxiv.org/abs/2609.19513.

[5] M-A-P, Ge Zhang, Xinrun Du, Zhimiao Yu, Zili Wang, et al. FineFineWeb: A comprehensive study on fine-grained domain web corpus, December 2024. URL https://huggingface.co/datasets/m-a-p/FineFineWeb.

[6] Chi Liu, Derek Li, Yan Shu, Robin Chen, Derek Duan, Teng Fang, and Bryan Dai. Fleming-R1: Toward expert-level medical reasoning via reinforcement learning, 2025. URL https://arxiv.org/abs/2509.15279.

[7] 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. doi: 10.3390/app11146421. URL https://doi.org/10.3390/app11146421.

[8] Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. MedMCQA: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pages 248–260. PMLR, 2022. URL https://proceedings.mlr.press/v174/pal22a.html.

[9] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2009.03300.

[10] Yubo Wang, Xueguang Ma, Ge Zhang, et al. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems*, volume 37, pages 95266–95290, 2024. doi: 10.52202/079017-3018. URL https://arxiv.org/abs/2406.01574.

[11] Yuxin Zuo, Shang Qu, Yifei Li, Zhang-Ren Chen, Xuekai Zhu, Ermo Hua, Kaiyan Zhang, Ning Ding, and Bowen Zhou. MedXpertQA: Benchmarking expert-level medical reasoning and understanding. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 80961–80990. PMLR, 2025. URL https://proceedings.mlr.press/v267/zuo25a.html.

[12] Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. PubMedQA: A dataset for biomedical research question answering. In *Proceedings of EMNLP-IJCNLP*, pages 2567–2577. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1259. URL https://aclanthology.org/D19-1259/.

[13] Charles Nimo, Tobi Olatunji, Abraham Toluwase Owodunni, et al. AfriMed-QA: A pan-african, multi-specialty, medical question-answering benchmark dataset. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pages 1948–1973. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.96. URL https://aclanthology.org/2025.acl-long.96/.

*References to Tether AI Research are references to Tether Data, S.A. de C.V.

[14] Rahul K. Arora, Jason Wei, Rebecca Soskin Hicks, et al. HealthBench: Evaluating large language models towards improved human health, 2025. URL https://arxiv.org/abs/2505.08775.

[15] Karan Singhal, Shekoofeh Azizi, Tao Tu, et al. Large language models encode clinical knowledge. *Nature*, 620:172–180, 2023. doi: 10.1038/s41586-023-06291-2. URL https://doi.org/10.1038/s41586-023-06291-2.

[16] Karan Singhal, Tao Tu, Juraj Gottweis, et al. Toward expert-level medical question answering with large language models. *Nature Medicine*, 2025. doi: 10.1038/s41591-024-03423-7. URL https://doi.org/10.1038/s41591-024-03423-7.

[17] Alexandre Sallinen, Antoni-Joan Solergibert, Michael Zhang, Guillaume Boyé, Maud Dupont-Roc, Xavier Theimer-Lienhard, Etienne Boisson, Bastien Bernath, Hichem Hadhri, Antoine Tran, et al. Llama-3-Meditron: An open-weight suite of medical llms based on llama-3.1. In *Workshop on Large Language Models and Generative AI for Health at AAAI 2025*, 2025. URL https://openreview.net/forum?id=ZcD35zKujO.

[18] Ankit Pal and Malaikannan Sankarasubbu. OpenBioLLMs: Advancing open-source large language models for healthcare and life sciences, 2024. URL https://huggingface.co/aaditya/OpenBioLLM-Llama3-70B.

[19] John Snow Labs. JSL-MedLlama-3-8B-v2.0, 2024. URL https://huggingface.co/johnsnowlabs/JSL-MedLlama-3-8B-v2.0.

[20] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, et al. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 13484–13508. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.754. URL https://aclanthology.org/2023.acl-long.754/.

[21] Can Xu, Qingfeng Sun, Kai Zheng, et al. WizardLM: Empowering large pre-trained language models to follow complex instructions. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2304.12244.

[22] Che Liu, Haozhe Wang, Jiazhen Pan, Zhongwei Wan, Yong Dai, Fangzhen Lin, Wenjia Bai, Daniel Rueckert, and Rossella Arcucci. Beyond distillation: Pushing the limits of medical llm reasoning with minimalist rule-based rl, 2025. URL https://arxiv.org/abs/2505.17952.

[23] Qiying Yu, Zheng Zhang, Ruofei Zhu, et al. DAPO: An open-source llm reinforcement learning system at scale. In *Advances in Neural Information Processing Systems*, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/a4277440d50f1f15d2cb4c14f7e0c0d2-Abstract-Conference.html.

[24] Gengyang Li, Yifeng Gao, Yuming Li, and Yunfang Wu. ThinkLess: A training-free inference-efficient method for reducing reasoning redundancy. *arXiv preprint arXiv:2505.15684*, 2025. URL https://arxiv.org/abs/2505.15684.

[25] Yunho Jin, Gu-Yeon Wei, and David Brooks. The energy cost of reasoning: Analyzing energy usage in LLMs with test-time compute. *arXiv preprint arXiv:2505.14733*, 2025. URL https://arxiv.org/abs/2505.14733.

[26] Georgi Gerganov et al. llama.cpp: Llm inference in c/c++. GitHub repository, 2023. URL https://github.com/ggml-org/llama.cpp.

[27] Baichuan-M3 Team, Chengfeng Dou, Fan Yang, et al. Baichuan-M3: Modeling clinical inquiry for reliable medical decision-making, 2026. URL https://arxiv.org/abs/2602.06570. Model checkpoint: https://huggingface.co/baichuan-inc/Baichuan-M3-235B.

[28] OpenAI, Sandhini Agarwal, Lama Ahmad, et al. gpt-oss-120b & gpt-oss-20b model card, 2025. URL https://arxiv.org/abs/2508.10925.

[29] Xiaotian Zhang, Yuan Wang, Zhaopeng Feng, et al. Med-U1: Incentivizing unified medical reasoning in llms via large-scale reinforcement learning, 2025. URL https://arxiv.org/abs/2506.12307.

[30] DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

[31] Maosong Cao, Kai Chen, Haodong Duan, Yixiao Fang, Zhiwei Fei, Tong Gao, et al. Opencompass: A universal evaluation platform for large language models. *arXiv preprint arXiv:2605.19276*, 2026. URL https://arxiv.org/abs/2605.19276.

[32] Qwen Team. QwQ-32B: Embracing the power of reinforcement learning. Hugging Face model card and Qwen blog, 2025. URL https://huggingface.co/Qwen/QwQ-32B.

[33] AI@Meta. Llama 3.3 model card, 2024. URL https://github.com/meta-llama/llama-models/blob/main/models/llama3_3/MODEL_CARD.md.

[34] Taolin Zhang, Maosong Cao, Alexander Lam, Songyang Zhang, and Kai Chen. CompassJudger-2: Towards generalist judge model via verifiable rewards, 2025. URL https://arxiv.org/abs/2507.09104.

[35] Benjamin Warner, Ratna Sagari Grandhi, Max Kieffer, Aymane Ouraq, Saurav Panigrahi, et al. Medmarks: A comprehensive open-source llm benchmark suite for medical tasks. *arXiv preprint arXiv:2605.01417*, 2026. URL https://arxiv.org/abs/2605.01417.

[36] Lisa Adams, Felix Busch, Tianyu Han, Jean-Baptiste Excoffon, Jana Ortmann, Markus Rockenbach, Moritz Ogrodnik, Danial Ruckert, et al. Longhealth: A question answering benchmark with long clinical documents. *arXiv preprint arXiv:2401.14490*, 2024. URL https://arxiv.org/abs/2401.14490.

[37] Yudong Wang, Zixuan Fu, Jie Cai, Peijun Tang, Hongya Lyu, Yewei Fang, Zhi Zheng, Jie Zhou, Guoyang Zeng, Chaojun Xiao, Xu Han, and Zhiyuan Liu. Ultra-fineweb: Efficient data filtering and verification for high-quality llm training data, 2025. URL https://arxiv.org/abs/2505.05427.

[38] UK AI Security Institute. Inspect AI: Framework for large language model evaluations. Software, 2024. URL https://inspect.aisi.org.uk/.

[39] Zhihong Shao, Peiyi Wang, Qihao Zhu, et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

[40] AI@Meta. Llama 3.2 model card, 2024. URL https://github.com/meta-llama/llama-models/blob/main/models/llama3_2/MODEL_CARD.md.

[41] 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*, 2025. doi: 10.1145/3689031.3696075. URL https://doi.org/10.1145/3689031.3696075.

A Technical appendices and supplementary material

Appendix contents

A.1 Data sources and prompts 14
A.2 Teacher-selection ablation 15
A.3 Additional large medical baselines 16
A.4 Backbone-selection ablation 16
A.5 Transfer to a Qwen2.5-derived reasoning checkpoint 18
A.6 Backbone architectures 18
A.7 Compute resources 18
A.8 Per-grader HealthBench results 19
A.9 HealthBench per-dimension and per-axis breakdowns 19
A.10 Per-benchmark response length 20
A.11 Evaluation Robustness and Decontamination 21
A.12 Quantization: imatrix ablation and per-benchmark tables 22
A.13 Real-device measurement protocol 23
A.14 Accuracy under explicit thinking budgets 24
A.15 Context-length scaling 25
A.16 Long-context accuracy on LongHealth 25

A.17 RL ablation studies 26
A.18 Existing assets, licenses, and terms of use 32

A.1 Data sources and prompts

Corpus 1 uses selected Genesis III biology/medicine seeds plus new health seeds. For the health slice, we select FineFineWeb health-domain passages after Ultra-FineWeb-classifier[37] filtering and apply the scaling-QA prompt below to generate self-contained MCQ user prompts. We keep only the question/options as the user turn, then use Baichuan-M3-235B to generate every assistant target; Corpus 2 uses only the public train sources listed in Tab. 5 as the user turn, then uses Baichuan-M3-235B as well to generate every assistant target.

Table 5: Final Data Mix and Source Licenses. SFT row counts represent final selected input for assistant target generation; RL counts describe the AlphaMed19K prompt pool used for DAPO.

Group	Source / Role	Rows	License / Note
Corpus 1: Broad SFT & Seeds			
Seeds	Genesis III (Biology/Medicine)	8,089,891	CC-BY-NC-4.0
Seeds	FineFineWeb (Health)	2,133,642	Apache-2.0
SFT-1	Synthetic Broad Med Corpus	10,223,533	Inherited
Corpus 2: Focused SFT			
SFT-2	AfriMedQA MCQ Train	124	CC-BY-4.0
SFT-2	MedMCQA Train	129,231	Apache-2.0
SFT-2	MedQA-USMLE Train	9,176	CC-BY-4.0
RL Prompt Pool			
RL	AlphaMed19K	$\sim$19,000	MIT

Scaling-QA prompt for new health-domain seeds. The prompt template used for generating samples starting from the seeds is as follows:

```
You are a {{level}} {{domain}} tutor. I will give you a context passage.
IGNORE the details of the passage. Use it only as inspiration for topics, but DO NOT refer to the
     passage, book, text, or "discussed/mentioned" material in your questions.
Context:
{{output}}

Generate EXACTLY 4 independent, self-contained questions suitable for {{level}} level {{domain}}.

Output rules (MUST be followed strictly):
- Output EXACTLY 4 samples.
- Each row = ONE question in the following format:
  "<question>","<choiceA>","<choiceB>","<choiceC>","<choiceD>","<correct_choice_letter>"
- Each row MUST end with a newline (line break).
- Do NOT join multiple questions into one line.
- Do NOT output any extra text, headers, or blank lines.

Field definitions:
- <question> = a fully self-contained {{level}}-level {{domain}} problem.
  It must NOT reference any passage, book, figure, table, or "as discussed" or "as mentioned" style
     wording.
  Include all necessary values, constants, or definitions directly inside the question.
- <choiceA> ... <choiceD> = four mutually exclusive, plausible answer choices.
- <correct_choice_letter> = exactly one of A, B, C, D.

Question variety:
- Use a mix of imperative (e.g., "Calculate..."), interrogative, completion/fill-in, and true/false style.
- Do NOT start all questions with the same word.
- Include realistic values/constants when needed (e.g., g = 9.8 m/s^2).
- Use standard SI units and symbols.

Hard constraints:
- Generate exactly 4 questions (4 samples).
- Correct answers must be evenly distributed: one A, one B, one C, one D.
- Randomize placement of correct answers; no clustering.
- Each row must contain exactly 6 comma-separated values.
- Do NOT add quotes around the entire block; only around individual fields.
```

```
Output ONLY the 4 samples in the required format, nothing else.
```

Closed-ended answer-extraction prompt. This is the MMLU extractor; the same template was used for each closed-ended dataset, changing only the option list and dataset-specific field names.

```
MMLU_ANSWER_EXTRACTOR_TEMPLATE = """
You are an expert answer extractor. Your ONLY job is to extract the final answer from the candidate's
     response.

CRITICAL INSTRUCTIONS - READ CAREFULLY:
- DO NOT solve the question yourself
- DO NOT generate a new answer
- DO NOT think about what the correct answer should be
- DO NOT evaluate whether the candidate's answer is right or wrong
- ONLY extract what the candidate actually wrote as their final answer

Your task is purely extraction, not generation or evaluation.

Here are the extraction guidelines for MMLU multiple choice questions:
1. Look for the candidate's final answer in their response. This should be one of: A, B, C, or D
   - Look for explicit statements like "ANSWER: A", "The answer is B", "Final answer: C"
   - Look for \boxed{A} format (extract what's inside the braces)
   - Look for standalone letters A, B, C, or D in their conclusion
   - The final choice they settle on in their reasoning

2. If the candidate's response contains multiple different answers:
   - If they state one answer but then correct/fix themselves, extract the corrected/final answer they
     provided
   - If they state multiple different answers without choosing one or correcting themselves, return
     "MULTIPLE_ANSWERS"

3. If you cannot clearly identify a single letter choice (A, B, C, or D) in the response, return
     "NO_ANSWER"

4. If the candidate generates new questions:
   - DO NOT extract the answer from the new generated questions
   - ONLY extract the answer from the original question

RESPONSE FORMAT:
First, provide a brief explanation of why you are extracting that particular answer (what indicators you
     found in the candidate's response).
Then, provide the extracted answer as a single letter.

Use this exact format:

Extraction Reasoning: [Brief explanation of what indicators led you to extract this answer from the
     candidate's response]

Extracted Candidate's Answer: [A single letter: A, B, C, or D. Use MULTIPLE_ANSWERS if the candidate
     provided multiple different answers, or NO_ANSWER if no clear answer was found.]

<Original Question Begin>:
Question: {input}

A. {A}
B. {B}
C. {C}
D. {D}

Answer:
<Original Question End>

<Candidate's Response Begin>:
{prediction}
<Candidate's Response End>
"""
```

A.2 Teacher-selection ablation

This appendix details the ablation study performed to determine the most effective teacher model for our distillation pipeline. We evaluate candidate models across several closed-ended medical benchmarks in Table 6 and report their overall HealthBench performance across three distinct graders in Table 7.

As shown in the results, Baichuan-M3-235B consistently outperforms the other candidates, which justifies its selection as the primary teacher for our experiments.

Table 6: Teacher closed-ended benchmark results. Bold marks best per column.

Teacher	Avg	AfriMedQA	MMLU-H	MedQA	PubMedQA	MedMCQA	MedXpertQA	MMLU-Pro-H
Baichuan-M3-235B	**74.83**	**74.58**	**93.08**	88.90	73.27	**76.88**	**40.91**	**76.20**
GPT-OSS-120B	72.73	73.28	91.85	**89.03**	75.20	73.26	35.30	71.15
Fleming-R1-32B	72.55	72.68	92.61	85.91	**79.20**	74.02	29.93	73.47

Table 7: Teacher HealthBench overall scores across three graders.

Teacher	CompassJudger	Llama-3.3-70B	GPT-OSS-120B
Baichuan-M3-235B	**77**	**71**	**58**
GPT-OSS-120B	69.67	63	52.33
Fleming-R1-32B	67	61	45.67

Student-level validation of teacher choice. To test whether the teacher-level HealthBench advantage transfers to the student, we trained three controlled 4B SFT arms from the same base model on the same 170K-prompt subset with the same training pipeline, changing only the source of the target responses. Table 8 reports the results.

Table 8: Student-level teacher ablation (4B students, identical 170K prompts, varying only the SFT target source). Closed-ended averages differ by at most 0.41 points, but HealthBench separates the arms by 11 points; the mixed model stays near Fleming-only rather than at the 65.25 midpoint of the single-teacher arms.

SFT target source	**Closed-ended avg**	**HealthBench**
Fleming-R1-32B only	63.43	59.5
Baichuan-M3-235B only	63.02	**71.0**
Both teachers, mixed	63.40	60.0

Closed-ended averages differ by at most 0.41 points, whereas HealthBench separates the arms by 11 points. Mixing heterogeneous target teachers provides no benefit here: the mixed arm (60.0) tracks the Fleming-only arm (59.5) rather than the midpoint of the two single-teacher arms (65.25), consistent with a less coherent SFT signal. In this controlled run, the Baichuan-only arm therefore best preserves the open-ended criterion used for teacher selection.

A.3 Additional large medical baselines

Table 9 reports the closed-ended benchmark comparison against additional open medical LLMs in the 8–70B range.

Table 9: Closed-ended comparison with additional open medical baselines. Average is computed over the seven closed-ended benchmarks.

Model	Avg	AfriMedQA	MMLU-H	MedQA	PubMedQA	MedMCQA	MedXpertQA	MMLU-Pro-H
MEDPSY-4B	**70.54**	71.50	89.70	**84.39**	75.00	72.15	**30.61**	70.45
MedGemma-27B-text-it	69.95	**73.07**	**90.48**	83.29	71.93	72.77	25.18	**72.94**
Meditron3-70B	69.57	71.65	88.93	80.65	**80.07**	70.74	23.66	71.31
OpenBioLLM-70B	67.46	67.46	90.10	76.12	77.93	**72.92**	19.22	68.46
Meditron3-8B	57.51	60.21	76.55	62.74	74.47	58.35	14.39	55.87
JSL-MedLlama-3-8B-v2.0	55.03	54.12	77.92	56.35	74.47	57.97	11.92	52.45
MedGemma-1.5-4B-it	51.20	54.38	67.69	64.39	58.73	50.08	15.80	47.31

A.4 Backbone-selection ablation

We evaluated multiple candidate backbones at each target size on our full medical benchmark suite *before* any post-training. Tables 10, 11, 12, 13 report the results.

Table 10: 4B-class backbone closed-ended results.

Model	Avg	MMLU-H	AfriMedQA	MMLU-Pro-H	MedMCQA	MedQA	MedXpertQA	PubMedQA
Qwen3-4B-Thinking-2507	**63.10**	**85.92**	**64.12**	**67.73**	**61.78**	**70.91**	**16.69**	74.53
Qwen3-4B (Thinking)	60.64	83.89	63.15	61.82	60.21	67.64	15.01	72.80
Llama-3.2-3B-Instruct	49.67	66.81	52.39	39.97	52.35	49.51	12.43	**74.20**
SmolLM3-3B	48.99	72.64	47.24	47.23	46.20	46.40	11.32	71.93
gemma-3-4b-it	42.59	62.72	45.46	34.68	45.08	40.51	10.52	59.13

Table 11: 4B-class backbone HealthBench results across three graders.

Model	CompassJudger	Llama-3.3-70B	GPT-OSS-120B
Qwen3-4B-Thinking-2507	**63**	**56**	36.67
Qwen3-4B (Thinking)	62	55	**37.67**
gemma-3-4b-it	59	53	33.33
SmolLM3-3B	50	43.67	24
Llama-3.2-3B-Instruct	37.33	32	14.67

Table 12: Sub-2B-class backbone closed-ended results.

Model	Avg	MMLU-H	AfriMedQA	MMLU-Pro-H	MedMCQA	MedQA	MedXpertQA	PubMedQA
Qwen3-1.7B (Thinking)	**49.95**	**72.49**	**51.87**	**45.07**	**49.14**	**47.18**	**11.60**	**72.33**
LFM2.5-1.2B-Thinking	44.15	63.48	45.07	37.81	42.11	39.85	11.54	69.20
Llama-3.2-1B-Instruct	36.18	49.34	36.84	23.88	37.89	34.04	10.25	61.00
SmolLM2-1.7B-Instruct	33.32	49.27	31.39	20.70	33.80	29.22	9.86	59.00

Table 13: Sub-2B-class backbone HealthBench results across three graders.

Model	CompassJudger	Llama-3.3-70B	GPT-OSS-120B
Qwen3-1.7B (Thinking)	**53**	**47.33**	**27.67**
LFM2.5-1.2B-Thinking	49	41.67	22.33
Llama-3.2-1B-Instruct	25	22	5.33
SmolLM2-1.7B-Instruct	23	18.67	6

Among the tested sub-2B candidates, Qwen3-1.7B provided the strongest pretrained starting point across the closed-ended suite and all three HealthBench graders, motivating its selection for the final model.

A.5 Transfer to a Qwen2.5-derived reasoning checkpoint

To test the portability of our post-training recipe beyond Qwen3, we applied the same four-stage pipeline to DeepSeek-R1-Distill-Qwen-1.5B [30], a compact reasoning model built on Qwen2.5-Math and distilled from DeepSeek-R1. Table 14 shows consistent gains at every training stage on both the seven closed-ended benchmarks and HealthBench, demonstrating that the recipe also transfers effectively to a Qwen2.5-derived reasoning checkpoint.

Table 14: Stage-wise transfer results for DeepSeek-R1-Distill-Qwen-1.5B. Scores have weighted mean output tokens from the same protocol in parentheses. The explicit-reasoning rows add a "Think step by step" instruction and required final-answer format; the paper-mirror rows do not.

Metric	Base	SFT-1	SFT-2	RL-1	RL-2
Closed-ended avg. (7), paper mirror	28.42 (1,209)	44.07 (2,430)	46.96 (2,394)	49.25 (1,107)	52.16 (1,601)
Closed-ended avg. (7), explicit reasoning	29.53 (3,007)	45.76 (2,730)	48.20 (2,664)	49.92 (1,068)	53.60 (1,476)
HealthBench	7.0 (1,292)	55.0 (2,530)	56.0 (2,450)	57.0 (2,071)	58.0 (2,089)

Evaluation behavior. During evaluation, we observed that DeepSeek-R1-Distill-Qwen-1.5B behaves differently depending on whether the prompt explicitly requests step-by-step reasoning. We therefore report two closed-ended settings: the paper-mirror setting uses the same prompts and answer format as the main experiments, whereas the explicit-reasoning setting adds "Think step by step" and requires the specified final-answer format. The explicit-reasoning setting is applied only to the closed-ended benchmarks; HealthBench is always evaluated under the standard protocol, without any added reasoning instruction or required answer format.

The recipe improves accuracy consistently in both settings. Under the paper-mirror evaluation, the seven-task average rises from 28.42 to 52.16, with gains on all seven tasks; after RL, output length is 33.1% below SFT-2 (1,601 vs. 2,394 tokens). Under explicit reasoning, the average rises from 29.53 to 53.60 while output length falls from 3,007 to 1,476 tokens, a 50.9% reduction. The two settings thus reach comparable final accuracy while exhibiting different length behavior. Under the standard HealthBench protocol, the score rises from 7.0 to 58.0, with output length increasing from 1,292 to 2,089 tokens. Overall, the transfer experiment shows strong gains across both elicitation styles, while explicit prompting primarily changes the token-efficiency profile.

A.6 Backbone architectures

Table 15 summarizes the architectural parameters of the two selected Qwen3 backbones.

Table 15: Architecture summary of the chosen Qwen3 backbones.

Parameter	Qwen3-1.7B	Qwen3-4B
Hidden size	2,048	2,560
FFN hidden size	6,144	9,728
Layers	28	36
Attention heads	16	32
KV groups (GQA)	8	8
Vocab size	151,936	151,936
Position embedding	RoPE	RoPE
Normalization	RMSNorm	RMSNorm
Activation	SwiGLU	SwiGLU

A.7 Compute resources

Cluster. 30 worker nodes; 8×NVIDIA H100 80GB HBM3 per node; InfiniBand interconnect (Mellanox ConnectX, 8 HCA ports/node); SLURM scheduler. Containers: NVIDIA NeMo 25.09 (Enroot), PyTorch 2.5.1, CUDA 12.1.

Per-job parallelism. 1.7B SFT: 4 nodes, 32 GPUs, TP=1, PP=1, DP=32. 4B SFT: 4 nodes, 32 GPUs, TP=1, PP=1, DP=32.

Training hyperparameters. Both SFT runs used 3 epochs, sequence length 4,096 with packed sequence size 4,096, global batch size 512 packed sequences, and micro batch size 1. Optimization used distributed fused Adam in bf16 mixed precision with a learning rate of 2×10^{-5}, minimum learning rate of 2×10^{-6}, cosine decay, and a linear warmup over 1% of training steps. For RL training, effective per-update batch is 512 prompts $\times$ n=16 rollouts = 8,192 sampled trajectories. We present the RL configuration in Table 16.

Table 16: RL training hyperparameters (DAPO).

Hyperparameter	S1 (1.7B)	S1 (4B)	S2 (1.7B)	S2 (4B)
Asymmetric clip $(\epsilon_{\text{low}}, \epsilon_{\text{high}})$	$(0.20,\ 0.28)$			
Entropy coefficient	0			
Optimizer / precision	AdamW, bf16 mixed			
Learning rate	1×10^{-6} (constant)			
Train batch (prompts)	512			
PPO mini-batch	128			
PPO micro-batch / GPU	8			
Gradient checkpointing	enabled			
Rollouts per prompt n	16			
Max prompt / response length	512 / 4,096 tokens			
vLLM max_model_len	4,608 tokens			
Tensor parallel (vLLM)	1	2	1	2
Buffer length L_{soft} / cap L_{max}	2,048 / 4,096 tokens			
Penalty factor λ	1.0			

Tracking and checkpoints. Weights & Biases plus TensorBoard. Checkpoints in Megatron `torch_dist` format (fully parallel save), converted to HuggingFace `safetensors` for evaluation and deployment.

Training time. The primary experimental pipeline required approximately 8,250 H100 GPU hours. The vast majority of this budget was allocated to the data generation phase ($\sim$8,000 hours), while downstream training stages were significantly less compute-intensive; Supervised Fine-Tuning (SFT) and Reinforcement Learning (RL) consumed approximately 100 and 150 hours, respectively. These figures represent the final end-to-end training of our primary model. Including preliminary ablations, failed runs, and evaluation, the total computational expenditure for the project is estimated at approximately 30,000 H100 GPU hours.

A.8 Per-grader HealthBench results

HealthBench's evaluation criteria are written by physicians; the model graders in our panel determine whether each response satisfies those fixed, conversation-specific criteria. The main paper reports overall scores under all three graders (CompassJudger-2-32B-Instruct, Llama-3.3-70B-Instruct, GPT-OSS-120B) directly in Tab. 2. Three-grader results are also reported for the teacher- and backbone-selection ablations (App. A.2, A.4).

Across all four uses of the panel (final-model comparison, teacher ablation, 4B-class backbone ablation, sub-2B-class backbone ablation), the main conclusions are preserved across the three graders: the selected teacher and backbone remain at or near the top, with only a small GPT-OSS swap between the two Qwen3-4B variants, and the released MEDPSY models keep the same ordering. Absolute-score levels are uniformly highest under CompassJudger-2-32B-Instruct, intermediate under Llama-3.3-70B-Instruct, and lowest under GPT-OSS-120B (the strictest of the three graders on open-ended clinical content).

A.9 HealthBench per-dimension and per-axis breakdowns

Tab. 17 and Tab. 18 report the per-dimension breakdown of HealthBench (Standard and Hard splits) under CompassJudger-2-32B-Instruct, the strongest dedicated model grader on evaluator benchmarks. The seven dimensions are expertise-tailored communication, response depth, context seeking, emergency referrals, global health, health-data tasks, and responding under uncertainty. The Overall column matches the corresponding CompassJudger column of Tab. 2. The 4B model's strongest dimensions are Emergency Referrals (81.67/62.33), Expertise-Tailored Communication (79.33/55.33), and Responding under Uncertainty (76.33/61.00; Standard/Hard).

Table 17: HealthBench (Standard) per dimension, CompassJudger-2-32B-Instruct.

Model	Overall	Expert. Comm.	Resp. Depth	Context Seeking	Emerg. Refer.	Global Health	Health Data	Resp. under Uncert.
MEDPSY-4B	**74.00**	**79.33**	**63.67**	**71.67**	**81.67**	**73.67**	**60.67**	**76.33**
MEDPSY-1.7B	70.33	76.33	56.33	69.33	80.00	68.33	57.00	74.00
MedGemma-27B-text-it	65.00	73.00	61.33	58.67	73.00	61.00	56.67	66.33
Qwen3-4B-Thinking-2507	63.00	71.00	58.00	57.67	74.00	59.00	54.67	64.33
MedGemma-1.5-4B-it	54.00	62.67	48.67	46.00	64.00	47.67	44.67	58.33
Qwen3-1.7B (Thinking)	53.00	63.67	49.67	48.33	64.67	45.67	42.33	56.33

Table 18: HealthBench-Hard per dimension, CompassJudger-2-32B-Instruct.

Model	Overall	Expert. Comm.	Resp. Depth	Context Seeking	Emerg. Refer.	Global Health	Health Data	Resp. under Uncert.
MEDPSY-4B	**58.00**	**55.33**	**47.67**	**63.33**	**62.33**	**60.00**	**46.67**	**61.00**
MEDPSY-1.7B	54.33	52.33	40.33	61.00	60.33	55.00	43.33	58.33
MedGemma-27B-text-it	42.67	44.67	38.67	42.00	39.67	42.67	39.33	42.67
Qwen3-4B-Thinking-2507	42.00	45.00	38.67	43.00	47.33	43.33	39.67	42.00
MedGemma-1.5-4B-it	29.67	31.67	29.00	28.00	29.00	29.00	23.67	35.00
Qwen3-1.7B (Thinking)	28.33	31.67	28.33	32.00	27.67	26.67	21.33	31.00

Per-axis breakdown. Where the per-dimension tables show *where* failures occur, Table 19 shows *how* response quality deteriorates: it reports the five behavior axes defined by the native HealthBench evaluator. Values are percentages on the 1,000-example Hard split, scored with CompassJudger-2-32B-Instruct and averaged over three independent generation runs. Context Awareness and Instruction Following are the two lowest-scoring axes for both MEDPSY model sizes.

Table 19: HealthBench-Hard behavior-axis scores under CompassJudger-2-32B-Instruct. Values are three-run means over the 1,000-example Hard split.

Model	Accuracy	Completeness	Context awareness	Instruction following	Communication quality
MEDPSY-4B	63.5	64.1	50.7	53.4	71.3
MEDPSY-1.7B	59.8	61.0	48.0	45.0	69.2
MedGemma-27B-text-it	57.2	43.9	23.0	54.6	71.4
Qwen3-4B-Thinking-2507	54.8	44.4	25.4	58.5	74.6
MedGemma-1.5-4B-it	45.5	29.6	11.8	46.5	71.7
Qwen3-1.7B (Thinking)	42.5	31.0	9.5	50.1	66.7

A.10 Per-benchmark response length

Table 20 reports the per-benchmark average output length. The first row reports the weighted average; the remaining rows are the per-benchmark values, sorted by Qwen3-4B-Thinking-2507 response length (descending). Reductions are MEDPSY-4B / MEDPSY-1.7B versus the corresponding Qwen3 backbone; values $< 1\times$ indicate that MEDPSY produces *longer* responses than the backbone on that benchmark. The only such case is HealthBench at the 1.7B class ($0.9\times$), reflecting a deliberate trade for the +17 pp HealthBench gain reported in Sec. 5.4.

Table 20: Per-benchmark average response length (tokens; lower is better).

Benchmark	4B class			1.7B class		
	Qwen3-4B-Thinking-2507	MEDPSY-4B	**Reduction**	Qwen3-1.7B (Thinking)	MEDPSY-1.7B	**Reduction**
Weighted avg. (all)	**2,953**	**909**	**3.2×**	**1,901**	**1,110**	**1.7×**
MedXpertQA	4,900	**1,152**	**4.3×**	2,641	**1,459**	**1.8×**
MedQA-USMLE	4,050	**810**	**5.0×**	2,886	**1,128**	**2.6×**
MMLU-Pro Health	3,396	**773**	**4.4×**	2,564	**994**	**2.6×**
AfriMedQA	2,969	**660**	**4.5×**	2,079	**875**	**2.4×**
PubMedQA	2,703	**527**	**5.1×**	998	**788**	**1.3×**
MMLU (Health)	2,534	**581**	**4.4×**	2,020	**749**	**2.7×**
MedMCQA	2,387	**597**	**4.0×**	1,636	**770**	**2.1×**
HealthBench	2,222	**1,405**	**1.6×**	**1,327**	1,534	0.9×

A.11 Evaluation Robustness and Decontamination

This appendix provides two robustness checks for the primary closed-ended results. First, we assess parser reliability by re-parsing the original model generations using three independent LLM parsers (Tab. 21). Second, we address potential train–test leakage by rerunning the evaluation suite after excluding test examples that overlap with our final synthetic training corpus (Tabs. 23 and 24).

Decontamination Methodology and Impact. Although our pipeline utilizes only public training splits, medical datasets are frequently not mutually disjoint. Our audit revealed cross-dataset overlaps: for example, MedQA training items appeared in the MedXpertQA test set, and MedMCQA training items overlapped with AfriMedQA test items. To ensure the integrity of our results, we performed a string-matching decontamination and excluded all overlapping test examples from the final scoring.

The decontaminated results utilize the same checkpoints as the original evaluation; only the overlapping benchmark examples are removed. As shown in the following tables, the percentage of excluded examples is minimal. The impact on performance across all models was found to be negligible, and the relative rankings between our models and the baselines remain unchanged. Full per-benchmark exclusion counts and the resulting decontaminated scores are detailed in Tables 23 and 24.

Table 21: Closed-ended parser robustness. Main-paper closed-ended tables use QwQ as the parser; this ablation re-scores the same generations with Llama-3.3-70B-Instruct, GPT-OSS-120B, and QwQ. Average is over the full seven-benchmark closed-ended suite.

Model	Parser	Avg	AfriMedQA	MMLU-H	MedQA	PubMedQA	MedMCQA	MedXpertQA	MMLU-Pro-H
MedGemma-27B-text-it	Llama	70.33	73.34	90.80	84.29	72.20	72.92	25.31	73.45
MedGemma-27B-text-it	GPT-OSS	70.22	73.31	90.64	84.11	72.20	72.84	25.31	73.13
MedGemma-27B-text-it	QwQ	69.95	73.07	90.48	83.29	71.93	72.77	25.18	72.94
MedGemma-27B-text-it	MAX–MIN	0.38	0.27	0.32	1.00	0.27	0.15	0.13	0.51
Qwen3-4B-Thinking-2507	Llama	63.49	64.32	86.18	71.95	74.73	62.06	17.07	68.12
Qwen3-4B-Thinking-2507	GPT-OSS	63.27	64.26	86.10	71.20	74.73	61.92	16.72	67.96
Qwen3-4B-Thinking-2507	QwQ	63.10	64.12	85.92	70.91	74.53	61.78	16.69	67.73
Qwen3-4B-Thinking-2507	MAX–MIN	0.39	0.20	0.26	1.04	0.20	0.28	0.38	0.39
MedGemma-1.5-4B-it	Llama	51.65	54.55	67.92	65.23	59.47	50.59	15.86	47.93
MedGemma-1.5-4B-it	GPT-OSS	51.20	53.95	67.65	64.99	58.73	49.95	15.84	47.29
MedGemma-1.5-4B-it	QwQ	51.20	54.38	67.69	64.39	58.73	50.08	15.80	47.31
MedGemma-1.5-4B-it	MAX–MIN	0.45	0.60	0.27	0.84	0.74	0.64	0.06	0.64
Qwen3-1.7B (Thinking)	Llama	50.09	51.99	72.60	47.47	72.27	49.25	11.65	45.40
Qwen3-1.7B (Thinking)	GPT-OSS	50.07	51.93	72.62	47.47	72.47	49.20	11.63	45.17
Qwen3-1.7B (Thinking)	QwQ	49.95	51.87	72.49	47.18	72.33	49.14	11.60	45.07
Qwen3-1.7B (Thinking)	MAX–MIN	0.13	0.12	0.13	0.29	0.20	0.11	0.05	0.33
MEDPSY-4B	Llama	70.89	71.71	89.78	85.47	75.10	72.55	30.84	70.78
MEDPSY-4B	GPT-OSS	70.85	71.70	90.07	85.15	75.07	72.37	30.74	70.85
MEDPSY-4B	QwQ	70.54	71.50	89.70	84.39	75.00	72.15	30.61	70.45
MEDPSY-4B	MAX–MIN	0.35	0.21	0.37	1.08	0.10	0.40	0.23	0.40
MEDPSY-1.7B	Llama	62.94	65.12	83.04	75.44	69.73	63.93	21.37	61.95
MEDPSY-1.7B	GPT-OSS	62.89	65.02	83.04	75.44	69.73	63.84	21.39	61.77
MEDPSY-1.7B	QwQ	62.62	64.84	82.72	75.05	69.53	63.56	21.28	61.37
MEDPSY-1.7B	MAX–MIN	0.32	0.28	0.32	0.39	0.20	0.37	0.11	0.58

Matched-instance pairwise comparison. Following the original Medmarks v0.1.0 implementation [35], we assign one point to the model with the higher reward on each matched example and half a point to each model for a tie, average these scores within each dataset, and combine them using log-dataset-size weighting. The analysis covers AfriMedQA, MMLU-Health, MMLU-Pro Health, MedMCQA, MedQA-USMLE, MedXpertQA, and PubMedQA, with the six MMLU-Health subsets pooled before weighting. The three-run result in Table 22 confirms matching closed-ended performance.

Table 22: Matched-instance pairwise scores by benchmark, averaged over three runs. W/T/L counts are summed across runs and shown from the MEDPSY-4B perspective.

Benchmark	MEDPSY-4B outcomes			Pairwise score	
	Win	Tie	Loss	MEDPSY-4B	MedGemma-27B-text-it
AfriMedQA	640	10,258	832	49.1816%	**50.8184%**
MMLU-Health	147	2,944	176	49.5562%	**50.4438%**
MedQA-USMLE	325	3,209	285	**50.5237%**	49.4763%
PubMedQA	123	1,298	79	**51.4667%**	48.5333%
MedMCQA	1,155	10,164	1,230	49.7012%	**50.2988%**
MedXpertQA	1,136	5,476	738	**52.7075%**	47.2925%
MMLU-Pro Health	179	2,032	243	48.6960%	**51.3040%**
Overall (log-weighted)	–	–	–	**50.2501%**	49.7499%

Table 23: Closed-ended decontamination results. "Decontam" evaluates the same checkpoint after excluding benchmark examples matched against the final synthetic training corpus.

Model	Eval set	Avg	AfriMedQA	MMLU-H	MedQA	PubMedQA	MedMCQA	MedXpertQA	MMLU-Pro-H
MEDPSY-4B	Original	70.54	71.50	89.70	84.39	75.00	72.15	30.61	70.45
MEDPSY-4B	Decontam	70.36	71.44	88.59	84.36	75.00	72.17	30.52	70.42
MedGemma-27B-text-it	Original	69.95	73.07	90.48	83.29	71.93	72.77	25.18	72.94
MedGemma-27B-text-it	Decontam	69.80	73.01	89.70	83.25	71.93	72.79	24.98	72.95
Qwen3-4B-Thinking-2507	Original	63.10	64.12	85.92	70.91	74.53	61.78	16.69	67.73
Qwen3-4B-Thinking-2507	Decontam	62.84	64.05	84.31	70.84	74.53	61.80	16.66	67.69
MEDPSY-1.7B	Original	62.62	64.84	82.72	75.05	69.53	63.56	21.28	61.37
MEDPSY-1.7B	Decontam	62.39	64.75	81.45	74.99	69.53	63.56	21.11	61.32
MedGemma-1.5-4B-it	Original	51.20	54.38	67.69	64.39	58.73	50.08	15.80	47.31
MedGemma-1.5-4B-it	Decontam	50.74	54.26	64.78	64.38	58.73	50.06	15.70	47.25
Qwen3-1.7B (Thinking)	Original	49.95	51.87	72.49	47.18	72.33	49.14	11.60	45.07
Qwen3-1.7B (Thinking)	Decontam	49.58	51.80	70.12	47.06	72.33	49.16	11.59	45.00

Table 24: Decontamination deltas and overlap counts. Deltas are decontaminated score minus original score.

Model / count	Avg	AfriMedQA	MMLU-H	MedQA	PubMedQA	MedMCQA	MedXpertQA	MMLU-Pro-H
MEDPSY-4B	−0.18	−0.06	−1.11	−0.03	0.00	0.02	−0.09	−0.03
MedGemma-27B-text-it	−0.15	−0.06	−0.78	−0.04	0.00	0.02	−0.20	0.01
Qwen3-4B-Thinking-2507	−0.26	−0.07	−1.61	−0.07	0.00	0.02	−0.03	−0.04
MEDPSY-1.7B	−0.23	−0.09	−1.27	−0.06	0.00	0.00	−0.17	−0.05
MedGemma-1.5-4B-it	−0.46	−0.12	−2.91	−0.01	0.00	−0.02	−0.10	−0.06
Qwen3-1.7B (Thinking)	−0.38	−0.07	−2.37	−0.12	0.00	0.02	−0.01	−0.07
Samples removed	157	10	125	3	0	1	17	1
Total count	14,223	3,910	1,089	1,273	500	4,183	2,450	818
Contamination %	1.10%	0.26%	11.48%	0.24%	0.00%	0.02%	0.69%	0.12%

A.12 Quantization: imatrix ablation and per-benchmark tables

This appendix supplements the six-format quantization study of Sec. 5.6, whose per-variant overview is main-paper Table 4, with the `Q4_K_M` imatrix ablation, the capacity-vs-precision comparison, and per-benchmark closed-ended scores. The comparison uses `Q8_0` without imatrix and all sub-8-bit variants (`Q4_K_M`, `IQ4_NL`, `IQ4_XS`, `IQ3_M`, `IQ3_XXS`) with imatrix calibration.

On the 1.7B model, imatrix calibration provides its largest closed-ended benefits on the most reasoning-intensive tasks: 4.69 points on MedQA-USMLE, 3.59 on MMLU-Pro Health, and 2.07 on MMLU (Health), the same benchmarks that benefited most from post-training (Sec. 5.4).

Table 25: Per-format AVG Score changes by model size (absolute change vs. BF16, on the 0–100 scale).

Format	Δ Score 4B	Δ Score 1.7B	1.7B / 4B ratio
`Q8_0`	−0.15	0.00	—
`Q4_K_M`	−0.81	−0.73	≈1×
`IQ4_NL`	−1.02	−1.70	1.7×
`IQ4_XS`	−1.08	−1.79	1.7×
`IQ3_M`	−1.50	−3.58	2.4×
`IQ3_XXS`	−5.56	−12.46	2.2×

Table 26: Closed-ended per-benchmark results, quantized MEDPSY-4B variants. Δ vs. BF16. "(+imatrix)" / "(no imatrix)" denote with / without imatrix calibration.

Variant	Avg	Δ	AfriMedQA	MMLU-H	MedQA	PubMedQA	MedMCQA	MedXpertQA	MMLU-Pro-H
4B (BF16)	**70.54**	n/a	71.50	89.70	84.39	75.00	72.15	30.61	70.45
4B-Q8_0 (+imatrix)	70.33	−0.21	71.15	89.19	84.57	74.73	71.82	30.05	70.82
4B-Q8_0 (no imatrix)	70.25	−0.29	71.42	89.70	84.31	73.60	71.95	30.01	70.78
4B-Q4_K_M (+imatrix)	69.92	−0.62	71.62	89.03	83.13	76.07	71.02	28.90	69.64
4B-Q4_K_M (no imatrix)	69.60	−0.94	71.15	88.73	83.09	74.93	70.57	28.79	69.92

Table 27: HealthBench per-dimension results, quantized MEDPSY-4B variants (CompassJudger).

Variant	Overall	Expert. Comm.	Resp. Depth	Context Seek.	Emerg. Refer.	Global Health	Health Data	Resp. Uncert.
4B (BF16)	**74.00**	79.00	63.00	72.00	82.00	74.00	60.00	76.00
4B-Q8_0 (+imatrix)	74.00	79.00	63.00	72.00	82.00	73.00	62.00	76.00
4B-Q8_0 (no imatrix)	74.00	79.00	62.00	72.00	82.00	74.00	60.00	76.00
4B-Q4_K_M (+imatrix)	73.00	79.00	62.00	72.00	81.00	72.00	60.00	76.00
4B-Q4_K_M (no imatrix)	73.00	79.00	62.00	71.00	81.00	72.00	59.00	76.00

Table 28: Closed-ended per-benchmark results, quantized MEDPSY-1.7B variants. Δ vs. BF16.

Variant	Avg	Δ	AfriMedQA	MMLU-H	MedQA	PubMedQA	MedMCQA	MedXpertQA	MMLU-Pro-H
1.7B (BF16)	**62.62**	n/a	64.84	82.72	75.05	69.53	63.56	21.28	61.37
1.7B-Q8_0 (no imatrix)	62.62	0.00	64.89	82.52	74.84	70.47	63.05	21.01	61.53
1.7B-Q8_0 (+imatrix)	62.44	−0.18	64.54	82.41	74.50	71.33	62.83	21.08	60.39
1.7B-Q4_K_M (+imatrix)	62.16	−0.46	64.40	82.15	74.00	71.20	62.72	20.16	60.52
1.7B-Q4_K_M (no imatrix)	60.58	−2.04	63.24	80.65	70.36	70.67	61.22	20.12	57.78

Table 29: HealthBench per-dimension results, quantized MEDPSY-1.7B variants (CompassJudger).

Variant	Overall	Expert. Comm.	Resp. Depth	Context Seek.	Emerg. Refer.	Global Health	Health Data	Resp. Uncert.
1.7B (BF16)	**70.00**	77.00	56.00	69.00	80.00	68.00	58.00	74.00
1.7B-Q8_0 (no imatrix)	70.00	77.00	57.00	69.00	80.00	69.00	57.00	73.00
1.7B-Q8_0 (+imatrix)	70.00	77.00	57.00	70.00	79.00	69.00	56.00	74.00
1.7B-Q4_K_M (+imatrix)	69.00	75.00	56.00	69.00	79.00	67.00	55.00	73.00
1.7B-Q4_K_M (no imatrix)	68.00	74.00	54.00	67.00	78.00	66.00	55.00	72.00

The per-variant overall scores for the I-quant variants (`IQ4_NL`, `IQ4_XS`, `IQ3_M`, `IQ3_XXS`) are reported in main-paper Table 4; Table 25 compares score changes by model size. The imatrix ablation details are split by model size and evaluation type: closed-ended results for 4B and 1.7B are in Tables 26 and 28, and HealthBench per-dimension results are in Tables 27 and 29.

A.13 Real-device measurement protocol

We benchmarked local llama.cpp inference on an Apple M5 Pro using Metal and on a OnePlus 13 (CPH2649IN, Android 16, Adreno 830) using Android build 9226 with Vulkan partial offload (`-ngl 20`) and six CPU threads. The Apple prompt set contains 20 MedQA and 10 MedXpertQA-Text questions. The phone measurements use three MedQA prompts per model.

Real-device measurements. We ran the quantized models locally with llama.cpp on an Apple M5 Pro and a OnePlus 13. Table 30 reports the Apple M5 Pro measurements over the 30-prompt set: matched `Q4_K_M` runs decode at comparable speed (83.9 vs. 82.2 tokens/s), but MedPsy-4B emits 738 rather than 2,727 tokens per answer, reducing median end-to-end time from 34.6 to 8.75 seconds (74.7%). Table 31 reports the OnePlus 13 measurements, where both model sizes run fully locally within 4 GB of measured peak RAM. Estimated energy per answer shows the same pattern: at matched `Q4_K_M`, MEDPSY-4B uses 232 J against the backbone's 917 J (74.7% lower). The accuracy of these quantized variants on the full benchmark suite is reported in main-paper Table 4.

Table 30: Apple M5 Pro local-inference measurements over the 30-prompt set (20 MedQA and 10 MedXpertQA-Text). Time per answer is the median; energy per answer is estimated as measured sustained SoC power (powermetrics, 26.5 W) times median end-to-end wall time.

Model/quant	Peak RAM	tok/s	tokens/answer	time/answer	energy/answer
MedPsy-1.7B/`Q4_K_M`	2.60 GB	171.0	994	5.9 s	156 J
MedPsy-4B/`IQ3_M`	3.66 GB	95.0	768	8.3 s	220 J
MedPsy-4B/`Q4_K_M`	4.29 GB	83.9	738	8.75 s	232 J
Qwen3-4B-Thinking-2507/`Q4_K_M`	4.28 GB	82.2	2,727	34.6 s	917 J

Table 31: OnePlus 13 local-inference measurements (three MedQA prompts per model). Both models use `Q4_K_M` GGUF builds with Vulkan partial offload (`-ngl 20`) and six CPU threads. Energy is from the Android battery charge counter (unplugged device at 49% charge, Wi-Fi debugging).

Model	Peak RAM	tok/s	tokens/answer	TTFT	time/answer	energy/answer
MedPsy-1.7B	2.22 GB	15.27	789	8.90 s	55.6 s	420 J
MedPsy-4B	3.98 GB	6.63	422	15.04 s	105.4 s	508 J

A.14 Accuracy under explicit thinking budgets

We reran the seven closed-ended tasks with external llama.cpp servers while capping tokens in the explicit thinking channel at 300, 600, 900, or 1,500; when the cap was reached, the runtime injected the instruction "Thinking budget exceeded. Please stop thinking and answer now." Each condition contains three sampled generations for all 12,549 examples, evaluated with the QwQ parser. Generation uses thinking mode with temperature 0.6, top-k 20, top-p 0.95, an 8,192-token output limit, and a 16,384-token sequence limit. Table 32 and Figure 2 reports the resulting average accuracy, the fraction of traces that reached the cap, and the mean length of the final answer channel.

Table 32: Seven-task closed-ended accuracy under explicit thinking-token caps. Parentheses show the percentage of traces reaching the cap and mean final-answer-channel tokens.

Model / budget	300	600	900	1,500	Unbounded
MEDPSY-4B	68.07 (68.0%, 152)	69.51 (25.1%, 156)	69.96 (9.6%, 159)	70.29 (1.3%, 163)	70.48 (0.0%, 162)
Qwen3-4B-Thinking-2507	59.45 (99.0%, 677)	60.07 (86.2%, 614)	61.04 (73.3%, 578)	62.04 (53.6%, 532)	63.15 (0.0%, 451)

MEDPSY-4B retains 96.6%, 98.6%, and 99.3% of its unbounded score at 300, 600, and 900 thinking tokens; Qwen3-4B-Thinking-2507 reaches 98.2% at 1,500. At 300 tokens, MEDPSY-4B scores 8.62 points above the backbone and 4.92 points above the backbone's unbounded score. These results show stronger accuracy under the tested explicit-reasoning budgets. Qwen3-4B-Thinking-2507 may move some computation into the answer channel under tighter caps, as suggested by its longer final answers.

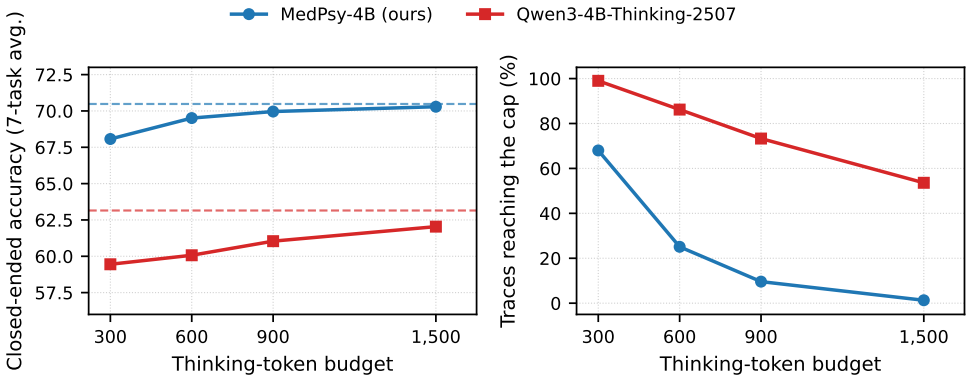


Figure 2: Accuracy and cap-hit rate under explicit thinking-token caps. Left: seven-task closed-ended accuracy at thinking budgets of 300–1,500 tokens; dashed lines mark unbounded generation. **Right:** fraction of traces reaching the cap. MEDPSY-4B degrades gracefully and rarely reaches the cap beyond a 600-token budget, whereas Qwen3-4B-Thinking-2507 hits the cap on 99% of traces at 300 tokens and is still cap-bound on over half of traces at 1,500.

A.15 Context-length scaling

On the laptop (Apple M5 Pro, Metal backend), decode throughput at 8K input retains 66.5–72.5% of its 512-token rate across the three tested variants, while peak RAM at 8K remains between 3.16 and 4.83 GB (Table 33); TTFT at 512 tokens is 0.06–0.12 s. These values come from a dedicated context-scaling probe and differ slightly from the smoke-set figures of Table 30; the 4.83 GB peak of MEDPSY-4B `Q4_K_M` at 8K context is the largest peak RAM we observed on the laptop.

Table 33: Apple M5 Pro context-length sweep (llama.cpp, Metal): decode throughput at 512-token and 8K-token inputs, peak RAM at 8K, and TTFT at 512 tokens.

Model/quant	Decode @512	Decode @8K	Peak RAM @8K	TTFT @512
MEDPSY-1.7B `Q4_K_M`	164 tok/s	109 tok/s	3.16 GB	0.06 s
MEDPSY-4B `IQ3_M`	91 tok/s	66 tok/s	4.32 GB	0.12 s
MEDPSY-4B `Q4_K_M`	89 tok/s	63 tok/s	4.83 GB	0.12 s

These measurements focus on edge context scaling. Table 34 reports a OnePlus 13 CPU run with MEDPSY-1.7B `Q4_K_M`. At 551 and 4,134 input tokens, TTFT is 12.88 and 93.85 seconds, respectively, while decode throughput is 21.07 and 13.88 tokens/s. These measurements provide practical bounds for selecting context budgets on a phone while process RAM remains stable. This sweep uses the CPU build with six threads; Table 31 instead reports generation with partial Vulkan offload (`-ngl 20`).

Table 34: OnePlus 13 CPU context-length scaling with MEDPSY-1.7B `Q4_K_M`: timing and throughput across four input lengths.

Context target	Input tokens	TTFT (s)	Prefill (tok/s)	Decode (tok/s)	Total time (s)
512	551	12.88	42.79	21.07	14.11
1K	1,062	21.23	50.04	17.82	21.95
2K	2,088	51.79	40.32	14.98	53.19
4K	4,134	93.85	44.05	13.88	95.01

A.16 Long-context accuracy on LongHealth

LongHealth [36] evaluates question answering over long clinical documents (5,090–6,754 words), probing the input-length regime in which retrieval-augmented deployment operates. Table 35 reports accuracy on Tasks 1 and 2. Both MEDPSY sizes outperform the MedGemma model of the next size class up: MEDPSY-4B exceeds MedGemma-27B-text-it by 2.7 and 7.3 points on the two tasks, and

MEDPSY-1.7B exceeds MedGemma-1.5-4B-it by 2.5 and 30.6 points, indicating that the accuracy reported in the main evaluation carries over to long-context inputs.

Table 35: LongHealth accuracy on long clinical documents (5,090–6,754 words). Best per size class in bold.

Model	Task 1	Task 2
MEDPSY-4B	**0.855**	**0.875**
MedGemma-27B-text-it	0.828	0.802
MEDPSY-1.7B	**0.590**	**0.670**
MedGemma-1.5-4B-it	0.565	0.364

A.17 RL ablation studies

We ran a controlled set of ablations to isolate which design choices in the MEDPSY RL recipe drive the gains reported in Tables 1–2. Each ablation modifies a single component of the released MEDPSY pipeline and trains for the same number of optimisation steps; all other hyperparameters are held constant. The ablations come in two natural groups: (i) *effectiveness* ablations that probe which components contribute to accuracy, and (ii) *token-efficiency* ablations that decompose the source of the response-length compression.

Evaluation suite used in this appendix. For every ablation we score checkpoints on the same three closed-ended benchmarks tracked online during training (MedQA-USMLE, MedXpertQA, PubMedQA) using the lightweight flexible scorer in the `inspect-ai` evaluation framework [38]. This is a strict subset of the full seven-benchmark suite scored under the LLM-as-Parser methodology of Sec. 5.1 and Table 20; the smoke suite was used to keep ablation compute and time within budget. Per-checkpoint absolute scores in this appendix therefore differ slightly from the LLM-as-Parser numbers in the main paper. For example, on MedQA-USMLE the released MEDPSY-4B S2 checkpoint scores 84.84% under `inspect-ai` vs. 84.39% under LLM-as-Parser, well within scorer noise. Output-token counts are also broadly consistent across the two scorers but can differ in absolute value due to different prompting conventions and post-generation truncation in the two frameworks; the per-ablation *deltas* reproduce on both. We use the smoke suite throughout this appendix and direct readers wanting released-checkpoint absolute numbers to the main-paper tables.

Ablation set. All ablations train from the same SFT-Stage-2 checkpoints used by the released MEDPSY models, except *RL-from-base*, which uses no SFT at all. We use short, descriptive names throughout this appendix:

- *Random S2.* Same Stage-2 setup as MEDPSY but the training data is a uniformly-random sample from AlphaMed19K [22], sized to match the mined 1:2 right:wrong subset (2,439 items at 4B, 4,317 at 1.7B). Tests whether the mining procedure itself contributes vs. training longer on more data.
- *Binary reward.* Replace the four-tier reasoning-aware reward (Eq. 1) with the binary $\{0, 1\}$ correctness reward used by AlphaMed [22]. Isolates the contribution of the gated reward.
- *No group filter.* Disable DAPO's dynamic filtering of zero-variance rollout groups (`filter_groups.enable=false`). Isolates the contribution of group filtering.
- *No overlong.* Remove the soft penalty applied to responses exceeding L_{soft}=2,048 tokens within the L_{max}=4,096-token budget; all other DAPO settings unchanged. Isolates the contribution of the soft length penalty.
- *Vanilla GRPO.* Replace DAPO with vanilla GRPO [39]: re-introduce the KL anchor (`kl_coef`$=0.001$), use symmetric PPO clipping (ϵ_{low}=ϵ_{high}=0.2), and disable the overlong penalty. Isolates the contribution of DAPO's full set of drift-enabling modifications.
- *RL-from-base.* RL Stage 1 initialised from the un-fine-tuned Qwen3 backbones (Qwen3-4B-Thinking-2507 for the 4B class, Qwen3-1.7B for the 1.7B class), with no SFT at all. Tests whether SFT is necessary before RL.
- *Difficult-only S1.* RL Stage 1 trained only on items the size-matched SFT-2 checkpoint fails on every attempt during pass-at-N annotation (n_c=0); 1,276 items at 4B and 1,888 items at 1.7B. Tests whether training on easy/moderate items is necessary to install short reasoning.

A.17.1 Effectiveness ablations

Mining drives hard-case generalisation. The clearest effectiveness finding is that the deterministic 1:2 right:wrong mining of Stage 2 (§4.2) accounts for the majority of the Stage-2 lift, and it does so almost entirely through gains on the hardest of our benchmarks (MedXpertQA). Holding the Stage-1 checkpoint and all hyperparameters fixed (Table 36), the mined Stage-2 set lifts MedXpertQA by $+6.1$ pp on the 4B model and $+4.3$ pp on the 1.7B model, whereas the *Random S2* ablation lifts it by only $+1.1$ pp and $+2.3$ pp respectively. Easy and medium benchmarks (MedQA, PubMedQA) are essentially indistinguishable between mined and random Stage-2 (within the per-benchmark stderr of ~ 1.0 pp). The mined Stage-2 also reaches its peak in fewer steps: for the 4B model the mined run peaks at step 190, while the random run requires the full 500 steps to reach a lower plateau.

Table 36: Mined vs. random Stage-2. Both runs initialise from the same Stage-1 best checkpoint and train for the same number of steps. Mining's benefit is concentrated on MedXpertQA at both scales.

	Best step	MedQA	MedXpertQA	PubMedQA	Avg	Δ vs. S1
4B class (S1 best avg = 62.34)						
S2 mined (MEDPSY)	190	84.76	**32.37**	76.40	**64.51**	+2.17
S2 random (*Random S2*)	500	85.31	27.39	77.00	63.23	+0.89
1.7B class (S1 best avg = 52.35)						
S2 mined (MEDPSY)	400	72.82	**20.94**	76.60	**56.79**	+4.44
S2 random (*Random S2*)	230	74.71	18.90	73.40	55.67	+3.32

This pattern matches the design intent: the 1:2 right:wrong subset over-samples failure cases that, by construction, require capabilities beyond surface pattern matching. MedXpertQA requires advanced reasoning patterns to pass; MedQA and PubMedQA are dominated by simpler factual recall and structured-text comprehension, which both Stage-1 RL and even random Stage-2 already saturate.

Reward shape and DAPO group filter are within smoke-eval noise. The *Binary reward* ablation, which replaces the four-tier reasoning-aware reward with AlphaMed's binary $\{0, 1\}$ correctness reward, leaves average accuracy within ± 1.0 pp of the MEDPSY Stage-1 best at both model sizes. The *No group filter* ablation, which disables DAPO's dynamic group filtering, is similarly within ± 1.0 pp (Table 38). We do not claim an accuracy contribution from either knob based on these results. The four-tier reward does, however, produce a small but measurable token-efficiency benefit relative to binary, which we attribute to the explicit penalisation of the "correct without thinking" shortcut.[5]

RL needs SFT. The *RL-from-base* ablation initialises RL Stage 1 from the un-fine-tuned Qwen3 backbones (Qwen3-4B-Thinking-2507 and Qwen3-1.7B), bypassing both SFT stages. Performance plateaus far below the MEDPSY Stage 1 at both model sizes (Table 37): the 4B *RL-from-base* run reaches a smoke-suite average of 55.89% vs. 62.34% for the MEDPSY Stage 1 (-6.45 pp), and the 1.7B *RL-from-base* run reaches 45.07% vs. 52.35% (-7.28 pp). Two observations clarify the gap. First, the absolute RL lift over the run's starting point is comparable to the MEDPSY recipe: the 4B *RL-from-base* run improves from 54.20% at step 0 to 55.89% at its best step ($+1.69$ pp), while MEDPSY Stage 1 improves from 60.69% to 62.34% ($+1.65$ pp); both gain ~ 1.7 pp from RL. What differs is the floor, which sits 6–7 pp lower without SFT. Second, the training reward signal in the *RL-from-base* runs is dominated by missing format compliance: at the 4B size the mean reward converges to ~ 0.43 vs. ~ 0.96 for MEDPSY, with the gap explained by the model failing to reliably emit the structured `<think>...</think>/<answer>...</answer>` format that the SFT pipeline installs (Eq. 1 returns 0.5 instead of 1.0 even when the answer is correct). We therefore conclude that SFT establishes the operating floor – broad medical coverage and the reasoning-trace format – on top of which RL adds its ~ 1.7 pp of consolidation; RL on the un-fine-tuned backbone is not unable to learn, but cannot recover the SFT-installed capabilities in the same number of optimisation steps. Per-benchmark numbers in Table 37 are consistent with the corresponding backbone numbers reported under LLM-as-Parser in Sec. 5.4 (e.g. MedQA-USMLE on the 4B Qwen3 backbone is 71.56% under LLM-as-Parser vs. 72.19% for *RL-from-base* baseline under `inspect-ai`; MedQA-USMLE on the 1.7B Qwen3 backbone is 47.18% under LLM-as-Parser vs. 47.13% under `inspect-ai`).

[5] For the 4B model, the average MedQA output tokens at the best checkpoint are 759 for MEDPSY Stage 1 (4-tier reward) vs. 836 for the *Binary reward* ablation, about 9% lower for the four-tier reward at matched accuracy. For the 1.7B model the gap is within noise (943 vs. 944 tokens).

Table 37: *RL-from-base* ablation: RL Stage 1 from the un-fine-tuned Qwen3 backbones. Best-step on the smoke-eval suite. *RL-from-base* converges 6–7 pp below the MEDPSY Stage 1 at both model sizes because it lacks the SFT-installed medical coverage and reasoning-trace format, even though the absolute RL lift over its starting point is comparable to MEDPSY.

	Best step	MedQA	MedXpertQA	PubMedQA	Avg	Δ vs. MEDPSY S1
4B class						
MEDPSY S1 (from SFT-2)	112	85.70	26.33	75.00	**62.34**	—
RL-from-base (no SFT, baseline)	0	72.19	18.20	72.20	54.20	−8.14
RL-from-base (no SFT, best)	120	74.47	18.41	74.80	55.89	−6.45
1.7B class						
MEDPSY S1 (from SFT-2)	96	69.29	16.57	71.20	**52.35**	—
RL-from-base (no SFT, baseline)	0	47.13	11.76	73.00	43.96	−8.39
RL-from-base (no SFT, best)	68	49.18	12.24	73.80	45.07	−7.28

Table 38: *Binary reward* and *No group filter* ablations on RL Stage 1. All numbers are best-step on the smoke-eval suite. Both ablations are within smoke-eval noise of the MEDPSY Stage-1 best at both model sizes; we report them as *neutral* accuracy findings.

	Best step	MedQA	MedXpertQA	PubMedQA	Avg	Δ vs. MEDPSY S1
*4B class (*MEDPSY *S1 best = 62.34, MedXpertQA = 26.33)*						
MEDPSY S1 (4-tier + filter + overlong)	112	85.70	26.33	75.00	62.34	—
Binary reward	84	84.92	25.55	74.60	61.69	−0.65
No group filter	124	85.07	26.33	75.60	62.33	±0.00
*1.7B class (*MEDPSY *S1 best = 52.35, MedXpertQA = 16.57)*						
MEDPSY S1 (4-tier + filter + overlong)	96	69.29	16.57	71.20	52.35	—
Binary reward	100	69.13	17.18	71.80	52.70	+0.35
No group filter	40	66.38	16.65	74.60	52.54	+0.19

A.17.2 Token-efficiency ablations: a clean three-way decomposition

The 1.7–3.2× token-efficiency improvements of the MEDPSY models reported in Sec. 5.5 are the most distinctive empirical property of our pipeline. The *Vanilla GRPO* and *No overlong* ablations, in combination with the MEDPSY runs, decompose this compression into three additive sources, each separately controllable from the others. Table 39 reports per-benchmark average output tokens at the best checkpoint of each variant; Table 40 reports the corresponding training-time response-length trajectory.

Three additive sources. Reading Table 39 top-to-bottom within each model class:

- **(1) RL with a KL anchor (*Vanilla GRPO*).** Vanilla GRPO compresses MedQA tokens by only 6% on the 4B model and 20% on the 1.7B model relative to the SFT-Stage-2 baseline. The KL anchor pulls the policy back toward the verbose SFT prior, and symmetric clipping limits how far the upper PPO bound can boost short-but-correct trajectories. The accuracy delta of *Vanilla GRPO* is also small (−0.5 pp on the 4B model, −0.6 pp on the 1.7B model vs. the MEDPSY Stage 1).
- **(2) DAPO drift modifications (*No overlong*).** Switching to DAPO, removing the KL term and using clip-higher= 0.28, without the overlong penalty pushes compression to 44% (4B) and 45% (1.7B). Allowing the policy to drift further from the SFT prior and biasing the upper PPO bound is responsible for the bulk of the compression, even though neither change references response length explicitly. This ablation also delivers the highest accuracy of any 4B Stage-1 variant (+1.0 pp vs. MEDPSY).
- **(3) The soft overlong-response penalty (MEDPSY).** Adding the linear penalty over $[L_{\text{soft}}, L_{\text{max}}]$ on top of DAPO drives compression the rest of the way to 66% (4B) and 65% (1.7B). The overlong penalty contributes its ∼ 22 pp of additional compression with a measured accuracy cost of 1.0 pp on the 4B model and 0.8 pp on the 1.7B model relative to *No overlong*; it is also the only component that prevents the right tail of the response-length distribution from re-expanding (Table 40, "Tail (end)" reports the fraction of rollouts at the 4,096-token cap). Removing the penalty leaves the mean below L_{soft} but allows individual rollouts to grow, increasing average output length at evaluation by approximately 1.6×.

Table 39: Token-efficiency decomposition (output tokens at evaluation). Per-benchmark average output tokens at the best Stage-1 checkpoint of each variant, plus the final MEDPSY Stage 2 row for reference. The three drift-side mechanisms (Vanilla GRPO → DAPO drift mods → overlong penalty) compose additively to the released-model token budget.

	MedQA	MedXpertQA	PubMedQA	vs. SFT-2
4B class				
SFT-2 baseline (no RL)	2,232	3,510	728	—
+ *Vanilla GRPO*	2,096	3,199	695	−6% / −9% / −5%
+ DAPO drift mods (*No overlong*)	1,259	1,986	555	−44% / −43% / −24%
+ overlong penalty (MEDPSY Stage 1)	**759**	**1,183**	**374**	−66% / −66% / −49%
+ Stage-2 mining (MEDPSY-4B S2)	653	1,066	356	−71% / −70% / −51%
1.7B class				
SFT-2 baseline (no RL)	2,674	3,642	938	—
+ *Vanilla GRPO*	2,142	2,792	805	−20% / −23% / −14%
+ DAPO drift mods (*No overlong*)	1,475	2,117	696	−45% / −42% / −26%
+ overlong penalty (MEDPSY Stage 1)	**943**	**1,204**	**464**	−65% / −67% / −51%
+ Stage-2 mining (MEDPSY-1.7B S2)	1,024	1,244	539	−62% / −66% / −43%

Table 40: Training-time response length and tail behaviour by Stage-1 variant. "Mean (start)" and "Mean (end)" are per-rollout response lengths averaged over the first/last five logged steps; "Tail (end)" is the fraction of rollouts that hit the $4{,}096$-token cap in the final five steps. Removing the overlong penalty keeps the tail wide and prevents the mean from compressing further.

	4B class			1.7B class		
Stage-1 variant	**Mean (start)**	**Mean (end)**	**Tail (end)**	**Mean (start)**	**Mean (end)**	**Tail (end)**
MEDPSY (DAPO + overlong)	1,128	633	0.00%	1,270	787	0.06%
DAPO, *No overlong*	1,482	1,154	0.16%	1,556	1,220	0.74%
Vanilla GRPO (KL + symmetric clip)	1,762	1,699	3.60%	1,901	1,760	3.99%

The accuracy/efficiency trade-off is modest. A common concern with length-controlled RL is that compression comes at the cost of accuracy on hard items. In our setting, the measured cost is at most approximately one point on the smoke suite. The *No overlong* ablation, which keeps DAPO but disables the soft length penalty, scores 63.34% avg on the 4B model (vs. 62.34% for the MEDPSY Stage 1, $+1.00$ pp) at the cost of $\sim 1.6\times$ longer outputs, and 53.15% on the 1.7B model (vs. 52.35%, $+0.80$ pp), again at $\sim 1.6\times$ length. The overlong penalty therefore buys the headline 1.7–$3.2\times$ token reduction reported in Sec. 5.5 for a worst-case accuracy cost of ~ 1 pp on the smoke suite. For an on-device deployment this trade-off is decisively in favour of the penalty; for a deployment that prioritises accuracy over latency, dropping the penalty is a defensible alternative.

Why mean tokens shrink even when the penalty is rarely triggered. A natural objection to the overlong-penalty story is that the policy's mean response length is well below the L_{soft}=2,048-token threshold for almost the entire run, so the penalty should "not matter". Table 40 resolves this: the penalty does not act on the mean, but on the per-rollout right tail. In the MEDPSY runs, the fraction of rollouts that hit the L_{max}=4,096-token cap drops from 2–3% in the first decile of training to $\sim$ 0–0.1% at the end. Removing the penalty (the *No overlong* ablation) keeps that tail wide (0.16–0.74% of rollouts at the cap), and the wider tail in turn pulls the mean upward via DAPO's drift dynamics. The penalty is best understood as an *anchor on the right tail of the response-length distribution*, not as a per-step length subtraction.

The full curriculum Pareto-strictly-dominates the single-stage Difficult-only shortcut. The cleanest counterfactual to the two-stage curriculum is to skip the easy/moderate Stage 1 entirely and run a single RL stage on hard data. The *Difficult-only S1* ablation does exactly this: it initializes from the size-matched SFT-2 checkpoint and trains DAPO on only the items the SFT-2 policy fails on every attempt during pass-at-N annotation (n_c=0); the same per-policy mining procedure used to construct the released Stage 2 set. Table 42, Table 41, and Figs. 3–5 compare it to MEDPSY S2 (MEDPSY = SFT-2 → RL Stage 1 on Easy/Moderate → RL Stage 2 on mined hard). The result is that *every* checkpoint along the *Difficult-only* trajectory is Pareto-dominated by MEDPSY S2 on accuracy and training-time response length, simultaneously. At the best-by-accuracy checkpoint MEDPSY S2

is better on *both* axes by a wide margin: $+2.69$ pp average accuracy and $-1{,}868$ training-time tokens on the 4B model (64.43% / 578 tokens vs. 61.74% / 2,446 tokens), and $+5.90$ pp / $-1{,}515$ tokens on the 1.7B model (56.46% / 864 tokens vs. 50.56% / 2,379 tokens). On the hardest benchmark MedXpertQA the curriculum's lead widens to $+4.33$ pp on the 4B model (31.27% vs. 26.94%) and $+4.82$ pp on the 1.7B model (21.31% vs. 16.49%).

Why the curriculum wins, and why *Difficult-only* validates the design. Fig. 4 traces the training-time mean response length over the full curriculum (the MEDPSY trace truncated at the released S2 checkpoint, S2 step 200 for the 4B model and S2 step 330 for the 1.7B model). The Easy/Moderate Stage 1 brings the policy from the verbose SFT-2 prior ($\sim$ 1.8–2.0K tokens) down to $\sim$ 620–760 tokens; Stage 2 then trains on hard items *starting from* that compressed checkpoint and converges to $\sim$ 580–870 tokens, gaining accuracy throughout. *Difficult-only S1*, by contrast, sees no items it can solve at the start of training (every prompt has $n_c{=}0$ under the SFT-2 policy by construction), so the available reward gradient is dominated by overlong-length and shortcut penalties rather than correctness signal. As a result the policy compresses tokens at essentially the same per-step rate as all DAPO+overlong runs (-65% to -77% across the run on training-time response length) but *loses* accuracy as it does so: on the 4B model the run peaks at step 4 (61.74%, still close to the SFT-2 starting accuracy of 60.69%) and ends at step 100 at 58.08%; on the 1.7B model it peaks at step 4 (50.56%) and ends at step 90 at 47.73%. By the end of training, training-time tokens are essentially matched between *Difficult-only S1* ($\sim$ 590 on the 4B model, $\sim$ 730 on the 1.7B model) and MEDPSY S2 (578 / 864), but accuracy is now 6–9 pp lower. The trajectory therefore validates the design in a stronger form than we initially anticipated: easy/moderate items are necessary not merely to install short reasoning, but to provide a learnable correctness signal at all – a single RL stage on the items the same SFT-2 policy never solves cannot make accuracy progress, and the only thing that does happen with more steps is that the soft-overlong penalty compresses tokens at the cost of accuracy. Easy-first followed by mined hard is what simultaneously installs the short-reasoning habit and gains the hard-item accuracy that defines MEDPSY S2.

A note on units. "Response length" refers to the mean number of tokens the policy emits when it generates a single answer. We report it in two forms that are conceptually the same and differ only in implementation: the *training-time* value is averaged over the $n{=}16$ rollouts per training prompt (sampled at temperature 0.7 during DAPO group sampling) and is logged every step in W&B; the *evaluation* value (used in the per-benchmark token tables of Sec. 5.5 and Table 42) is computed over the standard MedQA / MedXpertQA / PubMedQA prompts under deterministic generation. Both views agree on the headline finding (MEDPSY S2 Pareto-dominates the *Difficult-only* trajectory); we use training-time in Figs. 4–3 because it is per-step granular and free of evaluation-prompt confounds.

Would a longer *Difficult-only* run catch up? *Difficult-only S1* is trained for 50 epochs ($\sim$ 100 optimisation steps) on the 4B model and 30 epochs ($\sim$ 90 steps) on the 1.7B model, matched to the MEDPSY Stage 1 step count, with each model trained on the items its *own* size-matched SFT-2 policy fails on every $n{=}5$ rollout (1,276 items at 4B, 1,888 at 1.7B). Within the tested budget, two signals argue against near-term catch-up, while the smaller and less diverse training set remains a confound. First, the accuracy trajectory degrades after the first few steps (from 61.74% at step 4 down to $\sim$ 58% on the 4B model; from 50.56% at step 4 down to $\sim$ 48% on the 1.7B model). Second, the training-time response length is approximately saturated by step $\sim$ 80 at both sizes ($\sim$ 590–650 tokens on the 4B model, $\sim$ 700–750 tokens on the 1.7B model). The observed *Difficult-only* trajectory is therefore dominated by MEDPSY S2 within this training budget; this does not establish a universal result for data-volume-matched or substantially longer runs.

A.17.3 Limitations of these ablations

Three caveats apply. First, all numbers in this appendix are computed on the 3-benchmark smoke-eval suite (MedQA, MedXpertQA, PubMedQA) under the `inspect-ai` flexible scorer; the per-ablation deltas reproduce under the LLM-as-Parser scorer used for the released checkpoints (Sec. 5.1), but absolute scores differ slightly between the two scorers due to different prompting conventions and post-generation handling. Second, no single ablation in this appendix trains for substantially more compute than the MEDPSY component it replaces; we cannot rule out that with 5–10$\times$ more steps the gap between mined and random Stage-2, or between MEDPSY and *No overlong*, would close further than reported. Third, the *Difficult-only S1* ablation trains on a much smaller and less diverse corpus

Table 41: Headline ablation summary. Training-time mean response length at the best-accuracy checkpoint and best-checkpoint avg accuracy on the 3-benchmark smoke suite. For the SFT-2 baseline the response length is the training-rollout length at the first RL step (since SFT-2 has no RL trace itself). MEDPSY S2 has the shortest response length *and* the highest accuracy at both model sizes. Token reductions are relative to the SFT-2 baseline.

	4B class				1.7B class			
	Tokens	**↓ vs SFT-2**	**Acc**	**Δ vs SFT-2**	**Tokens**	**↓ vs SFT-2**	**Acc**	**Δ vs SFT-2**
SFT-2 baseline	1,826	—	60.69	—	2,053	—	51.62	—
Vanilla GRPO	1,676	−8%	61.80	+1.11	1,762	−14%	51.70	+0.08
No overlong	1,144	−37%	63.34	+2.65	1,203	−41%	53.15	+1.53
MEDPSY S1	618	−66%	62.34	+1.65	761	−63%	52.35	+0.73
Difficult-only S1	2,446	+34%	61.74	+1.05	2,379	+16%	50.56	−1.06
MEDPSY S2	**578**	**−68%**	**64.43**	+3.74	**864**	**−58%**	**56.46**	+4.84

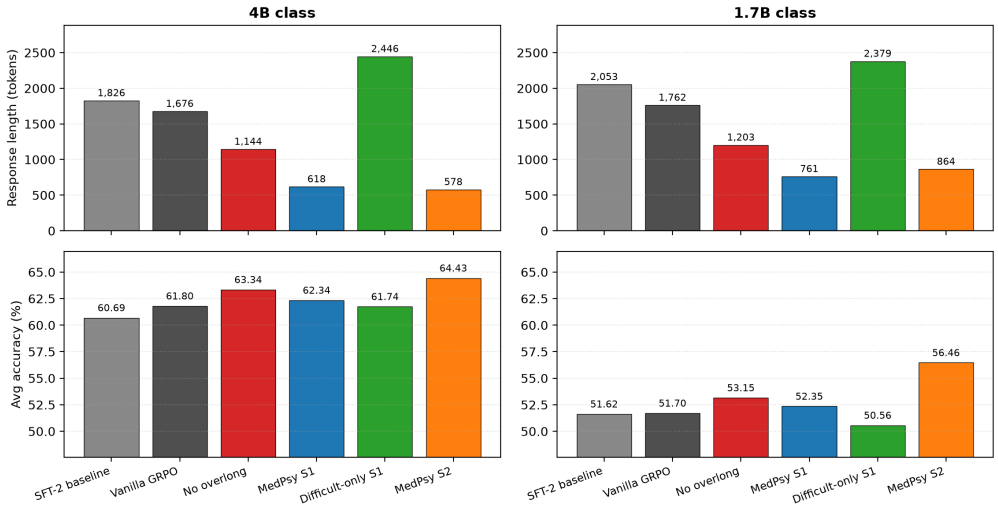


Figure 3: Training-time mean response length (top) and avg accuracy (bottom) at the best-accuracy checkpoint of each RL variant. MEDPSY S2 (orange) wins on both axes at both model sizes.

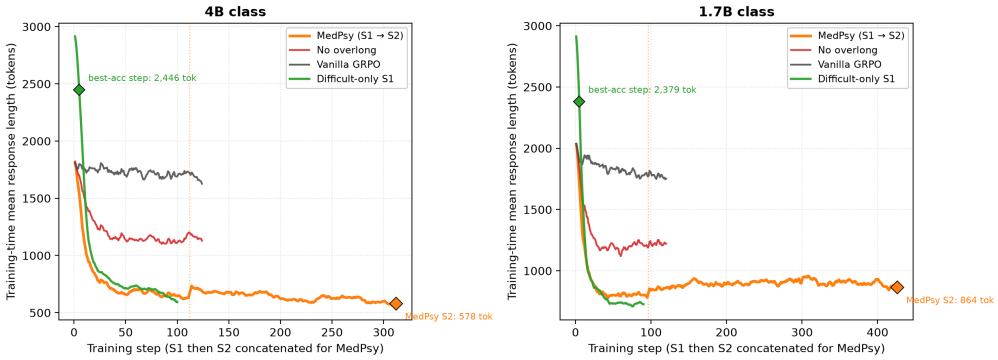


Figure 4: Training-time mean response length over training, traces truncated at the best-accuracy checkpoint. MEDPSY S2 (orange) traces the full pipeline: Easy/Moderate Stage 1 compresses the policy from ∼ 1.8–2.0K tokens down to ∼ 620–760 tokens, then mined-hard Stage 2 (after the dotted vertical line) converges to ∼ 580–870 tokens. The *Difficult-only S1* ablation (green) reaches its best-accuracy checkpoint very early at ∼ 2.4K tokens (diamond marker), and only matches MEDPSY S2's response length late in training, after losing 6–9 pp of accuracy in the process. *No overlong* (red) plateaus much higher; *Vanilla GRPO* (gray) barely moves. Diamond markers show the best-accuracy checkpoint values for MEDPSY S2 and *Difficult-only S1*.

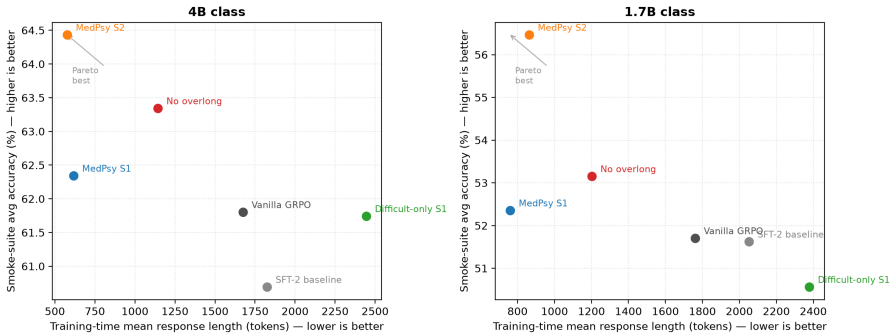


Figure 5: Token–accuracy Pareto across all RL variants at their best-accuracy checkpoints (upper-left is better; MEDPSY S2 is the orange diamond). Response length on the x-axis is the training-time mean at the same checkpoint used for the accuracy on the y-axis. MEDPSY S2 is the unique Pareto-dominant point at both model sizes: shortest responses and highest accuracy.

Table 42: Per-benchmark best-checkpoint detail. *Difficult-only S1* vs. MEDPSY S2 on the smoke-eval suite. MEDPSY S2 wins on the average and on every individual benchmark at both model sizes; the largest gap is on the hardest benchmark, MedXpertQA.

	Best step	MedQA	MedXpertQA	PubMedQA	Avg	med-tok	mxq-tok	pmd-tok
4B class								
Difficult-only S1	4	82.09	26.94	76.20	61.74	1,744	2,756	620
MEDPSY-4B S2 (step 200)	200	**84.84**	**31.27**	**77.20**	**64.43**	**641**	**1,041**	**343**
1.7B class								
Difficult-only S1	4	65.99	16.49	69.20	50.56	2,088	2,758	704
MEDPSY-1.7B S2 (step 330)	330	**74.86**	**21.31**	**73.20**	**56.46**	**1,020**	**1,251**	**534**

than MEDPSY (1,276 items at 4B and 1,888 at 1.7B vs. $\sim$ 15–16K items in MEDPSY Stage 1). We compensate via 50 vs. 4 epochs (4B) and 30 vs. 4 epochs (1.7B) to match the MEDPSY Stage-1 step count; nevertheless, the diversity of the training distribution remains a confound, and a fully matched-data-volume single-stage comparison is left to future work.

A.18 Existing assets, licenses, and terms of use

Tables 43, 44 and 45 report the licenses or governing terms for the existing datasets, benchmarks, model checkpoints, and software tools used in this paper.

Table 43: Existing data and benchmark assets. Licenses are listed for the source release or dataset card used in our experiments.

Asset type	Asset / source	License / terms	Use in this paper
Training seed data	Genesis III biology/medicine subsets [4]	CC-BY-NC-4.0	Prompt-side question/option seeds for Corpus 1.
Training seed data	FineFineWeb health subset [5]	Apache-2.0	Health-domain passages used to generate additional MCQ prompt seeds.
Filtering model/tool	Ultra-FineWeb-classifier [37]	Apache-2.0	Filtering FineFineWeb health-domain passages before prompt generation.
Focused SFT source / benchmark	AfriMedQA [13]	CC-BY-4.0	Focused clinical QA prompt source for Corpus 2 and closed-ended evaluation benchmark.
Focused SFT source / benchmark	MedMCQA [8]	Apache-2.0	Focused clinical QA prompt source for Corpus 2 and closed-ended evaluation benchmark.
Focused SFT source / benchmark	MedQA-USMLE [7]	CC-BY-4.0	Focused clinical QA prompt source for Corpus 2 and closed-ended evaluation benchmark.
RL prompt source	AlphaMed19K [22]	MIT	RL prompt pool for DAPO training and ablations.
Evaluation benchmark	MMLU health subdomains [9]	MIT	Closed-ended health-literacy evaluation.
Evaluation benchmark	MMLU-Pro Health [10]	MIT	Closed-ended health-literacy evaluation.
Evaluation benchmark	MedXpertQA [11]	MIT	Closed-ended expert-reasoning evaluation.
Evaluation benchmark	PubMedQA [12]	MIT	Closed-ended biomedical-research evaluation.
Evaluation benchmark	HealthBench and HealthBench-Hard [14]	MIT	Open-ended clinical-scenario evaluation.
Evaluation benchmark	LongHealth [36]	Apache-2.0	Long-context clinical-document evaluation (App. A.16).

Table 44: Existing model assets used as backbones, teachers, graders, parsers, and baselines. Custom model licenses and acceptable-use policies are reported by name where applicable.

Asset type	Asset / source	License / terms	Use in this paper
Student backbone / baseline	Qwen3-4B-Thinking-2507 / Qwen3-4B family [3]	Apache-2.0	4B student backbone and general baseline.
Student backbone / baseline	Qwen3-1.7B (Thinking) [3]	Apache-2.0	1.7B student backbone and general baseline.
SFT target teacher	Baichuan-M3-235B [27]	Apache-2.0	Assistant-target generation and teacher-selection candidate.
Prompt generator / teacher candidate	Fleming-R1-32B [6]	Apache-2.0	MCQ generation from FineFineWeb health passages and teacher-selection candidate.
Teacher candidate / grader / parser	GPT-OSS-120B [28]	Apache-2.0	Teacher-selection candidate, HealthBench grader, and parser-robustness model.
Closed-ended parser	QwQ-32B [32]	Apache-2.0	Main LLM-as-parser model for closed-ended answer extraction.
Grader / parser	Llama-3.3-70B-Instruct [33]	Llama 3.3 Community License	HealthBench grader and parser-robustness model.
HealthBench grader	CompassJudger-2-32B-Instruct [34]	Apache-2.0	Main HealthBench grader and per-dimension grader.
Medical baseline	MedGemma-27B-text-it [1]	Health AI Developer Foundations License / Google Health AI terms	Large medical baseline.
Medical baseline	MedGemma-1.5-4B-it [2]	Health AI Developer Foundations License / Google Health AI terms	Compact medical baseline.
Backbone-selection baseline	Qwen3-4B (Thinking) [3]	Apache-2.0	4B-class backbone-selection baseline.
Backbone-selection baseline	Llama-3.2-3B-Instruct [40]	Llama 3.2 Community License	4B-class backbone-selection baseline.
Backbone-selection baseline	Llama-3.2-1B-Instruct [40]	Llama 3.2 Community License	Sub-2B backbone-selection baseline.
Backbone-selection baseline	gemma-3-4b-it	Gemma Terms of Use	4B-class backbone-selection baseline.
Backbone-selection baseline	SmolLM3-3B	Apache-2.0	4B-class backbone-selection baseline.
Backbone-selection baseline	SmolLM2-1.7B-Instruct	Apache-2.0	Sub-2B backbone-selection baseline.
Backbone-selection baseline	LFM2.5-1.2B-Thinking	LFM1.0 custom model license	Sub-2B backbone-selection baseline.
Transfer-ablation backbone	DeepSeek-R1-Distill-Qwen-1.5B [30]	MIT	Qwen2.5-derived reasoning backbone (Qwen2.5-Math-1.5B base, Apache-2.0) for the recipe-transfer experiment (App. A.5).
Large medical baseline	Meditron3-70B [17]	Llama 3.1 Community License	Additional large medical baseline.
Medical baseline	Meditron3-8B [17]	Llama 3.1 Community License	Additional medical baseline.
Large medical baseline	OpenBioLLM-70B [18]	Llama 3 license	Additional large medical baseline.
Medical baseline	JSL-MedLlama-3-8B-v2.0 [19]	CC-BY-NC-ND-4.0	Additional medical baseline.

Table 45: Existing software and infrastructure assets. Commercial or hosted services are listed with their governing service terms.

Asset type	Asset / source	License / terms	Use in this paper
Evaluation software	Inspect AI [38]	MIT	Lightweight smoke-eval framework for RL ablations.
Evaluation software	OpenCompass [31]	Apache-2.0	LLM-as-Parser and model-grading evaluation codebase for the main benchmark suite.
Evaluation software	Medmarks v0.1.0 [35]	MIT	Matched-instance pairwise win-rate computation (App. A.11).
Inference / quantization software	llama.cpp [26]	MIT	GGUF conversion, quantization, and evaluation of quantized variants.
Training framework	NVIDIA NeMo 25.09	Apache-2.0 for NeMo framework	Distributed SFT training stack.
Training framework	verl [41]	Apache-2.0	Distributed RL training stack.
Training / inference runtime	PyTorch 2.5.1	BSD-style license	Model training and evaluation runtime.
Inference engine	vLLM	Apache-2.0	RL rollout and inference serving.
Experiment tracking	Weights & Biases	MIT	Experiment tracking and run logging.
Experiment tracking	TensorBoard	Apache-2.0	Training visualization and logs.
Container runtime	Enroot	Apache-2.0	Container runtime for training environment.
Cluster scheduler	SLURM	GPLv2-or-later	Cluster job scheduling.
GPU software stack	CUDA 12.1 / NVIDIA drivers	NVIDIA CUDA Toolkit EULA and NVIDIA driver terms	GPU training and inference software stack.

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: The abstract and Section 1 state four contributions: compact medical models that match or exceed much larger medical baselines, a four-stage post-training recipe, token-efficiency gains over the Qwen3 backbones, and quantized edge deployment with real-device measurements. These claims are supported by Sec. 3–4 and App. A.17 (recipe), Sec. 5 (Tables 1 and 2) and App. A.11 (accuracy), Sec. 5.5 and App. A.10 (token efficiency), and Sec. 5.6, App. A.12, App. A.13, and App. A.15 (quantization and on-device measurements).

   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: Section 6 discusses scope limitations (text-only models, predominantly English data, hallucination risk, no clinical validation), and App. A.17.3 discusses the limitations of the RL ablations.

   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: The paper presents an empirical contribution (models, training recipe, evaluation, quantization study). It contains no theoretical results requiring formal assumptions or proofs.

   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: Sec. 3–4 describes the data-construction principles, teacher and backbone selection, the SFT→SFT→RL→RL curriculum, difficulty annotation, reward function, and optimization choices; Sec. 5 describes the generation, parsing, grading, and decontamination protocols; and App. A.7, App. A.11, App. A.12, and App. A.17 provide additional experimental details and robustness checks. The released artifacts (see the open-access item) include the BF16 and GGUF checkpoints and the evaluation code, so the reported evaluations can be rerun on the released models; we plan to release the synthetic training corpus at a later date. The training code is not released; SFT and RL use the public NVIDIA NeMo and verl frameworks, and all training hyperparameters are listed in App. A.7 (Table 16 for RL).

   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 evaluation code, covering the seven closed-ended benchmarks and HealthBench with the generation, parser, and grader settings used in the paper, is released under Apache-2.0 at `https://github.com/tether-ai-research/qvac-medpsy`. The MedPsy collection at `https://huggingface.co/collections/qvac/medpsy` releases the MEDPSY-1.7B and MEDPSY-4B BF16 checkpoints and GGUF builds of all six evaluated formats. The synthetic SFT training corpus is not yet included; we plan to release it on the same collection at a later date. The training code is not released (see the reproducibility item).

   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: Sec. 4 specifies the four-stage recipe, RL difficulty annotation ($N{=}5$ rollouts), DAPO optimization, reward design, rollout count, clipping parameters, and response-length penalty; App. A.7 gives the SFT configuration (hardware, parallelism, sequence length, batch size, optimizer, learning-rate schedule, epochs, checkpoint format) and the full RL hyperparameter table (Table 16). Sec. 5 specifies generation settings, closed-ended parsing, HealthBench judging, and decontamination, while App. A.2, App. A.4, App. A.11, and App. A.17 document selection and robustness experiments.

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: [No]

   Justification: The main tables report point estimates, each the mean of three generation runs, without confidence intervals, error bars, or significance tests. We report several checks on evaluation variance instead: re-scoring with three closed-ended parsers (maximum average spread 0.45 points) and decontaminated scores (App. A.11), a three-run matched-instance pairwise comparison against MedGemma-27B-text-it (App. A.11), and HealthBench scores under three independent model graders in the main results (Table 2, App. A.8). These checks do not replace formal uncertainty estimates.

   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: In App. A.7, we provide a comprehensive breakdown of the computational resources. Specifically, we detail the hardware configuration (30 nodes, 8×NVIDIA H100 80GB GPUs), the software stack (NVIDIA NeMo, PyTorch, CUDA), and the memory specifications. We report the execution time for the primary pipeline (8,250 H100 GPU hours) with a granular breakdown across training stages, and provide an estimate for the total project compute (30,000 H100 GPU hours) including ablations and failed runs. On-device experiments run on an Apple M5 Pro laptop and a OnePlus 13 smartphone (App. A.13).

   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: We reviewed the NeurIPS Code of Ethics. The work uses public datasets and benchmarks under their stated licenses (App. A.18), synthetic prompt construction, and teacher-generated supervision; it collects no new human-subject data and involves no crowdsourcing. Medical-use risks, hallucination risk, and the lack of prospective clinical validation are discussed in Sec. 6 and Sec. 7.

   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: Positive impacts: the introduction and conclusion discuss private, low-latency on-device medical assistance that keeps protected health information on the device, with shorter generations lowering latency and energy per answer (Sec. 5.6). Negative impacts: Sec. 6 notes that the models can produce confident but incorrect medical statements, must not be used for autonomous diagnosis or treatment, and would need safety review, prospective testing, and regulatory clearance before clinical use. The release includes mitigations through the model cards (see the safeguards item).

    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: The models are released through the MedPsy collection linked in the paper, with model cards that state the intended use (research and prototyping of downstream healthcare tools, not clinical diagnosis or treatment), list known limitations (hallucination, English-only, text-only, knowledge cutoff, training-data bias, not for emergencies), and give safety recommendations for integrators (visible AI-generated disclaimers, clinician oversight, output monitoring and safety filters). Sec. 6 states the same intended-use boundaries. Access is not gated, and the model cards note that no dedicated red-teaming has been performed.

    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: The paper credits the main existing data, benchmark, model, and software assets through citations and repository/model/dataset-card references. App. A.18 explicitly lists the licenses or governing terms for the datasets and benchmarks, model checkpoints, evaluation tools, training frameworks, inference/quantization software, and hosted services used in the 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: [Yes]

    Justification: The MedPsy collection at `https://huggingface.co/collections/qvac/medpsy` releases two kinds of new assets. (i) Models: MEDPSY-1.7B and MEDPSY-4B BF16 checkpoints and GGUF builds under Apache-2.0, with model cards covering the training summary, benchmark results, intended use, limitations, safety recommendations, usage examples, and attribution of upstream data and models. (ii) The evaluation code at `https://github.com/tether-ai-research/qvac-medpsy` under Apache-2.0, with usage instructions. We plan to release the synthetic SFT training corpus on the same collection at a later date, with a dataset card describing its sources, construction, and licensing terms, including those inherited from the source datasets. The paper documents how each asset was built and evaluated (Sec. 3–5.6, App. A.1, App. A.12, App. A.13). No data were collected from people for this work; all sources are public datasets used under their licenses.

    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 involves no crowdsourcing and no research with human subjects. Training data come from public datasets and teacher-model outputs, and all evaluations are automatic: an LLM parser extracts closed-ended answers, and model graders score HealthBench responses against its existing physician-written rubrics.

    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 involves no human-subjects research, so IRB approval is not applicable. Sec. 6 notes that any prospective clinical deployment would require additional safety review, IRB-grade evaluation, prospective testing, and regulatory clearance, none of which the paper claims.

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: LLMs are core to the methodology and are explicitly described and cited. Baichuan-M3-235B generates all SFT reasoning targets (Sec. 3.2); Fleming-R1-32B generates the FineFineWeb-derived MCQ prompt seeds (Sec. 3); GPT-OSS-120B and Fleming-R1-32B are teacher-selection candidates (App. A.2); Qwen3 models are the student backbones (Sec. 4); QwQ-32B is the main closed-ended parser, with Llama-3.3-70B-Instruct and GPT-OSS-120B as robustness parsers (Sec. 5, App. A.11); and CompassJudger-2-32B-Instruct, Llama-3.3-70B-Instruct, and GPT-OSS-120B grade HealthBench responses against physician-written rubrics (App. A.8).

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.