

Identifying Student Real Weaknesses: Learnable Error-Aware Distillation for Medical Question Answering

Yonghe Chen^{1,3}, Xiaoyang Meng^{1,3}, Xiaoding Zhou^{1,3}, Wentian Zhang^{1,3}
Ruimeng Zhao^{1,3}, Ying Lian², Guoqiang Chen², Weiyu Zhang^{1,3*}, Wenpeng Lu^{1,3}

¹Key Laboratory of Computing Power Network and Information Security,
Ministry of Education, Shandong Computer Science Center
(National Supercomputer Center in Jinan),

Qilu University of Technology (Shandong Academy of Sciences), Jinan, China

²The First Affiliated Hospital of Shandong First Medical University
(Shandong Provincial Qianfoshan Hospital), Jinan, China

³Shandong Provincial Key Laboratory of Computing Power Internet and Service Computing,
Shandong Fundamental Research Center for Computer Science, Jinan, China

zwy@qlu.edu.cn

Abstract

Large language models (LLMs) show strong performance in medical question answering, but their computational costs limit deployment under resource constraints. Therefore, an increasing number of studies focus on distilling knowledge from LLMs into smaller models to improve performance in medical question answering. However, existing methods often fail to identify truly useful knowledge to distill, making it difficult to focus distillation on the student’s real weaknesses. As a result, much of the distillation budget is spent on already acquired knowledge rather than on weaknesses that still limit performance. To address this problem, we propose **LEAD**, a **Learnable Error-Aware Distillation** framework that precisely improves the weaknesses of student models. First, LEAD identifies questions answered incorrectly by the student model to determine which weaknesses require teacher guidance. Then, LEAD generates teacher rationales and similar samples to construct candidate supervision for these errors. Through supervision refinement, reliable and learnable training samples are retained from the candidate supervision. Finally, LEAD trains the student on a balanced sample mixture to correct learnable errors while preserving previously acquired knowledge. Across four medical QA benchmarks, LEAD achieves the highest macro-average accuracy across the evaluated student settings. The code is available at <https://github.com/Yh-C12138/LEAD>.

1 Introduction

Large Language Models (LLMs) show strong performance in medical question answering (Singhal

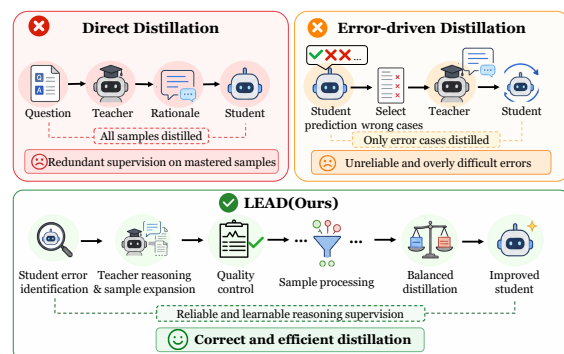


Figure 1: Comparison of distillation paradigms. Unlike direct distillation over all samples or error-driven distillation over all mistakes, LEAD focuses on quality-checked and learnable student errors for targeted rationale supervision.

et al., 2023; Lucas et al., 2024). However, their high computational and memory costs make them difficult to deploy in realistic medical or resource-constrained scenarios (Klang et al., 2024). Therefore, knowledge distillation has become a promising solution for transferring knowledge from LLMs to compact student models. In knowledge distillation, teacher models provide answers and rationales as supervision. Through such supervision, smaller models can improve task performance while maintaining lower inference and deployment costs (Hsieh et al., 2023).

A straightforward form of knowledge distillation is direct distillation. As shown in Figure 1, the teacher generates supervision for all medical questions. Then, the student is fine-tuned on these teacher outputs (Wang et al., 2025; Li et al., 2025a). This strategy relies on the assumption that all samples provide comparable training value for the current student. In practice, many samples may al-

*Corresponding author.

ready be answered correctly by the student, so additional teacher explanations provide limited guidance in correcting the student’s remaining errors. As a result, direct distillation tends to concentrate supervision resources on the student model’s already mastered knowledge, rather than focusing on its specific weaknesses.

Error-driven distillation offers a more targeted alternative to direct distillation (Agarwal et al., 2024). Instead of applying teacher supervision to all samples, error-driven distillation selects samples according to errors made by the student model. As shown on the right side of Figure 1, error-driven distillation identifies incorrect predictions of the student model. Then, teacher explanations are generated only for these incorrectly answered medical questions. Therefore, error-driven distillation directs teacher guidance toward medical questions answered incorrectly by the student model.

However, student-error selection alone remains insufficient for effective distillation. A wrong answer identifies a current failure but does not guarantee reliable or learnable supervision (Lucas et al., 2024). This distinction is especially important in medical QA, where teacher rationales may appear plausible despite flawed intermediate reasoning, and questions vary widely in specialty and difficulty. Some errors reflect prediction instability, whereas others require knowledge or reasoning beyond the student’s current capacity. Distilling all such errors can therefore introduce noisy or overly complex supervision. Effective medical QA distillation should instead prioritize candidate learnable weaknesses, using rationales matched to the student’s current learning capacity.

To address these challenges, we propose a **Learnable Error-Aware Distillation (LEAD)** framework for medical QA. Rather than transferring supervision from all training samples and all observed student errors indiscriminately, LEAD focuses on selected error cases. These cases provide reliable rationales and remain learnable for the current student. First, LEAD identifies questions answered incorrectly by the student to determine which weaknesses require teacher guidance. Then, teacher models generate medical rationales and similar samples for these identified errors. Through supervision refinement, unreliable content is filtered out and samples within the student’s learnable range are retained (Shen et al., 2025). Finally, LEAD trains the student with a balanced rationale distillation objective to correct learnable errors while

preserving previously acquired knowledge. Experiments on four medical QA benchmarks show that LEAD improves the accuracy of student models and achieves higher scores than most distillation baselines. Our contributions are as follows:

- We propose LEAD, a learnable error-aware distillation framework that constructs student-targeted rationale supervision from reliable and learnable error cases.
- We introduce a supervision refinement strategy that improves teacher rationales by removing unreliable content and retaining samples suitable for the current student.
- Experiments on four medical QA benchmarks show that LEAD improves the performance of compact student models and outperforms most supervised and distillation baselines.

2 Related Work

2.1 Language Model Reasoning in Medical Question Answering

Medical question answering provides a controlled setting for evaluating whether language models can acquire and apply clinical knowledge. MedQA, MedMCQA, and CMExam are widely used examination-style benchmarks for this purpose (Jin et al., 2021; Pal et al., 2022; Liu et al., 2023). Recent medical LLM studies improve clinical reasoning through instruction alignment and domain adaptation. Med-PaLM 2 demonstrates the benefits of medical instruction alignment, while BioMistral, PMC-LLaMA, and BianCang incorporate biomedical corpora or medical-domain supervision (Singhal et al., 2025; Labrak et al., 2024; Wu et al., 2024; Wei et al., 2025). However, the computational costs of large models remain a barrier to practical deployment (Klang et al., 2024). Beyond examination-style QA, recent systems address real-world clinical workflows: JointCoder employs compact multi-agent models for Chinese EHR-based ICD coding, while PG-Bench and its knowledge-evolvable assistant support conversation-based patient guidance (Liu et al., 2026; Lu et al., 2025). Retrieval-augmented methods such as Self-RAG and D²-RAG use adaptive or uncertainty-aware retrieval to reduce interference from irrelevant external evidence (Asai et al., 2024; Zhang et al., 2026). KARD instead transfers knowledge-augmented reasoning from stronger models to improve medical and knowledge-intensive QA (Kang et al., 2023).

Despite these advances, existing medical distilla-

tion methods often transfer final answers or noisy rationales indiscriminately. They rarely select supervision that is both quality-checked and learnable for the current student. Small medical QA models therefore still need targeted supervision under limited capacity.

2.2 Chain-of-Thought Distillation

Chain-of-thought (CoT) distillation transfers LLM reasoning behavior to smaller language models (SLMs). Early rationale distillation uses LLM-generated task rationales as supervised labels, enabling SLMs to imitate step-by-step reasoning (Ho et al., 2023; Magister et al., 2023). Mutual-information maximization frames CoT distillation information-theoretically by coupling rationale generation with answer prediction (Chen et al., 2024c). Program-aided Distillation (PaD) uses executable programs and feedback to reduce faulty synthetic reasoning (Zhu et al., 2024). Skip-Thinking segments long rationales and trains students to skip non-essential reasoning, improving efficiency and latency (Chen et al., 2025). Retrieval-augmented generation and tool-use learning further motivate distilling external-knowledge and tool-mediated reasoning into smaller models (Asai et al., 2024; Wei et al., 2025). Related work on rationale extraction has also explored denoising rationalization by retaining faithful, consistent, and salient evidence while suppressing irrelevant rationale content (Si et al., 2024).

However, these methods usually treat teacher rationales as directly usable supervision, which may introduce noisy signals in medical question answering.

2.3 Curriculum Learning and Sample Selection

Curriculum learning and sample selection assume that training examples differ in their optimization value. Classical curriculum learning orders examples from easy to hard using predefined difficulty measures (Bengio et al., 2009). Recent LLM training instead estimates sample utility automatically from multiple perspectives (Huang et al., 2024). Representative methods curate small alignment sets (Zhou et al., 2023), remove low-quality instruction data with LLM judges (Chen et al., 2024b), score complexity, quality, and diversity (Liu et al., 2024), select influential or weak-model-filtered examples (Xia et al., 2024; Li et al., 2024), and rank data by diversity–difficulty–

dependability, gradient-clustered coreset value, or response-style consistency (Zhang et al., 2025a,b; Li et al., 2025b).

However, these methods mainly estimate sample value from general attributes such as quality, diversity, difficulty, or influence. They do not explicitly ask whether a sample reflects a quality-checked and learnable error of the current student.

3 Method

In this section, we present **LEAD (Learnable Error-Aware Distillation)**, a framework that constructs quality-controlled and learnable rationale supervision for capacity-limited student models. As illustrated in Figure 2, the pipeline comprises six modules. The first two modules are student error identification (Section 3.1) and teacher rationale and sample expansion (Section 3.2). Teacher quality control (Section 3.3) and teacher rationale compression (Section 3.4) further refine the generated supervision. Learnability filtering (Section 3.5) selects samples suitable for the current student. Finally, balanced rationale distillation (Section 3.6) updates the student with the selected supervision.

3.1 Student Error Identification

The student error identification module identifies questions that the student model answers incorrectly. These questions form the candidate pool for subsequent corrective distillation.

Given a training set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, where $x_i = (q_i, A_i)$, the student model performs forward inference on each sample to obtain the prediction distribution:

$$p_i = S_{\theta_s}(x_i). \quad (1)$$

The predicted label is then computed as:

$$\hat{y}_i = \arg \max_y p_i(y). \quad (2)$$

By comparing $\hat{y}_i$ with the ground-truth label y_i , we define the student error set as:

$$\mathcal{D}_{\text{wrong}} = \{(x_i, y_i) \in \mathcal{D} \mid \hat{y}_i \neq y_i\}. \quad (3)$$

This set contains samples that the student has not yet handled correctly and serves as the candidate pool for teacher-rationale generation and learnability filtering. By focusing on these error cases, LEAD directs supervision toward the student’s current capability gaps.

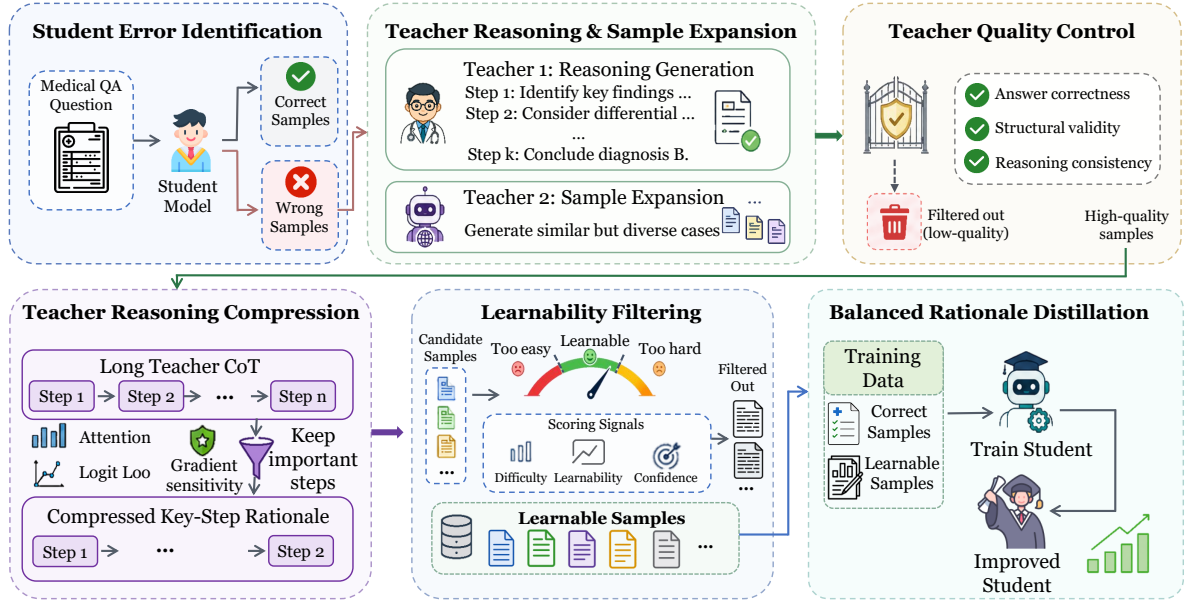


Figure 2: Overview of LEAD. The framework identifies student error samples, constructs teacher candidate rationales, filters low-quality outputs, compresses redundant rationale steps, selects learnable samples, and performs balanced rationale distillation.

3.2 Teacher Rationale and Sample Expansion

After identifying student error samples, LEAD uses teacher models to construct candidate medical rationale supervision. We assign two teacher roles: one for rationale generation and the other for semantics-preserving sample expansion.

Teacher rationale generation. The first teacher model, T_1 , constructs candidate rationale supervision for student error samples. For each error sample $x_i = (q_i, A_i)$ in $\mathcal{D}_{\text{wrong}}$, T_1 receives the question and candidate options as input. It produces a teacher answer $\bar{y}_i$ and a structured rationale r_i that explains the clinical evidence supporting its predicted answer. The pair $(\bar{y}_i, r_i)$ is treated as a candidate supervision signal.

Sample expansion. The second teacher model, T_2 , augments the preliminary outputs from T_1 in a semantics-preserving manner. It generates equivalent variants by rephrasing questions, reordering answer options, or reformulating rationales, while preserving the original clinical facts, correct answer, and answer-discriminative rationale. The original and augmented samples are then merged into the teacher candidate set for subsequent quality control.

The resulting teacher candidate set is defined as $\mathcal{D}_{\text{teach}} = \mathcal{D}_{T_1} \cup \mathcal{D}_{\text{aug}}$. Here, $\mathcal{D}_{T_1} = \{(q_i, A_i, y_i, \bar{y}_i, r_i) \mid i \in \mathcal{I}_{\text{err}}\}$ contains the outputs generated by T_1 for student error samples,

where $\mathcal{I}_{\text{err}}$ denotes the corresponding index set. $\mathcal{D}_{\text{aug}} = \{(\tilde{q}_j, \tilde{A}_j, \tilde{y}_j, \tilde{r}_j) \mid j \in \mathcal{J}_{\text{aug}}\}$ contains the augmented samples generated by T_2 , where $\mathcal{J}_{\text{aug}}$ denotes the augmented-sample index set. Each augmented sample preserves the medical semantics and answer-discriminative rationale of its source sample.

3.3 Teacher Quality Control

Before distillation, LEAD applies a quality-control gate to the teacher candidate samples. Given a teacher candidate sample $z_i = (q_i, A_i, y_i, \bar{y}_i, r_i)$, the gate evaluates supervision quality along three dimensions: answer correctness, structural validity, and rationale consistency. Answer correctness and structural validity are implemented as deterministic checks. Specifically, $C_{\text{ans}}(z_i) \in \{0, 1\}$ verifies whether the teacher-predicted answer $\bar{y}_i$ matches the ground-truth label y_i , and $C_{\text{fmt}}(z_i) \in \{0, 1\}$ checks whether the output contains parseable fields for the final answer, key evidence, and stepwise rationale.

For rationale consistency, we use an LLM-based verifier to assign a scalar score $s_{\text{reason}}(z_i) \in [0, 1]$. The verifier receives the question, answer options, ground-truth answer, teacher-predicted answer, and teacher rationale. It assesses whether the rationale is internally coherent, medically plausible, and aligned with answer-discriminative evidence for

the ground-truth answer. We then define the overall quality score as:

$$Q_{qc}(z_i) = C_{ans}(z_i) \cdot C_{fmt}(z_i) \cdot s_{reason}(z_i). \quad (4)$$

A teacher candidate sample is retained if its quality score meets the quality-control threshold:

$$G_{qc}(z_i) = \mathbb{I}[Q_{qc}(z_i) \geq \gamma_{qc}]. \quad (5)$$

The retained teacher samples are collected as:

$$\mathcal{D}_{qc} = \{z_i \in \mathcal{D}_{teach} \mid G_{qc}(z_i) = 1\}. \quad (6)$$

In our implementation, the rationale-consistency verifier is GPT-4o. It is independent of the open-source teacher models used for rationale generation and semantic augmentation. The verifier is queried with a fixed rubric-based prompt and used only for offline filtering. Appendix A.1 provides the full prompt, scoring rubric, and robustness analysis with alternative open-source and biomedical verifiers.

3.4 Teacher Rationale Compression

Teacher quality control filters invalid samples, but retained rationales may still contain redundant or low-value reasoning steps. LEAD therefore compresses each quality-controlled rationale by estimating the contribution of each step to the final answer decision.

Given a quality-controlled sample $z_i = (q_i, A_i, y_i, \bar{y}_i, r_i) \in \mathcal{D}_{qc}$, where r_i denotes the teacher-generated CoT, LEAD requires r_i to follow a structured format with explicit step boundaries. Before compression, any explicit final-answer field is removed from r_i to avoid direct answer leakage; this field is not treated as a rationale step and is not eligible for retention. The remaining rationale is segmented into $C_i = [s_{i,1}, s_{i,2}, \dots, s_{i,n}]$, where each $s_{i,k}$ corresponds to a rationale step with token span $\mathcal{T}_{i,k}$.

For each step $s_{i,k}$, LEAD computes attention, logit leave-one-out, and gradient sensitivity scores using the frozen teacher model T_1 in an offline teacher-forced scoring pass. The scoring sequence is $[q_i, A_i; C_i; \text{AnswerPrefix}; a_i]$, where $a_i = \text{tok}(v(y_i))$ is the tokenised gold-answer verbalization. The answer tokens a_i are used only as scoring targets; the compressible content remains C_i , and the compressed rationale c_i contains only selected rationale steps. The attention score $\text{attn}_{i,k}$ averages attention from all positions of a_i to $\mathcal{T}_{i,k}$

over all heads and the last four decoder layers. The logit leave-one-out score $\text{logit}_{i,k}$ is the decrease in the teacher-forced log probability of a_i after removing $s_{i,k}$ from C_i . The gradient score $\text{grad}_{i,k}$ is the average ℓ_2 norm of the gradient of the negative log-likelihood of a_i with respect to the final-layer hidden states in $\mathcal{T}_{i,k}$. For multi-token answers, all scores are aggregated over the tokens in a_i . No teacher parameters are updated during scoring.

For each score type $m \in \mathcal{M}$, where $\mathcal{M} = \{\text{attn}, \text{logit}, \text{grad}\}$, LEAD first applies min-max normalization within the same rationale:

$$\hat{s}_{i,k}^m = \frac{s_{i,k}^m - \min_j s_{i,j}^m}{\max_j s_{i,j}^m - \min_j s_{i,j}^m + \epsilon}. \quad (7)$$

The step importance score is then computed by weighted fusion:

$$F_{i,k} = \sum_{m \in \mathcal{M}} w_m \hat{s}_{i,k}^m. \quad (8)$$

where $F_{i,k}$ denotes the fused importance score for step $s_{i,k}$.

LEAD selects rationale steps whose fused scores exceed a threshold ρ to form the compressed rationale $c_i \subseteq C_i$. To avoid excessive pruning, at least $\lceil n/3 \rceil$ steps are retained for each rationale. If fewer steps meet the threshold, the top-scoring steps are added. The compressed distillation set is:

$$\mathcal{D}_{comp} = \{(q_i, A_i, y_i, c_i) \mid z_i \in \mathcal{D}_{qc}\}. \quad (9)$$

The resulting compressed set is then used as the input to learnability filtering.

3.5 Learnability Filtering

Existing quality and difficulty-based criteria characterize supervision without directly measuring its learning response for the current student. Learnability Filtering instead estimates this response from the held-out NLL change induced by lightweight calibration and retains samples within an intermediate-response region.

Given a compressed sample $(q_i, A_i, y_i, c_i) \in \mathcal{D}_{comp}$, where c_i denotes the compressed rationale, we form the target sequence $u_i = [c_i; y_i]$. The label y_i is represented by its answer token or verbalized answer. The student loss on this target is measured by the teacher-forced negative log-likelihood:

$$\mathcal{L}_{NLL}(x_i, u_i; S_\theta) = - \sum_{t=1}^{|u_i|} \log p_\theta(u_{i,t} \mid x_i, u_{i,<t}), \quad (10)$$

where $x_i = (q_i, A_i)$ and $u_{i,<t}$ denotes the preceding target tokens.

To estimate sample learnability for the current student, LEAD compares the NLL before and after lightweight calibration under held-out scoring. Let S_{θ_0} be the original student model. We partition $\mathcal{D}_{\text{comp}}$ into disjoint folds and, for each scoring fold, train a temporary LoRA-calibrated copy $S_{\theta_1}^{(-k)}$ only on calibration samples outside that fold. Thus, no sample is scored by a model that has been updated on that sample. The learnability score for d_i in the held-out fold is defined as:

$$\Delta\text{NLL}_i = \mathcal{L}_{\text{NLL}}(x_i, u_i; S_{\theta_0}) - \mathcal{L}_{\text{NLL}}(x_i, u_i; S_{\theta_1}^{(-k)}). \quad (11)$$

A larger positive value indicates that the student reduces its loss on the sample after calibration. Small or negative responses provide limited learning signal, whereas extremely large values may reflect unstable or overly update-sensitive supervision. For compact notation, let $d_i = (q_i, A_i, y_i, c_i)$ and $\Delta_i = \Delta\text{NLL}_i$. LEAD retains samples within an intermediate learnability range:

$$\mathcal{D}_{\text{learn}} = \{d_i \in \mathcal{D}_{\text{comp}} \mid \Delta_i \in [\tau_l, \tau_h]\}. \quad (12)$$

Here, τ_l and τ_h are lower and upper thresholds selected on a validation set or by quantiles over the held-out ΔNLL scores.

3.6 Balanced Rationale Distillation

After learnability filtering, LEAD performs balanced rationale distillation to update the student with learnable error supervision while preserving its existing correct-answer behavior. To this end, LEAD mixes the learnable error samples with a small set of samples that the student already answers correctly.

We define the correctly answered sample set as $\mathcal{D}_{\text{correct}} = \{(x_i, y_i) \in \mathcal{D} \mid \hat{y}_i = y_i\}$. A retention subset $\mathcal{D}_{\text{keep}} \subseteq \mathcal{D}_{\text{correct}}$ is sampled and combined with the learnable error set to form $\mathcal{D}_{\text{LEAD}} = \mathcal{D}_{\text{learn}} \cup \mathcal{D}_{\text{keep}}$. The final objective is $\mathcal{L}_{\text{LEAD}} = \lambda \mathcal{L}_{\text{learn}} + (1 - \lambda) \mathcal{L}_{\text{keep}}$, where $\mathcal{L}_{\text{learn}}$ denotes the rationale distillation loss on learnable error samples, $\mathcal{L}_{\text{keep}}$ denotes the answer-level retention loss on correctly answered samples, and λ controls the trade-off between error correction and capability retention.

We analyze the sensitivity to r_{keep} and λ in Appendix C.1.

4 Experiments

4.1 Experimental Setup

Datasets and Models. We evaluate LEAD on four medical multiple-choice QA benchmarks: MedQA (Jin et al., 2021), MedMCQA (Pal et al., 2022), CMExam (Liu et al., 2023), and MMLU-Pro-Med (Wang et al., 2024). MedQA and MedMCQA are English medical examination datasets. CMExam focuses on Chinese medical examinations, and MMLU-Pro-Med is constructed from medicine-related questions in MMLU-Pro. We use Qwen2.5-7B-Instruct (Qwen Team, 2025) and HuatuoGPT-o1-7B (Chen et al., 2024a) as teacher models. For students, we evaluate Qwen2.5-0.5B/1.5B/3B-Instruct (Qwen Team, 2025), Llama-3.2-1B-Instruct (Llama Team, 2024), Phi-4-mini-instruct (Abouelenin et al., 2025), and Gemma-2-2B-IT (Gemma Team, 2024). Accuracy is the primary metric on all datasets.

Baselines. Six types of baselines were compared. (1) **Prompting**, which includes Zero-shot, Few-shot, and CoT prompting. These methods evaluate inference-time prompting without parameter updates. (2) **SFT**, which uses the same 10k-sample source subset as LEAD. It trains the student with gold answers. (3) **Answer-KD**, which uses only final answers generated by the teacher. (4) **Step-KD**, which trains the student with teacher rationales in a step-by-step format (Hsieh et al., 2023). (5) **GKD**, which denotes generalized knowledge distillation (Agarwal et al., 2024). It trains the student with teacher outputs under the same source data budget. (6) **Raw Error-KD**, which trains the student on teacher rationales generated for the original student-error samples, without quality control, rationale compression, or learnability filtering.

Implementation details. For datasets with official training splits, all trainable methods use the same benchmark-specific 10k-sample subset. MMLU-Pro-Med is used only for held-out evaluation, as detailed in Appendix A.2. Teacher rationale generation uses greedy decoding with temperature 0.0 and top- p 1.0. The maximum generation length is 1024 tokens for structured step-by-step rationales. Sample expansion uses temperature 0.4, top- p 0.9, and maximum length 2048. Quality control uses deterministic answer and format checks plus an LLM verifier for rationale consistency, with $\gamma_{\text{qc}} = 0.55$. Rationale compression uses equal fusion weights $w_a = w_l = w_g = 1/3$,

Table 1: Main results on four medical QA benchmarks. Teacher models are reported as references. For each student model, the best result is shown in **bold** and the second-best result is underlined. Our method is highlighted in blue.

Model	Method	Medical Exam QA			Know. QA	Model	Method	Medical Exam QA			Know. QA
		MedQA	MedMCQA	CMEExam	MMLU-Pro-Med			MedQA	MedMCQA	CMEExam	MMLU-Pro-Med
Qwen2.5 7B-Inst.	Teacher1	52.71	55.80	79.66	60.65	HuatuoGPT o1-7B	Teacher2	62.61	62.58	84.28	63.39
Qwen2.5 0.5B-Inst.	Zero-shot	28.65	33.78	38.54	17.00	Qwen2.5 1.5B-Inst.	Zero-shot	36.61	42.46	56.40	39.22
	Few-shot	28.36	35.52	41.36	17.26		Few-shot	36.84	44.13	62.04	40.52
	CoT	20.82	23.26	27.91	13.81		CoT	37.47	41.55	60.13	41.50
	SFT	<u>31.81</u>	35.50	44.63	21.89		SFT	42.58	<u>47.16</u>	61.55	41.62
	Answer-KD	31.66	35.69	44.76	21.76		Answer-KD	<u>43.75</u>	<u>45.33</u>	62.02	42.41
	Step-KD	28.83	35.64	36.59	18.89		Step-KD	39.28	44.90	61.02	42.67
	GKD	30.87	36.10	<u>45.01</u>	23.19		GKD	41.08	45.11	<u>63.92</u>	42.21
	Raw Error-KD	29.73	<u>36.26</u>	38.36	18.87		Raw Error-KD	40.50	44.20	57.00	29.40
	LEAD	33.39	38.82	45.62	25.93		LEAD	46.03	50.32	64.59	45.15
		↑ 1.58	↑ 2.56	↑ 0.61	↑ 2.74			↑ 2.28	↑ 3.16	↑ 0.67	↑ 2.48
Qwen2.5 3B-Inst.	Zero-shot	42.26	49.10	66.11	47.43	Llama-3.2 1B-Inst.	Zero-shot	33.31	34.62	32.60	19.35
	Few-shot	44.46	49.77	<u>71.03</u>	53.55		Few-shot	32.84	38.82	36.09	23.39
	CoT	39.83	47.12	61.81	47.62		CoT	33.54	34.66	33.91	30.55
	SFT	<u>47.76</u>	51.26	67.91	45.99		SFT	37.86	39.56	37.46	32.57
	Answer-KD	47.29	<u>51.49</u>	72.35	52.25		Answer-KD	37.39	39.25	<u>37.58</u>	32.90
	Step-KD	42.89	48.15	64.78	48.53		Step-KD	35.90	39.73	36.42	30.49
	GKD	43.60	48.12	67.76	52.31		GKD	36.92	44.25	35.48	33.45
	LEAD	48.78	54.43	70.58	50.62		LEAD	39.20	46.64	38.51	35.11
		↑ 1.02	↑ 2.94	↓ 1.77	↓ 2.93			↑ 1.34	↑ 2.39	↑ 0.93	↑ 1.66
Phi-4 mini-Inst.	Zero-shot	42.81	45.64	33.17	39.10	Gemma-2 2B-IT	Zero-shot	36.61	42.44	35.86	44.43
	Few-shot	43.68	49.08	34.65	41.56		Few-shot	36.14	42.31	35.21	43.19
	CoT	48.15	48.31	34.68	45.15		CoT	36.84	42.67	35.35	44.95
	SFT	<u>50.27</u>	51.61	36.90	47.36		SFT	42.66	44.06	37.58	45.54
	Answer-KD	50.35	<u>51.64</u>	<u>37.20</u>	50.10		Answer-KD	44.70	<u>44.63</u>	37.86	44.36
	Step-KD	46.43	50.35	37.01	53.70		Step-KD	45.09	44.42	38.51	45.73
	LEAD	48.78	54.94	37.45	57.79		LEAD	47.37	47.98	42.74	47.10
		↓ 1.57	↑ 3.30	↑ 0.25	↑ 4.09			↑ 2.28	↑ 3.35	↑ 4.23	↑ 1.37

Note: ↑ and ↓ indicate the difference between LEAD and the best non-LEAD baseline under the same student model. “Inst.” abbreviates Instruct checkpoints.

retention threshold $\rho = 0.20$, minimum keep ratio 0.34, and maximum scoring length 2048. For learnability filtering, we use held-out cross-fitting: a fresh calibration LoRA adapter is trained on 10% of quality-passed samples outside the scoring fold, and ΔNLL is computed only on the held-out fold. We set τ_l, τ_h to the 10th and 90th percentiles of the resulting held-out scores within each benchmark. The distillation stage was conducted on a single NVIDIA RTX 3090 GPU, taking 1422.550 seconds (23.71 minutes, 0.395 GPU-hours), and the total cost including warmup calibration was 0.464 GPU-hours. Appendix B provides a full cost audit and distinguishes final-stage training efficiency from end-to-end construction cost.

4.2 Main Results

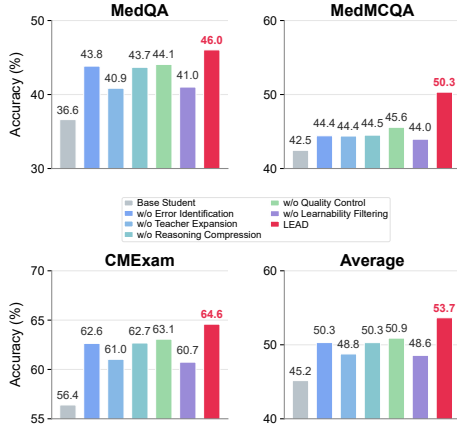
Table 1 summarises the main results under three in-domain benchmark-specific training settings and one MedQA→MMLU-Pro-Med held-out transfer setting. LEAD improves the performance of every student model over its zero-shot baseline. This suggests that the framework remains effective across model families and benchmark settings.

Compared with prompting-based methods,

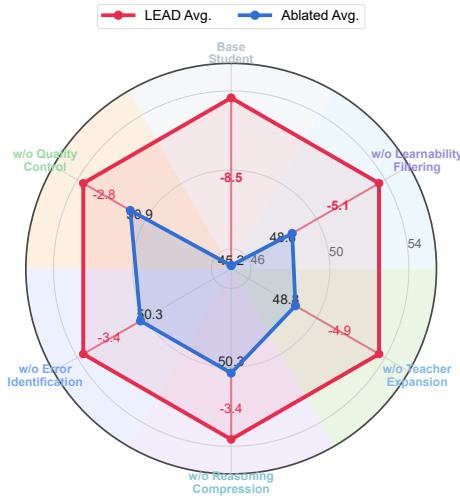
LEAD yields more stable accuracy gains. This is because it uses refined teacher rationales as training supervision. It does not rely on compact students to generate useful rationales only at inference time. The gains are also consistent across Qwen, Llama, Phi, and Gemma students. This indicates that LEAD is not tied to a single student architecture.

The results are especially meaningful for compact students. Their limited capacity makes direct rationale imitation difficult. By filtering unreliable rationales, compressing redundant steps, and retaining learnable samples, LEAD provides supervision that better matches compact student capacity. This explains why LEAD often improves over direct rationale distillation methods such as Step-KD.

Compared with supervised and distillation baselines, LEAD shows the advantage of selective rationale supervision. Although LEAD is not always the best method on every benchmark, it provides more stable overall improvements across students and evaluation settings. Appendix A.1 shows that the results are stable across verifier choices. LEAD reduces the final student-side fine-tuning token budget through targeted weakness-oriented data con-



(a) Accuracy of LEAD and its ablated variants.



(b) Performance profile across medical QA benchmarks.

Figure 3: Ablation study of LEAD. Panel (a) reports dataset-wise and average accuracies. Panel (b) summarises the average performance drop caused by each ablation.

struction, but it does not aim to minimise the total offline construction-plus-training cost. Appendix B clarifies the trade-off between final fine-tuning efficiency and extra offline construction cost. To further examine whether the aggregate gains correspond to corrections of specific student weaknesses, we report a fine-grained repair analysis by medical specialty, knowledge type, and error type in Appendix F.4.

4.3 Ablation Studies

We conduct ablation studies using Qwen2.5-1.5B-Instruct as the student model. Each variant removes one LEAD component while keeping the remaining settings unchanged.

As shown in Figure 3, removing any compo-

nent reduces average performance. This confirms that the modules provide complementary benefits. The largest drops come from removing Learnability Filtering and Teacher Expansion. The average accuracy decreases to 48.6 and 48.8, respectively. This suggests that both modules are central to constructing useful supervision. Learnability Filtering removes samples outside the student’s learnable range. Teacher Expansion improves supervision coverage through semantics-preserving variants.

Error Identification provides notable gains, especially on MedMCQA, by targeting student-specific failures. Reasoning Compression and Quality Control also improve performance by producing compact rationales and removing unreliable teacher outputs.

We also examine the effect of teacher model combinations. The results show that mixing different teacher models provides more complementary supervision. Detailed results are reported in Appendix F.3.

4.4 Comparison with Related Distillation Methods

To further evaluate whether LEAD improves over closely related rationale-distillation and data-selection approaches, we implement adapted versions of four representative methods under two Qwen2.5 student sizes. Specifically, QCRD applies quality-guided rationale filtering before student training (Wang et al., 2025). Skip-Thinking uses compressed teacher rationales as supervision (Chen et al., 2025). Superfiltering selects training samples with a weak-model filtering criterion (Li et al., 2024). DEITA selects instruction data by scoring sample quality and complexity while maintaining data diversity (Liu et al., 2024).

Student	Method	MedQA	MedMCQA	CMExam	MMLU-Pro-Med	Avg.
Qwen2.5-0.5B	Zero-shot	28.65	33.78	38.54	17.00	29.49
	QCRD	31.72	36.62	42.18	23.65	33.54
	Skip-Thinking	32.56	36.72	44.37	24.17	34.45
	Superfiltering	24.51	27.73	33.10	12.51	24.46
	DEITA	32.78	36.53	45.55	23.06	34.48
	LEAD	33.39	38.82	45.62	25.93	35.94
Qwen2.5-1.5B	Zero-shot	36.61	42.46	56.40	39.22	43.67
	QCRD	45.88	45.57	63.81	44.56	49.95
	Skip-Thinking	45.56	45.88	63.89	45.02	50.09
	Superfiltering	43.13	43.94	61.06	38.83	46.74
	DEITA	46.11	45.47	63.91	44.10	49.90
	LEAD	46.03	50.32	64.59	45.15	51.52

Table 2: Comparison with related distillation methods under two Qwen2.5 student sizes. Avg. denotes the macro average across four benchmarks.

As shown in Table 2, LEAD achieves the highest macro-average accuracy under both Qwen2.5-0.5B-

Instruct and Qwen2.5-1.5B-Instruct. These results show that targeting reliable and learnable student errors is more effective than relying on isolated filtering or data selection criteria.

5 Conclusion

In this paper, we introduced LEAD, whose primary technical contribution is a student-conditioned learnability filter for error-centered medical QA distillation. By combining held-out calibration responses with teacher-side supervision refinement and balanced training, LEAD constructs rationale supervision that is reliable and empirically learnable for the current student. Experimental results on four medical QA benchmarks demonstrate the effectiveness of LEAD. LEAD outperforms most supervised and distillation baselines. Moreover, ablation studies show that each major component contributes to the final performance. Future work will extend LEAD beyond answer selection to open-ended medical QA and evaluation for deployment.

Limitations

LEAD has several limitations. First, although LEAD is not intrinsically restricted to medicine, our experiments cover only medical multiple-choice QA. Thus, its cross-domain generality and performance in open-ended or clinical settings remain unverified. Second, LEAD relies on teacher-generated rationales and automatic proxy-based filtering. Although these mechanisms improve supervision quality, imperfect rationales may still pass the filtering process, and useful rationales may also be discarded. Third, rationale compression improves compactness but may remove contextual details that are useful for difficult cases. LEAD incurs additional offline computation for teacher rationale generation, sample expansion, verifier-based quality control, rationale-compression scoring, and learnability estimation. Its efficiency advantage therefore refers specifically to the reduced token budget for final student fine-tuning, rather than to lower end-to-end cost. Thus, LEAD trades a higher offline construction cost for a smaller, more targeted student fine-tuning set. Future work should evaluate LEAD in more realistic clinical tasks, incorporate clinician annotations, and examine robustness, calibration, and safety in deployment-oriented settings.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No.62376130, No.62402258), the Natural Science Foundation of Shandong Province (No.ZR2024MF088), Taisihan Scholars Program (No.TSQN202507242), Program of New Twenty Policies for Universities of Jinan (No.202333008), Key Research and Development Program of Shandong Province (Major Science and Technology Innovation Project) (No.2026CXGC010604) and the Pilot Project for Integrated Innovation of Science, Education, and Industry of Qilu University of Technology (Shandong Academy of Sciences) (Grant No.2025ZDZX01).

References

- Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, Dong Chen, Dongdong Chen, Jun-Kun Chen, Weizhu Chen, Yen-Chun Chen, Yi-ling Chen, Qi Dai, Xiyang Dai, Ruchao Fan, and 55 others. 2025. Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras. *arXiv preprint arXiv:2503.01743*.
- Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. 2024. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*.
- Akari Asai, Zeqiu Wu, Yizhong Wang, Avi Sil, and Hannaneh Hajishirzi. 2024. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *International conference on learning representations*.
- Yoshua Bengio, Jerome Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 41–48.
- Junying Chen, Zhenyang Cai, Ke Ji, Xidong Wang, Wanlong Liu, Rongsheng Wang, Jianye Hou, and Benyou Wang. 2024a. Huatuogpt-o1, towards medical complex reasoning with llms. *arXiv preprint arXiv:2412.18925*.
- Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srivasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. 2024b. Alpapasus: Training a better alpaca with fewer data. In *International Conference on Learning Representations*.

- Xiaoshu Chen, Sihang Zhou, Ke Liang, Xiaoyu Sun, and Xinwang Liu. 2025. Skip-thinking: Chunk-wise chain-of-thought distillation enable smaller language models to reason better and faster. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 12142–12157.
- Xin Chen, Hanxian Huang, Yanjun Gao, Yi Wang, Jishen Zhao, and Ke Ding. 2024c. Learning to maximize mutual information for chain-of-thought distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6857–6868.
- Gemma Team. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.
- Namgyu Ho, Laura Schmid, and Se-Young Yun. 2023. Large language models are reasoning teachers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14852–14882.
- Cheng-Yu Hsieh, Chun-Liang Li, Chih-kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8003–8017.
- Hui Huang, Bing Xu, Xinnian Liang, Kehai Chen, Muyun Yang, Tiejun Zhao, and Conghui Zhu. 2024. Multi-view fusion for instruction mining of large language model. *Information Fusion*, 110:102480.
- Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421.
- Minki Kang, Seanie Lee, Jinheon Baek, Kenji Kawaguchi, and Sung Ju Hwang. 2023. Knowledge-augmented reasoning distillation for small language models in knowledge-intensive tasks. In *Advances in Neural Information Processing Systems*, volume 36.
- Eyal Klang, Donald Apakama, Ethan E Abbott, Akhil Vaid, Joshua Lampert, Ankit Sakhuja, Robert Freeman, Alexander W Charney, David Reich, Monica Kraft, Girish N Nadkarni, and Benjamin S Glicksberg. 2024. A strategy for cost-effective large language model use at health system-scale. *NPJ digital medicine*, 7(1):320.
- Yanis Labrak, Adrien Bazoge, Emmanuel Morin, Pierre-Antoine Gourraud, Mickael Rouvier, and Richard Dufour. 2024. BioMistral: A collection of open-source pretrained large language models for medical domains. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5848–5864.
- Ming Li, Yong Zhang, Shwai He, Zhitao Li, Hongyu Zhao, Jianzong Wang, Ning Cheng, and Tianyi Zhou. 2024. Superfiltering: Weak-to-strong data filtering for fast instruction-tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pages 14255–14273.
- Yuetai Li, Xiang Yue, Zhangchen Xu, Fengqing Jiang, Luyao Niu, Bill Yuchen Lin, Bhaskar Ramasubramanian, and Radha Poovendran. 2025a. Small models struggle to learn from strong reasoners. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 25366–25394.
- Zhuang Li, Yuncheng Hua, Thuy Vu, Haolan Zhan, Lizhen Qu, and Gholamreza Haffari. 2025b. Scar: Data selection via style consistency-aware response ranking for efficient instruction-tuning of large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pages 12756–12790.
- Junling Liu, Peilin Zhou, Yining Hua, Dading Chong, Zhongyu Tian, Andrew Liu, Helin Wang, Chenyu You, Zhenhua Guo, Lei Zhu, and Michael Lingzhi Li. 2023. Benchmarking large language models on CMExam: A comprehensive chinese medical exam dataset. In *Advances in Neural Information Processing Systems*, volume 36.
- Kangjun Liu, Zhenyu Li, Jianlei Wang, Hongjiao Guan, Ying Lian, Guoqiang Chen, Tao Xin, and Wenpeng Lu. 2026. JointCoder: Exploring automated ICD coding on real-world Chinese EHRs with a multi-agent framework. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 263–274, San Diego, California, United States. Association for Computational Linguistics.
- Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. 2024. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. In *International Conference on Learning Representations*.
- Llama Team. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Wenpeng Lu, Kangjun Liu, Jianlei Wang, Xueping Peng, Tao Shen, Fa Zhu, Weiyu Zhang, Jiabing Zhu, Tao Xin, and Athanasios V. Vasilakos. 2025. Advancing Chinese conversation-based patient guidance with a benchmark and knowledge-evolvable assistant. *IEEE Journal of Biomedical and Health Informatics*, pages 1–12.
- Mary M. Lucas, Justin Yang, Jon K. Pomeroy, and Christopher C. Yang. 2024. Reasoning with large language models for medical question answering. *Journal of the American Medical Informatics Association*, 31(9):1964–1975.
- Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. 2023. Teaching small language models to reason. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 1773–1781.

- Ankit Pal, Logesh Kumar Umapathi, and Malaikanan Sankarasubbu. 2022. MedMCQA: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pages 248–260.
- Qwen Team. 2025. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.
- Zhenyi Shen, Hanqi Yan, Linhai Zhang, Zhanghao Hu, Yali Du, and Yulan He. 2025. CODI: Compressing chain-of-thought into continuous space via self-distillation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 677–693.
- Jiasheng Si, Yingjie Zhu, Wenpeng Lu, and Deyu Zhou. 2024. Denoising rationalization for multi-hop fact verification via multi-granular explainer. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 12593–12608, Miami, Florida, USA. Association for Computational Linguistics.
- Karan Singhal, Shekoofeh Azizi, Tao Tu, S. Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, Perry Payne, Martin Seneviratne, Paul Gamble, Chris Kelly, Abubakr Babiker, Nathanael Schärli, Aakanksha Chowdhery, Philip Mansfield, Dina Demner-Fushman, and 13 others. 2023. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180.
- Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Mohamed Amin, Le Hou, Kevin Clark, Stephen R. Pfohl, Heather Cole-Lewis, Darlene Neal, Qazi Mamunur Rashid, Mike Schaeckermann, Amy Wang, Dev Dash, Jonathan H. Chen, Nigam H. Shah, Sami Lachgar, Philip Andrew Mansfield, and 16 others. 2025. Toward expert-level medical question answering with large language models. *Nature Medicine*, 31(3):943–950.
- Wei Wang, Zhaowei Li, Qi Xu, YiQing Cai, Hang Song, Qi Qi, Ran Zhou, Zhida Huang, Tao Wang, and Li Xiao. 2025. QCRD: Quality-guided contrastive rationale distillation for large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 14334–14345.
- Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyang Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhui Chen. 2024. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems*, volume 37.
- Sibo Wei, Xueping Peng, Yifei Wang, Tao Shen, Jiasheng Si, Weiye Zhang, Fa Zhu, Athanasios V. Vasilakos, Wenpeng Lu, Xiaoming Wu, and Yinglong Wang. 2025. BianCang: A traditional chinese medicine large language model. *IEEE Journal of Biomedical and Health Informatics*, pages 1–12.
- Chaoyi Wu, Weixiong Lin, Xiaoman Zhang, Ya Zhang, Yanfeng Wang, and Weidi Xie. 2024. PMC-LLaMA: Toward building open-source language models for medicine. *Journal of the American Medical Informatics Association*, 31(9):1833–1843.
- Mengzhou Xia, Sathika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024. LESS: Selecting influential data for targeted instruction tuning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 54104–54132.
- Jia Zhang, Chen-Xi Zhang, Yao Liu, Yi-Xuan Jin, Xiao-Wen Yang, Bo Zheng, Yi Liu, and Lan-Zhe Guo. 2025a. D3: Diversity, difficulty, and dependability-aware data selection for sample-efficient LLM instruction tuning. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pages 8348–8356.
- Jinshuo Zhang, Xiaoding Zhou, Weiye Zhang, Guoqiang Chen, Ying Lian, Xiaoyang Meng, Yonghe Chen, Hongjiao Guan, Jiasheng Si, and Wenpeng Lu. 2026. D²-RAG: Dual-decision retrieval-augmented generation via multi-dimensional uncertainty and utility-aware decoding. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 15355–15382, San Diego, California, United States. Association for Computational Linguistics.
- Jipeng Zhang, Yaxuan Qin, Renjie Pi, Weizhong Zhang, Rui Pan, and Tong Zhang. 2025b. TAGCOS: Task-agnostic gradient clustered coreset selection for instruction tuning data. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4686–4701.
- Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. LIMA: Less is more for alignment. In *Advances in Neural Information Processing Systems*, volume 36.
- Xuekai Zhu, Biqing Qi, Kaiyan Zhang, Xinwei Long, Zhouhan Lin, and Bowen Zhou. 2024. PaD: Program-aided distillation can teach small models reasoning better than chain-of-thought fine-tuning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2571–2597.

Appendix Contents

A	Implementation Details	12
A.1	Verifier Specification	12
A.2	Dataset Configuration	12
B	Efficiency Evaluation	13
B.1	Cost Accounting Protocol	13
B.2	Stage-level Offline Construction Cost	14
B.3	Final Student-side Fine-tuning Efficiency	14
B.4	Matched-budget Controls	14
B.5	Deterministic Control Evaluation	15
B.6	Accuracy-cost Trade-off against Related Methods	15
B.7	Multi-seed Stability on the Main Benchmarks	15
C	Hyperparameter Sensitivity	16
C.1	Balanced Distillation	16
D	Sensitivity Analysis of Learnability Filtering Thresholds	17
D.1	Learnability Filtering Thresholds	17
D.2	Repeated-Inference Controls	18
E	Rationale Quality Assessment	19
E.1	Rationale Compression	19
E.2	Physician Evaluation	19
F	Generalization Analysis	19
F.1	Student Scale	19
F.2	Evidence Retention	20
F.3	Teacher Model Family	21
F.4	Weakness Repair	22

A Implementation Details

A.1 Verifier Specification

We use GPT-4o as the automatic verifier for rationale-consistency scoring. The verifier is independent of the open-source teacher models used for rationale generation and semantic augmentation. It is used only during offline quality control. It assigns a scalar rationale-consistency score to each teacher-generated rationale. Its outputs are not used as student training targets. The same prompt and rubric are used for all datasets and student models.

We use the scalar verifier score as $s_{\text{reason}}(z_i)$ in Eq. 4. This score corresponds to the JSON field `reasoning_consistency_score`. A teacher-generated rationale is retained only when three conditions are satisfied. First, it must pass the deterministic answer-correctness check. Second, it must pass the deterministic format-validity check. Finally, it must satisfy $Q_{\text{qc}}(z_i) \geq \gamma_{\text{qc}}$. In our experiments, $\gamma_{\text{qc}} = 0.55$. The field `brief_justification` is used only for auditing

and error analysis. It is not used as a student training target.

The verifier provides an automatic proxy for rationale quality, not clinical expert validation. Its score should therefore be interpreted as a filter for inconsistent, poorly grounded, or non-discriminative teacher rationales. It should not be treated as conclusive evidence of clinically faithful reasoning.

Table 3: Robustness to verifier choice. Pass denotes the percentage of samples retained after quality control. Overlap denotes retained-sample overlap with GPT-4o. ρ/τ denotes Spearman and Kendall correlation with GPT-4o scores.

Verifier	Pass	Overlap	ρ/τ	Acc.
GPT-4o	61.8	100.0	1.00/1.00	52.97
Llama-70B	59.6	86.4	0.81/0.64	52.74
BioMistral-7B	57.9	82.7	0.76/0.58	52.58
PMC-LLaMA-13B	63.2	84.1	0.78/0.60	52.83

Verifier robustness. Because LEAD uses GPT-4o as an automatic rationale-consistency verifier, we examine its sensitivity to verifier choice. We replace GPT-4o with three alternative verifiers. They include Llama-3.1-70B-Instruct as an open general verifier. They also include BioMistral-7B as a biomedical verifier. PMC-LLaMA-13B is used as a medical-domain verifier. All verifiers use the same scoring rubric and quality-control threshold. The deterministic answer-correctness and format-validity checks remain unchanged. Only the verifier model for $s_{\text{reason}}(z_i)$ is replaced.

As shown in Table 3, the alternative verifiers produce retention behavior similar to GPT-4o. Their retained-sample overlap with GPT-4o ranges from 82.7% to 86.4%. Their score correlations with GPT-4o remain moderate to high. Spearman correlations range from 0.76 to 0.81. Kendall correlations range from 0.58 to 0.64. The final student accuracy also remains stable across verifiers. Replacing GPT-4o changes macro accuracy from 52.97% to 52.58–52.83%. These results suggest that LEAD’s gains are not an artefact of a single closed-source verifier. Automatic verification should still be interpreted as a proxy for rationale quality. It should not be treated as clinical expert validation.

A.2 Dataset Configuration

Table 4 summarises the dataset usage in our experiments. For benchmarks with official training splits,

Rationale-Consistency Verifier Prompt

You are a strict medical rationale verifier. Your task is to evaluate the rationale consistency of a teacher-generated explanation for the given gold answer.

The answer correctness check and format-validity check are implemented separately. Do not judge whether the output format is valid. Do not decide whether the teacher-predicted answer matches the gold answer. Focus only on whether the rationale provides coherent, medically plausible, and answer-discriminative support for the gold answer.

Input: a medical multiple-choice question, answer options, the gold answer, the teacher-predicted answer, and the teacher-generated rationale.

Instruction: Evaluate whether the rationale supports the gold answer. Penalize rationales that are vague, internally inconsistent, medically implausible, contradictory to the question, unsupported by the given evidence, or unable to distinguish the gold answer from plausible distractors.

Question: {question}

Options: {options}

Gold answer: {gold_answer}

Teacher-predicted answer: {teacher_answer}

Teacher-generated rationale: {rationale}

Evaluation criteria: (1) *Internal coherence*: whether the reasoning is coherent, non-contradictory, and logically connected; (2) *Medical plausibility*: whether the clinical claims are medically plausible and consistent with standard medical knowledge; (3) *Gold-answer alignment*: whether the rationale supports the gold answer rather than another option.

Return only a JSON object:

```
{
  "reasoning_consistency_score": a number between 0 and 1,
  "brief_justification": "one short sentence"
}
```

Scoring guidance: 0.90–1.00: coherent, medically plausible, evidence-grounded, and clearly discriminative; 0.70–0.89: mostly coherent and plausible, with minor omissions; 0.55–0.69: partially useful but incomplete or weakly discriminative; 0.30–0.54: substantially flawed, weakly grounded, or weakly aligned with the gold answer; 0.00–0.29: contradictory, medically implausible, unsupported, or primarily supportive of a wrong option.

Figure 4: Prompt used for rationale-consistency scoring. The scalar field `reasoning_consistency_score` is used as $s_{\text{reason}}(z_i)$ in Eq. (4). Answer correctness and structural validity are checked separately by deterministic rules.

all trainable methods use the same 10k-example subset sampled from the corresponding training split. The test sets are used only for final evaluation. MMLU-Pro-Med is used as a held-out transfer benchmark because MMLU-Pro does not provide an official training split. Results for trainable methods are obtained by evaluating MedQA-trained checkpoints directly on MMLU-Pro-Med.

Dataset	Avail. Train	Source Train	Test
MedQA	10,178	10,000	1,273
MedMCQA	182,822	10,000	6,150
CMEexam	54,198	10,000	6,780
MMLU-Pro-Med	–	–	1,535

Table 4: Source Train denotes the training source used by trainable methods; MMLU-Pro-Med uses MedQA-trained checkpoints for held-out transfer evaluation.

B Efficiency Evaluation

This appendix clarifies the scope of LEAD’s efficiency claim. LEAD is not designed to minimise the total offline construction cost. Instead, it uses additional teacher-side processing to construct

higher-quality supervision for a compact student model. Therefore, any reduction in final student fine-tuning tokens should be interpreted as a final-stage training benefit, not as a reduction in the full offline-plus-training pipeline.

We separate three cost regimes: (i) offline supervision construction, which includes teacher generation, semantic augmentation, verifier-based quality control, rationale-compression scoring, and learnability estimation; (ii) final student fine-tuning, which trains the compact student on the constructed LEAD mixture; and (iii) deployment-time inference, which uses only the compact student. Teacher models, the verifier, step-level scoring, and the learnability module are not used at deployment time.

B.1 Cost Accounting Protocol

We use accounted tokenised operations as a device-independent cost measure. This measure includes teacher generation, semantic augmentation, verifier-based quality control, tokenised scoring for rationale compression and learnability estimation, and final student fine-tuning tokens.

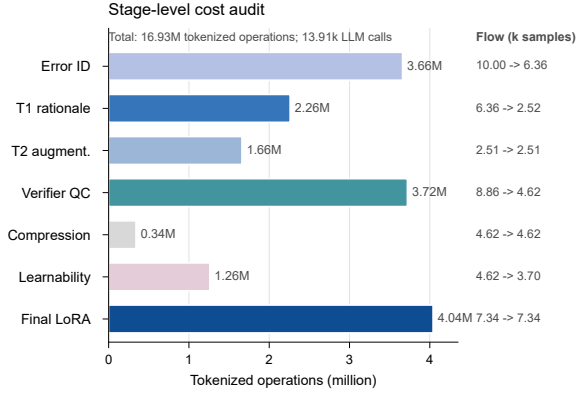


Figure 5: Stage-level cost audit for LEAD under the Qwen2.5-1.5B-Instruct student setting. Tok. denotes accounted tokenised operations, including offline supervision construction, tokenised scoring, and final student fine-tuning. Deployment uses only the compact student model and does not require teacher models, the verifier, step-level scoring, or the learnability module.

All cost analyses in this appendix use the Qwen2.5-1.5B-Instruct student setting unless otherwise stated. Accuracy denotes macro accuracy across MedQA, MedMCQA, CMExam, and MMLU-Pro-Med.

B.2 Stage-level Offline Construction Cost

Figure 5 reports a stage-level cost audit for a representative LEAD run. The full LEAD pipeline uses 16.93M accounted tokenised operations. This cost comes from the offline construction of targeted rationale supervision. Specifically, LEAD first identifies student errors, then generates and augments teacher rationales. Next, it applies verifier-based quality control, rationale-compression scoring, and learnability estimation. These stages are designed to convert raw teacher outputs into quality-checked and learnable supervision. After this construction process, deployment uses only the compact student model. Teacher models, the verifier, step-level scoring, and learnability estimation are not required during deployment.

B.3 Final Student-side Fine-tuning Efficiency

Although LEAD is more expensive in total offline-plus-training cost, it reduces the final student-side fine-tuning budget. Compared with answer-only SFT on the same 10,000-example source subset, LEAD retains 7,338 final training samples. It reduces two-epoch student fine-tuning tokens from 5.62M to 4.04M, a 28.1% reduction. At the same time, macro accuracy improves from 48.22% to

Method	Records	FT Tok.	Acc.	Correct
SFT	10,000	5.62M	48.22	8,254
LEAD	7,338	4.04M	51.52	8,753
Δ	-26.6%	-28.1%	+3.30	+499

Table 5: Final student-side fine-tuning efficiency. FT Tok. denotes two-epoch final student fine-tuning tokens and excludes offline construction cost. Acc. denotes macro accuracy across the four evaluation benchmarks.

51.52%. Table 5 summarises this final-stage comparison.

This result should not be read as evidence that LEAD is cheaper end-to-end. Rather, it shows that the final compact-student training stage becomes smaller and more accurate after offline selection, filtering, compression, and learnability estimation.

B.4 Matched-budget Controls

The comparison above uses SFT trained on the original 10,000-example source subset and LEAD trained on its filtered final mixture. To isolate whether LEAD’s gains come from the selection and rationale-processing pipeline rather than from the final training budget, we add matched-budget controls.

We consider two settings. In the matched-record setting, SFT, Answer-KD, Step-KD, and random-error distillation are trained with the same number of final records as LEAD. In the matched-token setting, each baseline is subsampled to match LEAD’s final student-side token budget. The random-error baseline uses student-error cases without LEAD’s quality control, compression, and learnability filtering.

Table 6 shows that LEAD achieves the highest macro accuracy under both controls. Under matched records, the strongest baseline is Answer-KD-7.3k, with 49.77% macro accuracy, while LEAD reaches 51.52%, a gain of 1.75 percentage points. Under matched tokens, the strongest baseline is Random-error-token, with 50.17% macro accuracy, while LEAD remains higher by 1.35 percentage points. These results indicate that LEAD’s improvements cannot be explained solely by using a different number of final training records or tokens. The learnable error-aware selection and rationale-processing pipeline provides additional benefit.

Method	Budget	Records	MedQA	MedMCQA	CMExam	MMLU-Pro-Med	Avg.	Δ vs. LEAD
Base student	–	–	36.61	42.46	56.40	39.22	43.67	-7.85
SFT-7.3k	Record	7,338	44.85	45.73	64.06	39.35	48.50	-3.02
Answer-KD-7.3k	Record	7,338	44.54	46.00	66.28	42.28	49.77	-1.75
Step-KD-7.3k	Record	7,338	43.75	45.45	63.72	44.30	49.30	-2.22
Random-error-7.3k	Record	7,338	44.07	45.21	63.50	46.19	49.74	-1.78
SFT-token	Token	7,201	45.09	46.40	64.62	41.24	49.34	-2.18
Answer-KD-token	Token	7,201	43.05	46.31	65.87	41.30	49.13	-2.39
Step-KD-token	Token	6,470	43.60	45.52	63.60	43.97	49.17	-2.35
Random-error-token	Token	6,811	43.83	46.09	64.03	46.71	50.17	-1.35
LEAD	–	7,338	46.03	50.32	64.59	45.15	51.52	0.00

Table 6: Matched-budget controls under the Qwen2.5-1.5B-Instruct student setting. Record-matched baselines use the same number of final training records as LEAD. Token-matched baselines are subsampled to match LEAD’s final student-side training-token budget. Avg. denotes macro accuracy across the four benchmarks.

B.5 Deterministic Control Evaluation

We further evaluate two intermediate controls under deterministic inference. QC-only retains student-error samples after quality control but does not apply rationale compression or learnability filtering. Matched Random Subset uses a randomly sampled student-error subset with the same training-set size as the corresponding controlled setting. Table 7 reports results for two Qwen2.5 student scales.

B.6 Accuracy-cost Trade-off against Related Methods

We further compare LEAD with related distillation and data-selection methods under the same Qwen2.5-1.5B-Instruct student setting. Table 8 reports macro accuracy and total accounted tokenised operations, including offline construction and final student fine-tuning.

SFT is the lowest-cost method because it does not perform teacher-side rationale construction, quality filtering, compression, or learnability estimation. LEAD is therefore not the cheapest method in total accounted cost. However, LEAD achieves the highest macro accuracy among the compared methods. Compared with stronger rationale- or selection-based alternatives, LEAD is more accurate than QCRD, Skip-KD, Superfiltering, and DEITA. Its total accounted budget is close to Skip-KD and lower than DEITA.

B.7 Multi-seed Stability on the Main Benchmarks

To examine whether LEAD’s performance is sensitive to random variation, we first rerun the Qwen2.5-0.5B-Instruct setting under three random seeds (42, 43, and 44) on the four main benchmarks.

Seed 42 corresponds to the main experimental run reported in Table 1. As shown in Table 9, LEAD remains stable across seeds. The macro accuracies are 35.9%, 35.7%, and 36.0% under seeds 42, 43, and 44, respectively, resulting in a standard deviation of only 0.2 percentage points. The dataset-level results also exhibit modest variation, suggesting that the reported performance is not driven by a single favorable run.

To provide broader stability evidence, we further extend the analysis to Qwen2.5-1.5B-Instruct, the primary student model used in our ablation studies, and Llama-3.2-1B-Instruct, which represents a different model family. For each model, we rerun Full LEAD and the corresponding ablation variants using the same three training seeds. Table 10 reports the dataset-level and macro accuracies as the mean and standard deviation over these seeds.

More importantly, the relative advantage of Full LEAD is consistent across random seeds. It achieves the highest macro accuracy under each of the three seeds on both Qwen2.5-1.5B-Instruct and Llama-3.2-1B-Instruct. Its average gains over the strongest ablation are 2.19 and 1.24 percentage points, respectively, which are larger than the observed across-seed standard deviations of 0.27 and 0.15. On Qwen2.5-1.5B-Instruct, removing filtering or weakness selection consistently reduces macro accuracy, while random selection also exhibits greater dataset-level variation. The same advantage over the filtering-free and random-selection variants is observed on Llama-3.2-1B-Instruct. Together with the Qwen2.5-0.5B-Instruct results, these consistent patterns suggest that LEAD’s improvements are not attributable to a favorable seed and generalize across model scales

Student	Method	MedQA	MedMCQA	CMExam	MMLU-Pro-Med	Macro
<i>Qwen2.5-0.5B-Instruct</i>	QC-only	31.80	38.40	43.10	20.30	33.40
	Matched Random Subset	29.10	34.60	40.50	17.20	30.35
<i>Qwen2.5-1.5B-Instruct</i>	QC-only	42.30	43.90	61.00	39.60	46.70
	Matched Random Subset	41.50	42.20	47.90	27.20	39.70

Table 7: Deterministic evaluation of intermediate control variants across two student scales. Macro denotes average accuracy across the four benchmarks.

Method	Macro Acc.	Total Tok.
SFT	48.22	5.62M
QCRD	49.95	12.82M
Skip-KD	50.09	16.80M
Superfiltering	46.74	11.75M
DEITA	49.90	18.23M
LEAD	51.52	16.93M

Table 8: Accuracy–cost trade-off under the Qwen2.5-1.5B-Instruct setting. Total Tok. includes offline construction and final fine-tuning.

Dataset	Seed 42	Seed 43	Seed 44	Mean $\pm$ Std.
MedQA	33.4	32.6	33.5	33.2 $\pm$ 0.5
MedMCQA	38.8	39.0	39.5	39.1 $\pm$ 0.4
CMExam	45.6	46.5	46.8	46.3 $\pm$ 0.6
MMLU-Pro-Med	25.9	24.7	24.1	24.9 $\pm$ 0.9
Macro	35.9	35.7	36.0	35.9 $\pm$ 0.2

Table 9: Multi-seed accuracy (%) of LEAD with Qwen2.5-0.5B-Instruct. Seed 42 is the main run, and Macro averages the four benchmarks.

and families.

C Hyperparameter Sensitivity

C.1 Balanced Distillation

We conduct a sensitivity analysis of balanced distillation using Qwen2.5-1.5B-Instruct as the student. We vary the keep ratio $r_{\text{keep}} = |\mathcal{D}_{\text{keep}}|/|\mathcal{D}_{\text{learn}}|$ and the loss weight λ , and report results on MedQA, MedMCQA, and CMExam. Macro denotes the unweighted average over the three benchmarks, and we include only configurations for which all three results are available.

As shown in Table 11, balanced distillation consistently improves over the base student across all complete configurations. The base student obtains a Macro accuracy of 45.16%, while balanced-distillation variants achieve Macro accuracies between 52.94% and 54.23%, corresponding to absolute gains of 7.78–9.07 percentage points. These results indicate that the improvement transfers across

(a) Qwen2.5-1.5B-Instruct				
Dataset	Full LEAD	w/o Filt.	Random	w/o Weak.
MedQA	46.37 $\pm$ 0.32	44.70 $\pm$ 0.31	43.81 $\pm$ 0.43	44.46 $\pm$ 0.21
MedMCQA	50.31 $\pm$ 0.16	46.11 $\pm$ 0.17	45.73 $\pm$ 0.17	45.25 $\pm$ 0.18
CMExam	64.59 $\pm$ 0.19	63.77 $\pm$ 0.19	63.56 $\pm$ 0.27	64.96 $\pm$ 0.63
MMLU-Pro-Med	45.93 $\pm$ 0.72	43.87 $\pm$ 0.53	43.87 $\pm$ 1.81	39.04 $\pm$ 0.95
Macro	51.80 $\pm$ 0.27	49.61 $\pm$ 0.05	49.24 $\pm$ 0.53	48.43 $\pm$ 0.18

(b) Llama-3.2-1B-Instruct			
Dataset	Full LEAD	w/o Filt.	Random
MedQA	39.86 $\pm$ 0.58	39.75 $\pm$ 0.28	38.65 $\pm$ 0.80
MedMCQA	46.23 $\pm$ 0.46	42.39 $\pm$ 0.19	42.23 $\pm$ 0.06
CMExam	38.27 $\pm$ 0.22	38.43 $\pm$ 0.10	38.18 $\pm$ 0.29
MMLU-Pro-Med	35.63 $\pm$ 0.53	34.46 $\pm$ 0.17	34.70 $\pm$ 0.10
Macro	40.00 $\pm$ 0.15	38.76 $\pm$ 0.18	38.44 $\pm$ 0.14

Table 10: Multi-seed accuracy on two additional student models. Values are mean $\pm$ standard deviation over seeds 42, 43, and 44; Macro averages the four benchmarks. Best results are in bold.

both English and Chinese medical QA benchmarks.

For the keep-ratio sweep with $\lambda = 0.5$, using only learnable-error samples already improves Macro accuracy to 53.09%. Adding retention samples further improves performance, reaching 53.51% at $r_{\text{keep}} = 0.5$ and 53.65% at $r_{\text{keep}} = 1.0$. Increasing the ratio to 2.0 gives a similar but slightly lower Macro accuracy of 53.59%, suggesting that moderate retention helps preserve existing capability, while excessive retention may dilute the error-correction signal.

For the λ sweep with $r_{\text{keep}} = 1.0$, Macro accuracy increases from 52.99% at $\lambda = 0.3$ to 54.23% at $\lambda = 0.9$. The expanded grid shows that balanced distillation remains stable across nearby configurations, while the best result in this sensitivity setting is obtained with a larger weight on learnable-error supervision.

In the main experiments, we use $r_{\text{keep}} = 1.0$ and $\lambda = 0.5$ for consistency with the primary experimental protocol. This default setting improves the base student from 45.16% to 53.65% Macro accuracy, while the $\lambda = 0.9$ result is reported only as an additional sensitivity point rather than the default

configuration.

D Sensitivity Analysis of Learnability Filtering Thresholds

D.1 Learnability Filtering Thresholds

We further examine the sensitivity of the learnability filtering thresholds τ_l and τ_h on MedQA using Qwen2.5-1.5B-Instruct. All other components remain fixed, including the teacher, quality control, rationale compression, LoRA configuration, and distillation hyperparameters. Here, ΔNLL denotes the NLL reduction after lightweight LoRA calibration. We test whether the default interval is robust and whether intermediate- ΔNLL samples provide a better stability-plasticity trade-off than extreme samples.

Threshold sensitivity. First, we vary the retained ΔNLL interval from loose to aggressive filtering. Table 12 reports the retained sample count and resulting accuracy. The default 10–90 interval achieves the highest accuracy of 45.09% while retaining 79.95% of QC-passed samples. Compared with no filtering, it removes 20.05% of samples and improves accuracy by 0.47 percentage points. The nearby 5–95 interval also slightly outperforms the no-filtering baseline. However, narrower intervals progressively reduce accuracy, suggesting that aggressive filtering discards useful training samples. Overall, the results support a moderate interval and indicate that the default setting is robust to nearby threshold changes.

Fixed-budget threshold-shift control. Changing τ_l and τ_h also changes the number of retained samples, which confounds threshold-location effects with sample-quantity effects. To control for this factor, we fix the budget at 1,978 QC-passed samples, matching the default 10–90 setting. We compare three equal-width windows: 0–80, 10–90, and 20–100. The 0–80 interval retains the lower- ΔNLL region, whereas the 20–100 interval retains the upper region. We also include a random 80% subset as a control. Repair counts initially wrong predictions that become correct, while regression counts initially correct predictions that become wrong. Net repair is the difference between repaired and regressed predictions.

Table 13 shows that the 10–90 interval achieves the highest accuracy and repair rate, together with the lowest regression rate. It produces 105 net repairs, compared with 92, 87, and 88 for the other

three variants. Since all variants use the same budget, this advantage cannot be attributed to retaining more samples. Instead, the results show that the selected ΔNLL region directly affects performance. Retaining either extreme yields a weaker repair-regression balance, while removing both extremes gives the strongest overall result.

Equal-budget ΔNLL region diagnostic. Finally, we compare low-, intermediate-, and high- ΔNLL regions under an equal budget of 494 QC-passed samples. We select the bottom 20%, middle 40–60%, and top 20% of the distribution. We also include a random 20% subset as a control.

Table 14 shows that the intermediate region achieves the highest accuracy and largest net repair. The bottom 20% obtains the lowest accuracy and repair rate. This result indicates that very small ΔNLL values provide limited learning signals. The top 20% achieves the highest repair rate but also causes the highest regression rate. Thus, very large ΔNLL values produce stronger but less stable updates. The intermediate region provides a better balance between error repair and prediction preservation. Although random selection yields the lowest regression rate, its weaker repair performance reduces both accuracy and net repair.

Together, these experiments provide empirical support for the learnability filtering heuristic in LEAD. However, they do not establish ΔNLL as a theoretically optimal measure of sample utility. Instead, LEAD treats ΔNLL as an operational signal of sample responses to lightweight calibration. Small values indicate weak responses, while very large values indicate stronger but less stable responses. On MedQA with Qwen2.5-1.5B-Instruct, the default 10–90 interval achieves the strongest overall result. It also remains competitive under nearby thresholds and outperforms the equal-budget alternatives. Overall, the region diagnostic supports selecting intermediate- ΔNLL samples to balance error repair and prediction preservation.

Sensitivity to upstream quality-control and compression thresholds. We also examine the quality-control threshold γ_{qc} and the compression hyperparameter ρ on MedQA. Table 15 shows that moderate γ_{qc} values produce similar accuracy. However, $\gamma_{qc} = 0.75$ removes many samples and substantially reduces performance. Accuracy varies by only 0.31 percentage points across the tested ρ values. This result indicates limited sensitivity to ρ within the evaluated range.

Setting	r_{keep}	λ	MedQA	MedMCQA	CMExam	Macro	Δ Macro
Base student	–	–	36.61	42.46	56.40	45.16	–
Keep-ratio / grid region: $r_{\text{keep}} = 0.0$							
Grid sweep	0.0	0.3	46.50	49.60	63.17	53.09	+7.93
Keep-ratio sweep	0.0	0.5	46.42	49.70	63.16	53.09	+7.93
Grid sweep	0.0	0.7	46.26	49.74	63.07	53.02	+7.86
Grid sweep	0.0	0.9	46.26	49.43	63.14	52.94	+7.78
Grid sweep	0.0	1.0	46.26	49.84	63.10	53.07	+7.91
Keep-ratio / grid region: $r_{\text{keep}} = 0.5$							
Grid sweep	0.5	0.3	45.56	50.34	63.44	53.11	+7.95
Keep-ratio sweep	0.5	0.5	46.58	50.15	63.79	53.51	+8.35
Grid sweep	0.5	0.7	46.34	50.56	63.66	53.52	+8.36
Grid sweep	0.5	0.9	46.26	51.32	63.31	53.63	+8.47
λ sweep at $r_{\text{keep}} = 1.0$							
λ sweep	1.0	0.3	45.63	49.43	63.91	52.99	+7.83
Main setting	1.0	0.5	46.03	50.32	64.59	53.65	+8.49
λ sweep	1.0	0.7	46.89	50.08	64.46	53.81	+8.65
λ sweep	1.0	0.9	47.75	50.61	64.32	54.23	+9.07
Keep-ratio extension: $r_{\text{keep}} = 2.0$							
Keep-ratio sweep	2.0	0.5	46.34	50.10	64.32	53.59	+8.43

Table 11: Sensitivity analysis of balanced distillation. r_{keep} denotes $|\mathcal{D}_{\text{keep}}|/|\mathcal{D}_{\text{learn}}|$. All task results are accuracies in percentage points. Macro denotes the unweighted average over MedQA, MedMCQA, and CMExam. Δ Macro reports the absolute improvement over the base student. Only configurations with all three dataset reports are included in the main comparison. The main experiments use $r_{\text{keep}} = 1.0$ and $\lambda = 0.5$; the other configurations are reported for sensitivity analysis.

Variant	Samples	Ratio (%)	Acc. (%)
No filtering	2474	100.00	44.62
5–95	2226	89.98	44.85
10–90	1978	79.95	45.09
20–80	1484	59.98	43.05
30–70	990	40.02	41.40
40–60	494	19.97	41.01

Table 12: Sensitivity to the learnability filtering thresholds on MedQA. Moderate filtering performs best, while aggressive filtering reduces accuracy.

Variant	Acc. (%)	Repair (%)	Regr. (%)	Net
0–80	44.07	20.40	15.35	92
10–90	45.09	21.64	14.71	105
20–100	43.68	19.53	14.93	87
Random 80%	43.75	20.40	16.20	88

Table 13: Fixed-budget threshold-shift control on MedQA. All variants retain 1,978 samples. Regr. denotes the regression rate, and Net denotes net repair.

D.2 Repeated-Inference Controls

To examine whether single-pass errors persist, we conduct a repeated-inference audit on the 1,273-example MedQA test set using Qwen2.5-1.5B-Instruct. All post-training checkpoints use training seed 42 and are trained for two epochs with AdamW, a learning rate of 2×10^{-5} , and LoRA with rank 64, scaling factor 128, and dropout 0.05.

For each checkpoint, we perform five stochas-

Variant	Acc. (%)	Repair (%)	Regr. (%)	Net
0–20	38.81	13.18	17.27	25
40–60	41.01	17.16	18.12	53
80–100	40.46	18.03	21.11	46
Random 20%	40.38	14.68	15.57	45

Table 14: Equal-budget diagnostic over different Δ NLL regions. All variants retain 494 samples. The intermediate region achieves the highest accuracy and net repair.

γ_{qc}	QC (%)	Acc. (%)	ρ	Steps (%)	Acc. (%)
0.35	100.00	43.99	0.10	99.95	43.91
0.45	100.00	43.75	0.15	99.95	43.68
0.55	100.00	43.91	0.20	99.92	43.83
0.65	93.98	44.38	0.25	99.79	43.99
0.75	38.92	40.77	0.30	99.42	43.68

Table 15: Sensitivity to γ_{qc} and ρ on MedQA. QC denotes the quality-control pass ratio, while Steps denotes the retained-step ratio.

tic inference runs using temperature 0.7, top- p 0.9, a maximum generation length of 96 tokens, and seeds 42–46. Prompts, option order, and answer extraction remain fixed. An example is a *persistent error* if at least four runs are incorrect and is *stably correct* if at least four runs are correct. A stable repair changes a persistent error into a stably correct prediction, whereas a stable regression denotes the reverse transition. We use 1,000 paired bootstrap repetitions for 95% confidence intervals and exact McNemar tests with Holm correction for paired

comparisons.

We compare raw error distillation, its record-matched subset, QC-only error distillation, LEAD using only selected error records, and Full LEAD. The record-matched raw baseline and LEAD-Error-Only use 1,978 error records with similar fine-tuning budgets. Full LEAD additionally includes 1,978 retention records.

Table 16 shows that error-only distillation improves error correction but causes substantial regression on previously correct predictions. Neither quality control nor learnability-based selection alone resolves this trade-off. Adding retention samples markedly improves behavior preservation, making Full LEAD the only method with positive net stable repair. Thus, the main benefit arises from balancing error correction with capability retention rather than from targeting errors alone.

E Rationale Quality Assessment

E.1 Rationale Compression

To examine whether the rationale compression module removes redundant teacher rationale steps, we conduct an additional audit on long teacher-generated rationales. We sample 151 student-error cases and instruct the teacher model to generate long rationales with 8–12 rationale steps. The generated rationales are then parsed, quality-controlled, and compressed using the same step-level scoring procedure described in Section 3.4.

Among the 151 generated rationales, all outputs are successfully parsed, 68 receive the correct teacher answer, and 67 pass the quality-control gate. We report compression statistics on the quality-passed rationales because these samples can be used as valid rationale supervision for distillation. As shown in Table 17, aggressive compression substantially reduces rationale length. The average number of tokens decreases from 260.4 to 56.6, and the average number of rationale steps decreases from 10.76 to 2.57. This corresponds to a 78.27% token reduction while preserving a compact set of answer-relevant rationale steps.

For completeness, we also report an extreme compression setting. Although extreme compression achieves a larger token reduction, it retains fewer rationale steps and may remove useful intermediate evidence. We therefore treat it as a stress-test setting rather than the default configuration.

E.2 Physician Evaluation

To further examine the medical reliability of teacher-generated rationales, we conducted a physician audit on a randomly sampled subset of medical QA items. The audit contains 100 paired questions, with 100 rationales from each teacher model. A physician annotated whether each rationale supports the gold answer, contains medical factual errors, includes misleading or potentially unsafe reasoning, omits key differential points, and is clinically acceptable. We report proportions with 95% Wilson confidence intervals.

The audit shows that most teacher-generated rationales support the gold answer and are clinically acceptable under physician review. However, it also reveals non-negligible medical factual errors and unsafe or misleading reasoning. Therefore, we treat teacher rationales as useful but imperfect supervision signals rather than clinical diagnostic evidence.

F Generalization Analysis

F.1 Student Scale

To examine whether LEAD remains effective beyond very compact student models, we conduct an additional student-scaling analysis. We use Qwen2.5-14B-Instruct as the teacher and evaluate four Qwen2.5 student models: Qwen2.5-0.5B-Instruct, Qwen2.5-1.5B-Instruct, Qwen2.5-3B-Instruct, and Qwen2.5-7B-Instruct. This controlled same-family setting keeps the model family fixed while varying the student scale, reducing confounding factors from architecture, tokenizer, and pretraining differences. We also include Llama-3.2-1B-Instruct as an additional cross-family sanity check. We evaluate all students on MedQA, CMExam, and MMLU-Pro-Med, and compare the original student baseline with the LEAD-enhanced student under the same training and evaluation protocol.

As shown in Figure 6, LEAD consistently improves all Qwen2.5 student scales. The macro-average gains are +7.42, +8.42, +5.26, and +3.44 points for the Qwen2.5-0.5B-Instruct, Qwen2.5-1.5B-Instruct, Qwen2.5-3B-Instruct, and Qwen2.5-7B-Instruct students, respectively. Although the improvement becomes smaller as the student model becomes stronger, which is expected because larger students leave less room for correction, LEAD still improves the 7B student on all three datasets. In particular, Qwen2.5-7B-Instruct improves by +6.18

Table 16: Repeated-inference stability audit on MedQA. Acc., Repair, Regr., and Net are averaged over five paired inference runs. S-Repair, S-Regr., and S-Net denote stable repair, stable regression, and net stable repair. All performance values are percentages.

Method	Records	FT tokens	Acc. ↑	Repair ↑	Regr. ↓	Net ↑	S-Repair ↑	S-Regr. ↓	S-Net ↑
Error-KD-Raw	2,474	525,041	36.64	24.97	42.09	1.26	6.55	16.26	-0.86
Error-KD-Raw-Matched	1,978	421,795	34.67	22.17	42.32	-0.63	4.41	15.72	-1.96
Error-KD-QC	2,474	525,041	35.06	22.41	41.87	-0.31	4.81	18.16	-2.44
LEAD-Error-Only	1,978	420,078	35.77	25.58	45.88	0.31	4.55	21.68	-3.61
Full LEAD	3,956	836,711	42.30	23.75	23.61	6.99	10.70	9.49	3.53

Compression setting	Number of tokens		Number of steps	
	Raw	Compressed	Raw	Retained
Aggressive	260.4	56.6	10.76	2.57
Extreme	260.4	39.2	10.76	1.54

Table 17: Compression audit on 67 quality-passed long teacher rationales. Aggressive compression reduces tokens by 78.27%, while extreme compression reduces tokens by 84.95%.

Student	MedQA	CMEExam	MMLU-Pro-med	Macro
Qwen2.5-0.5B-Instruct	28.65->34.52 (+5.87)	38.54->46.47 (+7.93)	17.00->25.45 (+8.45)	28.06->35.48 (+7.42)
Qwen2.5-1.5B-Instruct	36.61->47.56 (+10.95)	56.40->63.98 (+7.58)	39.22->45.97 (+6.75)	44.08->52.50 (+8.42)
Qwen2.5-3B-Instruct	42.26->48.94 (+6.68)	66.11->71.44 (+5.33)	47.43->51.21 (+3.78)	51.93->57.20 (+5.26)
Qwen2.5-7B-Instruct	52.71->58.89 (+6.18)	79.66->80.23 (+0.57)	60.65->64.23 (+3.58)	64.34->67.78 (+3.44)
Llama-3.2-1B-Instruct	33.31->37.39 (+4.08)	32.60->37.60 (+5.00)	19.35->34.98 (+15.63)	28.42->36.66 (+8.24)

Figure 6: Student-scale analysis with Qwen2.5-14B-Instruct as the teacher. Each cell reports baseline accuracy → LEAD accuracy, with the absolute gain shown in parentheses. The first four rows form the controlled Qwen2.5 same-family scaling analysis, while Llama-3.2-1B-Instruct provides an additional cross-family sanity check. Macro denotes the average accuracy across MedQA, CMEExam, and MMLU-Pro-Med.

points on MedQA and +3.58 points on MMLU-Pro-Med, with a smaller but still positive gain on CMEExam. The additional Llama-3.2-1B-Instruct result also shows a positive macro-average gain of +8.24 points, suggesting that the observed improvement is not restricted to a single student model family. These results indicate that LEAD is not limited to very compact students and remains useful in stronger open-weight student settings.

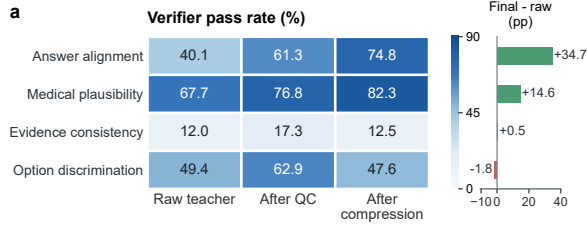
Metric	Model	Value
Factual error	Qwen	27% (19.3–36.4)
	Huatuo	33% (24.6–42.7)
Full support	Qwen	77% (67.8–84.2)
	Huatuo	81% (72.2–87.5)
Partial support	Qwen	15% (9.3–23.3)
	Huatuo	16% (10.1–24.4)
No support	Qwen	8% (4.1–15.0)
	Huatuo	3% (1.0–8.5)
Misleading/unsafe	Qwen	20% (13.3–28.9)
	Huatuo	17% (10.9–25.5)
Potentially dangerous	Qwen	1% (0.2–5.4)
	Huatuo	1% (0.2–5.4)
Missing differential	Qwen	21% (14.2–30.0)
	Huatuo	17% (10.9–25.5)
Major issue	Qwen	19% (12.5–27.8)
	Huatuo	16% (10.1–24.4)
No obvious issue	Qwen	67% (57.3–75.4)
	Huatuo	66% (56.3–74.5)
Clinically acceptable	Qwen	81% (72.2–87.5)
	Huatuo	84% (75.6–89.9)

Table 18: Physician audit of teacher-generated rationales on 100 randomly sampled paired medical QA items. Qwen denotes Qwen2.5-7B-Instruct; Huatuo denotes HuatuoGPT-o1-7B. Values are percentages with 95% Wilson confidence intervals.

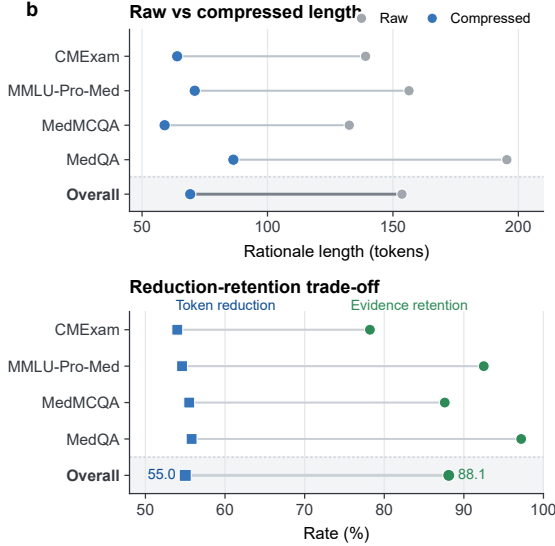
F.2 Evidence Retention

We conduct an additional rationale-quality audit to examine automatic proxies for the quality and compactness of teacher-generated rationale supervision. For sampled instances from the four benchmarks, we evaluate three stages of the LEAD pipeline: raw teacher rationales, quality-controlled rationales, and compressed rationales. We also assess whether the compressed rationales preserve the key clinical evidence required to justify the gold answers.

As shown in Figure 7, quality control improves the proxy quality of teacher rationales compared with raw teacher outputs. Verifier scores increase across all four evaluated dimensions after quality control, indicating that the quality-control gate removes many outputs with answer mismatches, invalid structures, or inconsistent rationale content. These results support the role of quality control in reducing low-quality teacher supervision before



(a) Verifier-based rationale quality scores



(b) Compression and evidence retention by dataset

Figure 7: Proxy rationale-quality and evidence-retention analysis. Panel (a) reports verifier pass rates for answer alignment, medical plausibility, evidence consistency, and option discrimination across raw teacher rationales, quality-controlled rationales, and compressed rationales. Panel (b) shows dataset-level compression statistics and evidence-retention rates after rationale compression.

distillation.

After compression, answer alignment and medical plausibility further improve, whereas evidence consistency and option discrimination decrease relative to the quality-controlled rationales. Together with the substantial reduction in rationale length, this pattern indicates a compactness-completeness trade-off: compression removes redundant content and retains a substantial portion of the key clinical evidence, but may also discard contextual or explanatory details needed to connect the evidence to the answer and distinguish it from alternative options.

Overall, these results suggest that LEAD improves several proxy quality dimensions, particularly answer alignment, medical plausibility, and compactness. However, these proxy improvements do not by themselves establish clinically faithful

reasoning in the resulting student model. The decreases in evidence consistency and option discrimination highlight the need to balance concise supervision against explanatory completeness.

F.3 Teacher Model Family

In addition to evaluating different teacher model families, this appendix also includes same-teacher controls, where both teacher roles in LEAD are instantiated with the same model, such as Qwen2.5-7B-Instruct only or Llama-3.1-8B-Instruct only. These settings remove cross-family teacher complementarity and help assess whether LEAD’s gains rely solely on using two heterogeneous teachers.

LEAD employs two teacher roles for constructing medical rationale supervision: one teacher generates candidate rationales for student error cases, and the other performs semantics-preserving sample expansion. This design raises a practical question: should the two roles use the same model, or can different teacher models provide stronger supervision?

To answer this question, we fix the student model as Qwen2.5-1.5B-Instruct and compare three teacher configurations. The first uses Qwen2.5-7B-Instruct for both teacher roles, denoted as the Qwen-only setting. The second uses Llama-3.1-8B-Instruct for both roles, denoted as the Llama-only setting. The third follows our main experimental setting, where Qwen2.5-7B-Instruct and HuatuoGPT-o1-7B are used as a mixed-teacher configuration. Qwen2.5 and Llama belong to different model families, while HuatuoGPT-o1 is a medical-domain model based on Qwen2.5-7B-Instruct.

As shown in Figure 8, all teacher configurations improve the student over the no-teacher baseline across the evaluated benchmarks. The no-teacher Qwen2.5-1.5B-Instruct student obtains 36.61, 42.46, 56.40, and 39.22 on MedQA, MedMCQA, CMExam, and MMLU-Pro-Med, respectively. The Qwen-only setting yields gains of 7.5, 4.0, 6.6, and 3.5 percentage points on the four benchmarks, while the Llama-only setting yields gains of 9.4, 4.4, 6.8, and 5.3 percentage points. These results show that useful teacher supervision is not restricted to the same model family as the student.

Compared with the same-model settings, the mixed-teacher configuration used in our main experiments achieves the strongest overall performance. It obtains 46.03 on MedQA, 50.32 on

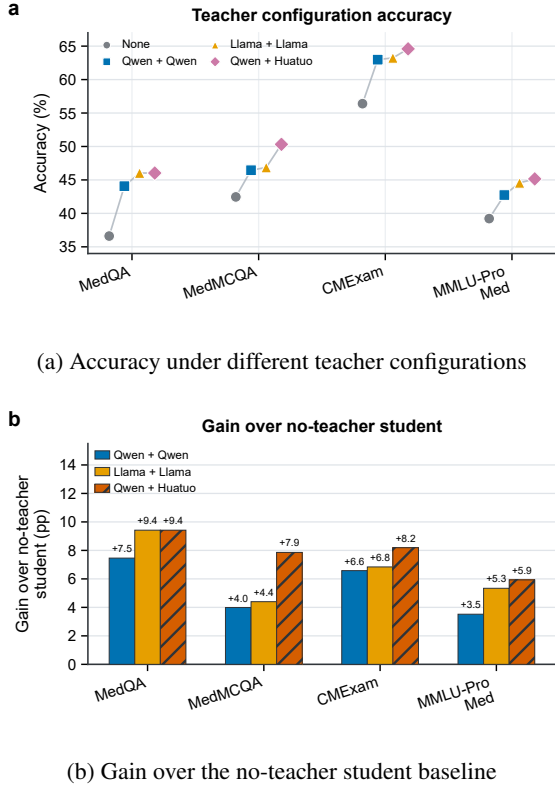


Figure 8: Effect of teacher model configuration. The student model is fixed as Qwen2.5-1.5B-Instruct. Qwen-only and Llama-only use the same teacher model for both teacher roles, while the mixed-teacher setting uses Qwen2.5-7B-Instruct and HuatuoGPT-o1-7B for different teacher roles.

MedMCQA, 64.59 on CMExam, and 45.15 on MMLU-Pro-Med, corresponding to gains of 9.4, 7.9, 8.2, and 5.9 percentage points over the no-teacher baseline. It ties with the best same-model setting on MedQA and performs best on MedMCQA, CMExam, and MMLU-Pro-Med. These results indicate that assigning different teacher models to different teacher roles is more effective than using the same teacher model twice. We attribute this to complementarity between a general instruction-following teacher and a medical-domain rationale teacher: the former provides broad instruction-following ability, while the latter supplies more domain-specific rationale supervision.

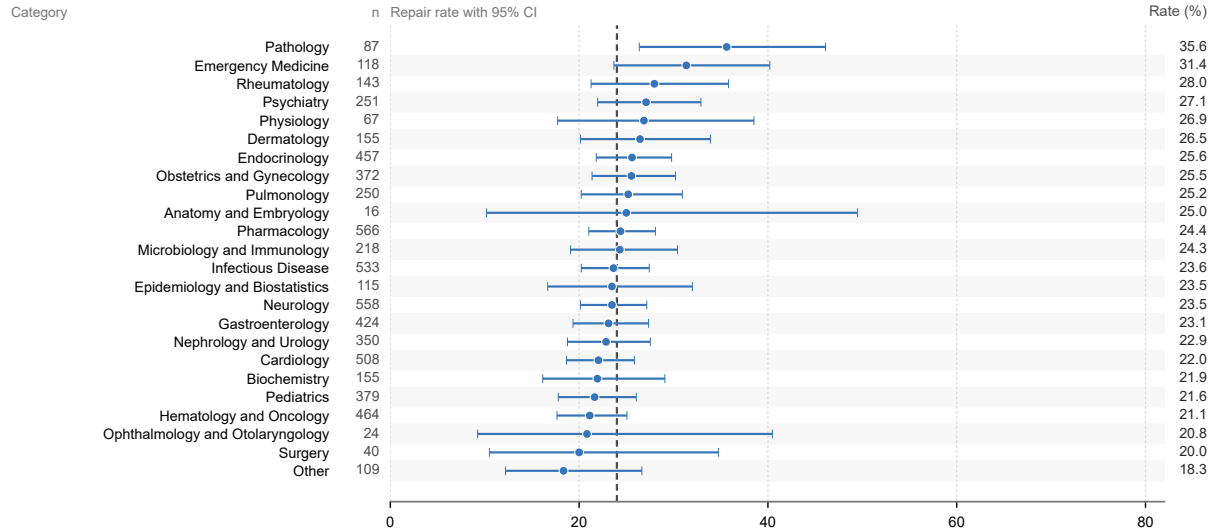
Overall, the results suggest that LEAD is not restricted to same-family teacher-student pairs. They also indicate that mixed-teacher configurations can be more beneficial than same-model configurations in our dual-teacher framework, especially when the teachers offer complementary general and medical-domain capabilities.

F.4 Weakness Repair

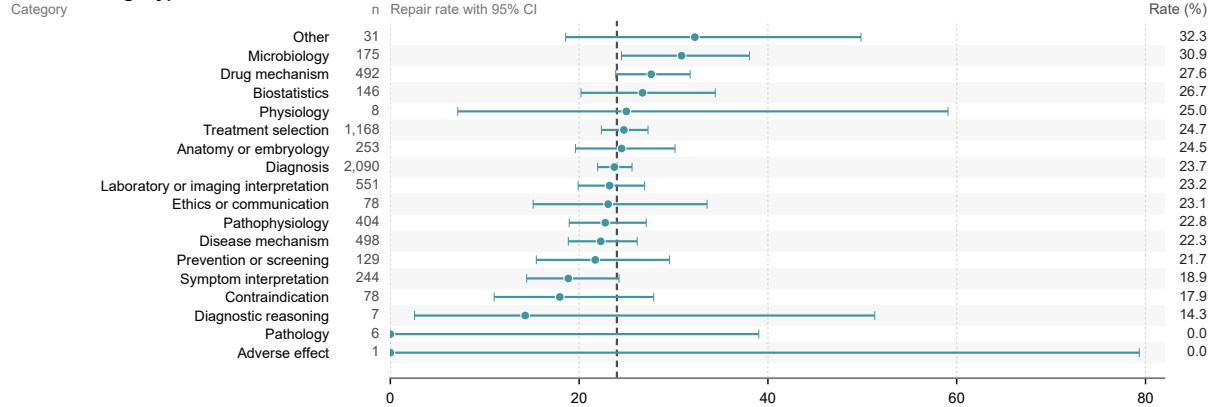
We analyze the 6,359 MedQA training questions that Qwen2.5-1.5B-Instruct answered incorrectly before distillation. We use deepseek-v4-pro to assign each question one primary label for medical specialty, knowledge type, and error type. Semantically equivalent labels are merged before aggregation. An error is considered repaired when the LEAD-trained student answers the same question correctly.

As shown in Figure 9, LEAD repairs 1,526 of the 6,359 original errors. The repaired questions span a broad range of medical specialties, knowledge types, and error types rather than concentrating in a single category. Among the major error categories, repair rates reach 26.01% for treatment selection errors and 25.52% for factual knowledge gaps, while diagnostic reasoning errors have a lower repair rate of 20.46%. These results provide a fine-grained view of both repaired and remaining student weaknesses. Because this diagnostic uses the original error set, it should not be interpreted as held-out test accuracy.

a Medical specialty



b Knowledge type



c Error type

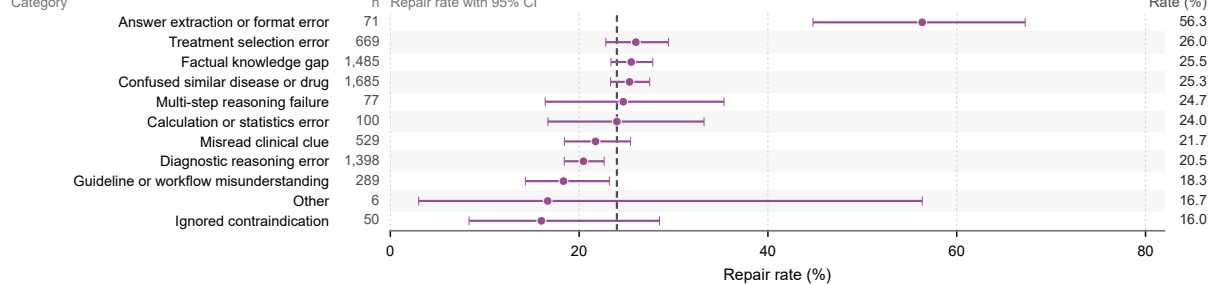


Figure 9: Fine-grained repair analysis across all annotated categories. Panels (a)–(c) report repair rates by medical specialty, knowledge type, and error type, respectively. Points denote category-level repair rates, horizontal bars show 95% Wilson confidence intervals, and the n column reports the number of samples in each category. The dashed line marks the overall repair rate of 24.00%. Each panel independently partitions the same 6,359 errors.