SCAFFOLD BUILDER: DIAGNOSIS-GUIDED DATA SYNTHESIS FROM INTERVENABLE FAILURES

Anonymous authors
Paper under double-blind review

ABSTRACT

The bottleneck of post-training is shifting from insufficient data scale to the scarcity of high-learning-value data. Existing data synthesis methods are often decoupled from the model's current capability state: the generated samples may be more complex or higher quality, yet they do not necessarily target the model's actual capability gaps. To address this issue, we propose **Scaffold Builder**, a diagnosis-guided data synthesis framework that formulates data construction as a teaching process centered on the model's capability boundary. Inspired by educational scaffolding and the Zone of Proximal Development (ZPD), Scaffold Builder organizes synthesis into an iterative loop of *boundary mining, scaffold construction, controlled synthesis, quality filtering, and feedback training*. It first identifies samples near the student model's capability boundary, after which a teacher model diagnoses the corresponding failures into a structured **RootCause–Fix–Scaffold** triplet. The effectiveness of each scaffold is then verified through student-model re-evaluation. Guided by the validated scaffolds and diagnosed root causes, the framework synthesizes targeted training data along two complementary paths and uses a three-stage curriculum to shift from scaffolded learning toward independent answering. Experiments across multiple reasoning benchmarks show that, with **only 300 seed examples**, Scaffold Builder achieves higher average performance than strong data synthesis and distillation baselines. Further ablation, iterative, model-scale, training-paradigm, and cross-domain experiments demonstrate the effectiveness, scalability, and transferability of the proposed framework.

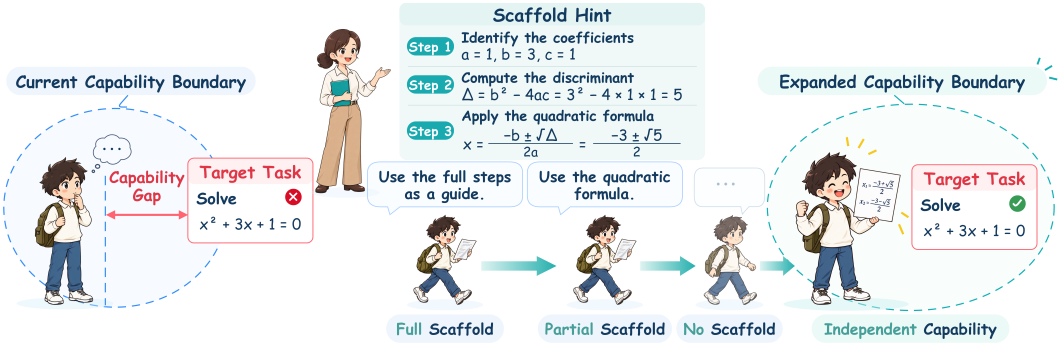

Figure 1: **Teaching intuition behind Scaffold Builder.** The child and teacher represent the student and teacher models. A target equation illustrates the gap between the student's current capability and the desired reasoning skill. Supported practice, reduced support, and independent answering depict the teaching progression that motivates the curriculum; dashed boundaries visualize the intended expansion of independent problem-solving capability.

1 INTRODUCTION

Post-training is an important means of improving the reasoning capabilities of large language models (LLMs) (Grattafiori et al., 2024; Yang et al., 2025), yet larger datasets, harder problems, or

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

higher-quality teacher solutions do not necessarily yield corresponding gains. The training value of an example depends on both its quality and its alignment with the model's current capabilities (Zhou et al., 2023; Xia et al., 2024): familiar examples offer limited new information, whereas those requiring large reasoning leaps can be difficult to learn from. A central question in post-training data construction is therefore how to continually provide valuable training examples that target the model's current capability gaps.

Data selection can retain useful examples from existing data (Xia et al., 2024), but has limited scope to fill gaps in the coverage of specific failure patterns and reasoning skills. When key reasoning steps are absent from the source data, selection alone cannot create the corresponding training examples, motivating targeted synthesis. Existing methods enrich training data through instruction augmentation (Wang et al., 2023), difficulty evolution (Xu et al., 2023), and teacher distillation (Ho et al., 2023; Wu et al., 2024), yet their generation objectives can remain misaligned with the student's specific weaknesses. Error-feedback-driven methods further adapt generation to student performance (Lee et al., 2024; Wang et al., 2025). A key question remains: **Which failures are amenable to targeted intervention, and how can such interventions be translated into transferable training data?** An incorrect answer or a plausible error explanation alone does not determine what subsequent synthesis should target.

We address this question by **modeling data synthesis as a teaching process**. Effective teaching centers on the student's specific difficulties: identifying knowledge that has not been reliably mastered, providing key hints to overcome obstacles, designing targeted practice, and gradually withdrawing assistance (Wood et al., 1976; Renkl et al., 2002). Inspired by the Zone of Proximal Development (ZPD) (Cole & SCRIBNER, 1978) and instructional scaffolding, we focus on problems where the student's independent performance is inconsistent but improves with appropriate hints. We use this intervenability to guide targeted synthesis, turning model failures into teaching signals. **Figure 1 illustrates the teaching intuition: targeted scaffolds help a student tackle a challenging problem, while progressively reduced support shifts practice toward independent solving. Scaffold Builder translates this process into diagnosis-guided data synthesis and training.**

To this end, we propose **Scaffold Builder**, a diagnosis-guided data synthesis framework. It first uses the student's multi-view responses to identify candidates near its capability boundary. A teacher then organizes each failure into a **RootCause–Fix–Scaffold** structure: RootCause characterizes the failure pattern, Fix specifies the missing key reasoning operation, and Scaffold expresses it as an actionable hint. The student answers again with the scaffold, and the gain from re-evaluation is used to validate the intervention's effectiveness. Guided by the validated scaffold strategies and diagnosed root causes, the teacher synthesizes targeted examples along two paths. After quality filtering, these examples support a three-stage curriculum that shifts from scaffolded learning toward independent problem solving. Student capabilities are reassessed after training to update the diagnostic and synthesis targets for the next iteration.

Our contributions are threefold:

1. **Teaching-driven closed-loop synthesis.** We propose a data synthesis framework grounded in teaching-process modeling, connecting capability-boundary mining, diagnosis-guided synthesis, and scaffolded curriculum training in an iterative loop that adapts its synthesis targets to the student's evolving capabilities.

2. **Intervention validation and dual-path synthesis.** We combine structured RootCause–Fix–Scaffold diagnosis with student re-evaluation to select effective scaffolds based on measured intervention gains. Scaffold strategies and diagnosed root causes then guide two synthesis paths, translating failure analysis into actionable criteria for data generation.

3. **Performance gains from few seed examples.** Starting from only 300 seed examples, Scaffold Builder achieves higher average performance than multiple synthesis and distillation baselines across two model families and multiple mathematical reasoning benchmarks. Ablations and extended experiments examine key design choices, model scale, and training paradigms. The results show effective capability improvement from few seeds, with gains across different model sizes and training paradigms.

2 Related Work

Instruction Data Synthesis. Instruction data synthesis expands a small set of seed examples into training data for language models. Self-Instruct (Wang et al., 2023) uses seed tasks to prompt models to generate new instructions and responses, followed by format checks and similarity filtering. Evol-Instruct further increases instruction complexity through additional constraints, deeper reasoning steps, and task composition, with WizardLM (Xu et al., 2023) and WizardCoder (Luo et al., 2024) extending this paradigm to general instruction following and code generation. These approaches move data synthesis beyond surface-level rewriting toward broader task coverage, expression diversity, and structural complexity.

Quality-Oriented Data and Reasoning Distillation. Beyond scale and complexity, another line of work emphasizes the quality and information content of training data. LIMA (Zhou et al., 2023) demonstrates the utility of a small set of carefully curated instruction examples for instruction following. Chain-of-thought distillation transfers teacher-generated step-by-step solutions to student models as reasoning supervision (Ho et al., 2023), while LaMini-LM (Wu et al., 2024) expands instruction-distillation data through example- and topic-guided generation. Pedagogically inspired distillation further organizes knowledge transfer around student knowledge gaps and progressive curricula (He et al., 2026). These methods highlight the importance of high-quality demonstrations and task coverage. Scaffold Builder further focuses on matching training signals to the student's current capabilities, using boundary mining and scaffold validation to determine where targeted synthesis should be applied.

Error Feedback and Iterative Learning. A model's own solving behavior provides a more targeted signal for data construction. STaR (Zelikman et al., 2022) filters reasoning trajectories by answer correctness and regenerates rationales for failed problems conditioned on the correct answer, while LEMA (An et al., 2023) uses a stronger model to identify erroneous reasoning steps, explain the failure, and construct correction data. LLM2LLM (Lee et al., 2024) generates supplementary examples from student errors and iteratively retrains the model, and Socratic-Zero (Wang et al., 2025) adjusts problem generation through the co-evolution of teacher, solver, and generator agents. These studies bring model errors and feedback into the data update loop. Scaffold Builder builds on this direction by asking whether a diagnosed hint actually improves student performance: scaffolds are validated through student re-evaluation before the validated scaffold strategies and diagnosed root causes are used to guide two complementary synthesis paths, connecting failure diagnosis, intervention validation, and training-data generation.

3 Method

3.1 Overview

Scaffold Builder models data synthesis as a teaching process centered on the student's capability gaps. Inspired by the Zone of Proximal Development and instructional scaffolding (Cole & SCRIBNER, 1978; Wood et al., 1976), it identifies unstable problem-solving behavior, validates targeted hints, and turns the resulting diagnoses into training examples. Figure 2 summarizes five components: **boundary mining, scaffold construction, controlled synthesis, quality filtering, and curriculum training**.

Let $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^{n}$ be the seed dataset, with problem x_i and reference answer y_i, and let M_k and T denote the student at iteration k and the teacher. Each iteration probes the same original seed pool, identifies boundary candidates $\mathcal{B}_k$, and constructs a filtered synthetic dataset $\widetilde{\mathcal{D}}_k$. A three-stage curriculum produces M_{k+1}. Training uses the current iteration's new examples, while the original seed pool remains fixed across iterations.

3.2 Multi-View Reasoning and Boundary Mining

We use eight reasoning views: normal, cautious, very cautious, confident, direct solving, arithmetic checking, constraint checking, and concise reasoning. For each problem, the student generates one response per view, giving $N = 8$ responses at temperature 0.7, top-p 0.95, and a maximum

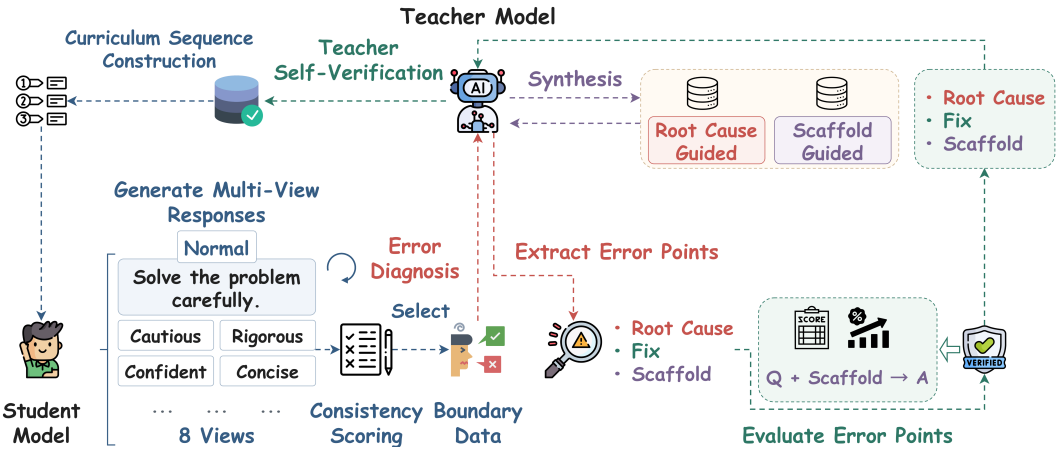

Figure 2: **The Scaffold Builder learning loop.** Multi-view student responses identify boundary candidates for teacher diagnosis. RootCause, Fix, and Scaffold describe the failure, corrective operation, and actionable hint, respectively. Student re-evaluation tests the hint before root-cause-guided and scaffold-guided synthesis. Quality filtering and a three-stage curriculum turn the generated examples into training data and an updated student, which is then probed on the same fixed seed pool.

generation length of 2,048 tokens. Let o_{ij} be response j and $c_{ij}^{(k)} \in \{0,1\}$ its correctness. We estimate

$$p_k(x_i) = \frac{1}{8}\sum_{j=1}^{8} c_{ij}^{(k)}, \qquad \mathcal{B}_k = \left\{(x_i, y_i) \in \mathcal{S} : \frac{3}{8} \le p_k(x_i) \le \frac{5}{8}\right\}. \tag{1}$$

Thus, problems with three to five correct responses are boundary candidates; zero to two and six to eight correct responses define the hard and easy groups, respectively. Every incorrect response on a boundary problem proceeds to diagnosis. Answer checking first normalizes the final `\boxed` answer for string matching and uses teacher-based mathematical equivalence evaluation when direct matching fails. Prompt templates and view-subset analyses appear in Appendices A and D.1.

3.3 Structured Error Diagnosis and Scaffold Construction

The teacher diagnoses every incorrect trajectory o_{ij}^{-} separately:

$$(r_{ij}, f_{ij}, h_{ij}) = T_{\text{diag}}(x_i, y_i, o_{ij}^{-}). \tag{2}$$

RootCause r_{ij} describes the underlying failure pattern, such as an omitted constraint or an invalid transformation. **Fix** f_{ij} specifies the reasoning operation to add or correct. **Scaffold** h_{ij} expresses that operation as an actionable hint. An auxiliary `evidence` field records the response span supporting the diagnosis. The prompt targets the first critical error and requests a reusable hint of one or two sentences, preferably within 40 tokens, excluding final answers and complete solutions.

3.4 Scaffold Validation

For each candidate scaffold, the student solves the original problem with the hint under the same eight views and decoding parameters. Baseline and scaffold-assisted responses are generated separately. Let $s_i = p_k(x_i)$ be the original score and s'_{ij} the correctness rate after adding h_{ij}. The scaffold gain and acceptance rule are

$$\Delta_{ij} = s'_{ij} - s_i, \qquad \text{retain } h_{ij} \text{ when } \Delta_{ij} \ge \tau, \qquad \tau = \frac{1}{8}. \tag{3}$$

The threshold requires a net increase of at least one correct response, the smallest positive gain resolved by eight responses. Validation operates on each problem–error-trajectory–scaffold instance: scaffolds for the same problem share s_i but receive separate re-evaluations. Accepted diagnoses supply the intervention targets for synthesis.

3.5 TARGETED DUAL-PATH SYNTHESIS

The teacher generates examples from each accepted diagnosis through two paths. **Scaffold-guided synthesis** constructs problems that require the reasoning strategy expressed by the validated hint. **RootCause-guided synthesis** constructs problems targeting the diagnosed failure pattern. Each output contains a new problem, a worked solution, a final answer, and two or three instructional hints. The paths organize practice around strategies to acquire and errors to correct.

Figure 3 at the beginning of the appendix illustrates the distinction for a sign error in Vieta's relations. The root-cause path targets the coefficient–root sign relation; the scaffold path transfers the strategy of expressing root quantities through symmetric sums and polynomial coefficients.

3.6 QUALITY CONTROL FOR SYNTHETIC DATA

Candidates undergo format checking, semantic deduplication, and answer verification. Empty fields and invalid JSON are rejected. MiniLM sentence embeddings compare candidate questions against a shared pool across both synthesis paths; candidates with cosine similarity at least 0.70 are discarded. The teacher then independently solves each retained question at temperature 0. We extract the last `\boxed` answer from both the independent solution and the generated solution, normalize their mathematical notation, and retain exact string matches. Boundary mining and scaffold validation provide student-specific guidance for synthesis, while the post-generation checks control format, redundancy, and answer agreement. The accepted examples form $\widetilde{\mathcal{D}}_k$.

3.7 THREE-STAGE CURRICULUM AND CLOSED-LOOP ITERATION

Following the principle of fading (Renkl et al., 2002), the curriculum progressively shifts its data composition toward independent answering. Each accepted scaffold-guided problem, termed a bridge example, has scaffolded and scaffold-free input versions.

Stage 1: Scaffolded learning. The student trains on all scaffolded bridge examples to learn the target reasoning operations with explicit support.

Stage 2: Scaffold-assisted learning. All scaffolded bridge examples are combined with a random 30% subset of their scaffold-free versions, introducing independent practice alongside supported learning.

Stage 3: Scaffold withdrawal and independent answering. Training uses all scaffold-free bridge examples and all root-cause-guided examples, together with a 10% subset of scaffolded bridge examples for replay.

Subsets are sampled without replacement with seed 42 and rounded down in size. Each SFT stage runs for five epochs, after which LoRA weights are merged. The merged model after stage 3 becomes M_{k+1}. The next iteration probes the same seeds and trains on newly synthesized data. Appendix C.4 gives the curriculum construction, and Algorithm 1 summarizes the complete procedure.

4 EXPERIMENTS

We aim to answer four questions:

- **Can a small seed set yield higher average performance than data-synthesis and distillation baselines across different student models?**
- **Which components drive the gains—multi-view probing, scaffold validation, dual-path synthesis, teacher verification, or curriculum training?**
- **How do seed size, category distribution, and repeated iterations affect learning outcomes?**
- **Do the gains extend across model scales, training paradigms, and tasks beyond mathematics?**

Main comparisons address the first question; component ablations examine the second; seed and iteration analyses study the third; and model-scale, training-paradigm, and cross-domain evaluations address the fourth. The appendix further analyzes how view count and selection thresholds affect diagnostic coverage and the pool of interventions available for synthesis.

4.1 Experimental Setup

Models and benchmarks. Our primary students are Qwen3-4B (Yang et al., 2025) and Llama3-8B (Grattafiori et al., 2024); Qwen3-8B and Qwen3-14B extend the model-scale evaluation. DeepSeek-V3.2 (Liu et al., 2025) serves as the teacher for error diagnosis, data synthesis, and answer evaluation. The main evaluation covers five mathematical reasoning benchmarks: GSM8K (Cobbe et al., 2021), AMC, AIME2024, AIME2025, and Olympiad (He et al., 2024). We also evaluate cross-domain transfer on MMLU (Hendrycks et al., 2020), GPQA-Diamond (Rein et al., 2023), and BBEH (Kazemi et al., 2025).

Seeds and baselines. Scaffold Builder starts from 300 problems sampled by subject category from the official MATH test split. These seeds form a fixed diagnostic pool, separate from the reported evaluation benchmarks. We compare with the initial model, Seed-SFT, SA, Self-Instruct (Wang et al., 2023), Evol-Instruct (Xu et al., 2023), LLM2LLM (Lee et al., 2024), CoT distillation (Ho et al., 2023), LaMini-LM (Wu et al., 2024), and Socratic-Zero (Wang et al., 2025). Appendix C.5 describes seed construction and evaluation-set checks.

Training and evaluation. SFT uses ms-swift (Zhao et al., 2025) and LoRA (Hu et al., 2021), with learning rate 10^{-5}, global batch size 256, and five epochs per curriculum stage. Benchmark inference uses ms-swift with temperature 0. We obtain eight responses per question and report their mean correctness over the benchmark as mean@8. Avg. is the arithmetic mean across the five mathematical benchmarks; scores are percentages and gains are percentage points. Table 1 reports the second-round observed peak, alongside the three-round trajectory in Table 5. Component, seed, and model-scale analyses use iteration 1. Detailed configurations appear in Appendix C.3.

4.2 Main Results

To evaluate whether organizing data synthesis as a teaching process can improve reasoning from a small seed set, we compare Scaffold Builder with direct seed training and representative synthesis and distillation methods across two student model families and five mathematical reasoning benchmarks (Table 1).

Scaffold Builder achieves the highest average scores of 40.02 on Qwen3-4B and 20.72 on Llama3-8B, exceeding Seed-SFT's 36.74 and 17.54 and outperforming LaMini-LM, the strongest comparison baseline by average score, by 0.17 and 1.64 percentage points, respectively. Notably, direct seed training lowers Llama3-8B's average from 18.16 to 17.54, whereas Scaffold Builder raises it to 20.72. This contrast highlights how expanding and organizing seed data around the student's current state affects post-training outcomes. **These results support organizing student failures into targeted practice and a teaching process as an effective way to construct post-training data from few seeds and improve reasoning across student models.**

4.3 Component Ablations

To evaluate the roles of student feedback, complementary practice, and staged learning in the teaching loop, we ablate multi-view probing, scaffold-effectiveness validation, each synthesis path, teacher self-verification, and curriculum training on Llama3-8B (Table 2). Appendix C.7 specifies the configurations.

The full method achieves an average of 20.31. Replacing multi-view probing with repeated sampling of one prompt lowers it to 14.39, while removing scaffold-effectiveness validation lowers it to 19.43, highlighting the value of student feedback in both identifying diagnostic targets and selecting interventions. Training on scaffold-guided or root-cause-guided data alone yields 18.84 and 18.81; combining both paths improves their averages by 1.47 and 1.50 points, respectively, and outperforms both single-path variants on all four benchmarks other than GSM8K. Removing teacher

Table 1: **Mathematical reasoning across two student model families.** Qwen3-4B and Llama3-8B are compared with their initial models, seed-only SFT, and synthesis or distillation baselines on five benchmarks. Entries report mean@8 (%), and Avg. is the unweighted benchmark mean. Scaffold Builder reports the observed peak at iteration 2; Table 5 shows all three rounds. Bold marks column maxima within each model group.

Method	GSM8K	AMC	AIME 2024	AIME 2025	Olympiad	Avg.
Qwen3-4B						
Initial model	84.00	**48.44**	8.33	7.08	23.56	34.28
Seed-SFT	86.62	44.06	7.50	8.75	36.76	36.74
SA	83.44	45.31	5.83	7.50	25.24	33.46
Self-Instruct	86.57	36.56	4.58	3.33	22.46	30.70
Evol-Instruct	87.77	40.94	9.58	6.67	33.54	35.70
LLM2LLM	80.10	42.81	11.25	5.83	32.83	34.56
CoT distillation	88.13	46.88	7.50	11.25	36.96	38.14
LaMini-LM	88.38	47.50	11.67	**12.08**	39.63	39.85
Socratic-Zero	**91.06**	44.37	10.83	10.83	**41.02**	39.62
Scaffold Builder	89.74	**48.44**	**12.50**	10.83	38.61	**40.02**
Llama3-8B						
Initial model	76.64	7.50	0.00	0.42	6.26	18.16
Seed-SFT	75.22	7.19	0.00	0.00	5.31	17.54
SA	64.56	9.69	0.00	0.42	6.20	16.17
Self-Instruct	55.91	10.31	0.00	0.00	4.65	14.17
Evol-Instruct	72.40	7.50	0.00	0.00	5.19	17.02
LLM2LLM	75.66	9.69	0.00	0.00	4.30	17.93
CoT distillation	76.28	8.75	0.42	0.42	5.74	18.32
LaMini-LM	75.60	12.19	0.00	0.42	7.20	19.08
Socratic-Zero	**78.20**	6.56	0.42	0.42	5.91	18.30
Scaffold Builder	77.99	**14.37**	**0.83**	**2.50**	**7.91**	**20.72**

Table 2: **Contributions of probing, validation, synthesis, and curriculum.** Llama3-8B results after iteration 1 compare the full method with single-view prompting, unvalidated scaffolds, each synthesis path alone, removal of teacher self-verification, and shuffled single-stage SFT. Scaffold-effectiveness validation screens interventions before synthesis; teacher self-verification checks generated answers afterward. The w/o scaffold-guided synthesis variant uses root-cause data only. Scores are mean@8 (%); Avg. is the five-benchmark mean. Bold marks column maxima. Configuration details appear in Appendix C.7.

Configuration	GSM8K	AMC	AIME 2024	AIME 2025	Olympiad	Avg.
Initial model	76.64	7.50	0.00	0.42	6.26	18.16
Full method	78.12	13.12	**0.83**	**1.67**	**7.83**	**20.31**
w/o multi-view prompts	57.57	8.75	0.00	0.42	5.22	14.39
w/o scaffold effectiveness validation	78.81	11.25	0.00	0.83	6.28	19.43
Only Scaffold-Guided Data	**79.21**	8.75	0.00	0.42	5.83	18.84
w/o scaffold-guided synthesis	78.89	8.12	0.00	0.83	6.20	18.81
w/o teacher self-verification	76.12	**14.37**	0.00	0.00	6.70	19.44
w/o scaffolded curriculum	77.98	12.50	0.00	0.83	7.54	19.77

self-verification or replacing the ordered curriculum with single-stage training lowers the average to 19.44 and 19.77, further supporting post-generation verification and curriculum organization. **These results show that combining student-feedback-driven intervention selection with complementary practice, quality verification, and staged training makes teaching-driven synthesis more effective, supporting a complete learning loop organized around the student's capability state.**

4.4 SEED SIZE AND DISTRIBUTION

To examine which seed compositions better support teaching-driven synthesis, we compare different seed counts and category-balanced versus unbalanced sets at the same count (Table 3).

Table 3: **Effects of seed quantity and category balance.** Results use Llama3-8B after iteration 1. The two 300-seed configurations compare category-stratified sampling with sampling without category quotas; 100- and 200-seed sets examine smaller starting pools. Scores are mean@8 (%), with Avg. across the five benchmarks. Bold marks column maxima.

Seed configuration	GSM8K	AMC	AIME 2024	AIME 2025	Olympiad	Avg.
300, balanced	78.12	**13.12**	**0.83**	**1.67**	**7.83**	**20.31**
300, unbalanced	**78.44**	11.88	0.00	0.83	7.04	19.64
200	77.62	7.50	0.00	0.43	6.96	18.50
100	78.02	7.50	0.00	0.83	6.63	18.60

The balanced 300-seed set achieves an average of 20.31, compared with 18.60 and 18.50 for 100 and 200 seeds. At a fixed count of 300, category balancing raises the average from 19.64 to 20.31, improving AMC, AIME2024, AIME2025, and Olympiad. GSM8K scores remain close across configurations, whereas competition-level tasks show more pronounced differences. Seed design therefore provides a starting point for broader task improvement, beyond maintaining existing proficiency. **These results show that effective teaching-driven synthesis benefits from attention to both seed quantity and category coverage: increasing the seed pool and improving its distribution provide a more comprehensive, balanced starting point for targeted generation.**

4.5 MODEL SCALE

To test cross-scale generalization, we apply Scaffold Builder to Qwen3-4B, 8B, and 14B (Table 4). One iteration improves average scores across all sizes, from 34.28 to 39.09 at 4B and from 40.41 to 41.16 at 14B; GSM8K, AIME2025, and Olympiad improve throughout. **Scaffolded synthesis thus benefits students beyond the smallest scale, supporting its cross-scale applicability and scalability.**

Table 4: **Generalization across student model sizes.** Initial and iter1 checkpoints are compared using mean@8 (%); Avg. averages five benchmarks. Sample sizes are listed for trained checkpoints. Bold marks the best scores within each model pair, including ties.

Student	Setting	GSM8K	AMC	AIME 2024	AIME 2025	Olympiad	Avg.	Sample size
Qwen3-4B	Initial	84.00	**48.44**	8.33	7.08	23.56	34.28	–
	iter1	**89.70**	46.88	**9.17**	**10.83**	**38.89**	**39.09**	641
Qwen3-8B	Initial	89.00	42.50	7.50	12.08	37.41	37.70	–
	iter1	**89.24**	**46.88**	**10.42**	**15.83**	**38.69**	**40.21**	1,123
Qwen3-14B	Initial	91.65	45.94	**11.25**	14.58	38.63	40.41	–
	iter1	**91.94**	**46.25**	**11.25**	**17.08**	**39.28**	**41.16**	611

4.6 ITERATIONS

To test progressive improvement through feedback-driven cycles, we compare three learning rounds (Table 5). Round 2 raises the average from 20.31 to 20.72 on Llama3-8B and from 39.09 to 40.02 on Qwen3-4B, with AMC improving on both students. Both averages decline in round 3. **Cyclic diagnosis and targeted synthesis therefore support round-by-round refinement, with useful gains realized in few iterations.**

Table 5: **Progressive refinement over three learning rounds.** Scores are mean@8 (%); Avg. averages five benchmarks. Both students reach their highest average in round 2.

Student	Round	GSM8K	AMC	AIME 2024	AIME 2025	Olympiad	Avg.
Llama3-8B	iter1	**78.12**	13.12	**0.83**	1.67	7.83	20.31
	iter2	77.99	**14.37**	**0.83**	**2.50**	**7.91**	**20.72**
	iter3	77.64	11.56	**0.83**	1.67	7.89	19.92
Qwen3-4B	iter1	89.70	46.88	9.17	**10.83**	**38.89**	39.09
	iter2	**89.74**	48.44	**12.50**	**10.83**	38.61	**40.02**
	iter3	89.51	**49.38**	10.42	10.42	37.65	39.48

4.7 Training Paradigms and Cross-Domain Transfer

Training paradigms. To test whether diagnosis-guided synthetic data support different training paradigms rather than supervised fine-tuning alone, we use the same synthetic examples for SFT and GRPO (Shao et al., 2024) (Table 6). Both paradigms improve Qwen3-4B's reasoning performance, raising its average from 34.28 to 40.02 and 40.25, respectively. **These results demonstrate the cross-paradigm applicability of Scaffold Builder's data, which provide useful learning signals for both supervised imitation and reward-based policy optimization.**

Table 6: **Synthetic data under SFT and GRPO.** Qwen3-4B uses the same synthesized examples and three-stage organization with supervised or reward-based optimization. The initial model provides the reference level. Entries report mean@8 (%) and their five-benchmark average. The training configurations, including five SFT epochs versus one GRPO epoch per stage, appear in Table 10.

Training paradigm	GSM8K	AMC	AIME 2024	AIME 2025	Olympiad	Avg.
Initial model	84.00	48.44	8.33	7.08	23.56	34.28
SFT	89.74	48.44	**12.50**	10.83	38.61	40.02
GRPO	**90.81**	**49.69**	9.17	**12.92**	**38.65**	**40.25**

Cross-domain transfer. To test whether teaching-driven synthesis produces benefits beyond its training domain, we evaluate the mathematically trained student on multi-subject knowledge, specialized question answering, and complex reasoning tasks (Table 7). Qwen3-4B improves from 44.8 to 52.9 on MMLU and from 31.0 to 32.6 on GPQA-Diamond. BBEH declines from 13.1 to 12.1, while the three-benchmark average rises from 29.6 to 32.6. Gains on multi-subject knowledge and specialized question answering show that targeted mathematical training can benefit tasks beyond those used for synthesis. **These results show that Scaffold Builder's benefits extend beyond mathematical problems to multi-subject knowledge and specialized question answering, demonstrating the approach's broader cross-domain transfer value across heterogeneous evaluation settings.**

Table 7: **Transfer beyond mathematical training.** Qwen3-4B is evaluated before and after Scaffold Builder on BBEH, GPQA-Diamond, and MMLU. Scores are percentages; Avg. averages these three benchmarks. Bold marks the higher score for each metric.

Model	BBEH	GPQA-Diamond	MMLU	Avg.
Initial model	**13.1**	31.0	44.8	29.6
Scaffold Builder	12.1	**32.6**	**52.9**	**32.6**

5 Conclusion

We presented Scaffold Builder, which turns intervenable failures into targeted training data through structured diagnosis, student-side scaffold validation, dual-path synthesis, and a three-stage curriculum. Starting from 300 seed examples, it achieves higher average performance than synthesis and distillation baselines across model families, with gains across model sizes and training paradigms. These findings support teaching-driven synthesis aligned with the student's current capabilities.

AI USE STATEMENT

We used ChatGPT to assist with manuscript drafting and revision, Chinese-to-English translation, literature discovery, methodological and experimental-design feedback, interpretation of reported results, and LaTeX preparation and table formatting. Generative image tools also assisted the development of conceptual teaching illustrations. Within Scaffold Builder, DeepSeek-V3.2 performs error diagnosis, scaffold construction, synthetic problem and solution generation, and answer verification, while the student models generate the responses used for boundary mining and scaffold evaluation. These roles and their prompts are described in Section 3 and Appendix A. The authors direct the research and take responsibility for the final text, experimental results, references, and visual materials.

ETHICS STATEMENT

This work studies post-training data synthesis using mathematical reasoning and general-domain benchmarks. The child-and-teacher illustrations are conceptual representations of model learning. Relevant ethical considerations include errors and inherited biases in synthetic data, the reliability of model-based evaluation, the usage terms of source models and datasets, and the computational cost of iterative synthesis and training. Teacher verification and quality filtering address unreliable synthetic examples. Responsible reuse of the data and models should preserve source attribution, respect applicable usage terms, and include validation appropriate to the intended application.

REPRODUCIBILITY STATEMENT

Section 3 presents the synthesis framework and its selection, validation, and curriculum components. Section 4.1 describes the models, benchmarks, baselines, and training and evaluation settings. Appendix A provides the prompt templates, and Algorithm 1 in Appendix B summarizes the iterative procedure. Appendix C reports computing resources, data flow, SFT and GRPO configurations, the three-stage curriculum, seed construction, ablation definitions, and the evaluation protocol. Appendix D analyzes reasoning-view subsets and selection thresholds using saved predictions. The experimental tables specify the model, configuration, and iteration associated with each reported result.

REFERENCES

Shengnan An, Zexiong Ma, Zeqi Lin, Nanning Zheng, Jian-Guang Lou, and Weizhu Chen. Learning from mistakes makes llm better reasoner. *arXiv preprint arXiv:2310.20689*, 2023.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Michael Cole and SYLVIA SCRIBNER. Vygotsky, lev s.(1978): Mind in society. the development of higher psychological processes, 1978.

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

Bowei He, Yankai Chen, Xiaokun Zhang, Linghe Kong, Philip Yu, Xue Liu, and Chen Ma. Pedagogically-inspired data synthesis for language model knowledge distillation. In *International Conference on Learning Representations*, volume 2026, pp. 112387–112424, 2026.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, 2024.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Namgyu Ho, Laura Schmid, and Se-Young Yun. Large language models are reasoning teachers. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 14852–14882, 2023.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Mehran Kazemi, Bahare Fatemi, Hritik Bansal, John Palowitch, Chrysovalantis Anastasiou, Sanket Vaibhav Mehta, Lalit K Jain, Virginia Aglietti, Disha Jindal, Yuanzhu Peter Chen, et al. Big-bench extra hard. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 26473–26501, 2025.

Nicholas Lee, Thanakul Wattanawong, Sehoon Kim, Karttikeya Mangalam, Sheng Shen, Gopala Anumanchipalli, Michael Mahoney, Kurt Keutzer, and Amir Gholami. Llm2llm: Boosting llms with novel iterative data enhancement. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 6498–6526, 2024.

Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, et al. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*, 2025.

Ziyang Luo, Can Xu, Pu Zhao, Qingfeng Sun, Xiubo Geng, Wenxiang Hu, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. Wizardcoder: Empowering code large language models with evol-instruct. In *International Conference on Learning Representations*, volume 2024, pp. 27168–27188, 2024.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

Alexander Renkl, Robert K Atkinson, Uwe H Maier, and Richard Staley. From example study to problem solving: Smooth transitions help learning. *The Journal of Experimental Education*, 70 (4):293–315, 2002.

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

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

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 13484–13508, 2023.

David Wood, Jerome S Bruner, and Gail Ross. The role of tutoring in problem solving. *Journal of child psychology and psychiatry*, 17(2):89–100, 1976.

Minghao Wu, Abdul Waheed, Chiyu Zhang, Muhammad Abdul-Mageed, and Alham Fikri Aji. Lamini-lm: A diverse herd of distilled models from large-scale instructions. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 944–964, 2024.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. Less: Selecting influential data for targeted instruction tuning. *arXiv preprint arXiv:2402.04333*, 2024.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. Wizardlm: Empowering large language models to follow complex instructions. *arXiv e-prints*, pp. arXiv–2304, 2023.

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

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

Yuze Zhao, Jintao Huang, Jinghan Hu, Xingjun Wang, Yunlin Mao, Daoze Zhang, Zeyinzi Jiang, Zhikai Wu, Baole Ai, Ang Wang, et al. Swift: a scalable lightweight infrastructure for fine-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 29733–29735, 2025.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36:55006–55021, 2023.

APPENDIX

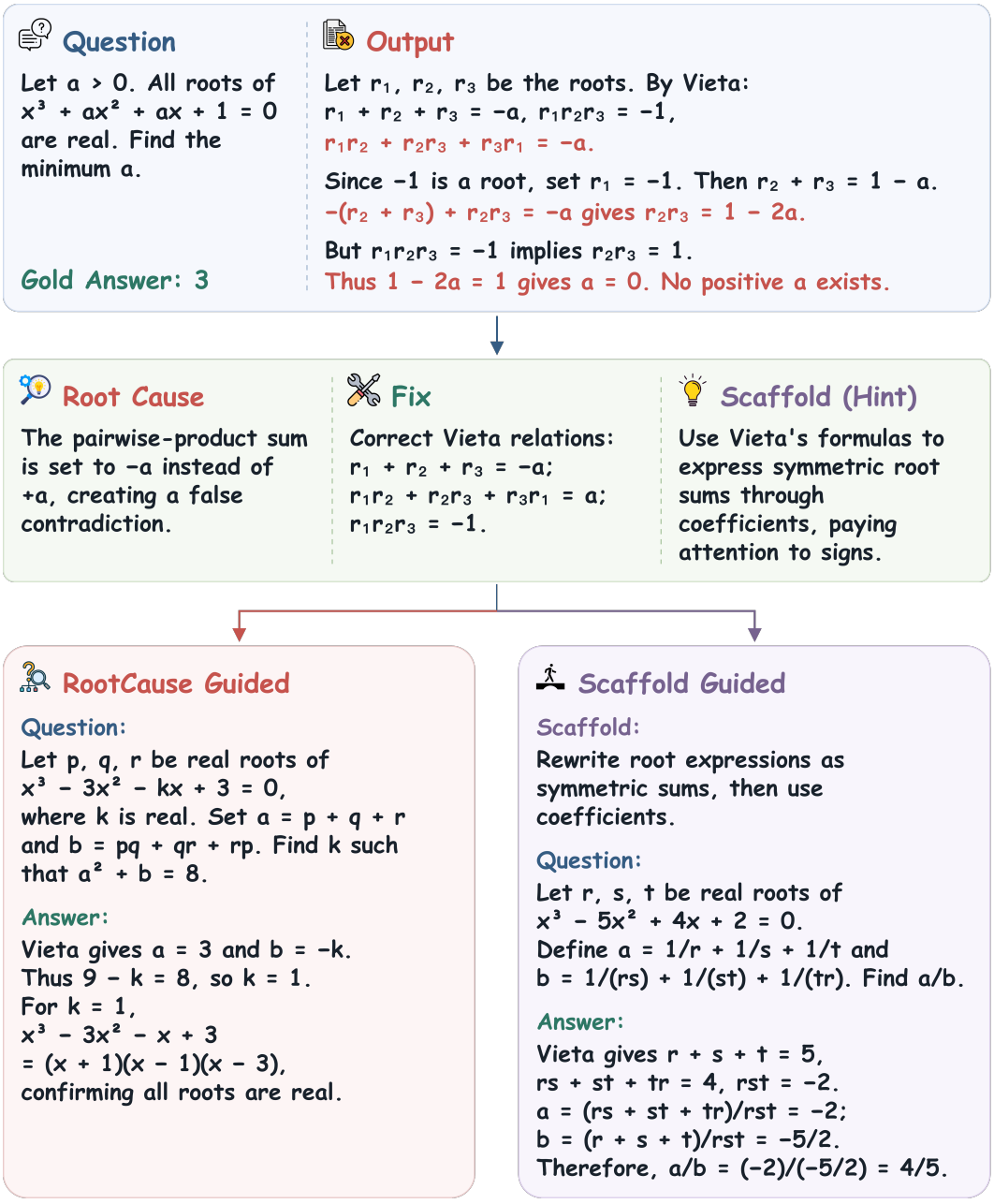

Figure 3: **From a diagnosed error to two synthesis targets.** Top: the student applies the wrong sign to the sum of pairwise root products in Vieta's relations, leading to a false contradiction. Middle: RootCause identifies the sign error, Fix restores the correct relation, and Scaffold expresses a reusable symmetric-sum strategy. Bottom: the RootCause-guided example on the left targets the same sign relation, while the Scaffold-guided example on the right transfers the strategy to expressions involving reciprocal roots.

This appendix provides implementation and evaluation material for Scaffold Builder. Appendix A collects the prompts for student responses, teacher evaluation, diagnosis, and synthesis. Appendix B presents the algorithm. Appendix C covers data flow, training configurations, the three-stage curriculum, datasets, evaluation, and ablations. Appendix D reports offline sensitivity analyses.

Appendix contents

Content	Page
Figure 3 Diagnosis and synthesis example	13
A Prompt templates	14
A.1 Mathematical equivalence evaluation	14
A.2 Eight reasoning views	15
A.3 Solver prompt	15
A.4 Structured error diagnosis	15
A.5 RootCause-guided synthesis	17
A.6 Scaffold-guided synthesis	18
B Algorithm	19
C Additional experimental details	20
C.1 Models and computing environment	20
C.2 Data flow across iterations	20
C.3 SFT and GRPO configurations	21
C.4 Three-stage curriculum construction	22
C.5 Seeds, benchmarks, and overlap checks	22
C.6 Generation and evaluation protocols	23
C.7 Ablation configurations	23
D Offline sensitivity analysis	25
D.1 Number of reasoning views	25
D.2 Boundary interval	26
D.3 Scaffold-gain threshold	26

A Prompt Templates

The student uses the reasoning-view and solver prompts to produce responses. The teacher evaluates answer correctness, constructs the RootCause–Fix–Scaffold diagnosis, and generates new training examples along the two synthesis paths.

A.1 Mathematical Equivalence Evaluation

For student probing and scaffold validation, direct answer matching is attempted first. The following teacher prompt judges mathematical equivalence when normalized strings differ.

Mathematical equivalence evaluation

```
You are a professional math competition judge.
Your task is to determine if a student's predicted answer (PRED) is mathematically
  equivalent to the ground truth (GOLD).
Rules:
1. Focus on mathematical equivalence, not LaTeX formatting (e.g., 0.5 is same as 1/2,
  \sqrt{2}/2 is same as 1/\sqrt{2}).
2. If the problem asks for a set or list, order doesn't matter unless specified.
3. Ignore minor differences in notation (e.g., units, '$' signs) unless they change the
  numerical value.
Return ONLY a valid JSON object: {"equal": true, "reason": "..."} or {"equal": false,
  "reason": "..."}.
```

A.2 Eight Reasoning Views

The student generates one response under each of the eight views, using temperature 0.7, top-p 0.95, and at most 2,048 new tokens. Boundary candidates have three to five correct responses. Scaffold validation uses the corresponding eight views with the hint added to the problem.

Eight reasoning views

```
[
  {"name": "normal", "instruction": "Solve the problem carefully."},
  {"name": "cautious", "instruction": "Solve the problem carefully and double-check key
    conditions before giving the final answer."},
  {"name": "very_cautious", "instruction": "Be very cautious. Verify important
    assumptions and check possible edge cases before answering."},
  {"name": "confident", "instruction": "Give your most confident answer to the
    problem."},
  {"name": "direct_reliable", "instruction": "Solve the problem directly, but make sure
    the final answer is reliable."},
  {"name": "check_arithmetic", "instruction": "Solve the problem carefully and pay
    special attention to arithmetic and symbolic manipulation."},
  {"name": "check_constraints", "instruction": "Solve the problem carefully and pay
    special attention to constraints, conditions, and definitions."},
  {"name": "concise_reasoning", "instruction": "Provide a concise but correct solution,
    and make sure the final answer is fully justified."}
]
```

A.3 Solver Prompt

Solver prompt

```
You are a careful mathematical solver.
Reason step by step when necessary, and conclude with \boxed{answer}.
```

A.4 Structured Error Diagnosis

Every incorrect response on a boundary problem is diagnosed separately. The teacher receives the question, reference answer, and response, and generates a diagnosis at temperature 0.7. The fields `root_cause`, `fix`, and `scaffold` correspond to r_{ij}, f_{ij}, and h_{ij} in Section 3.3; `evidence` stores the supporting response span.

RootCause–Fix–Scaffold diagnosis

```
You are an expert failure analyst for model-generated solutions.
Given a TASK (problem/question), a REFERENCE ANSWER (gold), and the MODEL'S RAW RESPONSE,
identify the FIRST critical point where the response goes wrong, and produce a
  structured diagnosis that is useful for generating new training data.
Return ONLY valid JSON with EXACT keys:
root_cause, evidence, fix, scaffold
Definitions:
- root_cause:
1-2 sentences describing the underlying reasoning failure at the first critical error
  point.
It must be mechanism-level and transferable to similar problems.
Do NOT describe a surface mistake tied only to this specific problem.
Prefer describing structural misunderstandings (e.g., misinterpreting constraints,
  ignoring conditions, invalid transformations, missing case distinctions).
- evidence:
A short verbatim quote (<= 200 characters) from the student response that supports the
  diagnosis and pinpoints the first critical error.
- fix:
The key reasoning operation that needs to be added or corrected to address the failure.
It should describe the key step that was missing or incorrectly applied, not a full
  solution.
```

RootCause–Fix–Scaffold diagnosis (continued)

```
- scaffold:
A concise scaffolded hint or guidance (1-2 sentences, <= 40 tokens recommended) that
  injects the fix while preserving the core skill.
It should be reusable for similar problems and avoid unnecessary surface details from
  the original task.
Prefer abstract mathematical phrasing over task-specific wording whenever possible.
Scaffold Constraints (Hard Rules):
- Do NOT include numeric answers, final results, or boxed values.
- Do NOT provide full step-by-step solutions or explicit full formulas.
- Do NOT leak the final answer.
- Keep it concise and focused on the key reasoning step.
- Avoid phrases like "therefore", "final answer", "computed", "explicitly write", or
  "step by step".
- The hint should guide thinking, not solve the problem.
Additional Constraints:
- root_cause must describe a reusable failure pattern, not a one-off local slip.
- Avoid referencing specific numbers, objects, or story elements unless absolutely
  necessary.
- Prefer abstract descriptions of reasoning (e.g., "fails to separate constrained
  choices by category" rather than "mixes meat and vegetable toppings").
- scaffold should remain valid if the surface context of the problem is changed.
- fix should correspond directly to the missing reasoning implied by root_cause.
General Constraints:
- Do NOT solve the full problem unless absolutely necessary to justify the diagnosis.
- Focus only on the FIRST critical failure point.
- Keep outputs concise.
- No extra keys. No markdown. No extra text.
```

A.5 ROOTCAUSE-GUIDED SYNTHESIS

The diagnosed failure pattern guides new problems and solutions targeting the erroneous reasoning step. Both synthesis paths use teacher temperature 1.0. The `stage_prompts` field stores instructional hints for constructing supported input versions; curriculum stages are defined in Appendix C.4.

RootCause-guided synthesis

```
You are an expert math pedagogical researcher and data synthesizer.
# Task
Generate ONE new math training example that is isomorphic to a target reasoning error
  pattern, and provide incremental scaffolding hints (stage prompts).
## INPUT
- Diagnosed Root Cause: {root_cause}
## INTERNAL PIPELINE
### Step 1: Cognitive Gap Abstraction
Identify the specific reasoning step or boundary condition the student missed. Abstract
  this into a "transferable logical pitfall" (e.g., "Neglecting the case where the base
  equals 1 in an exponential equation").
### Step 2: Isomorphic Problem Synthesis
Create a new problem with a completely different surface form (e.g., if the original was
  Geometry, make this Number Theory) that shares the EXACT same cognitive gap. The
  strategy to avoid the pitfall must be necessary for the solution.
### Step 3: Scaffolded Hint Design (Stage Prompts)
Develop 2-3 "Stage Prompts" that don't give away the answer but direct the solver's
  attention to the specific logic needed to overcome the gap.
- Hint 1: Nudge to identify the structural property.
- Hint 2: Direct focus to the specific reasoning pitfall.
- Hint 3: Guide the final execution or check.
## OUTPUT FORMAT
Return strict JSON with the following fields only:
{
"logic_pattern": "A short name for the reasoning failure pattern.",
"failure_mechanism": "1-2 sentences describing how a solver would likely fail here.",
"synthetic_problem": "The text of the new problem only.",
"thought_process": "The full, correct step-by-step solution. Conclude with the final
  answer in \\boxed{...} format.",
"gold_answer": "The final answer only.",
"stage_prompts": ["...","...","..."]
}
## HARD CONSTRAINTS
1. **No Spoilers**: Do not mention the root cause or give hints in the
  `synthetic_problem` text.
2. **Hidden Logic**: The problem must be solvable only by addressing the target
  reasoning weakness.
3. **Natural Scaffolding**: The `stage_prompts` must be actionable, small steps for a
  student to follow.
4. **Isomorphism**: The new problem must NOT be a simple numerical variation of the
  original.
5. **Exact Match**: The `gold_answer` must exactly match the value inside \boxed{...} in
  the `thought_process`
```

A.6 SCAFFOLD-GUIDED SYNTHESIS

The validated scaffold guides the generation of new problems and solutions that require the corresponding reasoning strategy.

Scaffold-guided synthesis

```
You are an expert math data synthesizer.
Your task is to first extract the transferable high-level thinking step contained in a
  scaffold hint, and then generate a new math problem that can naturally be solved using
  that thinking step.
## INPUT
- Scaffold Hint: {scaffold}
## TASK
Perform the following two stages internally:
### Stage 1: Extract the strategy step
Analyze the scaffold hint and identify a **transferable, general reasoning step or
  method** that can be applied to solve other problems of a similar type.
Do **not** use the problem category, topic, or surface details as the strategy.
Focus on the **cognitive move** or reasoning principle the student uses to make progress.
The extracted strategy step should:
1. Capture the key reasoning move needed to solve the problem.
2. Be abstract and transferable, applicable to multiple problems, not tied to original
  numbers or entities.
3. Remain concrete enough to guide solving other problems naturally.
### Stage 2: Generate a new problem
Generate ONE new math problem such that solving it **requires applying the extracted
  strategy step** in at least one key part of the solution.
The strategy must be **necessary** to reach the correct answer, not optional.
The new problem must:
1. Be mathematically valid, self-contained, and solvable.
2. Not explicitly reveal the strategy step in the problem statement.
3. Require genuine multi-step reasoning rather than trivial substitution.
## SOLUTION REQUIREMENTS
Provide a complete, correct, step-by-step solution to the new problem.
The reasoning should naturally reflect the extracted strategy step, but should not
  rigidly copy the scaffold wording.
End with the exact final answer in LaTeX: \boxed{...}.
Additionally, generate 2-3 scaffolded hints (stage_prompts) that reflect incremental
  steps in using this strategy to solve the synthetic problem. Each hint should be
  phrased as a small, actionable step for a student to follow.
## OUTPUT FORMAT
Return strict JSON with the following fields only:
{
"strategy_type": "...",
"strategy_step": "...",
"synthetic_problem": "...",
"thought_process": "...",
"gold_answer": "...",
"stage_prompts": ["...","...","..."]
}
## FIELD REQUIREMENTS
- "strategy_type": a short category name for the reasoning strategy
- "strategy_step": one sentence describing the transferable high-level thinking step
- "synthetic_problem": only the newly generated problem text
- "thought_process": the full solution to the synthetic problem, conclude with the exact
  answer in LaTeX: \boxed{...}.
- "gold_answer": only the final answer, as a single number or a very short phrase
- "stage_prompts": 2-3 scaffolded hints corresponding to incremental steps using the
  strategy
## HARD CONSTRAINTS
- Do not repeat the scaffold verbatim unless necessary.
- Do not explicitly expose the strategy step in the problem statement.
- The solution must correspond only to the synthetic_problem.
- The final answer in "gold_answer" must match the conclusion in "thought_process".
```

B ALGORITHM

Algorithm 1 summarizes the iterative procedure. The seed pool stays fixed, every incorrect boundary response is diagnosed, and only the current round's synthesized examples are used for training. The three curriculum sets follow Appendix C.4.

Algorithm 1: **Diagnosis-guided synthesis with three-stage curriculum training.** Each round probes the fixed seed pool, validates hints for individual error trajectories, and filters teacher-generated examples. The three curriculum stages train the student on the current round's data before the next boundary assessment.

Input: Initial student M_0; teacher T; fixed seed pool $\mathcal{S}$; eight reasoning views $\mathcal{V}$; iteration count K; gain threshold $\tau = 1/8$.

Output: Updated student M_K and round-specific training sets $\{\widetilde{\mathcal{D}}_k\}_{k=0}^{K-1}$.

for $k = 0, \ldots, K-1$ **do**

Boundary Mining

  for each $(x_i, y_i) \in \mathcal{S}$ **do**

    $\mathcal{O}_i \leftarrow \mathrm{Probe}_8(M_k, x_i, \mathcal{V})$ one response per view

    $c_{ij}^{(k)} \leftarrow \mathrm{Check}_T(o_{ij}, y_i)$ for $j = 1, \ldots, 8$

    $s_i = p_k(x_i) \leftarrow \frac{1}{8}\sum_{j=1}^{8} c_{ij}^{(k)}$

  end for

  $\mathcal{B}_k \leftarrow \{(x_i, y_i) \in \mathcal{S} : 3/8 \le p_k(x_i) \le 5/8\}$

  $\mathcal{D}_k \leftarrow \varnothing$

Per-Trajectory Diagnosis and Validation

  for each (i, j) with $(x_i, y_i) \in \mathcal{B}_k$ and $c_{ij}^{(k)} = 0$ **do**

    $(r_{ij}, f_{ij}, h_{ij}) \leftarrow T_{\mathrm{diag}}(x_i, y_i, o_{ij}^{-})$; store supporting evidence

    $\mathcal{O}'_{ij} \leftarrow \mathrm{Probe}_8(M_k, x_i \oplus h_{ij}, \mathcal{V})$

    $s'_{ij} \leftarrow \frac{1}{8}\sum_{\ell=1}^{8} \mathrm{Check}_T(o'_{ij\ell}, y_i)$

    if $s'_{ij} - s_i \ge 1/8$ **then**

      $\mathcal{D}_k \leftarrow \mathcal{D}_k \cup \mathrm{GenerateScaffoldGuided}(T, h_{ij})$

      $\mathcal{D}_k \leftarrow \mathcal{D}_k \cup \mathrm{GenerateRootCauseGuided}(T, r_{ij})$

    end if

  end for

Quality Control

  $\mathcal{D}_k^{\mathrm{format}} \leftarrow \mathrm{FormatFilter}(\mathcal{D}_k)$

  $\mathcal{D}_k^{\mathrm{unique}} \leftarrow \mathrm{Deduplicate}(\mathcal{D}_k^{\mathrm{format}}; 0.70)$

  $\widetilde{\mathcal{D}}_k \leftarrow \mathrm{VerifyAnswers}(T, \mathcal{D}_k^{\mathrm{unique}}; \mathrm{temperature} = 0)$

Three-Stage Curriculum

  Construct bridge versions $\mathcal{G}_k^{+}, \mathcal{G}_k^{0}$ and root-cause set $\mathcal{R}_k$

  $\mathcal{C}_k^{(1)} \leftarrow \mathcal{G}_k^{+}$

  $\mathcal{C}_k^{(2)} \leftarrow \mathcal{G}_k^{+} \cup \mathrm{Subsample}(\mathcal{G}_k^{0}, 0.30)$

  $\mathcal{C}_k^{(3)} \leftarrow \mathcal{G}_k^{0} \cup \mathcal{R}_k \cup \mathrm{Subsample}(\mathcal{G}_k^{+}, 0.10)$

  $M_k^{(0)} \leftarrow M_k$

  for $a = 1, 2, 3$ **do**

    $M_k^{(a)} \leftarrow \mathrm{MergeLoRA}(\mathrm{SFT}(M_k^{(a-1)}, \mathcal{C}_k^{(a)}; 5 \text{ epochs}))$

  end for; $M_{k+1} \leftarrow M_k^{(3)}$

end for; **return** $M_K, \{\widetilde{\mathcal{D}}_k\}_{k=0}^{K-1}$

`Probe`$_8$ uses temperature 0.7, top-p 0.95, and a 2,048-token generation limit; $\oplus$ adds the hint to the question. `Check` first attempts normalized answer matching and then invokes teacher equivalence evaluation if needed. `VerifyAnswers` independently solves a synthetic question and retains exact matches between normalized final answers. Deduplication uses MiniLM embeddings and a shared pool across synthesis paths. Subsampling is without replacement, uses seed 42, and for an input collection $\mathcal{A}$ retains $\lfloor \alpha|\mathcal{A}| \rfloor$ examples at fraction α.

C Additional Experimental Details

C.1 Models and Computing Environment

The primary students are Qwen3-4B (Yang et al., 2025) and Llama3-8B (Grattafiori et al., 2024); Qwen3-8B and Qwen3-14B extend the scale analysis. DeepSeek-V3.2 (Liu et al., 2025) performs teacher diagnosis, synthesis, and answer evaluation. Table 8 lists the computing environment.

Table 8: **Computing environment for the experiments.** The table lists the GPU configuration and operating-system, CUDA, Python, and Conda versions used in the reported setup. Training hyperparameters for SFT and GRPO are specified separately in Table 10.

Component	Configuration
Operating system	Ubuntu 22.04.1 LTS
GPUs	8× NVIDIA GeForce RTX 4090
CUDA	12.6
Python	3.12.9
Conda	25.1.1

C.2 Data Flow Across Iterations

To show how the teaching loop transforms a fixed seed pool into training data that adapt to the student's state, we track boundary problems, error trajectories, validated scaffolds, and retained synthetic examples across iterations (Table 9).

Problem and trajectory counts use different units: every incorrect response on a boundary problem is diagnosed, and a validated scaffold is a problem–trajectory–hint instance satisfying $\Delta_{ij} \geq 1/8$. Root and Bridge partition the retained synthetic questions into root-cause-guided and scaffold-guided examples.

Table 9: **From seed problems to validated synthetic examples.** Each row corresponds to one model and iteration on the fixed 300-problem seed pool. Boundary problems contribute their incorrect responses for diagnosis; valid scaffolds are problem–trajectory–hint instances with gain at least $1/8$. Synthesis candidates and retained examples are counted as synthetic questions. Root and Bridge partition the retained questions into root-cause-guided and scaffold-guided examples, respectively.

Model / round	Boundary problems	Error responses	Valid scaffolds	Synthesis candidates	Retained	Root	Bridge
Qwen3-4B / 1	33	123	31	480	313	176	137
Qwen3-4B / 2	31	123	59	800	486	236	250
Qwen3-4B / 3	38	152	93	3,259	2,022	991	1,031
Llama3-8B / 1	64	262	37	557	327	170	157
Llama3-8B / 2	53	212	43	999	616	251	365
Llama3-8B / 3	49	187	42	954	591	252	339

In Qwen3-4B round 1, 33 boundary problems are identified from 300 seeds, yielding 123 incorrect responses and 31 diagnoses that pass scaffold validation. The subsequent 480 synthesis candidates yield 313 retained examples after quality control: 176 root-cause-guided and 137 bridge examples. Each round re-diagnoses the same seed pool and constructs a new training set rather than simply accumulating previously synthesized data.

These statistics trace a complete path from student errors through validated interventions to targeted exercises, illustrating how the framework turns a small fixed seed pool into student-specific training data.

C.3 SFT AND GRPO CONFIGURATIONS

Both training paradigms use ms-swift (Zhao et al., 2025) with LoRA (Hu et al., 2021). SFT runs on eight GPUs with per-device batch size 2 and gradient accumulation 16, giving a global batch size of $2 \times 16 \times 8 = 256$. Each of the three SFT stages trains for five epochs. LoRA weights are merged after every stage, and the stage-3 model initializes the next iteration.

Table 10: **Optimization settings for SFT and GRPO.** Both paradigms use LoRA on eight GPUs. The global SFT batch contains 256 training records; the global GRPO batch contains 16 prompts, each producing four responses, for 64 completions. Epoch counts apply to each curriculum stage. Maximum length denotes the configured sequence limit; GRPO additionally uses a 1,024-token completion limit.

Parameter	SFT	GRPO
GPUs	8	8
Per-device batch	2	1 prompt
Gradient accumulation	16	2
Global batch	256	16 prompts
Epochs per stage	5	1
Learning rate	10^{-5}	5×10^{-6}
LoRA rank / alpha	8 / 32	16 / 32
Target modules	all-linear	all-linear
Maximum length	2,048	4,096
Weight decay	0.1	0.1
Warmup ratio	0.05	0.05
Scheduler	Cosine	Cosine
Precision	BF16	BF16
Gradient checkpointing	Enabled	Enabled

GRPO (Shao et al., 2024) initializes from the Qwen3-4B base model and follows the same three-stage data organization. Each prompt produces four responses, sampled at temperature 0.8 and top-p 0.9, with a completion limit of 1,024 tokens. The KL coefficient is $\beta = 0.04$, maximum gradient norm is 1, and distributed training uses DeepSpeed ZeRO-2. The reward is

$$R(o, y) = r_{\text{format}}(o) + 4r_{\text{answer}}(o, y). \tag{4}$$

The `math_format` component checks nonempty output, response formatting, and the final boxed answer. The `math_answer` component extracts and normalizes the final answer to assess mathematical correctness. SFT and GRPO share the synthetic data but use the optimization settings in Table 10.

C.4 THREE-STAGE CURRICULUM CONSTRUCTION

Let $\mathcal{G}_k^+$ and $\mathcal{G}_k^0$ be the scaffolded and scaffold-free versions of the bridge examples retained in round k, and let $\mathcal{R}_k$ contain its root-cause-guided examples. Each bridge problem has one supported and one unsupported input version, so $|\mathcal{G}_k^+| = |\mathcal{G}_k^0| = n_{B,k}$. The prompt field `stage_prompts` supplies instructional cues for supported inputs. Define $\mathrm{Subsample}(\mathcal{A}, \alpha)$ as sampling $\lfloor \alpha|\mathcal{A}| \rfloor$ records without replacement, using random seed 42. The curriculum is

$$\mathcal{C}_k^{(1)} = \mathcal{G}_k^+, \tag{5}$$

$$\mathcal{C}_k^{(2)} = \mathcal{G}_k^+ \cup \mathrm{Subsample}(\mathcal{G}_k^0, 0.30), \tag{6}$$

$$\mathcal{C}_k^{(3)} = \mathcal{G}_k^0 \cup \mathcal{R}_k \cup \mathrm{Subsample}(\mathcal{G}_k^+, 0.10). \tag{7}$$

Stage 1 provides scaffolded learning; stage 2 combines scaffold-assisted learning with unsupported bridge practice; stage 3 emphasizes independent answering through unsupported bridge and root-cause examples, with a small supported replay subset. The percentages refer to the bridge collections from which subsets are sampled. In Qwen3-4B round 3, 1,031 bridge and 991 root-cause examples yield 1,031, 1,340, and 2,125 training records for stages 1, 2, and 3, respectively.

C.5 SEEDS, BENCHMARKS, AND EVALUATION-SET CHECKS

The 300 seed questions are sampled from the official MATH test split. Subject-category stratification covers Algebra, Counting and Probability, Geometry, Intermediate Algebra, Number Theory, Prealgebra, and Precalculus, with approximately balanced category counts. Difficulty levels 1–5 contain 20, 143, 50, 44, and 43 questions, respectively. The unbalanced 300-seed condition samples directly from the candidate pool without category quotas. The original seeds are reused for every round of boundary mining and are not included in the reported evaluation scores.

The mathematical benchmarks are GSM8K (Cobbe et al., 2021), AMC, AIME2024, AIME2025, and Olympiad (He et al., 2024). External evaluation uses MMLU (Hendrycks et al., 2020), GPQA-Diamond (Rein et al., 2023), and BBEH (Kazemi et al., 2025). Seed questions were compared by normalized exact matching against a 2,094-question combined evaluation file and separately against the GSM8K evaluation file. Normalization standardizes case, whitespace, and leading or trailing characters; these checks found no identical seed-question text.

C.6 Generation and Evaluation Protocols

Probing and validation. Boundary mining generates one response per view using temperature 0.7, top-p 0.95, and a 2,048-token limit. For each erroneous boundary response, the teacher produces a diagnosis at temperature 0.7. Its scaffold is evaluated by generating eight new student responses under the corresponding views and decoding settings. Every scaffold for the same problem shares the original baseline correctness rate. We retain gains of at least $1/8$.

Answer checking and synthesis. Student responses are checked by extracting their last `\boxed` answer, normalizing notation, and attempting a direct match. If matching fails, DeepSeek-V3.2 judges mathematical equivalence with the prompt in Appendix A.1. Both synthesis paths use teacher temperature 1.0 and share the MiniLM-based deduplication pool with cosine threshold 0.70. Teacher self-verification uses temperature 0 and compares the normalized final answers of the independently generated and original synthetic solutions. Normalization removes LaTeX presentation markup and whitespace and standardizes multiplication notation. Only extractable, matching answers are retained.

Benchmark inference. Final evaluation uses ms-swift inference with temperature 0 and its default inference settings for the remaining parameters. Eight responses are generated per question, and mean@8 averages their correctness over the benchmark. With $n_{\mathcal{E}}$ questions and correctness judgments $b_{ij} \in \{0, 1\}$, the percentage score is

$$\text{mean@8}(\mathcal{E}) = \frac{100}{8n_{\mathcal{E}}} \sum_{i=1}^{n_{\mathcal{E}}} \sum_{j=1}^{8} b_{ij}. \tag{8}$$

Avg. averages the five mathematical or three cross-domain scores. The main comparison reports the round-2 observed peak of the multi-round benchmark trajectory; Table 5 includes rounds 1–3. Component, seed, and model-scale experiments report the first iteration.

C.7 Ablation Configurations

The six component ablations use Llama3-8B in iteration 1 with the same benchmark evaluation protocol. The variants distinguish complementary probing, student-side scaffold validation, the two synthesis paths, post-generation answer verification, and curriculum organization.

Without multi-view prompts. The student generates eight responses using one standard prompt instead of eight distinct views. Correctness remains $s_i = k_i/8$, and both boundary selection and scaffold-gain validation are retained. The scaffold-assisted condition likewise uses the standard prompt with the hint. This comparison tests the contribution of complementary prompts at a fixed response count.

Without scaffold effectiveness validation. Student-feedback-based effectiveness filtering is removed from the diagnosis stage. The teacher's diagnoses and scaffolds proceed to training-data construction without scaffold-gain screening. Basic format checks and answer-leakage checks are retained. This variant targets the intervention-selection step before synthesis.

Only scaffold-guided data. The student is trained only on scaffold-guided bridge data, with root-cause-guided examples excluded. This variant evaluates the scaffold-guided path on its own and complements the root-cause-only variant below.

Without scaffold-guided synthesis. The scaffold-guided generation branch is removed, and training uses only root-cause-guided examples.

Without teacher self-verification. Format checking and semantic deduplication are retained, but independent teacher re-solving and final-answer agreement filtering are omitted. The resulting candidate records are used for training. This variant concerns answer verification after synthesis, whereas the scaffold-effectiveness ablation concerns student feedback on the proposed hints before synthesis.

Without scaffolded curriculum. Scaffolded bridge, scaffold-free bridge, and root-cause examples are combined, shuffled, and used in a single SFT stage instead of the ordered curriculum.

D Offline Sensitivity Analysis

We reuse saved predictions and scaffold re-evaluations from Qwen3-4B and Llama3-8B over three iterations, giving six groups of 300 seed questions. The following analyses characterize selection behavior under alternative view counts and thresholds.

D.1 Number of Reasoning Views

To examine whether reducing the reasoning views changes the targets for subsequent diagnosis, we compare capability classifications and boundary sets from four-view and six-view subsets with the complete eight-view reference.

We enumerate all four-view and six-view subsets of the eight recorded views, giving $\binom{8}{4} = 70$ and $\binom{8}{6} = 28$ combinations. For a subset of size m, correctness rates at most 0.25 define hard questions, rates at least 0.75 define easy questions, and intermediate rates define boundary questions. The boundary counts are therefore 2 correct responses for $m = 4$, 2–4 for $m = 6$, and 3–5 for $m = 8$.

For each combination, we compute three-way classification agreement, boundary-set Jaccard similarity, and mean absolute error (MAE) in correctness rate relative to the complete eight-view result. Metrics are averaged over combinations within each group and then macro-averaged over the six model–iteration groups (Table 11).

Table 11: **Boundary-selection sensitivity to the number of reasoning views.** Four- and six-view subsets are enumerated from saved eight-view responses. Classification agreement compares hard, boundary, and easy labels; Jaccard compares selected boundary sets; MAE measures differences in estimated correctness. The full eight-view output is the reference. Metrics are averaged over subsets within each group and then over six model–iteration groups. Agreement and Jaccard are percentages; MAE uses the $[0, 1]$ correctness scale.

Views	Classification agreement	Boundary-set Jaccard	Correctness MAE
4	88.9	38.1	0.067
6	91.5	61.5	0.040
8	100.0	100.0	0.000

Four-view subsets achieve 88.9% three-way classification agreement with the eight-view reference, but only 38.1% boundary-set Jaccard similarity. Six views increase Jaccard similarity to 61.5% and reduce correctness MAE from 0.067 to 0.040. Similar overall classifications can therefore mask substantially different boundary sets: reducing the views changes the questions sent to diagnosis and synthesis, a difference obscured by the more numerous easy and hard questions.

These findings support retaining richer multi-view information so that targeted synthesis is grounded in a more comprehensive probe of student capabilities rather than a few prompt-specific observations.

D.2 BOUNDARY INTERVAL

To examine how boundary selection balances a focus on unstable student performance with diagnostic coverage, we compare the numbers of boundary problems and error trajectories obtained under different correctness intervals (Table 12).

Table 12: **Boundary interval and diagnostic coverage.** Saved eight-view predictions are reclassified using the exact midpoint or progressively wider symmetric intervals. Columns report the mean number of selected questions, their percentage of the 300-problem seed pool, and their incorrect responses across six model–iteration groups. The main experiments use $[3/8, 5/8]$; the other rows describe alternative selection rules.

Boundary interval	Boundary questions	Share of 300 seeds (%)	Incorrect responses
$p = 4/8$	15.8	5.3	63.3
$[3/8, 5/8]$	44.7	14.9	176.5
$[2/8, 6/8]$	86.7	28.9	319.8
$[1/8, 7/8]$	157.3	52.4	526.5

Selecting only questions with correctness $p = 4/8$ yields an average of 15.8 boundary problems and 63.3 incorrect responses per group. The $[3/8, 5/8]$ interval increases these counts to 44.7 and 176.5. It remains centered on the correctness midpoint while providing more error trajectories for diagnosis. Further widening the interval extends selection toward questions with more consistently successful or unsuccessful responses.

These results show that moderately widening the selection interval around the correctness midpoint balances target focus with diagnostic coverage, providing richer error information for subsequent targeted synthesis.

D.3 SCAFFOLD-GAIN THRESHOLD

To examine how scaffold selection balances positive intervention feedback with the pool of diagnoses needed for synthesis, we compare scaffold retention under different gain thresholds (Table 13).

Table 13: **Scaffold-gain threshold and available synthesis targets.** Each threshold is applied to the same 1,059 saved problem–trajectory–scaffold validation instances. Retained trajectories satisfy $\Delta_{ij} \geq \tau$, and the final column gives their percentage of the validation pool. The main threshold is $\tau = 1/8$; larger thresholds require larger increases in eight-response correctness and retain fewer instances.

Threshold τ	Retained trajectories	Share of validation instances (%)
0	384	36.3
$1/8$	305	28.8
$2/8$	232	21.9
$3/8$	186	17.6
$4/8$	95	9.0

Among 1,059 validation instances, a zero threshold retains 384 instances, including scaffolds with no gain. Setting $\tau = 1/8$ retains 305 instances (28.8%) and requires a net increase of at least one correct answer among the eight responses. Raising the threshold to $4/8$ reduces the retained set to 95 instances, showing that stronger gain requirements substantially restrict the targets available for subsequent synthesis.

Using the smallest resolvable positive gain as the admission criterion grounds synthesis in positive student feedback while preserving a substantial pool of diagnoses, providing an operational selection rule for intervention-validated data construction.