STEP: FROM LOCAL TEACHER SCAFFOLDS TO STUDENT-OWNED REASONING IN SMALL LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Improving the reasoning abilities of small language models requires successful reasoning experience that can be converted into independent problem-solving ability. Teacher distillation provides correct solutions, but teacher-generated reasoning paths are not always suitable for the student to learn from. Self-training on the student's own successful traces, in contrast, rarely provides successful experience beyond the student's current ability. We therefore ask how a small reasoning model can learn to reason beyond its current limits from self-generated experience with the help of temporary external scaffolds. Inspired by instructional scaffolding, we propose STEP (Scaffolded Training through Experience Production), in which a teacher provides a local scaffold at the reasoning obstacle exposed by a student's failed attempt, allowing the student to complete the remaining reasoning and produce a verified successful trace. During training, STEP includes the scaffold explicitly in the input and concentrates supervision on the scaffold and the student continuation that follows it, turning assisted experience into a learning resource. We evaluate STEP after removing the scaffold. Across two student sizes, two teacher models, and five mathematical reasoning benchmarks, STEP raises the scaffold-free MATH500 mean@4 of Qwen2.5-3B-Instruct from 57.15% to 62.05% in the Qwen3-235B teacher setting, exceeding P-ALIGN by 4.80 percentage points. These results suggest that local teacher scaffolds can turn successful experience that the student could not otherwise generate independently into stronger independent reasoning ability.

1 INTRODUCTION

Recent progress in the reasoning abilities of large language models offers new supervision signals for training small models. Chain-of-thought prompting improves complex reasoning by explicitly generating intermediate steps (Wei et al., 2022), and reinforcement learning for reasoning encourages behaviors such as verification and self-correction (DeepSeek-AI, 2025). However, a reasoning trace that solves a hard problem is not necessarily a good trace for a small model to learn from, because differences in reasoning ability and generation distribution between teacher and student affect how useful such traces are for training (Li et al., 2025). A central question in capability transfer is therefore how to turn the reasoning strength of a strong model into independent reasoning ability in a small one.

Existing work mainly follows two paradigms, and each has its own limitations. The first is teacher distillation, which supervises the student with reasoning traces generated by a teacher model and has become an effective way to transfer complex reasoning abilities (Hsieh et al., 2023; DeepSeek-AI, 2025). Yet a strong teacher does not automatically produce traces with high instructional value. A stronger teacher or more elaborate reasoning does not necessarily yield a better student, and effective supervision has to balance informativeness against learnability for the student (Li et al., 2025).

The second paradigm is self-training, which turns the student's own successful traces into supervision through repeated generation, verification, and learning (Zelikman et al., 2022; Singh et al., 2024). This reduces reliance on teacher demonstrations, but on hard tasks the student can only collect successful traces within the reach of its own exploration, and iterative self-training can face

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.

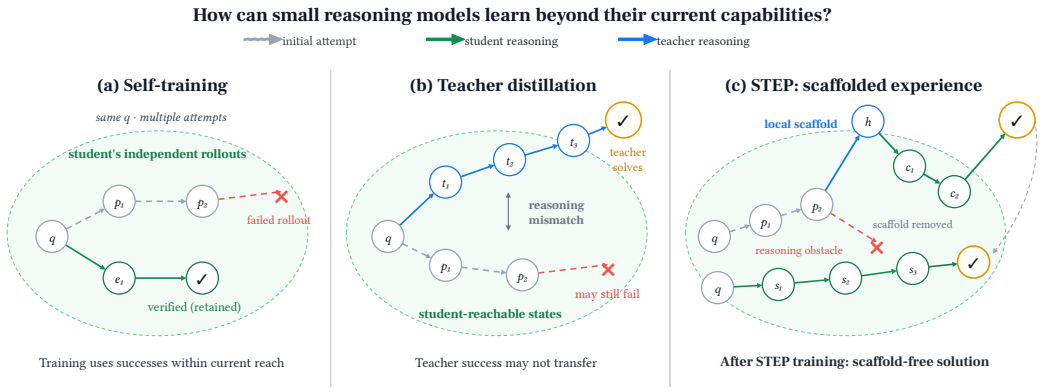

Figure 1: Three routes to reasoning experience. (a) Self-training retains successes the student can generate independently, leaving failed attempts without successful traces. (b) Teacher distillation supplies correct teacher-generated reasoning, but it may not transfer to independent student reasoning. (c) STEP uses a local teacher scaffold to enable a successful student continuation; after training, the student solves the illustrated question independently, without the scaffold.

diminishing gains as the available successful experience becomes limited (Singh et al., 2024). Both limitations lead to the same question: how can we obtain reasoning experience that extends the capability boundary of the student while remaining learnable for it? Figure 1 illustrates these limitations and the scaffolded alternative explored in this work.

To combine the strengths of both paradigms, prior work introduces teacher guidance into the generation and learning process of the student, for example through teacher prefixes, distribution-level supervision on student traces, and interleaved teacher-student generation (Liu et al., 2026; Agarwal et al., 2024; Xu et al., 2025). These methods raise a further question of how teacher guidance and autonomous student reasoning should be combined: where the teacher should intervene, how much help it should give, and which parts of the reasoning should be left to the student. We argue that this division of labor largely determines the learning value of assisted experience.

Instructional scaffolding offers a useful view of this division of labor. In scaffolding, a teacher provides temporary support tailored to the specific difficulties of a learner and gradually hands responsibility for the task back to the learner, thereby fostering independent competence (van de Pol et al., 2010). Following this idea, we use local teacher scaffolds to help the student get past reasoning obstacles and leave the subsequent reasoning to the student. Teacher guidance then does more than supply reasoning information, as it also creates conditions under which the student can obtain successful experience that it could not obtain on its own. From this perspective, we study how a small reasoning model, with the help of temporary external scaffolds, can learn from its own generated experience reasoning abilities that go beyond its current capability boundary.

We propose STEP (Scaffolded Training through Experience Production), which constructs successful reasoning experience for the student through local teacher scaffolds. Starting from a failed attempt of the student, STEP locates the critical reasoning obstacle, keeps the correct prefix before the obstacle, and asks the teacher to provide a local scaffold that connects naturally to this prefix. The student then completes the remaining reasoning, and we keep the resulting trace if it passes verification. During training, STEP treats the teacher scaffold as a conditional input and concentrates supervision on the scaffold and the student continuation that follows it, so that the training data preserve the role of the student in the subsequent reasoning.

We evaluate STEP on five mathematical reasoning benchmarks with two different teachers. The results show that STEP turns successful experience obtained with scaffold support into stronger independent reasoning ability. With the two teachers, STEP improves the unscaffolded MATH500 mean@4 of Qwen2.5-3B-Instruct by 4.90 and 3.92 percentage points, respectively, and outperforms P-ALIGN (Liu et al., 2026), a recent prefix-alignment distillation method, by 4.80 and 6.10 percentage points.

Our contributions are summarized as follows:

- **Experience-centered perspective.** We study how the organization of teacher guidance and autonomous student reasoning determines whether assisted reasoning can become useful training experience.
- **STEP framework.** We develop STEP to construct verified successful traces through targeted teacher scaffolds and student-generated continuations. During training, STEP conditions on the scaffold and masks the student prefix from the loss, concentrating supervision on the teacher repair and student continuation.
- **Analysis of experience transfer.** Across teachers, benchmarks, and controlled ablations, we show that STEP improves independent reasoning over representative distillation methods and identify local scaffolds and selective supervision as key factors in this transfer.

2 Related Work

Reasoning distillation and teacher–student alignment. Reasoning distillation uses reasoning generated by a teacher to give the student intermediate supervision beyond answer labels (Ho et al., 2023; Hsieh et al., 2023; DeepSeek-AI, 2025). A correct teacher trace, however, is not necessarily suitable for the student to learn from directly. Several methods address this gap between teacher and student. Orca 2 (Mitra et al., 2023) trains small models to choose a strategy for each task, such as step-by-step reasoning, recall-then-generate, or direct answering. Mix Distillation (Li et al., 2025) mixes long and short reasoning chains, or traces generated by models of different sizes, to adjust the reasoning complexity that the student is exposed to. GKD (Agarwal et al., 2024) distills on sequences generated by the student using token-level feedback from the teacher, and MiniLLM (Gu et al., 2024) optimizes a reverse KL objective on student samples. These methods improve the match between the teacher signal and the student, but they do not tie teacher support to the specific points where the student gets stuck in its own reasoning traces.

Learning to reason from self-generated experience. Self-training turns a model's own successful traces into new supervision through a loop of generation, verification, and training (Zelikman et al., 2022; Singh et al., 2024). When independent sampling rarely yields correct traces, prior work changes the conditions under which experience is generated. STaR (Zelikman et al., 2022) uses the correct answer as a hint to generate reasoning traces for problems the model initially fails, and ReST-MCTS* (Zhang et al., 2024) uses process rewards and tree search to expand the pool of usable traces. Another line of work draws learning signals directly from failed attempts. Self-Explore (Hwang et al., 2024) locates the first error to construct fine-grained feedback, while RISE (Qu et al., 2024) and SCoRe (Kumar et al., 2025) build self-correction ability through multi-turn training and reinforcement learning, respectively. Recent hint-based self-improvement likewise uses failed rollouts to synthesize targeted guidance before self-distillation (Wang et al., 2026). STEP sits between these two lines. Like the latter, it starts from a specific failure state of the student, but like the former, it turns this failure into a successful trace, one that the student itself completes.

Scaffolding and teacher-guided experience generation. Instructional scaffolding adjusts support to the current performance of the learner, and gradually withdraws this support and transfers responsibility for the task as learning progresses (Wood et al., 1976; van de Pol et al., 2010). This idea appears in several forms in model training. IOA (He et al., 2026) builds a progressive curriculum based on knowledge gaps and adjusts how knowledge is expressed. P-ALIGN (Liu et al., 2026) adaptively truncates teacher reasoning prefixes and pairs them with correct continuations generated by the student to form supervision samples. SKD (Xu et al., 2025) replaces student candidate tokens that fall outside the top-K of the teacher distribution during sampling, and then lets the student continue generating. These methods arrange teacher support at the level of the curriculum, the teacher prefix, and token sampling, respectively. Context-aware, stepwise mathematical hints provide a related inference-time setting (Hossain et al., 2026). STEP instead applies a local repair at the reasoning obstacle exposed by an actual student attempt, and uses both the teacher repair step and the verified student continuation as supervision targets.

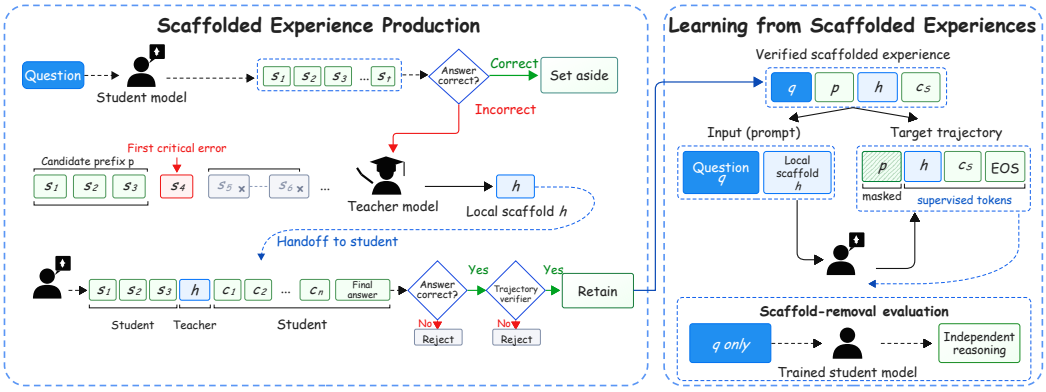

Figure 2: Overview of STEP. Left: the teacher identifies the first critical error in an unsuccessful student trace, retains the preceding student prefix, and provides a local scaffold. The student completes the reasoning; only traces passing answer and trajectory verification are retained. Right: training conditions on the question and an explicitly marked teacher scaffold. The prefix serves as context without loss, while supervision covers the scaffold, student continuation, and end-of-sequence token (EOS). Evaluation uses the question alone, without the scaffold.

3 STEP: Scaffolded Training through Experience Production

In this section, we present STEP (Scaffolded Training through Experience Production), which uses local teacher scaffolds to help student models generate successful reasoning experiences and learn from them to improve independent reasoning. Figure 2 provides an overview. We introduce the learning setup in Section 3.1, describe scaffolded experience production in Section 3.2, and present the training inputs and selective supervision objective in Section 3.3.

3.1 Learning Setup

We are given a dataset $\mathcal{D} = \{(q_i, a_i)\}_{i=1}^{N}$ of questions and reference answers, a student model π_S, and a teacher model π_T. STEP uses local scaffolds from the teacher to construct successful reasoning experience, and trains a model π_θ, initialized from π_S, on this experience to improve its independent reasoning ability.

We write a scaffolded candidate trace as $y = p \oplus h \oplus c_S$, where p is the prefix generated by the student before the reasoning obstacle, h is the local scaffold provided by the teacher, c_S is the continuation that the student generates after the scaffold, and $\oplus$ denotes sequence concatenation. A candidate trace is used for training only if it passes verification.

3.2 Scaffolded Experience Production

STEP starts from failed attempts of the student, adds a teacher scaffold aimed at the specific obstacle, and lets the student complete the remaining reasoning. Questions that the student already answers correctly on its own are set aside and used as a control in our analysis experiments.

Locating reasoning obstacles. For a question-answer pair (q, a), the student first generates a trace $\tilde{y} \sim \pi_S(\cdot \mid q)$ without assistance, and a programmatic answer verifier $V_{\text{ans}}(\tilde{y}, a)$ checks whether its final answer is correct. If the trace fails, we split it into reasoning steps $\tilde{y} = (s_1, \ldots, s_m)$ and ask the teacher to diagnose it, $(\hat{k}, d) = \text{Diag}_{\pi_T}(q, \tilde{y})$, where $\hat{k}$ is the position of the first critical error and d describes the obstacle. The prefix $p = s_1 \oplus \cdots \oplus s_{\hat{k}-1}$ is taken directly from the original student trace, so the erroneous step and everything after it are removed. If the error occurs at the first step, the prefix is empty. Samples in which no obstacle can be located are discarded.

Local scaffolds and student continuation. Given the question, the prefix, and the diagnosis, the teacher writes a local scaffold, and the student then continues the reasoning from the repaired

context:

$$h \sim \pi_T(\cdot \mid q, p, d),$$
$$c_S \sim \pi_S(\cdot \mid q, p \oplus h). \tag{1}$$

The scaffold follows the notation and context of the prefix and adds or corrects the key steps needed to resolve the obstacle, without giving a full solution or revealing the final answer. The prefix, the scaffold, and the student continuation together form the candidate trace.

Experience verification and collection. For a candidate trace y of (q, a), let $\tau = (q, \tilde{y}, p, h, c_S)$ denote the corresponding record. We keep a sample only if $V_{\text{ans}}(y, a) = 1$ and $V_{\text{traj}}(\tau) = 1$. The model-based trace verifier V_{traj} checks that the prefix is valid, that the scaffold addresses the obstacle without revealing the final answer, and that the student continuation follows naturally from the preceding context. We keep at most one successful trace per question, which gives the training set $\mathcal{D}_{\text{STEP}} = \{(q_i, p_i, h_i, c_{S,i})\}_{i=1}^{M}$, where M is the number of retained samples.

3.3 Learning from Scaffolded Experiences

STEP trains the student on the collected traces using scaffold-conditioned inputs and selective supervision.

Scaffold-conditioned input. For each sample, we build the input $x_i = \mathcal{I}(q_i, h_i)$ from the question and the local scaffold, and explicitly mark the scaffold as coming from the teacher. Unlike question-only training, STEP explicitly conditions learning on the teacher scaffold that enabled the student's successful continuation. The target is $y_i = p_i \oplus h_i \oplus c_{S,i}$. The scaffold therefore appears twice: in the input, it provides the support under which the trace was generated, and in the target, it contains the key reasoning steps that the model should learn.

Selective trace supervision. The student prefix serves as context, but we compute no loss on its tokens. Supervision covers only the teacher scaffold, the student continuation, and the end-of-sequence token EOS. Let $(y_{i,1}, \ldots, y_{i,T_i})$ denote the target token sequence with EOS appended, and let b_i be the number of prefix tokens in this sequence. For a minibatch $\mathcal{B}$, the training objective is

$$\mathcal{L}_{\mathcal{B}}(\theta) = -\frac{1}{Z_{\mathcal{B}}} \sum_{i \in \mathcal{B}} \sum_{t=b_i+1}^{T_i} \log \pi_\theta \left(y_{i,t} \mid x_i, y_{i,<t} \right), \tag{2}$$

where $Z_{\mathcal{B}} = \sum_{i \in \mathcal{B}} (T_i - b_i)$ is the total number of supervised tokens. The input prompt is excluded from the loss. Under this objective, the model learns both the key steps supplied by the teacher and how the student continues the reasoning from them.

4 Evaluation

Datasets. Following the procedure in Section 3.2, we construct training data from the MATH training set. Each reported teacher–student setting uses 330 training problems and one trajectory per problem. We evaluate on MATH500, AIME24, AIME25, AMC23, and OlympiadBench-EN. The training pools and benchmark details for each setting are specified in Appendix A.

Baselines. We compare STEP with a zero-shot baseline and three SFT-based baselines. In the zero-shot setting, Base generates solutions directly from the problems without additional training. Teacher-Direct supervises teacher-completed reasoning targets. P-ALIGN (Liu et al., 2026) adaptively selects a minimal sufficient prefix from a teacher trajectory, generates a student continuation conditioned on that retained prefix, and fine-tunes on the concatenation of the retained prefix and the student continuation with the full concatenated target supervised. In contrast, STEP obtains the prefix from the student's failed trace, inserts a teacher's local repair at the first diagnosed obstacle, masks the student prefix, and supervises the repair together with the student's continuation. SKD-Local, inspired by SKD (Xu et al., 2025), uses the same student prefix and local scaffold as STEP, but has the teacher generate the remaining reasoning. SKD-Local uses the same training input construction and supervised token range as STEP.

Evaluation. For each test problem, we sample four independent responses and report mean@4 and pass@4. mean@4 is the average accuracy over the four generations, while pass@4 is the proportion

Table 1: Overall scaffold-free reasoning performance (%) on five mathematical benchmarks. We report mean@4 and pass@4 averaged over three runs; the best result for each metric within a student–teacher setting is bolded. Dataset and experimental details are provided in Appendix A.

	MATH500		AIME24		AIME25		AMC23		OlympiadBench-EN	
Method	mean@4	pass@4	mean@4	pass@4	mean@4	pass@4	mean@4	pass@4	mean@4	pass@4
Student: Qwen2.5-3B-Instruct × Teacher: Qwen3-235B										
Base	57.15	72.07	6.67	12.22	1.11	4.44	34.58	59.17	24.54	39.65
Teacher-Direct	55.40	73.07	3.33	11.11	**3.89**	**12.22**	30.42	53.33	23.41	38.32
P-ALIGN	57.25	73.67	4.17	11.11	3.06	8.89	33.96	**61.67**	24.01	38.77
SKD-Local	54.73	73.40	6.39	**16.67**	3.33	7.78	31.04	55.83	22.44	37.19
STEP	**62.05**	**74.80**	4.44	11.11	1.94	6.67	**38.96**	**61.67**	**25.80**	**40.05**
Student: Qwen2.5-3B-Instruct × Teacher: DeepSeek-V4-Flash										
Base	57.15	72.07	6.67	12.22	1.11	4.44	34.58	59.17	24.54	**39.65**
Teacher-Direct	51.33	69.40	3.61	8.89	1.39	5.56	28.33	49.17	18.95	32.94
P-ALIGN	54.97	71.73	3.89	8.89	2.22	4.44	32.71	59.17	21.69	36.69
SKD-Local	55.88	71.73	2.50	8.89	3.06	7.78	33.33	56.67	22.51	37.04
STEP	**61.07**	**74.00**	**7.50**	**16.67**	**3.33**	**11.11**	**37.29**	**60.83**	**25.17**	38.42
Student: Qwen2.5-1.5B-Instruct × Teacher: DeepSeek-V4-Flash										
Base	42.70	62.67	1.67	4.44	0.83	3.33	18.88	38.96	15.34	29.57
Teacher-Direct	43.10	63.13	1.67	5.56	0.28	1.11	19.08	37.75	14.70	28.14
P-ALIGN	**47.52**	**66.47**	**2.78**	**11.11**	**1.39**	**4.44**	**22.89**	**42.97**	**16.48**	29.28
SKD-Local	41.98	61.80	1.39	4.44	0.56	2.22	17.37	34.14	13.18	26.16
STEP	45.73	64.20	**2.78**	**11.11**	0.56	2.22	18.27	37.75	16.18	**29.72**

Table 2: Ablation of scaffold conditioning (SC) and prefix masking (PM) on MATH500 (%). SC includes the local teacher scaffold in the training input; PM excludes the student prefix from the loss. The all@4 metric counts problems answered correctly in all four samples. Parenthesized values give changes from Base in percentage points. Results are averaged over three runs. Detailed settings are in Appendix A.5.

Method	Training input	Loss on p	mean@4	pass@4	all@4
Base	–	–	57.15	72.07	39.40
STEP w/o SC & PM	q	✓	57.63 (+0.48)	73.20 (+1.13)	40.13 (+0.73)
STEP w/o SC	q	×	59.90 (+2.75)	74.07 (+2.00)	44.67 (+5.27)
STEP w/o PM	$q + h$	✓	58.42 (+1.27)	73.33 (+1.26)	41.67 (+2.27)
STEP	$q + h$	×	**62.05** (+4.90)	**74.80** (+2.73)	**48.20** (+8.80)

of problems for which at least one generation produces a correct answer. We compare generated answers with the reference answers using a programmatic verifier. The decoding and scoring settings are given in Appendix A.4.

Implementation Details. We use Qwen2.5-3B-Instruct and Qwen2.5-1.5B-Instruct as student models, and Qwen3-235B-A22B-Instruct-2507 and DeepSeek-V4-Flash as teacher models, yielding the three student–teacher settings in Table 1. All training methods use full-parameter supervised fine-tuning in BF16 for three epochs. We use AdamW with a learning rate of 1×10^{-5}, a warmup ratio of 10%, a per-device batch size of 1, and 16 gradient-accumulation steps. We use training seeds 0, 1, and 2. Complete hyperparameter settings are provided in Appendix A.2.

4.1 Main Results

Table 1 compares STEP with Base, Teacher-Direct, P-ALIGN, and SKD-Local on five mathematical reasoning benchmarks. On Qwen2.5-3B-Instruct, STEP achieves the highest MATH500 mean@4 and pass@4 with both teachers. With Qwen3-235B and DeepSeek-V4-Flash as teachers, STEP reaches a mean@4 of 62.05% and 61.07%, respectively, which is 4.90 and 3.92 percentage points above Base and 4.80 and 6.10 points above P-ALIGN. Teacher-Direct, by comparison, obtains

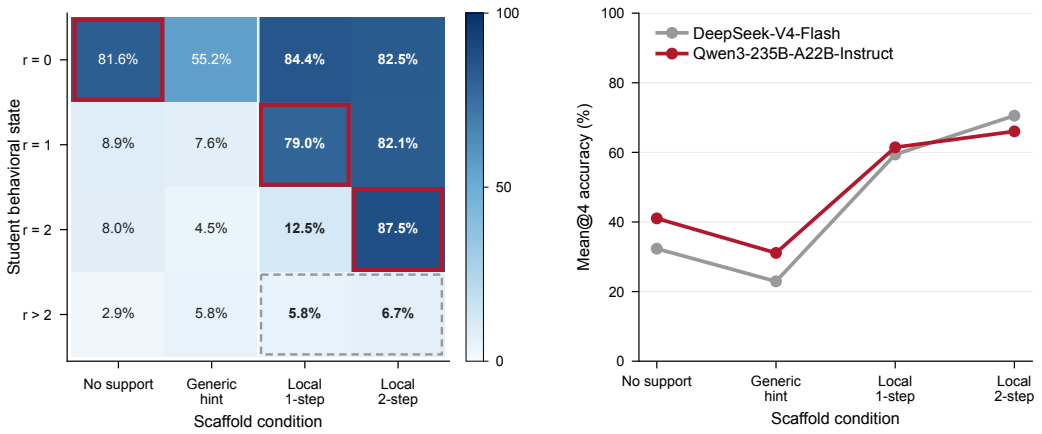

Figure 3: Behavior state and scaffold dose. The left panel shows the untuned student's scaffold-response states under the DeepSeek-V4-Flash teacher, and the right panel reports mean@4 as the amount of local support increases for the two teacher models. Here, $r = 0$ denotes questions mastered without additional scaffold; a matched first step restores many blocked problems, while a second step primarily helps cases that remain unresolved after the first intervention.

MATH500 mean@4 of 55.40% and 51.33%, respectively, both below Base's 57.15% by 1.75 and 5.82 percentage points. Training on correct teacher solutions can therefore hurt the student when the reasoning paths do not fit it, which underlines the need to match supervision to the student.

We next compare STEP with SKD-Local to examine who should complete the reasoning after the scaffold. The two methods use the same training questions, student prefixes, local scaffolds, input construction, and supervision scope. They differ only in whether the rest of the trace is generated by the student or by the teacher. STEP outperforms SKD-Local on MATH500 mean@4 by 7.32 and 5.19 percentage points with the two teachers, suggesting that local teacher help turns into independent problem-solving ability more effectively when the student carries out the subsequent reasoning itself. The gains are not limited to MATH500. In both 3B settings, STEP achieves the highest mean@4 on AMC23 and OlympiadBench-EN, and with the DeepSeek teacher it achieves the highest mean@4 on all five benchmarks. On the smaller Qwen2.5-1.5B-Instruct with the DeepSeek teacher, STEP also improves MATH500 mean@4 over Base, from 42.70% to 45.73%.

4.2 Ablation Study

We conduct an ablation study on the MATH500 test set to measure the contribution of two key training designs in STEP, namely scaffold-conditioned input (SC) and prefix masking (PM). We compare five settings. Base is the zero-shot baseline. STEP w/o SC & PM removes both scaffold-conditioned input and prefix masking. STEP w/o SC removes only scaffold-conditioned input and keeps prefix masking. STEP w/o PM keeps scaffold-conditioned input but disables prefix masking, so that the loss is also computed on the prefix. The full STEP uses both designs. Apart from the frozen Base model, all settings use the same training data and training configuration, and all are evaluated without scaffolds. Detailed experimental settings are provided in Appendix A.5.

Removing scaffold-conditioned input lowers MATH500 mean@4 from 62.05% to 59.90%, a drop of 2.15 percentage points relative to the full STEP. This suggests that keeping the teacher scaffold explicit during training helps the model learn how the subsequent reasoning depends on the scaffold. Removing prefix masking leads to a larger drop of 3.63 percentage points, to 58.42%. Supervising the student prefix thus appears to dilute the training signal, and concentrating the loss on the teacher scaffold and the student continuation is more effective for learning the reasoning that the student is responsible for after the scaffold. When both designs are removed, mean@4 falls to 57.63%, which is 4.42 percentage points below the full STEP and close to the 57.15% of Base. Since scaffold-conditioned input and prefix masking act on the input conditioning and the supervision scope of

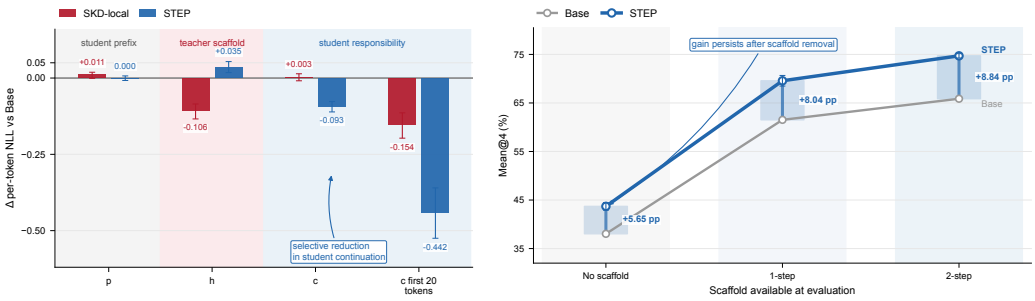

Figure 4: Responsibility-selective learning and scaffold-dose transfer. (a) Segment-level teacher-forced NLL shows that STEP's largest change is concentrated in the student continuation c, rather than the student prefix p. (b) STEP–Base mean@4 on a fixed MATH500-derived subset under no scaffold, one-step scaffold, or two-step scaffold; blue lifts denote the corresponding gains. STEP retains a positive gain in the no-scaffold condition.

the training experience, respectively, these results suggest that the two designs complement each other and together account for most of the gain in turning assisted traces into independent reasoning ability.

4.3 Scaffold Dose and Student State

We examine whether the amount of local scaffolding should match the student's current reasoning state. Using the untuned Qwen2.5-3B-Instruct, we fix the questions and student prefixes and compare immediate continuation under four conditions: no additional support, a generic hint, a one-step local scaffold, and a two-step local scaffold. We report results separately for the problem pools built with DeepSeek-V4-Flash and Qwen3-235B-A22B-Instruct. Detailed configuration is provided in Appendix A.6.2.

As shown in Figure 3, the generic hint instead lowers mean@4, from 32.32% to 22.93% on the DeepSeek pool and from 41.03% to 31.11% on the Qwen3 pool. A one-step local scaffold targeting the specific obstacle raises mean@4 to 59.39% and 61.45%, and a two-step scaffold further raises it to 70.54% and 66.03%. This pattern indicates that the benefit of local scaffolding comes from an actionable repair matched to the current obstacle, rather than from simply adding context.

To test whether the appropriate amount of scaffolding depends on the student's state, we stratify the DeepSeek pool and define the operational recovery level r as the smallest scaffold level at which the untuned student reaches mean@4 $\geq 50\%$. For $r = 1$ problems, one step raises mean@4 from 8.93% to 79.02%; for $r = 2$ problems, the second step raises it from 12.50% to 87.50%. For $r > 2$ problems, which remain unrecovered after two steps, the two-step condition reaches only 6.73%. Thus, the first step mainly crosses the initial obstacle, while additional support benefits cases that remain unresolved after it rather than all problems uniformly. This supports STEP's conditional-scaffold design: scaffold effectiveness depends on matching the student's operational recovery level, rather than on scaffold quantity alone.

4.4 Learning the Scaffold Handoff

STEP divides the assisted trace between teacher support and student responsibility. We first ask whether training changes the model primarily at the student-owned continuation. As shown in Figure 4a, STEP leaves the prefix NLL essentially unchanged while reducing the NLL of c, especially its first 20 tokens after the scaffold handoff. SKD-Local does not show a comparable reduction in the responsibility segment; its overall c-segment NLL does not decrease.

Figure 4b tests how the trained model uses different scaffold doses on a fixed MATH500-derived subset. Under no additional scaffold, one-step support, and two-step support, Base reaches mean@4 scores of 38.04%, 61.52%, and 65.87%, while STEP reaches 43.70%, 69.57%, and 74.71%, corresponding to gains of $+5.65$, $+8.04$, and $+8.84$ percentage points. STEP therefore improves both

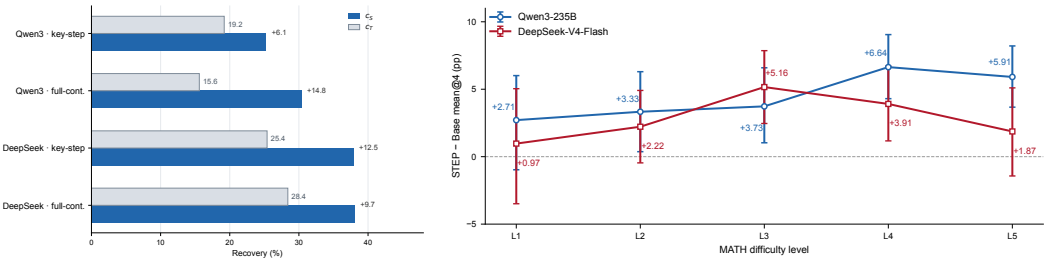

Figure 5: From student-owned continuations to scaffold-free transfer. (Left) On matched error states, we compare supervised training on student-completed continuations (c_S) and teacher-completed continuations (c_T). Bars show key-action and full-continuation recovery, and labels show the paired $c_S - c_T$ difference. (Right) We report the STEP–Base mean@4 gain under scaffold-free MATH500 evaluation across difficulty levels 1–5; curves correspond to Qwen3-235B and DeepSeek-V4-Flash, with error bars denoting paired 95% confidence intervals. Detailed settings are provided in Appendix A.6.4.

scaffold-free and scaffold-assisted behavior on this subset, with the largest gain under the two-step condition. The detailed subset and evaluation protocol are provided in Appendix A.6.1.

4.5 From Student-Owned Continuations to Scaffold-Free Transfer

What part of a scaffolded trajectory is most useful for training? We compare two matched targets: c_S, completed by the student after receiving the local scaffold, and c_T, completed by the teacher from the same diagnosed state. The two targets use the same questions, diagnosed states, update setting, and evaluation protocol. As shown in Figure 5, the student-completed target yields higher recovery on both teacher lines. With Qwen3-235B, the paired gain is $+6.1$ percentage points for key-action recovery and $+14.8$ points for full-continuation recovery; with DeepSeek-V4-Flash, the corresponding gains are $+12.5$ and $+9.7$ points. Under this controlled target comparison, the student-completed continuation is the more effective local training target. Detailed target construction and pairing rules are provided in Appendix A.6.4.

We next examine how this experience transfers to independent solving at different difficulty levels. We compare STEP and Base on scaffold-free MATH500 evaluation, stratified post hoc by Levels 1–5. STEP has a positive point estimate of the STEP–Base mean@4 difference at every level for both teacher lines. The largest gain occurs at Level 4 for Qwen3-235B ($+6.64$ points) and at Level 3 for DeepSeek-V4-Flash ($+5.16$ points). This pattern shows that locally scaffolded, student-completed experience supports scaffold-free reasoning across the difficulty range, with the magnitude of transfer varying by teacher and difficulty level.

5 Conclusion

We presented STEP (Scaffolded Training through Experience Production), which uses a local teacher scaffold targeted at the student's reasoning obstacle and lets the student complete the subsequent reasoning, thereby constructing successful experience that is suitable for learning.

Across different teacher models and multiple mathematical reasoning benchmarks, STEP improves the student's independent problem-solving ability. Our analyses further show that supervising the student-owned continuation and enabling the model to use local scaffolds effectively are central to converting temporary teacher support into independent reasoning ability.

AI Use Statement

During the preparation of this manuscript, the authors used AI tools to assist with language polishing, LaTeX formatting, and the organization of experimental data. All AI-assisted outputs were reviewed

and independently verified by the authors, who take full responsibility for the final manuscript and reported results.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2306.13649.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=5h0qf7IBZZ.

Bowei He, Yankai Chen, Xiaokun Zhang, Linghe Kong, Philip S. Yu, Xue Liu, and Chen Ma. Pedagogically-inspired data synthesis for language model knowledge distillation. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2602.12172.

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)*, 2023. URL https://aclanthology.org/2023.acl-long.830/.

Jawad Hossain, Xiangyu Guo, Jiawei Zhou, and Chong Liu. HintMR: Eliciting stronger mathematical reasoning in small language models. *arXiv preprint arXiv:2604.12229*, 2026. URL https://arxiv.org/abs/2604.12229.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. *arXiv preprint arXiv:2305.02301*, 2023.

Hyeonbin Hwang, Doyoung Kim, Seungone Kim, Seonghyeon Ye, and Minjoon Seo. Self-Explore: Enhancing mathematical reasoning in language models with fine-grained rewards. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024. URL https://aclanthology.org/2024.findings-emnlp.78/.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D. Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, Lei M. Zhang, Kay McKinney, Disha Shrivastava, Cosmin Paduraru, George Tucker, Doina Precup, Feryal Behbahani, and Aleksandra Faust. Training language models to self-correct via reinforcement learning. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2409.12917.

Yuetai Li, Xiang Yue, Zhangchen Xu, Fengqing Jiang, Luyao Niu, Bill Yuchen Lin, Bhaskar Ramasubramanian, and Radha Poovendran. Small models struggle to learn from strong reasoners. *arXiv preprint arXiv:2502.12143*, 2025.

Zhenghao Liu, Zhuoyang Wu, Xinze Li, Yukun Yan, Shuo Wang, Zulong Chen, Yu Gu, Ge Yu, and Maosong Sun. Long-chain reasoning distillation via adaptive prefix alignment. *arXiv preprint arXiv:2601.10064*, 2026.

Arindam Mitra, Luciano Del Corro, Shweti Mahajan, Andres Codas, Clarisse Simoes, Sahaj Agarwal, Xuxi Chen, Anastasia Razdaibiedina, Erik Jones, Kriti Aggarwal, Hamid Palangi, Guoqing Zheng, Corby Rosset, Hamed Khanpour, and Ahmed Awadallah. Orca 2: Teaching small language models how to reason. *arXiv preprint arXiv:2311.11045*, 2023.

Yuxiao Qu, Tianjun Zhang, Naman Garg, and Aviral Kumar. Recursive introspection: Teaching language model agents how to self-improve. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2407.18219.

Avi Singh, John D. Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J. Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron Parisi, Abhishek Kumar, Alex Alemi, Alex Rizkowsky, Azade Nova, Ben Adlam, Bernd Bohnet, Gamaleldin Elsayed, Hanie Sedghi, Igor Mordatch, Isabelle Simpson, Izzeddin Gur, Jasper Snoek, Jeffrey Pennington, Jiri Hron, Kathleen Kenealy, Kevin Swersky, Kshiteej Mahajan, Laura Culp, Lechao Xiao, Maxwell L. Bileschi, Noah Constant, Roman Novak, Rosanne Liu, Tris Warkentin, Yundi Qian, Yamini Bansal, Ethan Dyer, Behnam Neyshabur, Jascha Sohl-Dickstein, and Noah Fiedel. Beyond human data: Scaling self-training for problem-solving with language models. *Transactions on Machine Learning Research*, 2024. URL https://arxiv.org/abs/2312.06585.

Janneke van de Pol, Monique Volman, and Jos Beishuizen. Scaffolding in teacher–student interaction: A decade of research. *Educational Psychology Review*, 22(3):271–296, 2010. doi: 10.1007/s10648-010-9127-6.

Shengjie Wang, Guanghe Li, Zonghan Yang, and Yang Gao. Hindsight hint distillation: Scaffolded reasoning for swe agents from cot-free answers. *arXiv preprint arXiv:2605.11556*, 2026. URL https://arxiv.org/abs/2605.11556.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2201.11903.

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. doi: 10.1111/j.1469-7610.1976.tb00381.x.

Wenda Xu, Rujun Han, Zifeng Wang, Long T. Le, Dhruv Madeka, Lei Li, William Yang Wang, Rishabh Agarwal, Chen-Yu Lee, and Tomas Pfister. Speculative knowledge distillation: Bridging the teacher-student gap through interleaved sampling. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.11325.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. Star: Bootstrapping reasoning with reasoning. *arXiv preprint arXiv:2203.14465*, 2022.

Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. ReST-MCTS*: LLM self-training via process reward guided tree search. *arXiv preprint arXiv:2406.03816*, 2024.

A Experimental Details

A.1 Datasets and Benchmarks

Training data. All training examples are constructed from MATH-train following the experience-production procedure in Section 3.2. Each teacher–student setting uses 330 questions and retains one training trajectory per question. The same 330-question set and training trajectory count are used by Teacher-Direct, P-ALIGN, SKD-Local, and STEP within a setting.

Evaluation benchmarks. We evaluate scaffold-free reasoning on five mathematical benchmarks:

- **MATH500**: the held-out test split of MATH, covering 500 competition-style problems.
- **AIME24 and AIME25**: recent AIME competition problem sets that test challenging olympiad-style mathematical reasoning.
- **AMC23**: AMC competition problems used to evaluate generalization to a different contest format.
- **OlympiadBench-EN**: the English mathematical-olympiad subset, which contains more advanced problems and longer reasoning requirements.

A.2 Training Protocol

All trained models use the following common protocol:

- **Models:** Qwen2.5-3B-Instruct and Qwen2.5-1.5B-Instruct are used as students; Qwen3-235B-A22B-Instruct-2507 and DeepSeek-V4-Flash are used as teachers.
- **Optimization:** full-parameter supervised fine-tuning in BF16 with AdamW, learning rate 1×10^{-5}, warmup ratio 0.1, per-device batch size 1, and 16 gradient-accumulation steps.
- **Training budget:** three epochs, corresponding to 63 optimizer steps for the 330-example training set.
- **Randomness:** training seeds 0, 1, and 2 are used for the reported trained methods.

Teacher-Direct uses teacher-completed target trajectories. P-ALIGN follows its prefix-conditioned training construction. STEP and SKD-Local use the same questions, student prefixes, local scaffolds, input construction, and supervised token range; SKD-Local replaces the student continuation c_S with a teacher continuation c_T.

A.3 Prompt Templates

The scaffold-conditioned input used by STEP is instantiated with the following template. The model-specific chat template wraps this content before tokenization.

```
{problem}

Teacher scaffold (local repair):
{h}

Solve the problem yourself from this point.
```

The target sequence is the concatenation

```
{p}{h}{c_S}
```

where p is the student's prefix before the diagnosed obstacle, h is the local teacher scaffold, and c_S is the student's verified continuation after the scaffold. The prefix is retained as context and excluded from the training loss, while the scaffold and continuation are supervised as described in Section 3.3.

A.4 Evaluation and Scoring

All main evaluations use a natural question-only prompt. The decoding and scoring protocol is:

- temperature 0.7 and top-p 0.95;
- maximum generation length of 3,072 tokens;
- four independent responses per problem;
- final-answer verification with the programmatic MATH verifier.

We report mean@4, pass@4, and all@4. mean@4 is the average correctness across the four responses, pass@4 is the fraction of problems with at least one correct response, and all@4 is the fraction with four correct responses. Trained methods are summarized over the three training seeds; Base is evaluated with independent frozen-model repeats under the same decoding protocol.

A.5 Ablation Protocol

The ablation study uses Qwen2.5-3B-Instruct with Qwen3-235B-A22B-Instruct as the teacher. All trained variants share the same 330 MATH-train questions and the same verified $p \oplus h \oplus c_S$ target for each question. They differ only in two training choices:

- **Scaffold conditioning (SC):** whether the local scaffold h is included in the training input;

- **Prefix masking (PM):** whether the student prefix p is excluded from the loss.

All variants supervise the scaffold, the student continuation, and EOS. The full STEP model uses both SC and PM. Each trained variant follows Appendix A.2 and is evaluated without a scaffold.

A.6 Analysis Experiment Protocols

The analysis experiments use the same student and teacher models as the main study and isolate three aspects of STEP: the amount of support, the source of the post-scaffold continuation, and scaffold-free transfer. Unless noted otherwise, correctness is determined with the official MATH verifier.

A.6.1 Responsibility Profile and Scaffold-Dose Transfer

Figure 4a uses the Qwen2.5-3B-Instruct student with the Qwen3-235B-A22B-Instruct-2507 teacher. The held-out analysis set contains 60 problems with no overlap with the training pool or the main MATH500 evaluation set. We score Base, three STEP training seeds, and three SKD-Local training seeds on the same q, p, h, c_S records. For each checkpoint, the target is teacher-forced as $p \oplus h \oplus c_S$; segment boundaries are token-level, the prefix p is not used as a supervised segment, and EOS is excluded from the reported NLL. We report question-level means for p, h, c, and the first 20 tokens of c, together with paired bootstrap confidence intervals and sign-flip permutation tests for the STEP–SKD-Local interaction.

Figure 4b evaluates scaffold-dose transfer on a fixed 115-question MATH500-derived diagnostic subset. The student is Qwen2.5-3B-Instruct and the teacher is Qwen3-235B-A22B-Instruct-2507. Base is a frozen checkpoint and STEP is evaluated across three training seeds. Each problem is evaluated with four independent responses under three fixed prefix conditions: p_0 contains the student prefix without an additional scaffold, p_1 contains the prefix followed by one local scaffold, and p_2 contains the prefix followed by two local scaffolds. Decoding uses temperature 0.7, top-p 0.95, and a 4,096-token generation limit; all outputs are scored with the official MATH verifier. We report mean@4, STEP seed standard deviations, and paired 95% confidence intervals for STEP–Base differences. This diagnostic-subset analysis is distinct from the full 500-question natural-prompt MATH500 results in Table 1.

A.6.2 Behavior State and Scaffold Dose

- **Student:** untuned Qwen2.5-3B-Instruct.
- **Teachers:** DeepSeek-V4-Flash and Qwen3-235B-A22B-Instruct, analyzed in separate teacher-specific pools.
- **Conditions:** continuation from the student prefix p with no additional support, a generic hint, one local scaffold h_1, or two local scaffolds $h_1 \oplus h_2$.
- **Sampling:** four responses per condition, temperature 0.7, top-p 0.95, and a maximum of 4,096 generated tokens; matched conditions share sampling seeds.
- **Metrics:** mean@4 is the primary measure, with pass@4 and all@4 as supplementary measures.

The effective diagnostic pools contain 157 questions for DeepSeek and 131 for Qwen3. For the state-stratified analysis, the DeepSeek pool is partitioned by the smallest scaffold level whose pre-training mean@4 reaches 0.5: $r = 0$ (no additional scaffold required; 53 questions), $r = 1$ (recoverable after one step; 56 questions), $r = 2$ (recoverable after two steps; 22 questions), and $r > 2$ (not recovered within two steps; 26 questions). The label r is computed from the diagnostic rollout and is not a native difficulty label from MATH.

A.6.3 Student versus Teacher Continuation

- **Paired data:** for each teacher, the question q, student prefix p, and local scaffold h are fixed; only the post-scaffold continuation source changes.

- **Continuation sources:** c_S is completed by the student after the scaffold, while c_T is completed by the teacher from the same state.
- **Distributional analysis:** the untuned student teacher-forces both continuations under the same q, p, h context; token NLL and target-token top-1 support are averaged within question and then across questions.
- **Training comparison:** STEP uses c_S and SKD-Local uses c_T. Both use input $q + h$, target $p \oplus h \oplus c$, prefix masking, and supervision on h, c, and native EOS.
- **Ability evaluation:** both models are evaluated on MATH500 with the scaffold-free protocol in Appendix A.4.

A.6.4 From Student-Owned Continuations to Scaffold-Free Transfer

This analysis has two linked parts. First, for each teacher line, we retain 330 questions from its candidate pool for training. For each diagnosed error state, c_S and c_T are constructed from the same question, student prefix, and local scaffold. After training, the scaffold is removed and we measure:

- **Key-action recovery:** whether the model completes the diagnosed next critical action;
- **Full-continuation recovery:** whether the model completes the remaining reasoning and passes the official answer verifier.

The reported values are paired point estimates and $c_S - c_T$ differences within each teacher line.

Second, we reuse the completed STEP and Base evaluations on MATH500. We report scaffold-free mean@4 separately for official MATH difficulty Levels 1–5, using three STEP training seeds per teacher line and four natural responses per problem with temperature 0.7, top-p 0.95, and a 3,072-token generation limit. Error bars are question-level paired 95% confidence intervals for STEP–Base differences. Difficulty level is used only for this post-hoc analysis and does not determine training-question selection or ordering.

A.7 Qualitative Case Study

We provide a qualitative example to make the structure of a STEP training trace concrete. The problem asks for the area of the square inscribed in the region bounded by the parabola $y = x^2 - 8x + 12$ and the x-axis. The student first identifies the vertex $(4, -4)$, recognizes the symmetry about $x = 4$, and observes that the upper vertices of a square with side length s should lie at height $-s$. These statements form the student prefix p.

The teacher scaffold h supplies only the missing geometric constraint. It connects the height of the square to the parabola by writing

$$-s = x^2 - 8x + 12,$$

and then relates the horizontal distance between the two intersections to the side length:

$$\sqrt{16 - 4s} = s.$$

The scaffold resolves the modeling obstacle without completing the algebra or revealing the final answer.

The student continuation c_S then carries out the remaining derivation:

$$16 - 4s = s^2, \qquad s^2 + 4s - 16 = 0,$$

selects the positive root $s = 2(\sqrt{5} - 1)$, and computes the area as

$$s^2 = 24 - 8\sqrt{5}.$$

Thus, the resulting training trace has the form $p \oplus h \oplus c_S$: the teacher contributes the obstacle-specific modeling step, while the student completes the subsequent algebra and reaches the verified answer.