Failures are Treasures: Constructing a Pedagogical Bridge for Agentic Strategy Distillation

Jiaxin Guo[1,2] **Hao Sun**[1,2] **Wenhao Zhang**[3] **Chunyu Yang**[4] **Yan Zhang**[1,2*]

[1]School of Intelligence Science and Technology, Peking University
[2]State Key Laboratory of General Artificial Intelligence, Peking University, Beijing, China
[3]School of Computer Science, Shanghai Jiao Tong University
[4]Ucap Cloud

guojiaxin@stu.pku.edu.cn, zhyzhy001@pku.edu.cn

Abstract

While Large Language Models (LLMs) excel in reasoning tasks, small language models (SLMs) remain fragile, often collapsing after encountering errors. Traditional knowledge distillation focuses on imitating successful trajectories, while existing "learning from mistakes" methods treat errors as auxiliary signals rather than states requiring recoverable policies, leaving the dynamics of failure and recovery in agent settings largely unexplored. Inspired by Donald Schön's theory of reflective practice, we propose P-BRIDGE (Pedagogical Bridge for Reflective Insight and Distillation of Guiding Errors). P-BRIDGE combines reflection-in-action with reflection-on-action, enabling agents to diagnose and correct critical errors during execution while abstracting transferable strategies from contrastive student–teacher trajectories. Experiments across eight benchmarks in controlled environments demonstrate that P-BRIDGE significantly elevates SLM performance—e.g., raising the 2WikiMultiHopQA accuracy of a 0.6B model from 6.2% to 34.2%.

1 Introduction

Recent advances in Large Language Models (LLMs) have enabled a transition from static question answering toward agentic reasoning (Xi et al., 2025; Wang et al., 2024b), where models interact with external tools (Qin et al., 2023) and multi-step decision processes to accomplish tasks. While small language models (SLMs) offer clear advantages in efficiency and deployability, they are notably more fragile in these interactive settings (Shen et al., 2024; Valmeekam et al., 2023).

Despite this shift, existing approaches for transferring reasoning capabilities from large to small models are rooted in the paradigm of knowledge distillation (KD) (Xu et al., 2024). Most of the existing KD methods operate on the assumption that teacher's successful trajectories provide sufficient supervision (Li et al., 2025; Tian et al., 2025). In agentic settings, the dominant objective remains the imitation of what to do when things go right (Kang et al., 2025). However, as illustrated in Figure 1, the student model is only exposed to correct trajectories, making it fragile to the errors it will inevitably encounter in interactive environments.

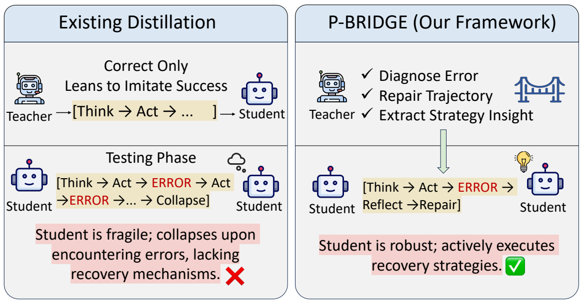

Figure 1: From fragile imitation to recoverable agent policies. P-BRIDGE transforms student failures into reflective supervision, enabling robust error recovery beyond imitating correct trajectories.

Recognizing that robust reasoning requires not just following a correct path, a growing body of research has begun to challenge this assumption through learning from mistakes(LfM) (Sun et al., 2024; Yu et al., 2025; Weng et al., 2025). These approaches acknowledge the value of failures and demonstrate promising gains in performance. However, they treat mistakes as auxiliary training signals—standalone corrections or post-hoc explanations—rather than as states that require recoverable policies. First, corrections are often instance-level, making them difficult to generalize to new problems. Second, they lack a clear notion of recovery dynamics, failing to teach the model how to systematically exit an error state. Third, most LfM research focuses on Chain-of-Thought (CoT) scenarios rather than the more diverse and complex failure patterns found in interactive agent settings.

[*]Corresponding authors

Findings of the Association for Computational Linguistics: ACL 2026, pages 18808–18823

Inspired by Donald Schön's theory of reflective practice (Schön, 2017; Ramage, 2017), which characterizes human expertise as arising from two complementary forms of reflection: *reflection-in-action*, the ability to diagnose and correct errors while acting, and *reflection-on-action*, the ability to abstract strategic insights after the fact. In this paper, we introduce P-BRIDGE (Pedagogical Bridge for Reflective Insight and Distillation of Guiding Errors), a reflective, failure-centered distillation framework on agentic learning. Our framework constructs a pedagogical bridge between agent failures and transferable reasoning competence. It empowers students with three high-level capabilities: failure diagnosis within the execution context, reflexive correction grounded in prior interaction history, and strategic abstraction across diverse failure cases.

Furthermore, motivated by Donald Schön's cognitive parsimony—the principle that effective reasoning favors concise and economical solutions (Schön and Rein, 1994), we incorporate an efficiency-driven selection mechanism. In contrast to existing distillation frameworks, which rarely consider the efficiency of trajectories, our mechanism prioritize correction trajectories that resolve failures with minimal steps and actions, thereby encouraging student models to internalize how to decisively and concisely recover from errors.

We evaluate our approach across multi-hop question answering and mathematical reasoning benchmarks, demonstrating consistent performance improvements over baselines. We also conduct behavioral analyses that measure explicit error recovery, showing that our framework is more capable of correcting mistakes after encountering environmental failures. Notably, these gains are achieved with high data efficiency, relying on a limited number (less than 2k) of error-driven training trajectories.

We summarize our contributions as follows:

- We propose P-BRIDGE, a framework for Recoverable Policy Distillation in multi-step reasoning. Inspired by Schön's reflective practice, it transforms agent failures into high-quality pedagogical signals via trajectory remediation and strategy extraction.
- P-BRIDGE equips models with execution-time reflection for failure diagnosis and promotes concise, non-redundant recovery behaviors by optimizing for trajectory efficiency.
- Extensive experiments across eight benchmarks demonstrate P-BRIDGE consistently elevates the reasoning performance of SLMs.

2 Related Work

2.1 Knowledge Distillation for Reasoning Capabilities

In traditional knowledge distillation (Xu et al., 2024; Do et al., 2025), the objective is to supervise the student model using soft labels or internal representations from the teacher model (Gou et al., 2021; Hinton et al., 2015). In recent years, researchers have leveraged teacher-generated reasoning traces (Zhu et al., 2024; Tian et al., 2025; Zhou et al., 2025b) and Chain-of-Thought (CoT) as supervisory signals. (Li et al., 2024a; Ho et al., 2023; Lee et al., 2024; Wang et al., 2023) This allows student models to learn the underlying reasoning process itself rather than just the final answer (Chih-Yao Chen et al., 2024; Feng et al., 2024). Recently, Knowledge Distillation has been extended to more complex agent architectures. For example, Agent Distillation (Kang et al., 2025) employs CoT trajectories generated by teacher LLMs to guide student reasoning. Nonetheless, the primary objective of these methods remains the replication of the teacher's correct reasoning chains. In contrast, our approach enriches the supervision signal by incorporating teacher insights from student errors.

2.2 Learning from Mistakes and Feedback

Beyond standard KD, an emerging research direction investigates error-aware learning and feedback integration to improve reasoning capabilities (Sun et al., 2024; Yu et al., 2025; Wang et al., 2024a; Li et al., 2024b; Weng et al., 2025). For example, LLMs-as-Instructors (Ying et al., 2024) use teacher models to analyze a student model's incorrect outputs and generate targeted training data. MAPD (Li et al., 2025) asks multiple teachers to identify and explain student mistakes through a simulated peer-review process. SALAM (Wang and Li, 2023) introduce auxiliary agents that collect and summarize LLM error cases to provide guidelines. Unlike existing methods that treat error feedback as an auxiliary signal, we unify teacher-generated insights with correct reasoning trajectories into a single, cohesive supervision signal.

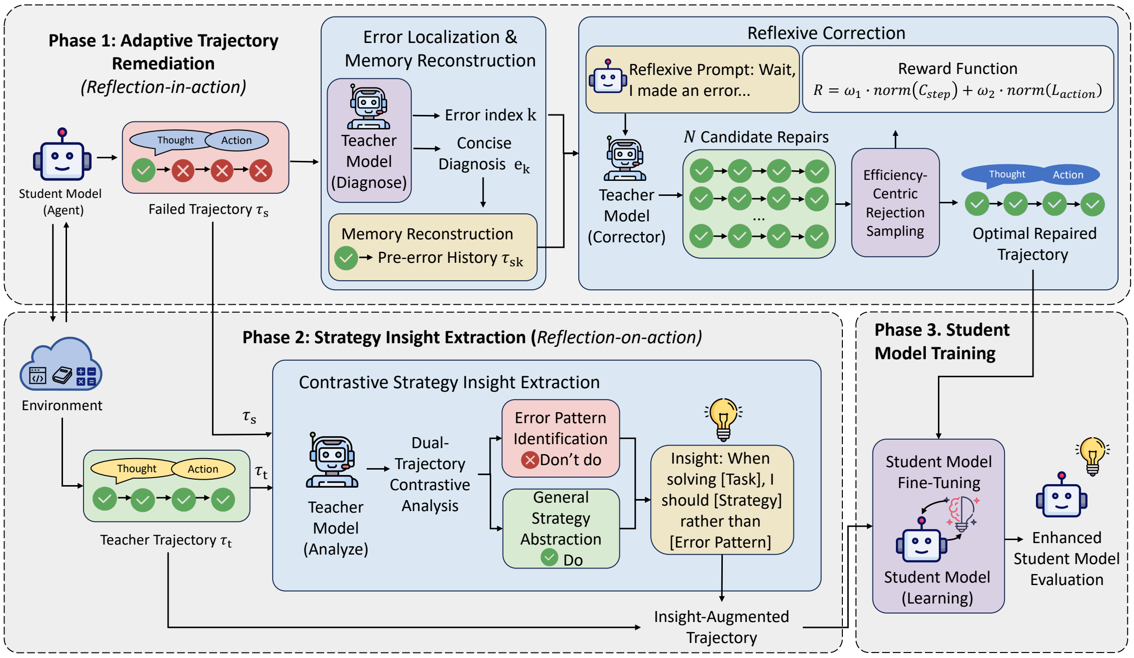

Figure 2: Overview of the proposed P-BRIDGE framework. P-BRIDGE integrates adaptive trajectory remediation, contrastive strategy insight extraction, and student model training to distill efficient error recovery and transferable reasoning strategies from failed agent trajectories.

3 Methodology

3.1 Framework Overview

We introduce P-BRIDGE, an error-aware distillation framework for agentic reasoning that simulates Donald Schön's theory of cognitive evolution through a dual-loop architecture. As in Figure 2, it is designed to enhance agent performance by converting failed student trajectories into high-fidelity pedagogical signals. Our framework comprises two components:

- Adaptive Trajectory Remediation. Representing *reflection-in-action*, a teacher model intervenes in failed student trajectories by identifying the erroneous decision point and performing corrective reasoning while preserving the student's prior interaction history.
- Cross-Trajectory Strategy Insight Extraction. Functioning as *reflection-on-action*, we contrast a student's erroneous trajectory with the teacher's repaired (successful) counterpart for the same task, we extract abstract, transferable problem-solving principles that go beyond instance-level corrections.

Together, these components enable the student model to learn not only what went wrong, but also why it went wrong and how to reason more effectively.

3.2 Adaptive Trajectory Remediation

Guided by the concept of *reflection-in-action*—the expert's ability to stop and rethink while doing, we design a locate–reconstruct–correct pipeline. This mechanism embeds self-correction inductive biases, enabling students to internalize *reflection-in-action* and systematically recover from error states.

3.2.1 Error Localization and Memory Reconstruction

Let a student-generated erroneous trajectory be denoted as $\tau_s = \{(t_i, a_i, o_i)\}_{i=1}^{n}$, where t, a, o represent the agent's thought process, action, and the environmental observation, respectively. We first employ a heuristic filter to detect explicit environment-returned failure signals. In cases where the failure is latent or logical, the teacher model analyzes the student's trajectory step-by-step to pinpoint the error in τ_s. The teacher outputs both the erroneous step index k and a concise diagnosis e_k of the underlying failure. The detailed prompts utilized in our method are provided in Appendix D

Next, we restore contextual memory by preserving the student's pre-error logs ($i < k$) to ensure decision-making continuity. This prefix is re-serialized as the teacher's "antecedent memory," which guarantees that any subsequent correction is strictly grounded in the Agent's original environmental state. Importantly, during the supervised fine-tuning phase, the tokens corresponding to this student-generated prefix are masked from the loss calculation, ensuring that the model optimizes its parameters exclusively based on the teacher-rectified reasoning steps.

3.2.2 Reflexive Correction by Efficiency-Driven Rejection Sampling

During correction, the teacher model continues the trajectory from step k and produces a repaired solution trajectory. We prepend a specialized reflexive prompt to the teacher's step k: "Wait, I made e_k in the previous step. Let me correct this." This design is intended to instill a capacity for *reflexive monitoring* in student models, explicitly signaling a shift in reasoning logic.

Adhering to the principle of *cognitive parsimony*—which suggests that effective reasoning should favor the most economical solution, we implement an Efficiency-Driven Rejection Sampling (EDRS) mechanism. This approach involves performing N independent correction attempts and subsequently employing an objective reward function R to identify the optimal trajectory

$$R = w_1 \cdot \text{norm}(C_{step}) + w_2 \cdot \text{norm}(L_{action}) \quad (1)$$

C_{step} and L_{action} denote the total step count and cumulative action length, respectively. We apply a min-max normalization where the value for the step count is derived as:

$$\text{norm}(C_{step}) = 1 - \frac{C_{step} - \min(\mathbf{C})}{\max(\mathbf{C}) - \min(\mathbf{C})} \quad (2)$$

where $\mathbf{C}$ represents the set of step counts observed across all N candidate trajectories generated during the sampling phase. The normalization for the L_{action} follows an identical logic. Such a selection mechanism prioritizes trajectories that are not only correct but also computationally efficient, which facilitates the distillation of a more concise and robust policy into the student model.

3.3 Strategy Insight Extraction from Contrastive Trajectories

While fine-grained corrections offer local alignment, *reflection-on-action* is required to generalize principles from experience and prevent overfitting to specific trajectories. To address this, we introduce a macro-level Strategic Insight Extraction mechanism to transform implicit teacher logic into explicit and transferable behavioral guidelines.

Concretely, for each paired sample (τ_s, τ_t), where the student fails and the teacher succeeds on the same task, we prompt the teacher model to perform a dual-trajectory contrastive analysis. Unlike conventional summarization, we require the model to adhere to the principle of actionable abstraction. The extracted insights are formalized into a canonical, decision-oriented template:

"When solving [Task Type], I should [Strategy] rather than [Error Pattern]."

By integrating stylized insights into the teacher's successful trajectories, we construct a specialized corpus of Insight-Augmented Demonstrations that differs fundamentally from the reactive error-recovery data in Section 3.2: rather than remedying failures, it provides proactive policy guidance, serving as a "pedagogical bridge" linking environmental observations to high-level reasoning.

4 Experiments

4.1 Experimental Datasets

We conduct experiments across two domains: multi-hop question answering(QA) and mathematical reasoning. Partially following the experimental protocols established in prior studies(Kang et al., 2025), we select eight benchmark datasets, including HotPotQA, MuSiQue, 2WikiMultiHopQA, Bamboogle for multi-hop QA, and MATH, GSM-Hard, MathQA, SVAMP for mathematics.

We adopt a high-efficiency training setup using only 2k samples each from MATH and HotPotQA. Our method generates two refined trajectory types — Adaptive Trajectory Remediation (ATR) and Strategy Insight Extraction (SIE). To ensure fair evaluation, we strictly control data size: if our method produces more trajectories than a baseline, we randomly downsample to match exactly, ensuring performance gains stem from reasoning quality, not scale. Detailed training and testing dataset statistics are provided in Appendix A.

4.2 Experimental Setups

In our distillation framework, we employ Qwen3-32B(Yang et al., 2025) as the teacher model due to its superior reasoning and code-generation capabilities. To evaluate the effectiveness of our method across various scales, we select three lightweight models as students: Qwen3-0.6B, Qwen3-1.7B, and Llama 3.2-1B(Grattafiori et al., 2024).

Following the agentic formulation established in CodeAct (Wang et al., 2024c), the interaction trajectory is a sequence of Thought, Action, and Observation. And we use Wikipedia as the knowledge base following (Kang et al., 2025). This iterative loop continues until the agent reaches a final answer or exceeds the maximum step limit.

For Error-aware Correct Trajectory Sampling (ECRS), we sample 5 trajectories per instance to construct a sufficiently diverse candidate pool. The reward function hyper-parameters are configured as $w_1 = 0.7$ and $w_2 = 0.3$. Specifically, accuracy in mathematical reasoning is determined via strict matching with ground truth; for multi-hop QA, we employ Qwen3-32B as a judge to assess the correctness of reasoning chains and report F1 scores for granular word-level assessment. Besides, we report the Pass@1 metric for all evaluations.

4.3 Experimental Baselines

We compare our framework against the following methods: (1) CoT Prompting (CoT) (Wei et al., 2022), a static reasoning paradigm utilizing system prompts and user prompt; (2) Agent Prompting (Agent) (Wang et al., 2024c), an interactive baseline employing a CodeAgent framework with a 5-step budget and tool access (e.g., Wikipedia); (3) AgentDistill, an imitation-learning-based framework that distills agentic behavior from teacher trajectories; and (4) AgentDistill-FTP (Kang et al., 2025), which enhances trajectory quality by incorporating a first-thought prefix during teacher generation. Critically, we maintain strictly identical system prompts and toolsets across all agentic configurations to ensure that performance gains are solely attributable to the distillation strategy.

5 Overall Performance

5.1 Performance on Multi-hop QA

Table 1 summarizes the experimental results on QA tasks, where our proposed framework consistently achieves superior performance across all evaluated student models and most of the benchmarks. The most substantial gains are observed in smaller-scale student architectures. For instance, on the 2Wiki-MultiHopQA dataset, our framework elevates the Qwen3-0.6B model from a baseline agentic accuracy of 6.20% to an impressive 34.20%, effectively bridging the capability gap between compact models and high-parameter teachers. Similarly, for the Qwen3-1.7B student, our approach achieves 48.20% accuracy on HotPotQA, outperforming the AgentDistill baseline by 3.4% absolute.

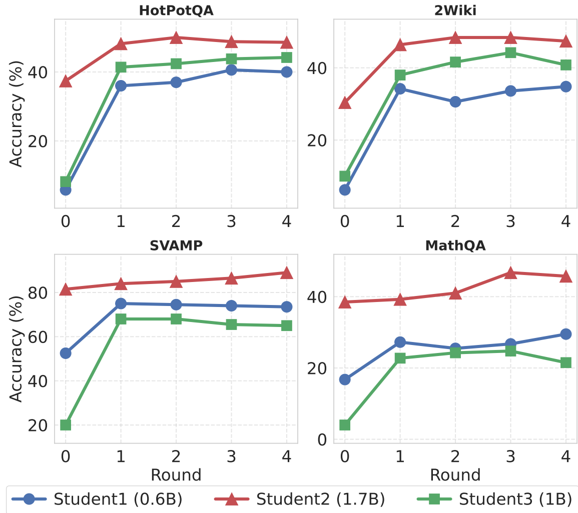

Figure 3: Accuracy across five iterative training rounds on representative QA and math benchmarks. Improvements stabilize after several rounds.

The robustness of our framework is further evidenced by the F1-score improvements. In the MuSiQue benchmark, while our accuracy matches the AgentDistill baseline at 14.20%, it achieves a higher F1-score (12.64 vs. 9.82), indicating superior precision and answer quality. These results underscore that by integrating Trajectory Remediation and Strategic Insight Extraction, our framework enables students to go beyond simple action imitation and develop a more coherent reasoning process, leading to more accurate and better-justified answers in multi-hop question answering.

5.2 Performance on Mathematical Reasoning

Table 2 presents the performance of our framework across mathematical reasoning benchmarks. The Qwen3-1.7B student model achieves an accuracy of 84.00% on SVAMP, nearing teacher-level performance. However, the improvements on the high-difficulty MATH dataset are more modest. We find that this performance bottleneck is closely linked to the teacher model's own capabilities in an agentic setting. Specifically, on the MATH benchmark, the teacher model's Agent mode (63.25%) performs significantly worse than Chain-of-Thought (CoT) performance of Qwen3-1.7B (65.80%). This drop in teacher quality directly impacts the student's learning process.

To verify that this suboptimal performance stems from the teacher model's capacity limits rather than algorithmic constraints, we conducted an additional experiment by replacing the original teacher

Model	Method	HotPotQA		MuSiQue		Bamboogle		2Wiki		AVG
		Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	-
Teacher (Qwen3-32B)	CoT	38.60	31.50	14.00	11.53	60.80	57.82	34.60	33.88	35.34
	Agent	60.20	47.97	28.00	22.73	64.00	61.08	59.60	47.45	48.88
Student1 (Qwen3-0.6B)	CoT	13.00	9.73	2.20	2.95	5.60	6.15	26.80	18.88	10.66
	Agent	5.80	3.45	0.60	1.67	4.80	1.61	6.20	6.38	3.81
	AgentDistill	19.40	11.10	4.20	4.31	11.20	11.16	26.40	8.29	12.01
	AgentDistill-FTP	18.40	14.25	3.80	4.06	11.20	9.80	25.40	21.78	13.59
	Ours	**36.00**	**29.60**	**6.20**	**6.64**	**21.60**	**19.98**	**34.20**	**31.56**	**23.22**
Student2 (Llama3.2-1B)	CoT	20.20	3.45	3.60	1.27	22.40	4.21	15.60	4.15	9.36
	Agent	8.20	1.94	1.60	1.07	3.20	0.69	10.00	4.85	3.94
	AgentDistill	40.80	31.81	12.00	10.68	40.00	**38.25**	**39.80**	**33.74**	30.89
	AgentDistill-FTP	35.20	27.69	6.80	8.18	29.60	27.36	37.20	32.51	25.57
	Ours	**41.40**	**34.91**	**12.40**	**10.85**	**42.40**	37.87	38.00	33.25	**31.39**
Student3 (Qwen3-1.7B)	CoT	21.60	16.26	2.60	5.71	16.00	15.81	19.60	17.30	14.36
	Agent	37.40	30.97	6.00	5.28	32.00	23.58	30.40	26.35	24.00
	AgentDistill	44.80	32.20	**14.20**	9.82	40.00	34.65	43.00	33.60	31.53
	AgentDistill-FTP	29.40	22.42	9.00	9.91	27.20	24.11	34.60	29.81	23.31
	Ours	**48.20**	**38.24**	**14.20**	**12.64**	**43.20**	**40.07**	**46.40**	**40.62**	**35.45**

Table 1: Performance comparison on multi-hop question answering benchmarks. Accuracy (Acc.) and F1-score (F1) are reported. Results compare Chain-of-Thought (CoT), Agent prompting, and distillation-based baselines with the proposed P-BRIDGE framework across different student model sizes. P-BRIDGE consistently achieves the strongest performance, with particularly large gains on smaller student models.

(Qwen3-32B) with a stronger model (DeepSeek-V3) and re-training P-BRIDGE under the identical setup. As shown in Table 6, the performance of the distilled student improves consistently when guided by a more capable teacher (e.g., average accuracy increasing from 60.61% to 65.71%). This demonstrates a clear positive correlation between teacher capability and downstream distilled performance. Furthermore, these findings confirm that P-BRIDGE does not inherently underperform on complex mathematical reasoning tasks; rather, when the teacher exhibits more robust reasoning and superior failure-correction behaviors, the student correspondingly internalizes more effective recovery policies.

Besides, an observation arises when comparing CoT and Agent modes for stronger student models. For datasets like MathQA, the student model's CoT performance (48.50%) significantly exceeds its Agent performance (38.00%). We attribute this to the "Agentic Overhead" and the inherent formalized symbolic nature of mathematics. Mathematics is a formal, self-consistent system where internal logical derivations are often sufficient for resolution. Forcing such tasks into an agentic workflow—which requires external tool invocation and adherence to interaction protocols—can increase the learning cost and introduce unnecessary complexity (Zhou et al., 2025a; Lightman et al., 2023). In such cases, the additional steps of tool use may actually serve as a distraction rather than a support, leading to a performance degradation compared to direct CoT reasoning.

6 Analysis

6.1 Ablation Study

To investigate the contribution of each component within our framework, we conduct a series of ablation experiments across all benchmarks. The results of Qwen3-1.7B are summarized in Table 3.

Turning "Failure into Treasure" via Trajectory Remediation (w/o TR) As our most fundamental component, Trajectory Remediation (TR) enables the student model to reflection-in-action and learn directly from its failed attempts. Excluding TR results in a performance degradation across the board. Specifically, in the 2WikiMultiHopQA task, the accuracy decreases from 46.40% to 38.80%.

The Role of Macro-Strategy Guidance (w/o SIE) Removing the Strategy Insight Extraction (SIE) component also leads to a significant performance decline, particularly in multi-hop reasoning tasks. This empirical evidence suggests that merely providing micro-level corrective actions is insufficient. Instead, the abstract problem-solving principles provided by SIE significantly enhance the model's logical consistency and generalization capabilities.

Model	Method	MATH	GSM	SVAMP	MathQA	AVG
Teacher	CoT	82.00	73.00	91.50	59.50	76.50
	Agent	63.25	80.00	85.50	54.75	70.27
Student1 (Qwen3-0.6B)	CoT	39.60	36.00	74.00	27.75	44.34
	Agent	20.20	28.40	52.50	16.75	29.46
	AgentDistill	37.40	45.40	**75.00**	**29.50**	46.83
	AgentDistill-FTP	38.48	**47.20**	70.50	24.50	45.17
	Ours	**40.60**	46.60	**75.00**	27.25	**47.36**
Student2 (Llama3.2-1B)	CoT	25.40	19.60	59.00	20.00	31.00
	Agent	10.20	10.00	20.00	4.00	11.05
	AgentDistill	30.20	**42.20**	63.00	21.00	39.10
	AgentDistill-FTP	**33.00**	41.00	64.00	21.50	39.88
	Ours	30.60	38.80	**68.00**	**22.75**	**40.04**
Student3 (Qwen3-1.7B)	CoT	**65.80**	58.80	82.50	**48.50**	**63.90**
	Agent	46.00	62.40	81.00	38.00	56.85
	AgentDistill	50.80	61.20	79.00	38.25	57.31
	AgentDistill-FTP	52.00	64.80	83.00	40.00	59.95
	Ours	54.00	**65.20**	**84.00**	39.25	60.61

Table 2: Performance comparison on mathematical reasoning benchmarks. Accuracy (%) is reported on MATH, GSM-Hard, SVAMP, and MathQA. Results demonstrate that P-BRIDGE improves reasoning performance across student models, while highlighting the impact of teacher agent quality on harder benchmarks such as MATH.

Impact of Data Quality and Efficiency-driven Rejection Sampling We observe a significant performance degradation in this configuration (Row 4), where both SIE and EDRS are removed. This configuration tests the critical impact of training data quality. Without EDRS, the distilled model tends to learn redundant and verbose solution paths. As illustrated in Figure 5, the "With EDRS" approach yields a much higher concentration of concise trajectories, with over 50% of samples resolved within 2 steps (one from the student and one from the teacher). In contrast, the "Without EDRS" version produces a significantly higher proportion of longer trajectories (e.g., lengths 4 to 6), leading to a less efficient policy. These results underscore the unique value of EDRS in ensuring that the distilled knowledge is not only correct but also computationally parsimonious.

6.2 Analysis of Iterative Improvement

To evaluate the Learning Persistence of our framework, we conduct an iterative study for five rounds. Figure 3 illustrates the performance evolution of three different student models.

A primary finding is the high data efficiency of our method, evidenced by a sharp performance surge within the first iteration. Specifically, on the 2WikiMultiHopQA benchmark, the smallest model (Student 1, 0.6B) begins with a baseline accuracy of only 6.20% at Round 0. However, after just a single round of instructor-guided training, its performance jumps to 34.20%. These results strongly support the hypothesis that targeted, instructive signals derived from failures can empower small models to achieve—and even exceed—the baseline capabilities of models with significantly higher parameter counts in practice.

Besides, The improvement trends are remarkably consistent across different model families, including both Qwen and Llama. All student models exhibit nearly identical improvement trends regardless of their starting baseline, suggesting that the framework's ability to extract transferable strategic insights is architecture-agnostic. Following the initial surge, performance typically stabilizes after the second or third round. Such saturation implies that the student model has effectively absorbed the useful guidance available within the existing teacher-student capability gap.

6.3 Analysis of Agentic Error-Recovery Dynamics

While the previous sections demonstrate significant gains in end-to-end performance, a fundamental question remains: does the student model truly internalize the ability to recover from failure? To verify this, we conduct a behavioral analysis focused on Explicit Error Correction. We define the Explicit Error Correction Rate as the rate at which an agent corrects its mistakes after encountering an explicit error from the environment (e.g., a Python compilation error). Notably, a trajectory is counted

	HotPotQA		MuSiQue		Bamboogle		2WikiMultiHopQA		MATH	GSM	SVAMP	MathQA
	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	Acc.	Acc.	Acc.
Ours (Full Framework)	48.20	38.24	14.20	12.64	43.20	40.07	46.40	40.62	54.00	65.20	84.00	39.25
w/o SIE	42.40	36.11	7.80	8.25	30.40	27.25	40.20	36.98	44.40	62.00	84.00	37.75
w/o TR	47.60	37.80	10.60	9.92	39.20	35.09	38.80	34.33	50.40	60.00	82.00	37.75
w/o EDRS	47.40	37.87	10.40	9.52	39.20	36.73	45.40	40.17	47.60	65.00	82.50	41.00
w/o SIE & EDRS	38.60	31.10	6.20	6.56	28.00	24.35	36.00	30.66	45.80	59.80	82.50	33.25

Table 3: Ablation study of the proposed framework. We report Accuracy (%) and F1-score for QA tasks, and Accuracy (%) for Math tasks. Removing either Trajectory Remediation (TR), Strategy Insight Extraction (SIE), or Efficiency-Driven Rejection Sampling (EDRS) leads to consistent performance degradation.

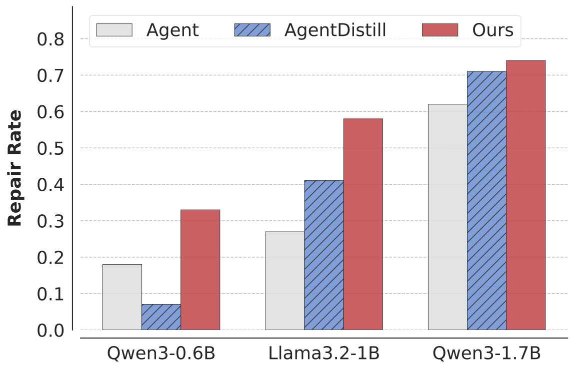

Figure 4: Comparison of the explicit error correction rate. The metric measures the probability that an agent recovers from an environment-returned error (e.g., tool or code failure). P-BRIDGE exhibits substantially higher recovery rates than baselines, indicating successful internalization of reflexive self-correction behaviors.

as "corrected" if the immediate execution error is resolved through reflexive reasoning, regardless of whether the final answer to the task is correct. The experimental results are shown in Figure 4.

As evidenced by our empirical results, the proposed framework substantially improves the error correction rate across all student models. The distilled agents actively leverage such signals as informative cues to trigger self-correction, indicating successful internalization of the reflexive reasoning patterns encoded during training. In contrast, Qwen3-0.6B fine-tuned using only flawless teacher demonstrations attains a markedly low recovery rate of 0.07. This gap suggests that exclusive distillation of successful trajectories can be detrimental for compact models, as the absence of error-to-resolution transitions prevents the student from learning how to escape sub-optimal interaction states. Collectively, these findings demonstrate that for small-scale models, acquiring the "path to recovery" is as critical as learning the correct solution itself, as it mitigates collapse under unexpected environmental feedback.

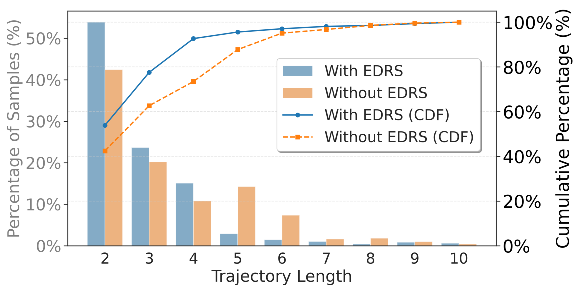

Figure 5: Effect of efficiency-driven rejection sampling (EDRS) on repaired trajectory length, removing EDRS leads to longer and more redundant trajectories.

6.4 Case Studies on Strategic Logic Shifts

The quantitative improvements in explicit error correction rates presented in Section 6.3 provide evidence of the model's enhanced Robustness. To fully understand the underlying reasoning mechanisms and illustrate how our error-aware distillation reshapes agent behavior, we select two examples—one from the QA benchmark (Figure 7) and one from mathematical reasoning (Figure 8) —to highlight the shift in reasoning patterns before and after P-BRIDGE distillation.

In the QA task, the original model repeatedly misuses Python's datetime despite persistent TypeErrors; the distilled model instead starts with the strategic insight and successfully switches to web_search. In the math problem, the untrained model cycles through confused calculations, while the distilled version, though initially producing an error, reflexively diagnoses—"I made a syntax error... I will correct this"—fixes the bug, and outputs the correct answer. These cases confirm our core motivation: failures are transformed into pedagogical signals that encode both concrete corrections and abstract strategies, enabling agents to internalize why an approach fails and how to adapt—moving beyond imitation toward robust, generalizable reasoning.

7 Conclusion

In this work, we present P-BRIDGE, a framework that transforms agentic failures into "pedagogical treasures." By grounding distillation in Schön's reflective practice, we move beyond simple imitation to empower SLMs with both the immediate capability to rectify execution errors and the strategic foresight to generalize lessons across tasks. Our results highlight that P-BRIDGE consistently improves the performance of lightweight agents and distilled models exhibit substantially stronger explicit error recovery capabilities, actively engaging with environmental failures rather than being derailed by them. P-BRIDGE highlights the importance of recoverable policies in agentic reasoning and offers a principled path toward more robust and deployable small-scale agents.

Limitations

Despite its effectiveness, P-BRIDGE has several limitations. First, The framework relies on a high-capacity teacher agent to diagnose failures and generate repaired trajectories, which means that the quality of distilled supervision is bounded by the teacher's own agentic competence. Second, The current scope of our evaluation is also limited to the specific domains of multi-hop question answering and mathematics. Finally, while our method demonstrates remarkable data efficiency using fewer than 2,000 samples, such a constrained training set may not fully account for all possible errors that an agent might encounter in real-world interactive scenarios.

Ethical Consideration

This work adheres to the ACL Ethics Policy. All eight datasets used, such as HotPotQA and MATH, are publicly available and widely adopted in the research community. P-BRIDGE is designed to enhance the robustness and error-recovery capabilities of small language models in agentic settings. Our methodology promotes high data efficiency and "cognitive parsimony," requiring fewer computational resources for training. We do not anticipate any direct negative societal or ethical impacts from this work. Furthermore, we are committed to research transparency and welcome community discussion regarding our methodology.

Acknowledgments

This work is supported in part by Ucap Cloud.

References

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers)*, pages 2357–2367.

Justin Chih-Yao Chen, Swarnadeep Saha, Elias Stengel-Eskin, and Mohit Bansal. 2024. Magdi: Structured distillation of multi-agent interaction graphs improves reasoning in smaller language models. *arXiv e-prints*, pages arXiv–2402.

Cong Thanh Do, Rama Sanand Doddipatla, and Kate Knill. 2025. Effectiveness of chain-of-thought in distilling reasoning capability from large language models. In *Proceedings of the 18th International Natural Language Generation Conference*, pages 833–845.

Tao Feng, Yicheng Li, Chenglin Li, Hao Chen, Fei Yu, and Yin Zhang. 2024. Teaching small language models reasoning through counterfactual distillation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5831–5842.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Pal: Program-aided language models. In *International Conference on Machine Learning*, pages 10764–10799. PMLR.

Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International journal of computer vision*, 129(6):1789–1819.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Namgyu Ho, Laura Schmid, and Se-Young Yun. 2023. Large language models are reasoning teachers. In

Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers), pages 14852–14882.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. *arXiv preprint arXiv:2011.01060*.

Minki Kang, Jongwon Jeong, Seanie Lee, Jaewoong Cho, and Sung Ju Hwang. 2025. Distilling llm agent into small models with retrieval and code tools. *arXiv preprint arXiv:2505.17612*.

Hojae Lee, Junho Kim, and SangKeun Lee. 2024. Mentor-kd: Making small language models better multi-step reasoners. *arXiv preprint arXiv:2410.09037*.

Xiang Li, Shizhu He, Jiayu Wu, Zhao Yang, Yao Xu, Yang jun Jun, Haifeng Liu, Kang Liu, and Jun Zhao. 2024a. Mode-cotd: Chain-of-thought distillation for complex reasoning tasks with mixture of decoupled lora-experts. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 11475–11485.

Zhuochun Li, Yuelyu Ji, Rui Meng, and Daqing He. 2024b. Enhance reasoning by learning from mistakes: Peer-review knowledge distillation from multiple large language models. *arXiv e-prints*, pages arXiv–2410.

Zhuochun Li, Yuelyu Ji, Rui Meng, and Daqing He. 2025. Learning from committee: Reasoning distillation from a mixture of teachers with peer-review. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 4190–4205, Vienna, Austria. Association for Computational Linguistics.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are nlp models really able to solve simple math word problems? *arXiv preprint arXiv:2103.07191*.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, and 1 others. 2023. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789*.

Magnus Ramage. 2017. Learning and change in the work of donald schön: Reflection on theory and theory on reflection. In *The Palgrave handbook of organizational change thinkers*, pages 1159–1172. Springer.

Donald A Schön. 2017. *The reflective practitioner: How professionals think in action*. Routledge.

Donald A. Schön and Martin Rein. 1994. *Frame Reflection: Toward the Resolution of Intractable Policy Controversies*. BasicBooks, New York.

Weizhou Shen, Chenliang Li, Hongzhan Chen, Ming Yan, Xiaojun Quan, Hehong Chen, Ji Zhang, and Fei Huang. 2024. Small llms are weak tool learners: A multi-llm agent, 2024. *URL: https://arxiv.org/abs/2401.07324*.

Hao Sun, Yong Jiang, Bo Wang, Yingyan Hou, Yan Zhang, Pengjun Xie, and Fei Huang. 2024. Retrieved in-context principles from previous mistakes. *arXiv preprint arXiv:2407.05682*.

Yijun Tian, Yikun Han, Xiusi Chen, Wei Wang, and Nitesh V Chawla. 2025. Beyond answers: Transferring reasoning capabilities to smaller llms using multi-teacher knowledge distillation. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, pages 251–260.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. musique: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Karthik Valmeekam, Matthew Marquez, Sarath Sreedharan, and Subbarao Kambhampati. 2023. On the planning abilities of large language models-a critical investigation. *Advances in Neural Information Processing Systems*, 36:75993–76005.

Danqing Wang and Lei Li. 2023. Learning from mistakes via cooperative study assistant for large language models. *arXiv preprint arXiv:2305.13829*.

Haorui Wang, Rongzhi Zhang, Yinghao Li, Lingkai Kong, Yuchen Zhuang, Xiusi Chen, and Chao Zhang. 2024a. Tpd: Enhancing student language model reasoning via principle discovery and guidance. *arXiv preprint arXiv:2401.13849*.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, and 1 others. 2024b. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345.

Peifeng Wang, Zhengyang Wang, Zheng Li, Yifan Gao, Bing Yin, and Xiang Ren. 2023. Scott: Self-consistent chain-of-thought distillation. *arXiv preprint arXiv:2305.01879*.

Xingyao Wang, Yangyi Chen, Lifan Yuan, Yizhe Zhang, Yunzhu Li, Hao Peng, and Heng Ji. 2024c. Executable code actions elicit better llm agents. In *Forty-first International Conference on Machine Learning*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Fenghua Weng, Chaochao Lu, Xia Hu, Wenqi Shao, and Wenjie Wang. 2025. Think-reflect-revise: A policy-guided reflective framework for safety alignment in large vision language models. *arXiv preprint arXiv:2512.07141*.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, and 1 others. 2025. The rise and potential of large language model based agents: A survey. *Science China Information Sciences*, 68(2):121101.

Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. 2024. A survey on knowledge distillation of large language models. *arXiv preprint arXiv:2402.13116*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2369–2380.

Jiahao Ying, Mingbao Lin, Yixin Cao, Wei Tang, Bo Wang, Qianru Sun, Xuan-Jing Huang, and Shuicheng Yan. 2024. Llms-as-instructors: Learning from errors toward automating model improvement. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 11185–11208.

Erxin Yu, Jing Li, Ming Liao, Qi Zhu, Boyang Xue, Minghui Xu, Baojun Wang, Lanqing Hong, Fei Mi, and Lifeng Shang. 2025. Self-error-instruct: Generalizing from errors for llms mathematical reasoning. *arXiv preprint arXiv:2505.22591*.

Jianpeng Zhou, Wanjun Zhong, Yanlin Wang, and Jiahai Wang. 2025a. Adaptive-solver framework for dynamic strategy selection in large language model reasoning. *Information Processing & Management*, 62(3):104052.

Xiaofeng Zhou, He-Yan Huang, and Lizi Liao. 2025b. Debate, reflect, and distill: Multi-agent feedback with tree-structured preference optimization for efficient language model enhancement. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 9122–9137.

Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. 2024. Distilling mathematical reasoning capabilities into small language models. *Neural Networks*, 179:106594.

A Dataset Details

To evaluate the effectiveness of our proposed framework, we conduct extensive experiments across two primary reasoning domains: Multi-hop Question Answering (QA) and Mathematical Reasoning. We evaluate our model on eight benchmark datasets to ensure a comprehensive assessment of its generalization capabilities.

We select four representative datasets for each domain. The detailed statistics of the evaluation sets are summarized in Table 4.

- Multi-hop Reasoning: This category includes HotPotQA(Yang et al., 2018), MuSiQue(Trivedi et al., 2022), Bamboogle(Press et al., 2023), and 2WikiMultiHopQA(Ho et al., 2020), which require the model to aggregate information across multiple documents or reasoning steps.
- Mathematical Reasoning: We utilize MATH(Hendrycks et al., 2021), GSM-Hard(Gao et al., 2023), SVAMP(Patel et al., 2021), and MathQA(Amini et al., 2019) to test the model's ability to perform logical deduction and numerical calculation.

Domain	Dataset	Test Size (Samples)
Multi-hop QA	HotPotQA	500
	MuSiQue	500
	2WikiMultiHopQA	500
	Bamboogle	125
Mathematics	MATH	500
	GSM-Hard	500
	MathQA	400
	SVAMP	200

Table 4: Test set sizes for multi-hop question answering and mathematical reasoning benchmarks.

To demonstrate the data efficiency of our approach, we deliberately use a constrained training set. Specifically, we collect from 1,000 samples of the training splits of MATH and 1,000 samples of HotPotQA. This sparse-data setting is designed to evaluate whether our method can effectively distill reasoning capabilities without relying on massive amounts of high-cost data.

To ensure a fair comparison with baseline methods, we implement a strict data-size control strategy. In cases where our proposed framework generates more synthetic reasoning trajectories than the baseline method, we perform random down-sampling on our generated data to match the exact volume used by the baseline. This ensures that any observed performance gains are attributed to the quality and insights of our data rather than simply an increase in training quantity.

Our training set is constructed from 1,000 samples each of MATH and HotPotQA. Specifically, our method generates two types of refined trajectories: Adaptive Trajectory Remediation (ATR) and Strategy Insight Extraction (SIE). The specific distribution of training samples for different backbone models is presented in Table 5

Notably, our framework achieves superior performance while utilizing a data volume that is comparable to, or even strictly less than, that of the baseline methods. This high data efficiency demonstrates our performance gains stem from the intrinsic quality of the extracted reasoning insights rather than simple numerical scaling of the training set.

Model	ATR: SIE
Llama3.2-1B(Grattafiori et al., 2024)	686: 713
Qwen3-0.6B(Yang et al., 2025)	568: 830
Qwen3-1.7B(Yang et al., 2025)	519: 476

Table 5: Statistics of training data of P-BRIDGE used in the first round.

B Robustness Analysis of the LLM Judge

To ensure the reliability of the LLM-as-a-judge for QA accuracy evaluation, we randomly sample 625 test cases and compare the judgments of the state-of-the-art closed-source model GPT-5 against Qwen3-32B. Our analysis shows a high inter-evaluator agreement of 89.92%. Upon manual inspection of the 63 inconsistent cases, we observe that GPT-5 exhibits excessive rigidity in semantic matching tasks. For instance, it flags "Alfonso IV King of Aragon" and "Alfonso IV of Aragon" as mismatched despite referring to the same entity. In contrast, Qwen3-32B demonstrates superior instruction-following capabilities by handling reasonable variations such as case sensitivity and stylistic differences with greater flexibility. This evidence confirms that Qwen3-32B is better aligned with our specific evaluation prompt, which emphasizes semantic consistency over literal string matching.

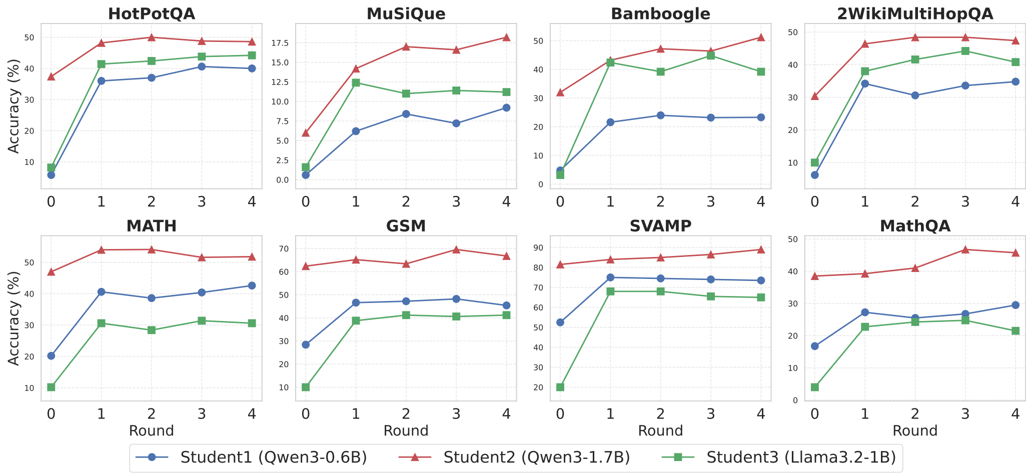

Figure 6: Performance trends across iterative rounds (extended results)

C More Experiment Results

C.1 Results of Iterative Improvement

Figure 6 presents a comprehensive view of the learning trajectories across eight benchmarks for three different student architectures. We observe a consistent "initial surge" phenomenon across all tasks: student models exhibit a sharp performance jump within the first iteration of instructor-guided training.This rapid improvement underscores the high data efficiency of the P-BRIDGE framework. Furthermore, the improvement trends are remarkably architecture-agnostic, with both Qwen and Llama model families exhibiting nearly identical learning curves.

C.2 Ablation Study on Teacher Capacity

As shown in Table 6, the performance of the distilled student improves consistently when guided by a more capable teacher.

C.3 Case Study

We analyze two representative cases where P-BRIDGE distillation corrected systematic failures in Figure 8 and Figure 7. They both demonstrate that the student model has internalized a recoverable policy rather than just memorizing a solution.

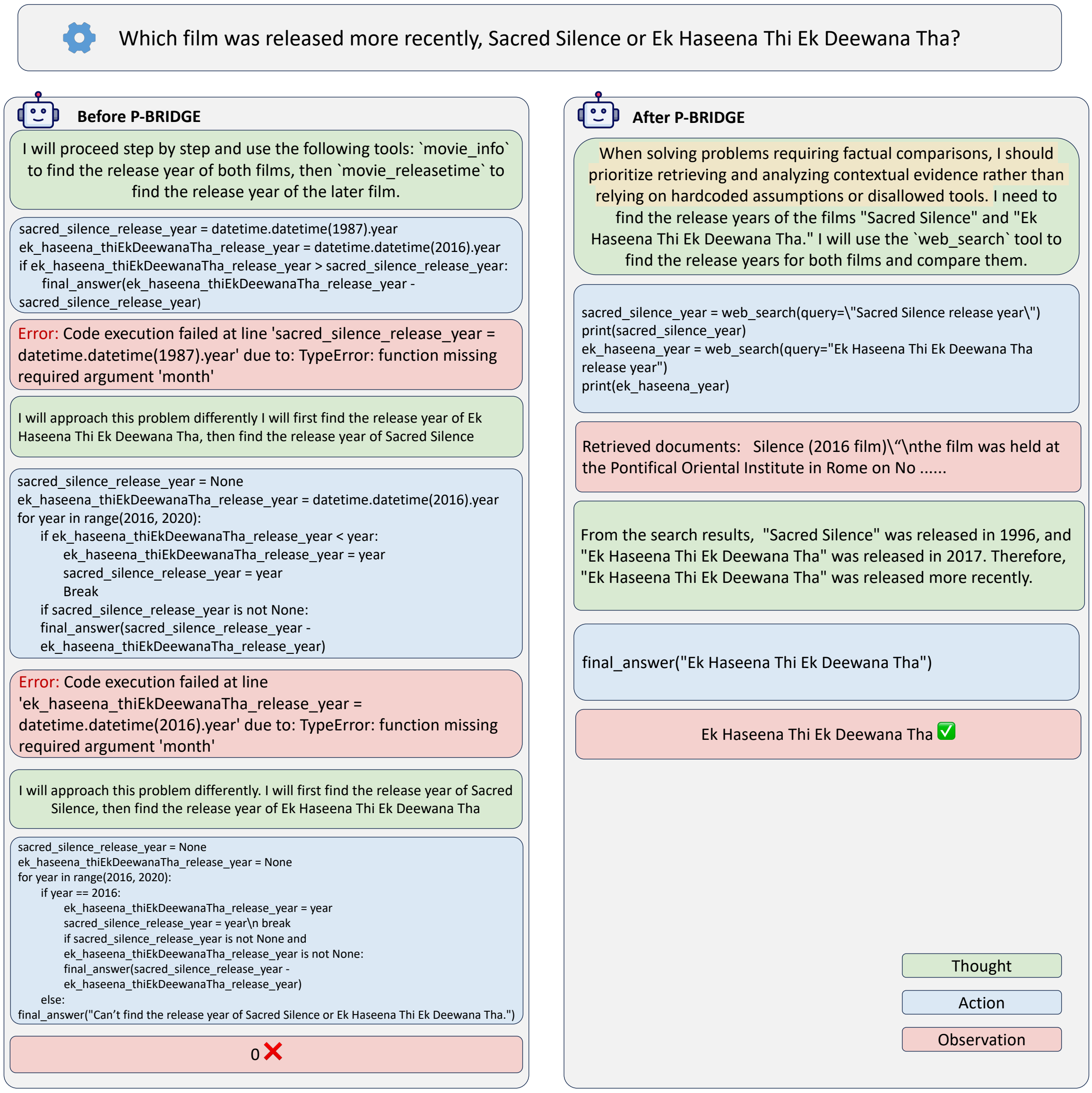

Figure 7: Case study on strategic logic shifts in multi-hop QA. Illustrating how the distilled model switches from failed code execution to a more effective web-search strategy based on extracted insights.

Category	Model / Method	MATH (%)	GSM (%)	SVAMP (%)	MathQA (%)	AVG (%)
Teacher	Qwen3-32B	63.25	80.00	85.50	54.75	70.88
	DeepSeek-V3	89.40	72.60	92.00	59.25	78.31
Student (Qwen3-1.7B)	CoT	65.80	58.80	82.50	48.50	63.90
	Agent	46.00	62.40	81.00	38.00	56.85
After P-BRIDGE	Qwen-Teacher	54.00	65.20	84.00	39.25	60.61
	DeepSeek-Teacher	62.40	63.20	88.00	49.25	65.71

Table 6: Impact of teacher model capability. Using a stronger teacher (DeepSeek-V3) consistently improves the student's mathematical reasoning performance compared to the original teacher (Qwen3-32B).

Figure 8: Case study on mathematical reasoning. A comparison of reasoning patterns before and after P-BRIDGE distillation, highlighting how the model internalizes the "path to recovery" after a syntax error.

D Prompts

Error Localization

You are an error diagnostician for AI reasoning agents. Your task is to carefully analyze the following problem-solving trajectory and identify where the error occurred.
Question: [question]
Correct Answer: [$true_{answer}$]
Generated Answer: [$generated_{answer}$]
Trajectory Summary:[summary]
Please analyze this trajectory step by step and identify:
1. At which step did the reasoning go wrong? (provide the step number, 0-indexed. Step 0 is the first thought/action)
2. What type of error occurred? For example: wrong reasoning, incorrect tool usage, wrong search query, premature conclusion, calculation error, other
Respond in JSON format:
{{"error step": <$step_{number}$>, "error reason": $< error_{type} >$}}

Strategy Insight Extraction

You are an educational reasoning analyst tasked with extracting high-level, transferable insights from student-teacher solution comparisons. Your role is to analyze the following two solution attempts and extract a GENERAL, TRANSFERABLE insight that can guide solving similar problems in the future. Think step by step before giving the final insight.
Question: [question]
Incorrect Solution: [$student_{answer}$]
Incorrect Trajectory: [$student_{summary}$]
Correct Solution: [$teacher_{answer}$]
Correct Trajectory: [$teacher_{summary}$]
Reason step-by-step:
1. Identify the key mistake pattern in the incorrect attempt (summarize in 1 line).
2. Identify the core strategy that made the correct attempt succeed (summarize in 1 line).
3. Generalize to a principle that is transferable to similar problems.
4. Craft the final insight sentence.
Insight Examples: [examples]
Keep it concise and actionable. Focus on the general principle, not specific details.
Output format:
Reasoning: <1-3 short lines capturing steps 1-3>
Insight: "When solving [problem type/situation], I should [correct strategy] rather than [general mistake pattern]."