FROM TEACHER IMITATION TO FAILURE DIAGNOSIS: CAUSALITY-INSPIRED TEST-TIME EVOLUTION OF SMALL LANGUAGE MODELS WITH LLM FEEDBACK

Anonymous authors
Paper under double-blind review

ABSTRACT

Test-time evolution of small language models (SLMs) with large language model (LLM) guidance has emerged as an important paradigm for LLM-SLM collaboration. Existing methods typically distill complete LLM-generated solutions into SLMs, while recent approaches selectively transfer knowledge at difficult tokens or intermediate reasoning steps identified from the SLM's reasoning state. Despite differing in what to transfer, these methods largely follow the same paradigm: when an SLM fails, it imitates the correct solution or reasoning provided by a stronger LLM. However, substantial differences in model capacity, training data, and acquired knowledge may make such direct imitation difficult for SLMs to effectively absorb. We take a different perspective: rather than only showing an SLM what the correct solution is, the LLM should first diagnose why the SLM fails. Such diagnosis serves two purposes. First, it guides the LLM to minimally revise the SLM's erroneous reasoning, preserving what the SLM already does correctly while correcting only failure-relevant parts. This provides a learning target closer to the SLM's current reasoning capability than an independently generated LLM solution, facilitating more effective knowledge transfer. Second, the diagnosis itself provides an additional learning signal, enabling the SLM to learn not only how its reasoning should be corrected but also why it failed. Based on this insight, we model LLM-guided SLM evolution from a causal perspective and propose CausalBridge. Reasoning State Probing samples and clusters multiple reasoning trajectories to characterize the SLM's reasoning state. Causal Diagnostic Mediation diagnoses the underlying failures and generates minimal, diagnosis-guided revisions. Finally, Mediator-Guided Evolution uses the diagnostic information as a mediator to improve failure-relevant reasoning. Extensive experiments across diverse reasoning tasks show that CausalBridge consistently outperforms both independent SLM evolution and existing LLM-SLM collaborative evolution methods.

1 INTRODUCTION

Test-time evolution of small language models with large language model guidance provides a promising paradigm for combining the efficient deployment of SLMs with the stronger reasoning ability of LLMs. Existing LLM-SLM collaborative inference methods operate at both token and task levels. At the token level, an SLM can propose candidate tokens, request LLM assistance at selected positions, or trigger LLM takeover to balance performance and inference cost (Chen et al., 2023; Huang et al., 2026; Lyu et al., 2026). At the task level, an SLM can request assistance during reasoning or route uncertain samples to an LLM for stronger predictions (Zeng et al., 2026; Zhang et al., 2026). Beyond improving the current prediction, test-time learning enables language models to adapt through lightweight parameter updates on newly observed data (Feng et al., 2026; Hu et al., 2025; Hübotter et al., 2025). Building on this capability, collaborative evolution methods use LLM outputs as external supervision for SLM adaptation, transferring difficult tokens, intermediate reasoning steps, or complete outputs (Zhong et al., 2026; Chen et al., 2024b; Wang et al., 2024). Despite differing in transfer granularity, these approaches share a direct imitation paradigm. When an SLM fails, it is trained to imitate the correct solution or reasoning generated by a stronger LLM.

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.

However, substantial differences in model capacity, training data, and acquired knowledge can make complete LLM solutions difficult for an SLM to absorb effectively.

We observe this limitation empirically on SAT-Bench in Fig. 1. Starting from the same unadapted path accuracy of 49.08, Answer-only and CoT-only adaptation peak at 54.50 and 55.10, respectively, before declining to 53.47 and 52.79 as the support set grows to 200 examples. In contrast, CausalBridge improves overall to 56.50, reaching 56.21 at 120 support examples. This trend suggests that adding more answer-level or complete teacher supervision can accumulate information that is not aligned with the SLM's failure, whereas diagnosis-guided revision provides a more targeted signal for adaptation.

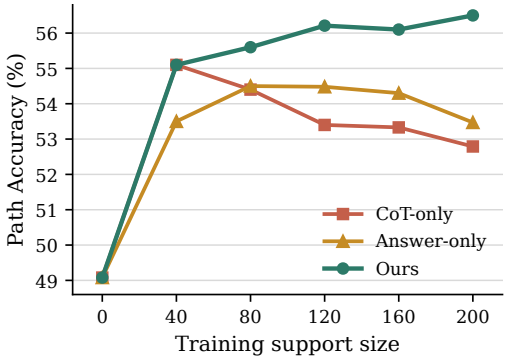


Figure 1: SATBench path accuracy as the training support set grows. Answer-only and CoT-only adaptation plateau or decline at larger support sizes, whereas CausalBridge improves overall and remains stable.

We take a different perspective. Rather than only showing an SLM the correct solution, the LLM should first diagnose why the SLM fails. Direct LLM corrections may alter reasoning that is already valid, introduce an alternative solution strategy, or include knowledge that is unrelated to the observed failure. These signals are especially problematic at test time, where the support data are limited and each update immediately affects subsequent predictions. Diagnosis provides two complementary benefits. First, it guides the LLM to minimally revise an erroneous SLM trajectory, preserving valid reasoning while correcting only failure-relevant parts. The resulting target is closer to the SLM's current reasoning capability than an independently generated LLM solution. Second, the diagnosis provides explicit supervision about why the trajectory failed, allowing the SLM to learn both how to correct its reasoning and why the correction is necessary.

We formulate this diagnostic process from a causal perspective. An SLM's domain bias can jointly affect its reasoning state and its final output, creating a confounding path that direct output imitation does not resolve. We therefore introduce LLM-generated diagnosis as a mediator between an erroneous reasoning state and a corrected outcome. Given an erroneous trajectory, the diagnosis identifies the critical failure and specifies the information required for correction. This mediated structure enables us to use the front-door criterion to identify failure-relevant corrective information rather than treating an entire LLM output as equally useful supervision.

Based on this formulation, we propose **CausalBridge**, a causality-inspired framework for LLM-guided SLM test-time evolution. **Reasoning State Probing** samples and clusters multiple trajectories to characterize the SLM's reasoning state and expose recurring failure modes. **Causal Diagnostic Mediation** asks the LLM to identify the underlying failure and generate a minimal diagnosis-guided revision. **Mediator-Guided Evolution** uses the diagnostic information as a mediator to improve failure-relevant reasoning through preference optimization. After adaptation, the evolved SLM performs independently without the diagnostic mediator or further cloud calls. This separates one-time cloud-assisted evolution from inference-time offloading while retaining low-cost local deployment. Extensive experiments across diverse reasoning tasks show that CausalBridge consistently outperforms both independent SLM evolution and existing LLM-SLM collaborative evolution methods.

Our main contributions are summarized as follows:

- We identify a limitation of direct LLM imitation for SLM test-time evolution. Complete LLM outputs may contain already-mastered knowledge, alternative reasoning plans, and information unrelated to the observed failure. We further show that this mismatch can limit adaptation as the support set grows.
- We formulate LLM-guided SLM evolution from a causal perspective and introduce diagnosis as a mediator for identifying failure-relevant corrective information through the front-door criterion. The formulation distinguishes corrective information from confounded signals in complete teacher outputs.

- We propose **CausalBridge**, which combines Reasoning State Probing, Causal Diagnostic Mediation, and Mediator-Guided Evolution to construct diagnosis-guided preference pairs for lightweight SLM adaptation. Experiments show consistent improvements over independent and collaborative SLM evolution baselines without deployment-time cloud calls.

2 Related Work

2.1 Test-Time Learning for Large Language Models

Test-time learning for LLMs can be viewed along two complementary axes, namely the parameters that are adapted and the signals that drive adaptation. Fast weights and prompt-only LoRA update limited model components to enable efficient parameter adaptation (Feng et al., 2026; Xu et al., 2026). Restricting updates to lightweight components is particularly useful when only a small target support set is available. Such choices control the cost and flexibility of adaptation, but do not by themselves determine which target-domain information should guide an update. A complementary line of work therefore focuses on constructing adaptation signals from newly observed data. Existing methods derive signals from input perplexity (Hu et al., 2025), informative example selection (Hübotter et al., 2025), or interaction feedback (Wei et al., 2025b). Reasoning-specific methods further use majority-voted pseudo-labels, tool verification, or confidence-conditioned verification to estimate the reliability of candidate answers (Zuo et al., 2026; Liao et al., 2026; Li et al., 2026). Feedback-driven approaches additionally show that weak answers, student trajectories, and language criticism can serve as supervision for model updates (Choi et al., 2025; Zhong et al., 2026; Klissarov et al., 2026). These methods differ in signal granularity, ranging from answer-level rewards to trajectory-level feedback. Verification can establish whether a candidate output should be trusted, whereas trajectory-based feedback provides contextual evidence about how it was produced. Neither property, however, necessarily identifies which inference within an erroneous trajectory should be corrected. Our work addresses this signal-design question. We use a stronger model to generate structured diagnoses of erroneous SLM trajectories and condition a preference-learning objective on the identified corrective information. The resulting update is tied to a path-specific failure state rather than to self-generated confidence, generic teacher outputs, or answer-level agreement alone.

2.2 Small-Large Language Model Collaboration

Small-large model collaboration at test time can be organized into collaborative inference and collaborative evolution. Collaborative inference improves the current response through SLM-LLM cooperation. At the token level, an SLM can propose draft tokens, request assistance at selected positions, or trigger LLM takeover to trade accuracy against latency or cost (Chen et al., 2023; Huang et al., 2026; Lyu et al., 2026). At the task level, an SLM can request help during reasoning or route low-confidence samples to an LLM (Zeng et al., 2026; Zhang et al., 2026). These methods use the LLM to complete, verify, or route the current task, and thus primarily improve the response generated during the ongoing interaction.

Collaborative evolution instead uses LLM feedback to improve the SLM for subsequent queries. On-policy distillation, cloud-device adaptation, and cross-device knowledge transfer use student trajectories, selected samples, or shared domain knowledge as signals for SLM updates (Zhong et al., 2026; Gan et al., 2023; Chen et al., 2024b; Wang et al., 2024; Chen et al., 2024a). While these approaches determine which samples or representations should be transferred, cloud outputs commonly serve as whole supervision targets. Our work studies causally motivated test-time collaborative evolution: an LLM diagnoses a specific erroneous SLM reasoning path, and the resulting corrective information conditions a preference update for the SLM.

3 Problem Formulation

3.1 SLM Evolution under LLM Feedback

We consider a collaborative evolution setting in which a SLM is progressively improved using feedback from a more capable LLM. Let the SLM and LLM be denoted by f_{θ_s} and f_{θ_l}, respectively. Given an input query x, the SLM first generates a reasoning trajectory and prediction

$(h_s, y_s) = f_{\theta_s}(x)$, where h_s denotes the reasoning trajectory and y_s denotes the final output. Conditioned on the query and the SLM response, the LLM produces feedback $z_l = f_{\theta_l}(x, h_s, y_s)$, which may contain diagnostic information, corrected reasoning, revised predictions, or other auxiliary supervision.

Existing collaborative evolution methods typically optimize the SLM directly on LLM feedback z_l:

$$\theta_s' = \arg\min_{\theta_s} \mathbb{E}_{(x,z_l)} \left[\mathcal{L}\big(f_{\theta_s}(x), z_l\big) \right]. \tag{1}$$

However, z_l may contain both failure-relevant information z_{rel} and irrelevant information z_{irr}, where the latter may arise from already-mastered knowledge, alternative reasoning paths, or additional content introduced by the LLM. Directly learning from the complete feedback therefore does not guarantee that the update targets the SLM's actual failure.

We instead formulate collaborative evolution as identifying failure-relevant supervision from LLM feedback. Given the SLM reasoning state (h_s, y_s) and LLM feedback z_l, we seek $z^* = g(h_s, y_s, z_l)$ that captures the information responsible for correcting the failure, and update the SLM as $\theta_s^* = \mathcal{A}(\theta_s, z^*)$. The central problem is thus to determine which information in LLM feedback should drive SLM evolution.

3.2 Deconfounding Collaborative Evolution via Causal Mediation

We formulate SLM evolution under LLM feedback from a causal perspective. Let X denote the erroneous reasoning state of the SLM and Y the corrective outcome induced by model evolution. Their relation can be confounded by latent factors U, such as domain-specific biases induced by the SLM's pretrained knowledge distribution, which simultaneously affect how the SLM reasons and how easily its prediction can be corrected. This introduces a backdoor path $X \leftarrow U \rightarrow Y$, such that the observed association $P(Y \mid X)$ does not necessarily reflect the causal effect of correcting the reasoning state.

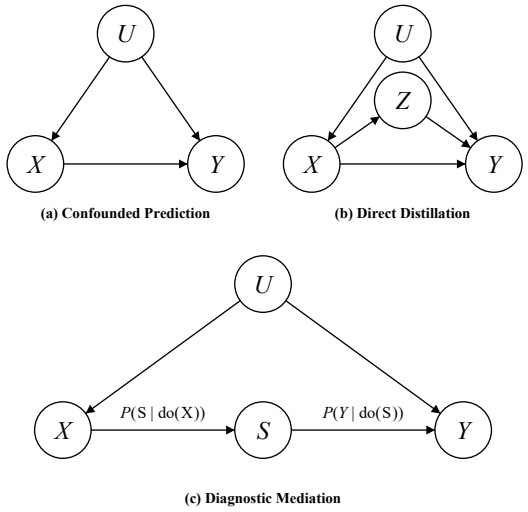


Figure 2: Causal formulation of diagnosis-mediated SLM evolution. **(a)** Confounded prediction. **(b)** Direct LLM supervision. **(c)** Diagnosis-mediated correction.

As shown in Fig. 2, the three settings illustrate confounded prediction, direct LLM supervision, and diagnosis-mediated evolution, respectively. Direct LLM supervision does not explicitly resolve this confounding. Given an erroneous state X, the LLM typically provides a corrected output Z, which may include both failure-relevant corrective information and additional content unrelated to the SLM's actual knowledge gap. Moreover, direct distillation does not remove the original dependence of the evolved prediction on X, but instead introduces an additional supervision pathway through Z. Therefore, complete LLM feedback is not explicitly designed to isolate the mechanism responsible for correcting the failure.

To make this correction mechanism explicit, we introduce LLM-generated diagnosis S as a mediator of the *corrective effect*, rather than of the entire SLM generation process. Specifically, S identifies the critical failure in X and specifies the corrective information required to repair it.

$$X \rightarrow S \rightarrow Y, \qquad X \leftarrow U \rightarrow Y. \tag{2}$$

Here, Y characterizes the correction induced by the adaptation process, rather than the original prediction directly generated from X. Under the front-door assumptions, the causal effect of the erroneous reasoning state on the corrective outcome can be identified through S as

$$P(Y \mid do(X = x)) = \sum_s P(s \mid x) \sum_{x'} P(Y \mid s, x') P(x'). \tag{3}$$

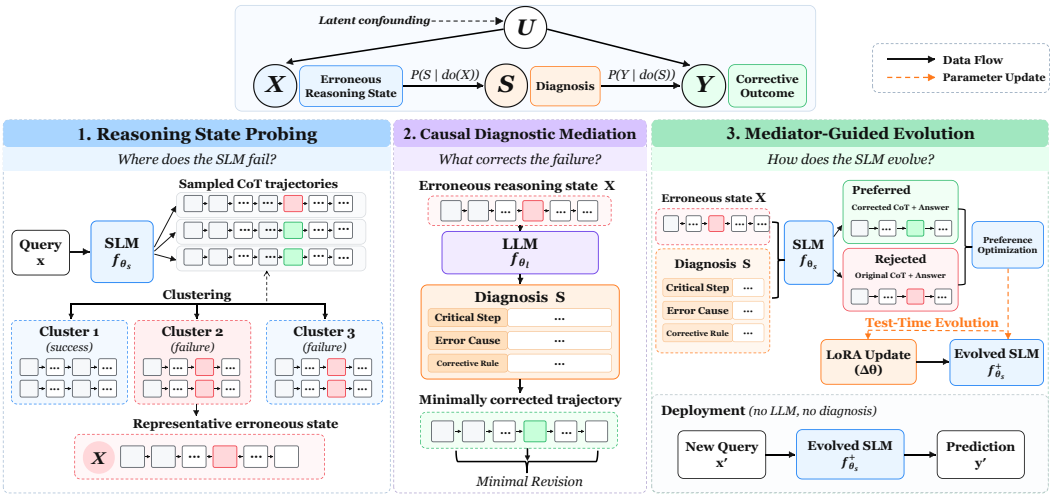


Figure 3: Overview of **CausalBridge**.

This formulation therefore uses diagnosis as an explicit causal bottleneck for modeling how an SLM failure is corrected, rather than treating the complete LLM output as undifferentiated supervision.

4 METHOD

4.1 FRAMEWORK OVERVIEW

CausalBridge consists of three components: **Reasoning State Probing**, **Causal Diagnostic Mediation**, and **Mediator-Guided Evolution**. As shown in Fig. 3, for each support query, Reasoning State Probing samples multiple reasoning trajectories from the SLM and groups similar trajectories to expose representative reasoning states and recurring failure patterns. Based on the identified erroneous states, Causal Diagnostic Mediation queries the LLM to diagnose the critical failure and generate a locally revised reasoning trajectory. The resulting diagnosis serves as privileged supervision that characterizes the corrective information connecting the erroneous state to the corrected outcome. Mediator-Guided Evolution jointly trains on diagnosis-conditioned repair pairs and matched query-only direct pairs. The former teaches the model to revise an observed failure, while the latter trains the query-only behavior used at deployment.

4.2 REASONING STATE PROBING

The front-door adjustment in Eq. 3 requires estimating causal quantities over the SLM reasoning state X. However, X is an internal model state and cannot be directly observed. We therefore use the SLM's stochastic reasoning trajectories as observable probes of its latent reasoning states. Given the same query, different decoding outcomes reflect different reasoning modes of the model, allowing multiple CoT samples to approximate the support of $P(X)$ and reveal where the SLM repeatedly succeeds or fails.

For each support query x, the SLM samples N reasoning trajectories,

$$\mathcal{C}(x) = \{(c_1, a_1), \ldots, (c_N, a_N)\}, \tag{4}$$

where c_i denotes a CoT trajectory and a_i its final answer. We treat each trajectory as an observable realization of the latent reasoning state, i.e., $c_i \simeq X_i$. Sampling multiple trajectories therefore exposes heterogeneous reasoning states rather than relying on a single generation. Importantly, an erroneous reasoning state is not determined solely by answer correctness, since a trajectory may reach the correct answer through an invalid derivation; whether a state requires correction is subsequently determined by diagnostic mediation.

To obtain representative reasoning states while avoiding redundant diagnosis, we group semantically similar trajectories into $\mathcal{G}_1, \ldots, \mathcal{G}_K$ and select the medoid of each group,

$$c_k^\star = \arg\min_{c_i \in \mathcal{G}_k} \sum_{c_j \in \mathcal{G}_k} \mathrm{dist}(c_i, c_j). \tag{5}$$

Each representative state is assigned a group proportion $w_k = |\mathcal{G}_k|/N$, yielding an empirical approximation of the reasoning-state distribution,

$$\hat{P}(X) = \sum_{k=1}^{K} w_k \, \delta(X - c_k^\star). \tag{6}$$

This probing process preserves recurring reasoning modes and failure patterns while avoiding redundant cloud diagnoses. The group proportions describe the coverage of the selected states and are not used as explicit optimization weights in the DPO objective.

4.3 Causal Diagnostic Mediation

Given the representative reasoning states obtained above, the next step is to construct the failure-relevant mediator S. We design S to explicitly intercept the corrective pathway from the erroneous reasoning state X to the corrective outcome Y. Specifically, S identifies the primary failure in X and specifies the corresponding corrective rule, while the subsequent correction is constrained to modify only the reasoning component indicated by S and preserve the remaining valid reasoning. In this way, the change from the erroneous state to the corrected outcome is explicitly mediated through S, forming the pathway $X \to S \to Y$. We therefore construct S as a compact diagnostic variable that captures the failure-specific intervention required to repair X.

For each representative erroneous trajectory c^-, the LLM produces a diagnosis s that identifies one primary reasoning failure, including its critical step, error cause, and corrective rule,

$$s = f_{\theta_l}^{\mathrm{diag}}(x, c^-). \tag{7}$$

We restrict s to a single primary failure so that multiple downstream symptoms caused by the same error are not treated as independent mediators. Under the front-door assumptions, the resulting diagnostic distribution provides an observable approximation to the first mediation term, $P(S \mid do(X)) = P(S \mid X)$.

To characterize how the diagnosed mediator changes the reasoning outcome, we construct a locally revised trajectory c^+. The LLM is prompted to preserve valid reasoning in c^- and modify only the portion identified by s. This locality is implemented through output validation rather than a global constrained optimization. A local revision must declare one to three edit operations, anchor every edit to a nonempty span in c^-, retain a copy ratio of at least 0.55, and end with a final answer that matches the trusted support label. When the model declares that the original approach cannot be repaired locally, it may return a concise global revision. Thus, c^+ is a validated diagnosis-guided revision rather than the exact minimizer of an edit-distance objective.

The resulting diagnosis-conditioned adaptation set is

$$\mathcal{D}_{\mathrm{diag}} = \left\{ (x, c^-, s, c^+) \;\middle|\; \mathrm{Invalid}(c^-, s) = 1,\ A(c^+) = a_{\mathrm{gold}} \right\}, \tag{8}$$

where the current implementation retains parsable trajectories whose final answer differs from a_{gold} and whose cloud diagnosis identifies an error. Malformed, truncated, or unverifiable diagnoses and corrections are removed before adaptation.

4.4 Mediator-Guided Evolution

The final stage internalizes diagnostic feedback into the SLM with two complementary types of preference pairs. Diagnosis-conditioned pairs teach repair under an observed failure state, while query-only direct pairs train the behavior available at deployment.

For each record in $\mathcal{D}_{\mathrm{diag}}$, we construct the prompt $p_s = \mathrm{format}(x, c^-, s)$, where c^- serves as the observable realization of reasoning state X and s specifies its failure-relevant correction. The revised

trajectory is the preferred outcome $y^+ = \mathrm{format}(c^+, a^+)$, while the original erroneous trajectory is the rejected outcome $y^- = \mathrm{format}(c^-, a^-)$. The resulting pair contrasts a corrected and an erroneous outcome under the same diagnosis-conditioned reasoning state.

For every support query with at least one accepted diagnosis-guided revision, we additionally construct at most one direct pair. Its prompt $p_x = \mathrm{format}(x)$ contains only the query. The preferred response is a verified reference trajectory with the correct final answer, and the rejected response is an SLM trajectory selected from the same query. These direct pairs provide an explicit query-only learning path and prevent deployment from relying on training-only diagnostic context.

Let π_{ref} denote the frozen base SLM and π_θ the SLM with trainable LoRA parameters. We define the relative log-likelihood as:

$$r_\theta(p, y) = \log \pi_\theta(y \mid p) - \log \pi_{\mathrm{ref}}(y \mid p), \tag{9}$$

and optimize the pairwise preference objective:

$$\mathcal{L}_{\mathrm{pref}} = -\log \sigma\left(\beta\left[r_\theta(p, y^+) - r_\theta(p, y^-)\right]\right). \tag{10}$$

For a pair (p, y^+, y^-), this objective increases the relative likelihood of the preferred response over the rejected response. We train on the uniformly shuffled union of diagnosis-conditioned and direct pairs,

$$\mathcal{L}_{\mathrm{ours}} = \mathbb{E}_{(p,y^+,y^-)\sim\mathcal{D}_{\mathrm{diag}}\cup\mathcal{D}_{\mathrm{dir}}}\left[\mathcal{L}_{\mathrm{pref}}(p, y^+, y^-)\right]. \tag{11}$$

The empirical objective does not estimate the front-door adjustment directly. Instead, the causal formulation motivates the diagnostic bottleneck, while the two-branch DPO objective learns both failure-conditioned repair and query-only generation from the retained supervision.

4.5 Independent SLM Deployment

After adaptation, the diagnostic mediator is no longer required at inference. Given an unseen query x_{test}, the evolved SLM independently generates its reasoning trajectory and final answer as

$$(c_{\mathrm{test}}, a_{\mathrm{test}}) \sim \pi_{\theta_s^*}(\cdot \mid x_{\mathrm{test}}), \tag{12}$$

where θ_s^* denotes the parameters obtained by Mediator-Guided Evolution. In contrast to the adaptation stage, no erroneous trajectory, diagnosis S, corrected trajectory, or trusted answer is provided for the test query.

The diagnostic information affects test-time prediction only through the evolved parameters:

$$\theta_s^* = \mathcal{A}(\theta_s, \mathcal{D}_{\mathrm{diag}} \cup \mathcal{D}_{\mathrm{dir}}), \qquad P(Y_{\mathrm{test}} \mid x_{\mathrm{test}}, \theta_s^*) = \pi_{\theta_s^*}(Y_{\mathrm{test}} \mid x_{\mathrm{test}}). \tag{13}$$

Thus, diagnosis serves as privileged supervision during evolution rather than an additional input during deployment. The evaluation therefore measures whether the failure-specific corrective effects identified through diagnostic mediation have been internalized into the SLM parameters and can generalize to unseen queries without external feedback.

5 Experiments

5.1 Experimental Protocol

We evaluate medical question answering, mathematical reasoning, and logical reasoning on MedQA, MATH, SATBench, BIG Bench Logical Deduction, LogiQA2.0, and FOLIO (Jin et al., 2020; Hendrycks et al., 2021; Wei et al., 2025a; Srivastava et al., 2023; Liu et al., 2023; Han et al., 2024). We use Qwen2.5-7B-Instruct (Qwen Team, 2025) as the device model and DeepSeek-V4-Flash (DeepSeek AI, 2026) as the cloud teacher in the main experiments.

For each evaluation query, we generate five CoT paths with a shared decoding configuration and score the parsed final answer from each path. Main results report mean path accuracy and standard deviation across the five paths. The ablation results report accuracy under the stated evaluation protocol. All trainable methods use one adaptation epoch and the same LoRA budget unless stated otherwise.

5.2 BASELINES

We compare the Base SLM with centralized adaptation methods and teacher supervision methods. TLM, SRPO, and ActiveFT represent test time learning, robust preference optimization, and active test time fine tuning (Hu et al., 2025; Choi et al., 2025; Hübotter et al., 2025). DSS supervises the device model with large model CoT rationales and final answers (Hsieh et al., 2023). LLM2LLM first fine tunes on seed data, asks a large model to augment corrective examples for student errors, and then retrains the student (Lee et al., 2024). SuperCorrect combines hierarchical CoT supervision, error localization, large model correction, and preference optimization (Yang et al., 2025). The centralized methods update a single model from target data or self-generated feedback, whereas the teacher-supervision methods transfer complete LLM rationales or corrected examples. Neither group conditions adaptation on a structured diagnosis tied to the observed SLM trajectory. The comparison therefore distinguishes the effect of test-time self-adaptation, direct teacher imitation, and failure-specific diagnostic feedback under the shared adaptation protocol. It tests whether access to a stronger output alone is sufficient, or whether the information that identifies the underlying failure provides an additional benefit. Our method uses diagnosis guided corrected CoT preference pairs as described in Section 4.

5.3 MAIN RESULTS

Table 1 compares diagnosis-guided adaptation with centralized adaptation and teacher-supervision baselines on reserved evaluation data. Our method achieves the highest mean accuracy on five of the six benchmarks, covering mathematical, commonsense, and logical reasoning tasks. These results suggest that selectively learning from failure-specific diagnostic feedback can provide a more stable adaptation signal than directly absorbing complete teacher-generated supervision.

Table 1: Main results of different adaptation methods across six reasoning benchmarks.

Method	MedQA	MATH	SATBench	BIG-Bench LD	LogiQA2.0	FOLIO
Base SLM	58.72 ± 1.24	57.50 ± 1.97	49.08 ± 2.84	52.00 ± 5.80	61.20 ± 1.96	60.80 ± 3.03
TLM	59.02 ± 2.53	58.00 ± 1.58	50.45 ± 1.56	54.80 ± 3.25	62.20 ± 2.42	65.20 ± 6.26
SRPO	60.20 ± 1.88	58.40 ± 2.41	50.75 ± 1.72	54.80 ± 3.25	61.70 ± 1.78	64.40 ± 3.85
ActiveFT	58.80 ± 3.54	58.30 ± 2.46	47.44 ± 3.87	51.60 ± 5.28	61.80 ± 1.57	66.00 ± 2.83
DSS	**62.24** ± 1.13	53.90 ± 3.85	55.40 ± 4.02	52.00 ± 1.79	60.10 ± 1.28	65.60 ± 4.34
LLM2LLM	61.04 ± 0.92	49.60 ± 3.75	48.40 ± 2.61	49.20 ± 4.12	61.10 ± 1.16	64.80 ± 3.81
SuperCorrect	57.08 ± 1.20	44.20 ± 1.72	50.60 ± 2.88	51.60 ± 3.44	63.00 ± 1.52	65.20 ± 2.04
Ours	61.92 ± 1.34	**59.80** ± 2.25	**56.50** ± 1.50	**56.80** ± 3.25	**63.63** ± 1.43	**69.50** ± 1.48

5.4 ABLATION STUDY

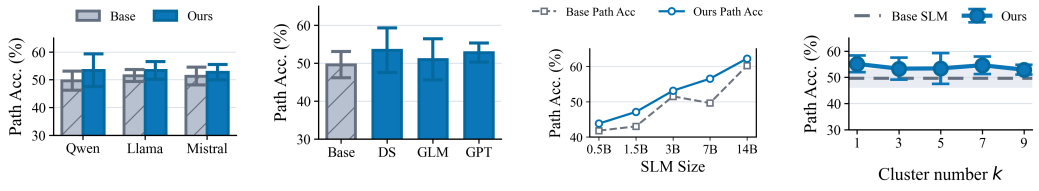
To isolate the adaptation signal, we keep the device model, LoRA budget, and evaluation protocol fixed while varying only the supervision used for adaptation. Answer-only LoRA uses the final answer, LLM-CoT SFT LoRA uses a complete teacher rationale, and our method conditions preference learning on the observed device trajectory and its diagnosis.

Table 2: Ablation study of adaptation signals across datasets. Each entry reports accuracy.

Method	MedQA	FOLIO	MATH	SATBench
Base SLM	58.72	60.80	57.50	49.08
Answer-only LoRA	59.40	64.00	58.80	55.70
LLM-CoT SFT LoRA	60.12	65.60	53.90	55.40
TLM	59.02	65.20	58.00	50.45
Diagnosis-guided LoRA (Ours)	**61.92**	**69.50**	**59.80**	**56.50**

Table 3: Path accuracy across SLM backbones and cloud LLM APIs on SATBench. The Base SLM does not invoke a cloud teacher and is repeated in the right panel for direct comparison.

SLM Backbone				**Cloud LLM API**			
Method	Qwen2.5 7B	Llama 3 8B	Mistral 7B	Method	DeepSeek V4 Flash	GLM 5.3 Flash	GPT 5.6 Luna
Base SLM	49.60	51.50	51.30	Base SLM	49.60	49.60	49.60
Ours	**53.40**	**53.34**	**52.70**	**Ours**	**53.40**	**51.00**	**52.80**



(a) SLM Backbone **(b)** Cloud Teacher **(c)** SLM Scale **(d)** Cluster Number k

Figure 4: Scalability and sensitivity analysis. **(a)** Performance across different SLM backbones. **(b)** Performance with different cloud teachers, including DeepSeek-V4-Flash (DeepSeek AI, 2026), GLM-5.3-Flash (GLM-5 Team, 2026), and GPT-5.6-Luna (OpenAI, 2026). **(c)** Performance across different SLM scales. **(d)** Sensitivity to the cluster number k. All results report path accuracy using five decoding paths per test query.

Table 2 shows that diagnosis-guided LoRA achieves the strongest result on every reported benchmark. This pattern is consistent with the benefit of conditioning adaptation on failure-specific diagnostic information rather than answer supervision or complete teacher CoT alone.

5.5 Scalability and Sensitivity Analysis

Table 3 tests whether diagnosis-guided adaptation generalizes beyond the default device and cloud model pair. Figure 4 reports the corresponding variation across decoding paths, model scale, and cluster number.

Across Qwen2.5-7B, Llama 3-8B, and Mistral-7B, our method improves mean Path Acc. by 1.40 to 3.80 points over the corresponding Base SLM. When the device model is fixed, DeepSeek-V4-Flash, GLM-5.3-Flash, and GPT-5.6-Luna all improve over the same local baseline, with DeepSeek-V4-Flash achieving the highest mean Path Acc. of 53.40. These results suggest that the adaptation signal is not restricted to a particular device backbone or cloud teacher.

Figure 4(c) shows positive gains from 0.5B to 14B parameters. Figure 4(d) shows that every evaluated cluster number improves over the Base SLM, indicating that the method is not sensitive to a precisely tuned value of k.

6 Conclusion

We presented a diagnosis guided framework for cloud assisted test time adaptation of small language models. The framework begins with erroneous device model reasoning paths and asks a stronger cloud model to identify corrective information for each failure. It then converts this information into preference pairs that update the device model through lightweight LoRA adaptation.

Experiments across medical question answering, mathematical reasoning, and logical reasoning show that diagnosis guided adaptation improves held out performance without cloud calls during deployment. The ablation results further indicate that answer supervision and teacher CoT supervision alone do not provide the same gains on the reported reasoning benchmarks. These findings support the value of connecting cloud feedback to the specific reasoning failure that produced it.

Future work will study more efficient selection of adaptation examples, stronger mechanisms for verifying cloud diagnoses, and continual adaptation under changing deployment distributions.

AI USE STATEMENT

In this work, we used generative AI tools to help develop conceptual frameworks and refine hypotheses, design and provide feedback on research methodology and experiments, implement methods, assist with translation, and interpret experimental results. A cloud language model is also a research component of CausalBridge. It generates structured diagnoses and minimally corrected trajectories that are filtered and used only as adaptation supervision. We did not use generative AI to generate or alter evaluation labels, reported metrics, or final empirical claims. Generative AI was not used to formulate mathematical claims, provide critical ingredients for proofs, or write proofs. Dataset cleaning and qualitative or thematic data analysis are not applicable to this work.

Additionally, we used generative AI tools to assist with code editing, literature search and summarization, manuscript drafting and editing, figure preparation, and reference formatting. The authors manually reviewed AI assisted writing and checked literature summaries against the cited papers. AI assisted code was inspected and its outputs were verified through the reported experimental runs. The authors reviewed all AI assisted work and take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The Method and Experiments sections specify the model implementation, training hyperparameters, datasets, baseline configurations, and evaluation protocol. Appendix A documents the cloud prompt, feedback filtering rules, dataset processing, adaptation pair construction, data splits, and evaluation details. The Theoretical Analysis appendix states the assumptions and provides the proof of the main theoretical result. We will submit an anonymous supplementary archive containing the source code, configuration files, and scripts needed to construct adaptation pairs and reproduce the reported tables and figures.

REFERENCES

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*, 2023.

Guohao Chen, Shuaicheng Niu, Deyu Chen, Shuhai Zhang, Changsheng Li, Yuanqing Li, and Mingkui Tan. Cross-device collaborative test-time adaptation. *Advances in Neural Information Processing Systems*, 37:122917–122951, 2024a.

Yaofo Chen, Shuaicheng Niu, Yaowei Wang, Shoukai Xu, Hengjie Song, and Mingkui Tan. Towards robust and efficient cloud-edge elastic model adaptation via selective entropy distillation. In *International Conference on Learning Representations*, volume 2024, pp. 41226–41256, 2024b.

Eugene Choi, Arash Ahmadian, Matthieu Geist, Olivier Pietquin, and Mohammad Gheshlaghi Azar. Self-improving robust preference optimization. In *International Conference on Learning Representations*, volume 2025, pp. 4047–4068, 2025.

DeepSeek AI. Deepseek v4 technical documentation. Model card, 2026. URL https://fe-static.deepseek.com/chat/transparency/deepseek-V4-model-card-EN.pdf.

Guhao Feng, Shengjie Luo, Kai Hua, Ge Zhang, Wenhao Huang, Di He, and Tianle Cai. In-place test-time training. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 114452–114472, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/ba6f5879d7f382241316f7dda7e6b8ac-Paper-Conference.pdf.

Yulu Gan, Mingjie Pan, Rongyu Zhang, Zijian Ling, Lingran Zhao, Jiaming Liu, and Shanghang Zhang. Cloud-device collaborative adaptation to continual changing environments in the real-world. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12157–12166. IEEE, 2023.

GLM-5 Team. GLM-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026.

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Wenfei Zhou, James Coady, David Peng, Yujie Qiao, Luke Benson, et al. FOLIO: Natural language reasoning with first-order logic. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 22017–22031, 2024.

Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Cheng-Yu Hsieh, Chun-Liang Li, Chien-yu Yu, Anik Nuyten, Jiatao Fu, Dror Rotem, Yuxiao Ke, Tushar Khot, et al. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 8003–8017, 2023.

Jinwu Hu, Zhitian Zhang, Guohao Chen, Xutao Wen, Chao Shuai, Wei Luo, Bin Xiao, Yuanqing Li, and Mingkui Tan. Test-time learning for large language models. *arXiv preprint arXiv:2505.20633*, 2025.

Chengsong Huang, Tong Zheng, Langlin Huang, Jinyuan Li, Haolin Liu, and Jiaxin Huang. Relayllm: Efficient reasoning via collaborative decoding. *arXiv preprint arXiv:2601.05167*, 2026.

Jonas Hübotter, Sascha Bongni, Ido Hakimi, and Andreas Krause. Efficiently learning at test-time: Active fine-tuning of llms. In *International Conference on Learning Representations*, volume 2025, pp. 74978–75035, 2025.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *arXiv preprint arXiv:2009.13081*, 2020.

Martin Klissarov, Jonathan Cook, Diego Antognini, Hao Sun, Jingling Li, Natasha Jaques, Claudiu Musat, and Edward Grefenstette. Improving interactive in-context learning from natural language feedback. *arXiv preprint arXiv:2602.16066*, 2026.

Nicholas Lee, Thanakul Wattanawong, Sehoon Kim, Karttikeya Mangalam, Sheng Shen, Gopala Anumanchipalli, Michael Mahoney, Kurt Keutzer, and Amir Gholami. LLM2LLM: Boosting LLMs with novel iterative data enhancement. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024.

Jiahui Li, Jianfeng Shan, Wenpei Chen, Shunyu Wu, Jian Lou, Wenjie Feng, Dan Li, and See-Kiong Ng. Exploiting verification-generation gap: Test-time reinforcement learning with confidence-conditioned verification. *arXiv preprint arXiv:2606.03608*, 2026.

Ruotong Liao, Nikolai Röhrich, Xiaohan Wang, Yuhui Zhang, Yasaman Samadzadeh, Volker Tresp, and Serena Yeung-Levy. Tool verification for test-time reinforcement learning. *arXiv preprint arXiv:2603.02203*, 2026.

Hanmeng Liu, Jian Liu, Leyang Cui, Zhiyang Teng, Nan Duan, Ming Zhou, and Yue Zhang. LogiQA 2.0—an improved dataset for logical reasoning in natural language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:2947–2962, 2023. doi: 10.1109/TASLP.2023.3293046.

David Lopez-Paz, Léon Bottou, Bernhard Schölkopf, and Vladimir Vapnik. Unifying distillation and privileged information. *arXiv preprint arXiv:1511.03643*, 2015.

Niqi Lyu, Pengtao Shi, Wei Qiu, Jianlin Zhong, Sicong Xia, Jianyao Ma, and Yicheng Ding. Pyrodash: Cost-efficient token-level small-large language model collaborative inference. *arXiv preprint arXiv:2607.20327*, 2026.

OpenAI. GPT-5.6 luna model. OpenAI API documentation, 2026. URL `https://developers.openai.com/api/docs/models/gpt-5.6-luna`.

Mary Phuong and Christoph Lampert. Towards understanding knowledge distillation. In *International conference on machine learning*, pp. 5142–5151. PMLR, 2019.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025.

Aarohi Srivastava et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*, 2023.

Vladimir Vapnik and Rauf Izmailov. Learning using privileged information: similarity control and knowledge transfer. *The Journal of Machine Learning Research*, 16(1):2023–2049, 2015.

Guanqun Wang, Jiaming Liu, Chenxuan Li, Yuan Zhang, Junpeng Ma, Xinyu Wei, Kevin Zhang, Maurice Chong, Renrui Zhang, Yijiang Liu, et al. Cloud-device collaborative learning for multimodal large language models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12646–12655. IEEE, 2024.

Anjiang Wei, Yuheng Wu, Yingjia Wan, Tarun Suresh, Huanmi Tan, Zhanke Zhou, Sanmi Koyejo, Ke Wang, and Alex Aiken. SATBench: Benchmarking LLMs' logical reasoning via automated puzzle generation from SAT formulas. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025a.

Chenxing Wei, Hong Wang, Ying He, Fei Yu, and Yao Shu. Test-time policy adaptation for enhanced multi-turn interactions with llms. *arXiv preprint arXiv:2509.23166*, 2025b.

Cindy Wu, Ekdeep Singh Lubana, Bruno Kacper Mlodozeniec, Robert Kirk, and David Krueger. What mechanisms does knowledge distillation distill? In *Proceedings of UniReps: the First Workshop on Unifying Representations in Neural Models*, pp. 60–75. PMLR, 2024.

Longhuan Xu, Cunjian Chen, and Feng Yin. Unsupervised layer-wise dynamic test time adaptation for llms. *arXiv preprint arXiv:2602.09719*, 2026.

Ling Yang, Zhaochen Yu, Tianjun Zhang, Minkai Xu, Joseph E. Gonzalez, Bin Cui, and Shuicheng Yan. SuperCorrect: Supervising and correcting language models with error-driven insights. In *International Conference on Learning Representations*, 2025.

Hang Zeng, Xiangyu Liu, Yong Hu, Chaoyue Niu, Jiarui Zhang, Shaojie Tang, Fan Wu, and Guihai Chen. Learning to seek help: Dynamic collaboration between small and large language models. *arXiv preprint arXiv:2604.17827*, 2026.

Chuang Zhang, Zizhen Zhu, Yihao Wei, Bing Tian, Junyi Liu, Henan Wang, Wang Xavier, and Yaxiao Liu. Confidence-calibrated small-large language model collaboration for cost-efficient reasoning. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4480–4501, 2026.

Qiyong Zhong, Mao Zheng, Mingyang Song, Xin Lin, Jie Sun, Houcheng Jiang, Xiang Wang, and Junfeng Fang. Sod: Step-wise on-policy distillation for small language model agents. *arXiv preprint arXiv:2605.07725*, 2026.

Zhiyuan Zhou, Andy Peng, Charles Xu, Qiyang Li, Tobias Springenberg, Kevin Frans, and Sergey Levine. Test-time gradient guidance of flow policies in reinforcement learning. *arXiv preprint arXiv:2606.11087*, 2026.

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Xinwei Long, Ermo Hua, Biqing Qi, et al. Ttrl: Test-time reinforcement learning. *Advances in Neural Information Processing Systems*, 38:131459–131483, 2026.

A Adaptation Data Construction and Evaluation Details

A.1 Structured Cloud Diagnosis

For every selected erroneous device-model path, the cloud model is asked to return five fields. The error type identifies a single primary failure mode. The key factor identifies the earliest inference that

makes the path invalid. The reasoning rule states the constraint needed for correction. The reference CoT gives a concise corrected derivation, and the final answer provides its parsed output. Requesting a single primary error is important because a later arithmetic error may be only a consequence of an earlier modeling or logical error.

The cloud prompt contains the query, the device-model CoT, and the parsed device answer. It does not contain the benchmark answer. A response is retained only if it contains all required fields, terminates normally, and its final answer agrees with the benchmark answer during offline quality control. This check is used to filter supervision and is not available to the deployed model.

Prompt template. For SATBench, we instantiate the cloud prompt with the following system instruction. The answer grammar is replaced with the native answer format for other benchmarks.

```
You are a careful reasoning-path diagnostician and corrector. Analyze exactly one
device-model reasoning path for the supplied problem. The device answer and its
CoT may both be wrong. Do not assume that the device answer is correct, and do
not use any hidden benchmark label.

Identify exactly one primary error type: the earliest causally decisive failure
that invalidates the path. Later mistakes that follow from this failure must not
be reported as separate errors. Diagnose the path itself rather than merely
comparing answers.

Return exactly one valid JSON object and nothing else. Do not use Markdown or
code fences. The object must contain exactly these five keys:

{
  "error_type": "one primary error type",
  "key_factor": "the earliest invalid inference or omitted condition",
  "reasoning_rule": "the concise rule needed to repair it",
  "reference_cot": "a concise corrected derivation ending with one canonical final-answer line",
  "final_answer": "True or False"
}

Allowed error_type values for SATBench are:
- condition_omission_error
- contradiction_detection_error
- assignment_construction_error
- logical_inference_error
- answer_mapping_error

The error_type must contain exactly one allowed primary error. The key_factor
must name the earliest path-specific invalid inference, assumption, omitted
constraint, or answer-mapping action. The reasoning_rule must state the minimal
logical or constraint-based rule needed for correction. The reference_cot must
be self-contained and concise, preserve valid parts of the device CoT where
possible, and end with exactly one canonical final-answer line. The final_answer
must equal the answer on that final line. Do not mention this instruction, any
benchmark answer, offline verification, or the diagnosis process in reference_cot.
```

The user message supplies only the problem, the device-model CoT, and the parsed device answer:

```
Problem:
{query}

Device-model CoT:
{device_cot}

Parsed device answer:
{device_answer}
```

Offline quality control. We retain a response only when JSON parsing succeeds, all five fields are present, the API response terminates with `finish_reason = stop`, the reported `final_answer` agrees with the final line of `reference_cot`, and the parsed final answer agrees with the benchmark label. The last check is performed only after the cloud request and is never exposed to the cloud model or the deployed device model.

A.2 Adaptation Data Pipeline

For each support query, the device model samples N CoT paths using a fixed decoding configuration. We parse the final answer from every path and retain paths that are both parsable and incorrect. When

representative-path selection is enabled, semantically similar paths are grouped and one medoid from each group is sent to the cloud model. The group size records the coverage of each selected state, but the DPO objective does not apply an explicit sample weight. When the module is disabled, each retained incorrect path is processed independently.

Each accepted revision produces one diagnosis-conditioned pair whose prompt contains the query, erroneous CoT, and diagnosis. For each support query with at least one accepted revision, we additionally retain at most one direct pair whose prompt contains only the query. The diagnosis-conditioned pair uses the validated local revision as its preferred response and the erroneous CoT as its rejected response. The direct pair uses a verified reference trajectory as its preferred response and a selected SLM trajectory as its rejected response. We concatenate and shuffle the two pair types before DPO training, so the model learns both observed-failure repair and query-only behavior without diagnostic context.

A.3 Diagnostic Intervention Protocol

The diagnostic intervention keeps the query, erroneous path, reference CoT, final answer, LoRA budget, and optimization schedule fixed. It changes only the diagnosis field in the diagnosis-conditioned preference pair. We compare the original diagnosis with an empty diagnosis, a diagnosis matched from another support example, and a diagnosis drawn from a distinct reasoning-path group for the same query. A reduction in accuracy under these replacements indicates that the original diagnosis carries path-specific corrective information rather than acting only as additional prompt text.

A.4 Evaluation Protocol

The support and test sets are disjoint. The cloud model accesses only the support set, where it constructs adaptation data. All methods are evaluated on the same held-out test set without cloud calls. For SATBench, the first 200 examples form the support set and the following 200 examples form the test set.

For each test query, a model generates five CoT paths with the same decoding configuration. Accuracy is computed from the parsed final answer for each path. The main table reports mean accuracy and standard deviation across the five independently sampled paths. The ablation table reports final-answer accuracy for its stated configurations. All training-based methods use the same LoRA budget and one adaptation epoch.

B Extended Related Work

Our work is related to test-time learning for language models and small-large model collaboration. The former studies how deployed models adapt with target data or feedback, while the latter studies how weak and strong models cooperate during inference or adaptation.

B.1 Test-Time Learning and Model Adaptation

Test-time learning can be studied from the updated structure and the adaptation signal. Fast weights update Transformer MLP projection matrices for test-time adaptation (Feng et al., 2026). Prompt-only LoRA uses layer-wise and step-wise learning-rate scaling (Xu et al., 2026). Input-perplexity minimization updates LoRA parameters on unlabeled test prompts (Hu et al., 2025), while active fine-tuning selects informative examples (Hübotter et al., 2025). Online interaction feedback can also adapt a small set of policy parameters (Wei et al., 2025b).

Reasoning-oriented methods further emphasize feedback reliability. Majority-voted pseudo-labels can be converted into reinforcement learning rewards (Zuo et al., 2026). Tool verification reweights reasoning trajectories (Liao et al., 2026), and confidence-conditioned verification filters pseudo-labels while preserving candidate diversity (Li et al., 2026). Broader guidance methods also use external value signals at test time to guide flow-policy sampling with critic gradients (Zhou et al., 2026).

Feedback-based adaptation also uses weak answers, student trajectories, or natural-language criticism. Preference optimization can improve a weak answer (Choi et al., 2025). On-policy distilla-

tion combines student trajectories with teacher distributions and tool feedback (Zhong et al., 2026), while language-feedback training teaches models to revise answers and later internalize feedback patterns (Klissarov et al., 2026).

Our focus is the adaptation signal. Instead of using likelihood, selected examples, interaction rewards, pseudo-labels, tool verification, or teacher distributions, we use structured diagnoses generated by a stronger language model for erroneous reasoning paths from the small model. These diagnoses condition preference learning without requiring the small model to reproduce the diagnosis text.

B.2 Small-Large Model Collaboration

Small-large model collaboration can be organized into collaborative inference and collaborative training. Collaborative inference includes token-level and task-level cooperation.

At token level, speculative sampling uses a draft model to propose tokens and a target model to verify them with rejection sampling (Chen et al., 2023). RelayLLM trains the small model to call the large model at key token positions (Huang et al., 2026), while PyroDash uses control tokens to trigger large-model takeover under a cost-aware GRPO objective (Lyu et al., 2026).

At task level, a small model can be trained to ask for help and continue reasoning after feedback (Zeng et al., 2026). Confidence calibration can instead transfer low-confidence samples to the large model (Zhang et al., 2026).

Collaborative training asks whether weak models can learn from stronger models or external feedback. Step-wise distillation combines student trajectories, teacher distributions, and tool feedback (Zhong et al., 2026). Teacher criticism can also be internalized through training (Klissarov et al., 2026).

Cloud-device adaptation studies a related deployment setting. Uncertain device samples can be uploaded for visual prompt learning (Gan et al., 2023). Selective entropy distillation reduces uploads through unreliable and low-information sample filtering before cloud-side distillation (Chen et al., 2024b). Multimodal cloud-device collaboration selects difficult tokens from compressed on-device models and transfers cloud knowledge through adapter-based distillation (Wang et al., 2024). Cross-device adaptation instead shares domain knowledge vectors across devices (Chen et al., 2024a).

In contrast, we treat erroneous chain-of-thought paths from the small model as learnable failure states. The large model diagnoses error types and causal reasoning rules, and the small model converts the diagnosis into LoRA preference updates. The large model does not replace final inference or simply provide standard distillation targets. It provides structured causal feedback that improves later reasoning accuracy and multi-path stability.

C Theoretical Analysis

The SCM in Section 3 identifies a failure-relevant causal target through the diagnostic mediator S. We connect this target to local parameter geometry to compare CausalBridge with direct distillation (DD) under finite-sample adaptation and diagnosis-free deployment.

DD imitates complete LLM feedback Z_l, which contains the failure-relevant component Z_{rel} and an irrelevant component Z_{irr}. CausalBridge instead uses a diagnosis S, a validated local revision when available, and the two-branch preference objective in Section 4. The comparison balances the nuisance cost of learning from Z_{irr} against the error of estimating and internalizing the S-mediated correction.

C.1 Local Parameter Geometry

Let θ_s denote the pre-adaptation SLM parameters and let $R(\theta)$ be the cloud-free population risk on unseen queries. For an adaptation increment Δ, consider the local quadratic model

$$R(\theta_s + \Delta) = R(\theta_s) - g^\top \Delta + \frac{1}{2}\Delta^\top H \Delta, \tag{14}$$

where g is the locally beneficial direction and $H \succeq 0$ is the local curvature.

Let $\mathcal{V}_{\mathrm{rel}}$ be the parameter subspace that changes the S-mediated corrective mechanisms identified by the SCM, and let P_{rel} be its orthogonal projector. We assume that (i) the locally beneficial direction satisfies $g \in \mathcal{V}_{\mathrm{rel}}$; and (ii) $P_{\mathrm{rel}}H = HP_{\mathrm{rel}}$, so local curvature preserves $\mathcal{V}_{\mathrm{rel}}$ and its orthogonal complement.

Decompose the DD increment as

$$\Delta_{\mathrm{DD}} = \Delta_{\mathrm{rel}} + \Delta_{\mathrm{irr}}, \Delta_{\mathrm{rel}} = P_{\mathrm{rel}}\Delta_{\mathrm{DD}}, \Delta_{\mathrm{irr}} = (I - P_{\mathrm{rel}})\Delta_{\mathrm{DD}}. \tag{15}$$

Here Δ_{rel} is the parameter effect associated with the failure-relevant target, whereas Δ_{irr} captures teacher style, already-mastered content, alternative reasoning plans, and other effects of Z_{irr}. Model the CausalBridge increment as

$$\Delta_{\mathrm{CB}} = \Delta_{\mathrm{rel}} + \varepsilon_{\mathrm{med}}, \tag{16}$$

where $\varepsilon_{\mathrm{med}}$ collects errors from probing $P(X)$, generating S, approximating the corrective relation $P(Y \mid S, X)$, finite-sample optimization, and internalizing training-only diagnosis for diagnosis-free deployment.

C.2 Risk Advantage

Proposition 1 (mediation advantage). Under equation 14 and the SCM-to-optimization compatibility assumptions above,

$$R(\theta_s + \Delta_{\mathrm{DD}}) - R(\theta_s + \Delta_{\mathrm{CB}}) = \frac{1}{2}\Delta_{\mathrm{irr}}^{\top} H \Delta_{\mathrm{irr}} - \Gamma(\varepsilon_{\mathrm{med}}; \Delta_{\mathrm{rel}}), \tag{17}$$

where

$$\Gamma(\varepsilon; \Delta_{\mathrm{rel}}) = -g^{\top}\varepsilon + \Delta_{\mathrm{rel}}^{\top} H \varepsilon + \frac{1}{2}\varepsilon^{\top} H \varepsilon. \tag{18}$$

Consequently, CausalBridge has strictly lower local deployment risk than DD whenever

$$\Gamma(\varepsilon_{\mathrm{med}}; \Delta_{\mathrm{rel}}) < \frac{1}{2}\Delta_{\mathrm{irr}}^{\top} H \Delta_{\mathrm{irr}}. \tag{19}$$

The same statement holds in expectation for stochastic support sets and teacher responses after taking expectations on both sides.

Proof. Substitute equations 15 and 16 into equation 14. Because $g \in \mathcal{V}_{\mathrm{rel}}$ and $\Delta_{\mathrm{irr}} \in \mathcal{V}_{\mathrm{rel}}^{\perp}$, $g^{\top}\Delta_{\mathrm{irr}} = 0$. Because $P_{\mathrm{rel}}H = HP_{\mathrm{rel}}$, H preserves the two orthogonal subspaces and $\Delta_{\mathrm{rel}}^{\top} H \Delta_{\mathrm{irr}} = 0$. Expanding both quadratic forms and subtracting gives equation 17; condition 19 gives the strict comparison. □

The two terms have direct causal interpretations. The nuisance term $\Delta_{\mathrm{irr}}^{\top} H \Delta_{\mathrm{irr}}/2$ measures the local risk induced by learning from Z_{irr}, outside the S-mediated target. The mediation penalty $\Gamma(\varepsilon_{\mathrm{med}}; \Delta_{\mathrm{rel}})$ measures the cost of estimating and internalizing that target. With an oracle mediator, $\varepsilon_{\mathrm{med}} = 0$, and CausalBridge improves over DD whenever DD has a nonzero irrelevant component in a positive-curvature direction.

If $P_{\mathrm{rel}}H \neq HP_{\mathrm{rel}}$, the risk gap additionally contains $\Delta_{\mathrm{rel}}^{\top} H \Delta_{\mathrm{irr}}$. The sufficient condition becomes $\Gamma(\varepsilon_{\mathrm{med}}; \Delta_{\mathrm{rel}}) < \Delta_{\mathrm{irr}}^{\top} H \Delta_{\mathrm{irr}}/2 + \Delta_{\mathrm{rel}}^{\top} H \Delta_{\mathrm{irr}}$.

C.3 Connection to the Training Objective

The three method components correspond to the SCM terms and reduce different parts of $\varepsilon_{\mathrm{med}}$. Reasoning State Probing selects representative trajectories from the sampled state distribution. Causal Diagnostic Mediation uses $s = f_{\theta_l}^{\mathrm{diag}}(x, c^-)$ to construct diagnosis-guided revisions with validated local-edit constraints. Mediator-Guided Evolution trains on both diagnosis-conditioned repair pairs and query-only direct pairs. The resulting DPO objective is an empirical surrogate for internalizing corrective information, rather than an estimator of the front-door adjustment.

At the optimization level, validated local revisions preserve anchored source spans and maintain a minimum lexical copy ratio when a local repair is feasible. Their score difference can therefore focus the preference gradient on a smaller correction than complete-feedback distillation. Schema

and answer-agreement filtering further reduce mediator and correction error. Together, these mechanisms are intended to suppress nuisance updates and reduce $\varepsilon_{\mathrm{med}}$ in equation 16.

Diagnosis acts as privileged information during adaptation (Vapnik & Izmailov, 2015; Lopez-Paz et al., 2015) and is internalized through parameter updates for query-only deployment. We include the residual internalization error in $\varepsilon_{\mathrm{med}}$. Randomly matched diagnoses, unfiltered feedback, and full rewrites should increase this term, whereas stable gradients under diagnosis paraphrases and stronger alignment with held-out loss reduction indicate successful mediation. Existing distillation theory emphasizes data geometry and optimization bias (Phuong & Lampert, 2019), while mechanism-level studies show that output matching may miss the teacher's mechanism (Wu et al., 2024). Proposition 1 connects these observations to the SCM through a sufficient local condition for improving upon complete-feedback imitation.