DiT-OPD: Diagnose, then Teach in On-Policy Distillation

Anonymous authors
Paper under double-blind review

Abstract

On-policy distillation (OPD) enables a student model to learn from teacher feedback on its own generated trajectories. Both the selection of supervision positions and the construction of teacher feedback affect answer accuracy and the reasoning length required to solve a problem. Existing approaches rely solely on statistics to determine where supervision is needed and apply globally uniform teacher feedback, resulting in weak error-correction capabilities and lengthy reasoning chains. To address this problem, we propose DiT-OPD (Diagnose, then Teach in On-Policy Distillation), which extracts guidance needs from the student's available prefix and uses the same diagnostic action to jointly select subsequent blocks and condition teacher scoring. The teacher identifies four candidate guidance needs: knowledge recall, contradiction resolution, derivation progress, and verification guidance. When no clear need is identified, the subsequent block receives no direct distillation supervision; tokens in selected blocks receive action-conditioned supervision. With Qwen3-8B supervising Qwen3-1.7B, DiT-OPD improves average pass@1 and pass@8 across six competition mathematics benchmarks by -0.25% and 0.48%, respectively, over standard OPD, while reducing reasoning length by 12.95%. On LiveCodeBench, it improves pass@8 over the base model by 6.26% and reduces reasoning length by 11.95% relative to standard OPD. In self-repair evaluation, it also improves repair pass@1 over standard OPD by 1.86%, enhancing the student's error-correction capability.

1 Introduction

Large language models often require long reasoning chains and multiple deduction steps to solve competition mathematics and complex reasoning tasks. Larger models incur substantial inference time and latency, whereas smaller models offer faster inference but lower accuracy on competition-level mathematics. On-policy distillation (OPD) offers a solution by training a smaller student on teacher feedback for its own trajectories, transferring reasoning capabilities from a larger model (Agarwal et al., 2024; Gu et al., 2024). However, local cognitive shortcomings can readily disrupt the student's progress through a long derivation: it may misremember a key formula, develop an unnoticed logical contradiction, or engage in inefficient trial and error without an intermediate subgoal. These heterogeneous, local reasoning errors require a distillation system to address two interdependent questions at each supervision decision: where to supervise (*Where*) and how to construct targeted teacher feedback (*How*). This paper therefore focuses on using the current reasoning state to jointly organize supervision positions and feedback conditions.

For supervision selection, prior work uses teacher–student distributional differences or predictive uncertainty to select tokens and adjust supervision weights (Wang et al., 2021; Huang et al., 2026; Jin et al., 2026). These methods identify high-disagreement tokens through statistics such as predictive entropy or teacher–student divergence (e.g., reverse KL). Such scalar statistics, however, reveal where the student disagrees substantially with the teacher without accurately identifying the specific type of error. They cannot diagnose whether disagreement stems from a missing formula, a logical conflict, or a valid alternative solution, making it difficult for the teacher to provide targeted corrective feedback. Meanwhile, local OPD signals can reinforce repetitive continuations and create a self-reinforcing cycle through on-policy sampling (Luo et al., 2026). When the teacher conditions on the same repetitive prefix, even a globally degenerate response can exhibit low teacher–student divergence (Yu et al., 2026), leading to lengthy student reasoning chains. These observations indi-

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.

cate that local statistical agreement does not fully capture effective reasoning progress, motivating supervision organized around the current reasoning state.

For feedback construction, existing methods provide teachers with additional information, such as reference answers or environmental evaluation feedback, to construct task-specific conditional supervision (Zhao et al., 2026; Hübotter et al., 2026). This external information specifies the target answer or reveals shortcomings in the student's output, enabling more targeted corrective feedback. Although these methods recognize the need for additional semantic guidance, their feedback depends heavily on external privileged information, such as reference answers or environmental evaluators. Consequently, they often apply a single, globally uniform prompt condition to the entire sequence. Such supervision struggles to accommodate frequent changes in local cognitive state along a long reasoning chain and can over-intervene when the student's derivation is progressing smoothly, disrupting its problem-solving process.

To address these problems, we implement Diagnostic Block Selection (DBS) and Action-Conditioned Feedback (ACF). DBS extracts discrete guidance needs from the student's prefix, uses them to select subsequent supervision blocks, and supplies the corresponding guidance conditions to the teacher. By drawing on the teacher's strong semantic understanding, it accurately diagnoses student errors and alleviates overly lengthy reasoning chains. ACF uses the student's currently available prefix as the basis for diagnosis and connects diagnostic actions to both block selection and teacher scoring prompts. By injecting the teaching strategy corresponding to the student's error type as privileged information into the teacher, ACF enables more fine-grained corrective supervision, improving the student's independent reasoning and error-correction capabilities.

Building on this analysis, we propose DiT-OPD (Diagnose, then Teach in On-Policy Distillation). The teacher identifies guidance needs using only the student prefix preceding the target reasoning block. When a clear need is identified, it selects the corresponding teaching action—knowledge recall, contradiction resolution, derivation progress, or verification guidance; otherwise, it skips direct distillation supervision of the subsequent block. The selected action both determines whether the next block contributes to the distillation loss and changes the conditions under which the teacher scores the student's original tokens through a fixed prompt. Diagnosis and scoring occur only during training, with no additional teacher calls at student inference. This design turns prefix-level needs into an executable supervision rule; we assess the student's reasoning capability through solution quality and generation length.

Our contributions are as follows:

- We introduce DiT-OPD, which couples DBS and ACF through the same prefix-level diagnostic action to implement need-driven on-policy distillation on original student trajectories.
- Across six competition mathematics benchmarks, DiT-OPD improves average pass@1 and pass@8 over standard OPD by -0.25% and 0.48%, respectively, while reducing reasoning length by 12.95%. On LiveCodeBench, it improves pass@8 over the base model by 6.26% and reduces reasoning length relative to standard OPD by 11.95%.
- Comparisons with Where-only, How-only, and statistical selection combined with ACF provide experimental evidence for coupling DBS and ACF. LiveCodeBench code-generation and self-repair results further show that the benefits extend to code-based problem solving and improved self-correction.

2 Related Work

On-policy distillation and selective supervision. GKD distills on student-generated sequences, MiniLLM studies reverse-KL optimization, and DistiLLM combines divergence design with data reuse (Agarwal et al., 2024; Gu et al., 2024; Ko et al., 2024). Selective distillation has several forms: Wang et al. select tokens by student cross-entropy in machine translation (Wang et al., 2021); SelecTKD weights tokens (Huang et al., 2026); other methods train on reasoning prefixes or adapt objectives to token difficulty (Zhang et al., 2026; Zhong et al., 2024). EOPD adds a forward-KL term at high-teacher-entropy positions (Jin et al., 2026). StableOPD addresses length inflation with a reference-distribution constraint and a rollout mixture, while TIDE adjusts excess and deficit in token probabilities (Luo et al., 2026; Yu et al., 2026). These studies rely solely on statistics to iden-

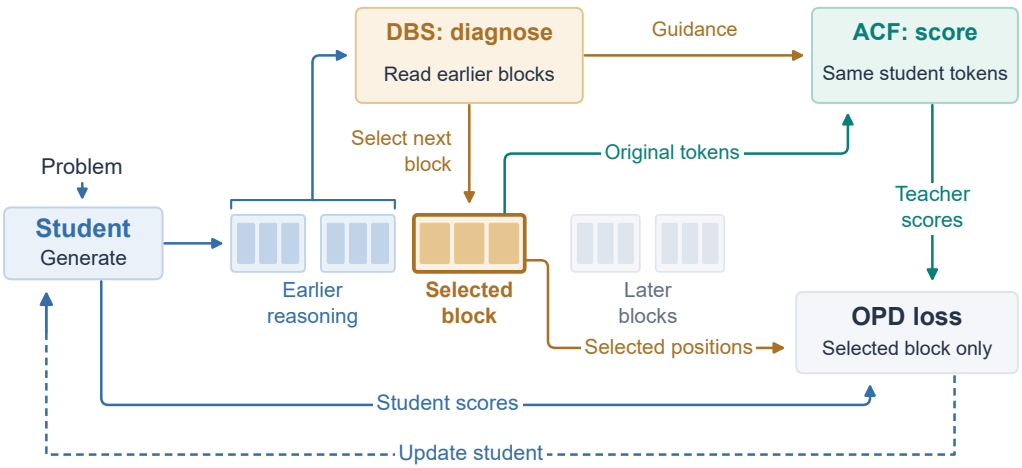

Figure 1: DiT-OPD architecture. Blue, amber, and grey indicate earlier reasoning, the current selected block, and later blocks. DBS reads the problem and earlier blocks; the same diagnosis selects the next block and conditions ACF. ACF scores original student tokens causally. Teacher and student scores are sampled-token log-probabilities. Diagnosis and scoring share a frozen teacher; only the student is updated (dashed arrow). Block and token counts are schematic.

tify positions of substantial teacher–student disagreement and apply supervision at those positions. However, they do not account for the different teaching actions required by different student error types, which degrades the student's error-correction capability.

Conditioned teachers and feedback distillation. OPSD conditions on reference solutions, SDFT uses contextual demonstrations, and SDPO distills predictions conditioned on environmental feedback or successful rollouts (Zhao et al., 2026; Shenfeld et al., 2026; Hübotter et al., 2026). These studies show that teacher-side context can change the supervision signal, although its effects depend on the task and implementation. DiT-OPD derives its conditioning information from prefix diagnosis, with the same action controlling subsequent-block selection and scoring prompts. Table 8 compares these information interfaces. However, these methods offer a limited way of constructing teacher supervision from additional context and consequently often apply a globally uniform prompt condition to the entire sequence. This can disrupt correct student reasoning steps and lead to longer reasoning chains.

Reasoning behaviors and process guidance. Heuristic problem solving and metacognitive monitoring provide conceptual background for our guidance categories (Pólya, 1945; Schoenfeld, 1985). SkillFactory studies self-distillation of cognitive behaviors, TriPosT trains self-improvement through interactive demonstrations, and Math-Shepherd provides step-level process supervision (Sprague et al., 2026; Yu et al., 2024; Wang et al., 2024a). Inspired by these studies, DiT-OPD defines four types of reasoning errors and corresponding teaching actions, using those actions to condition teacher scoring during training and thereby improve the student's reasoning patterns.

3 Method: DiT-OPD

3.1 OPD preliminaries and overview

Student trajectories and teacher supervision. On-policy distillation (OPD) first samples responses from the student, then obtains teacher supervision on those responses (Agarwal et al., 2024). Let x denote a problem and its original input messages, π_θ the trainable student, and q_ϕ the frozen teacher. At iteration k, the student before the update generates $y = (y_1, \ldots, y_T) \sim \pi_{\theta_k}(\cdot \mid x)$.

Write $h_t = (x, y_{<t})$ for the context at position t, and $q_\phi^0(\cdot \mid h_t)$ for the ordinary teacher without an additional guidance prompt. Both models predict the next token from the student's prefix.

Using reverse-KL distillation (Gu et al., 2024), a local objective on this iteration's fixed student prefixes is

$$\mathcal{L}_{\mathrm{OPD}}(\theta) = \mathbb{E}_{x,\, y\sim\pi_{\theta_k}}\left[\frac{1}{T}\sum_{t=1}^{T} D_{\mathrm{KL}}\big(\pi_\theta(\cdot \mid h_t) \,\|\, q_\phi^0(\cdot \mid h_t)\big)\right], \tag{1}$$

where $D_{\mathrm{KL}}(p \,\|\, q) = \sum_{v\in\mathcal{V}} p(v)\log[p(v)/q(v)]$ and $\mathcal{V}$ is the vocabulary. Equation (1) specifies the ordinary teacher's target distribution. In practice, training constructs feedback from the log-probability difference on the sampled token:

$$A_t^0 = \mathrm{stopgrad}\big[\beta\,\big(\log q_\phi^0(y_t \mid h_t) - \log \pi_{\theta_k}(y_t \mid h_t)\big)\big]. \tag{2}$$

Here $\mathrm{stopgrad}$ stops gradients and β is the distillation coefficient. At $\theta = \theta_k$ on a fixed prefix, the expected policy gradient weighted by A_t^0 follows the negative local reverse-KL gradient. Freezing this feedback and applying selection and clipping defines the implemented surrogate. The feedback is positive when the teacher assigns higher probability to the original token than the student, and negative otherwise. It expresses a relative probability preference, not a correctness label. Section 3.4 specifies the student update.

The two components of DiT-OPD. *Diagnostic Block Selection (DBS)* identifies the guidance needed from what the student has already written and selects the subsequent contiguous block for supervision. *Action-Conditioned Feedback (ACF)* maps that guidance category to a teacher prompt and scores the block's original tokens under this condition. Let a_j denote the guidance category for block j. DBS produces a supervision mask m_t, while ACF provides the conditional teacher probability $q_\phi^{a_j}(y_t \mid h_t)$; together they determine the distillation update (Figure 1). An "action" is a teacher-guidance category, not a sampled student token.

3.2 DBS: Diagnosing guidance needs and selecting blocks

Blocks and visible context. We partition the student response into contiguous token blocks:

$$B_j = \{b_j, \ldots, \min(b_j + B - 1, T)\}, \qquad b_j = (j-1)B + 1. \tag{3}$$

We use $B = 256$ and retain the final block's actual length. A block is a supervision unit; it need not coincide with a complete sentence or reasoning step. For $j > 1$, diagnosis reads only the problem x and the prefix preceding B_j:

$$(a_j, e_j) = D_\phi(x, y_{<b_j}), \qquad a_j \in \mathcal{A} \cup \{\texttt{skip}\}. \tag{4}$$

D_ϕ calls the frozen teacher with a diagnostic prompt. The action set $\mathcal{A}$ contains knowledge recall, contradiction resolution, derivation progress, and verification guidance. The diagnostic record e_j includes a short reason and, for a non-`skip` action, a visible block index; `skip` uses index -1. Diagnosis is performed after the complete trajectory has been generated.

When guidance is needed. DBS asks whether the prefix contains concrete evidence of an unresolved guidance need. Table 1 summarizes the four categories and corresponding instructions. Ordinary exploration, useful verification, resolved contradictions, and incomplete sentences or formulas caused by a block boundary should not by themselves trigger guidance. For example, a student still expanding a quadratic expression may simply continue; repeatedly relying on incompatible conclusions can instead call for contradiction resolution. The four prompt-defined categories may overlap; the teacher selects the most relevant action, or `skip` if no clear need is identified.

From diagnosis to a supervision mask. A non-`skip` action selects every token in B_j; otherwise the block is skipped:

$$m_t = \begin{cases} 0, & t \in B_1, \\ \mathbb{I}[a_j \neq \texttt{skip}], & t \in B_j,\ j > 1, \end{cases} \tag{5}$$

Here $\mathbb{I}[\cdot]$ is the indicator function. The first block receives no direct distillation loss. Evidence comes from preceding blocks, while supervision applies to the next block (Figure 2a).

Table 1: DBS guidance categories and their fixed ACF instructions. Each template combines shared requirements with one action-specific instruction.

Action a_j	Evidence in the prefix	Guidance in $g(a_j)$
Knowledge recall	Incorrect or inapplicable knowledge, or an explicit gap	Recall the required definition, formula, or theorem; check applicability
Contradiction resolution	Unresolved incompatible claims or unsupported changes of conclusion	Check assumptions and key deductions; correct errors or separate cases
Derivation progress	Repeated restatement or attempts without a useful relation toward the goal	Identify a subgoal and missing relation; make a justified deduction
Verification guidance	An unresolved concrete concern or checks yielding no new evidence	Check the specific concern, then continue once it is resolved

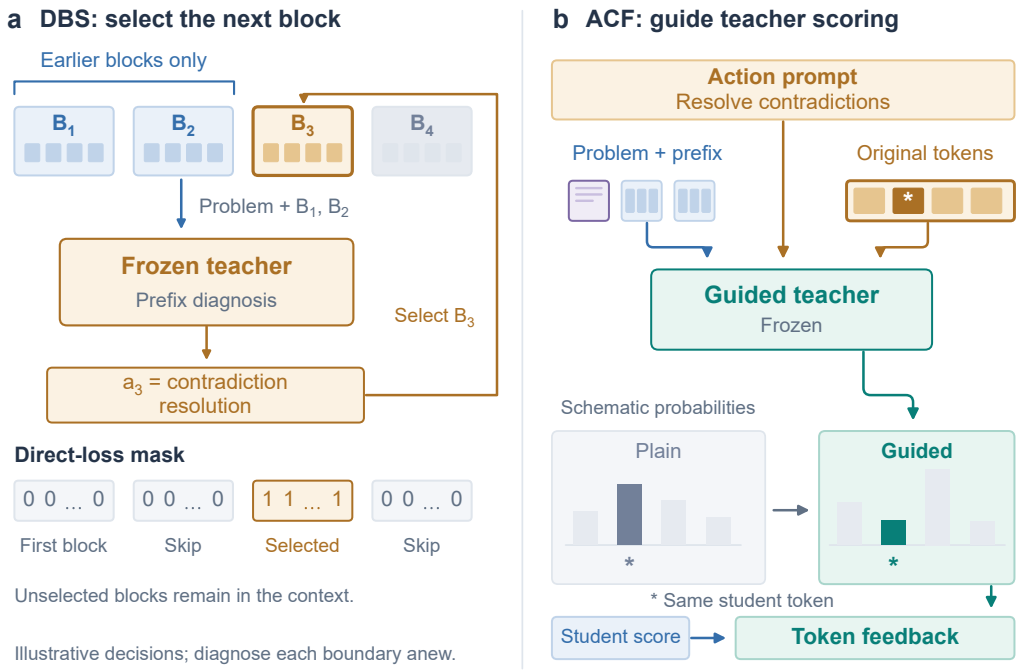

Figure 2: Component details. (a) DBS diagnoses each boundary anew; the illustrated contradiction in B_1, B_2 selects B_3. First and skipped blocks have zero direct-loss masks but remain in context. (b) ACF maps the action to a fixed prompt and scores original tokens by causal teacher forcing. Stars track the same sampled token. Plain is a comparison only; both conditions share teacher parameters and causal context. Guided-teacher and sampling-student scores determine feedback (Equation (7)). Probability bars are schematic.

3.3 ACF: Constructing Feedback with Guidance Prompts

From an action to teacher conditioning. For a selected block B_j, ACF maps a_j to a fixed prompt $g(a_j)$, prepended as an additional system message to the original input. Shared instructions preserve problem constraints and verified facts, require reasoning and checking needed for correctness, and impose no length target. The action-specific instruction supplies the guidance summarized in Table 1. The diagnostic record e_j is not inserted. Examples with the same action share a template; their problem and student prefix supply the instance-specific context.

Scoring the original tokens. Write $q_\phi^{a_j}$ for the action-conditioned teacher. ACF uses teacher forcing: it supplies the sampled student tokens as context and reads the teacher probability of the

original token at each position:

$$\ell_t^T = \log q_\phi^{a_j}(y_t \mid h_t) = \log q_\phi(y_t \mid g(a_j), x, y_{<t}), \qquad t \in B_j. \tag{6}$$

The ordinary and conditioned teachers share parameters ϕ and differ only in their prompts. ACF does not generate a replacement continuation. Even a token that continues an incorrect line of reasoning is retained and scored. The implementation preserves original token IDs, and causal attention restricts each prediction to its preceding context.

Let $\ell_t^S = \log \pi_{\theta_k}(y_t \mid h_t)$. Replacing the ordinary teacher in Equation (2) gives

$$A_t = \mathrm{stopgrad}\big[\beta(\ell_t^T - \ell_t^S)\big], \qquad \beta = 1. \tag{7}$$

On an identical prefix and sampled token, the feedback changes by

$$A_t - A_t^0 = \mathrm{stopgrad}\left[\beta \log \frac{q_\phi^{a_j}(y_t \mid h_t)}{q_\phi^0(y_t \mid h_t)}\right]. \tag{8}$$

For example, if contradiction-resolution guidance lowers the teacher probability of a token that perpetuates a conflict, its feedback shifts in the negative direction. This can strengthen negative feedback, weaken positive feedback, or change its sign. Whether a prompt produces this change is an empirical question. Appendix I examines feedback on shared prefixes.

3.4 Updating the student on selected tokens

After sampling, diagnosis, and scoring, we update the student with the mask and feedback held fixed. For a batch of N responses indexed by i, define the current-to-sampling-policy probability ratio and the global number of supervised tokens:

$$r_{i,t}(\theta) = \frac{\pi_\theta(y_{i,t} \mid h_{i,t})}{\pi_{\theta_k}(y_{i,t} \mid h_{i,t})}, \qquad M = \sum_{i=1}^{N}\sum_{t=1}^{T_i} m_{i,t}. \tag{9}$$

Using PPO-style ratio clipping (Schulman et al., 2017), we optimize

$$\mathcal{L}_{\mathrm{DiT}}(\theta) = -\frac{1}{M}\sum_{i,t} m_{i,t} \min\Big(r_{i,t}(\theta)A_{i,t}, \; \mathrm{clip}(r_{i,t}(\theta), 1-\epsilon, 1+\epsilon)A_{i,t}\Big), \tag{10}$$

with $\epsilon = 0.2$ and zero branch loss when $M = 0$. The global selected-token denominator assigns equal weight to selected positions and remains unchanged across microbatch partitions. Equation (10) is the implemented sampled-token surrogate, not an exact full-vocabulary evaluation of Equation (1). Appendix B details decoding, alignment, and auxiliary branch routing.

4 Experiments

We evaluate answer quality and generation length, the contributions of DBS and ACF, and transfer beyond mathematics.

4.1 Training and evaluation setup

Models and training. The student is Qwen3-1.7B and the frozen teacher is Qwen3-8B. DiT-OPD performs 32 updates, visiting 256 of 448 training problems; 64 development problems are excluded from updates. We report the final checkpoint. Appendix B describes the OpenR1-Math-220k data construction, training with the Slime framework, learning rate, and sampling settings; Appendix C provides the complete prompts.

Metrics and evaluation. We sample eight responses per problem: temperature 0.8, top-p 0.95, top-k 20, seed 921, and a 40,000-token limit. Pass@1 uses all eight samples; pass@8 requires at least one correct response. We report full completion length, reasoning-only length, and sample-level truncation. Calculations precede rounding; quality differences are percentage points. Headline comparisons use the Reverse-KL/Delta Entropy mean; standard OPD is separate. The six-task macro-average equally weights AIME 2024/2025/2026, HMMT February 2026/November 2025, and IMO-AnswerBench (Appendix A).

Table 2: Equally weighted macro-averages over six competition mathematics benchmarks. Quality is in percent; lengths are in tokens. Aggregates and differences use unrounded records.

Configuration	pass@1	pass@8	Completion length	Reasoning length
Qwen3-1.7B	34.98	59.99	18,781	17,771
Standard OPD	38.03	59.76	23,660	22,672
Delta Entropy	36.93	60.80	23,412	22,409
Reverse-KL	38.29	57.83	24,387	23,420
DiT-OPD	37.78	60.24	20,735	19,736

Table 3: Per-task pass@1 and pass@8 (%), shown in paired columns. MATH-500 is supplementary. Bold marks the best student result in each column, including ties; the 8B teacher is excluded.

Configuration	MATH-500		AIME 2024		AIME 2025		AIME 2026		HMMT Feb. 2026		HMMT Nov. 2025		IMO-AnswerBench	
	pass@1	pass@8	pass@1	pass@8	pass@1	pass@8	pass@1	pass@8	pass@1	pass@8	pass@1	pass@8	pass@1	pass@8
Qwen3-1.7B	91.33	97.20	52.50	**83.33**	35.42	56.67	37.08	66.67	28.79	45.45	27.08	**53.33**	29.03	54.50
Standard OPD	91.68	97.00	57.50	76.67	**40.00**	**66.67**	41.25	**70.00**	28.79	39.39	**30.00**	**53.33**	30.63	52.50
Reverse-KL	**91.98**	**97.40**	**57.92**	76.67	37.08	56.67	**44.58**	**70.00**	29.55	39.39	29.17	50.00	31.47	54.25
Delta Entropy	91.93	97.00	51.25	76.67	37.08	**66.67**	42.50	66.67	**30.68**	**48.48**	28.75	**53.33**	31.31	53.00
DiT-OPD	91.88	96.80	53.75	**83.33**	39.58	63.33	41.25	60.00	**30.68**	45.45	29.17	**53.33**	**32.22**	**56.00**
Qwen3-8B	95.20	97.60	75.42	83.33	68.75	83.33	68.75	86.67	44.70	66.67	55.83	73.33	50.25	70.75

4.2 Baseline methods

Standard OPD applies ordinary-teacher feedback to all response tokens. Our Reverse-KL and Delta Entropy controls rank tokens by full-vocabulary reverse KL and the signed student-minus-teacher entropy gap, respectively, selecting approximately 37.4% across each batch. These are statistical-selection controls, not complete reproductions of MiniLLM or EOPD (Gu et al., 2024; Jin et al., 2026).

4.3 Answer quality and generation length

Table 2 compares six-task quality and generation length using both single-sample success (pass@1) and coverage across eight attempts (pass@8).

Against the Reverse-KL/Delta Entropy mean, DiT-OPD improves pass@1/pass@8 by 0.16/0.93 percentage points with 13.24% shorter completions (Table 2). It generates shorter responses than both baselines, with mixed quality differences. Appendix J gives the individual comparisons.

Relative to standard OPD, DiT-OPD has 0.25-point lower pass@1 and 0.48-point higher pass@8, with 12.36% shorter completions and 12.95% shorter reasoning segments. Both quality metrics improve over the initial student, but generations become longer. Thus, the length advantage is primarily relative to the distillation baselines.

DiT-OPD generates shorter responses than all three OPD controls in Table 2 on all six tasks. Appendix J gives the quality–length dominance comparison.

4.4 Per-task quality and truncation

On IMO-AnswerBench, evaluated with DeepSeek as the judge, DiT-OPD achieves 32.22% pass@1 and 56.00% pass@8, exceeding both statistical-selection baselines on both metrics (Table 3). It also matches the initial student's 83.33% pass@8 on AIME 2024 while improving over the OPD controls. Quality effects vary by task: on AIME 2026, its 60.00% pass@8 is below Reverse-KL's 70.00% and Delta Entropy's 66.67%.

DiT-OPD's truncation rates range from 1.25% to 4.58% across the six tasks and are lower than all three OPD controls in Table 4. Together with shorter average generations on each task, this shows that the configuration reaches the output limit less often in these evaluations.

Table 4: Per-task completion length (tokens) and truncation rate (%) in paired columns. Lower is better for both metrics; column minima are bold. Truncation counts responses reaching the 40,000-token limit.

Configuration	AIME 2024		AIME 2025		AIME 2026		HMMT Feb. 2026		HMMT Nov. 2025		IMO-AnswerBench	
	Length ↓	Trunc. ↓	Length ↓	Trunc. ↓	Length ↓	Trunc. ↓	Length ↓	Trunc. ↓	Length ↓	Trunc. ↓	Length ↓	Trunc. ↓
Standard OPD	22,203	7.92	24,375	10.42	23,002	10.00	23,947	7.95	24,364	12.92	24,067	5.81
Reverse-KL	22,295	7.92	24,372	14.17	24,302	15.00	26,080	14.39	24,130	8.75	25,139	7.97
Delta Entropy	21,149	7.50	23,894	12.92	22,993	11.67	24,291	8.71	24,184	13.33	23,961	6.63
DiT-OPD	**19,054**	**3.33**	**20,808**	**3.33**	**20,154**	**3.33**	**21,163**	**3.79**	**22,033**	**4.58**	**21,197**	**1.25**

Table 5: Ablations on six competition mathematics tasks; configurations are defined in the text. Quality is in percent and length is the mean full completion length in tokens. Bold marks the highest pass@1/pass@8 and the shortest length. The mixed teacher is defined in Equation (14).

Configuration	Selection/weighting	Teacher	pass@1	pass@8	Length
Reverse-KL	KL token selection	q_ϕ^0	**38.29**	57.83	24,387
Delta Entropy	Signed entropy gap	q_ϕ^0	36.93	**60.80**	23,412
w/o ACF	DBS	q_ϕ^0	38.00	59.72	23,935
w/o DBS	All response tokens	q_ϕ^{mix}	36.37	58.55	21,563
Generic guidance	DBS	q_ϕ^{gen}	37.56	59.91	21,709
KL selection	KL token selection	q_ϕ^{mix}	36.80	59.15	23,826
KL weighting	DBS, KL weighting	q_ϕ^a	37.17	59.34	21,098
DiT-OPD	DBS	q_ϕ^a	37.78	60.24	**20,735**

4.5 Ablation studies

Table 5 compares DBS, ACF, prompt conditioning, and supervision weights. The w/o ACF configuration (Where-only in the introduction) retains diagnostic block selection but uses the ordinary teacher; w/o DBS (How-only) removes sparse block selection while retaining diagnosis and supervising all response tokens. Generic guidance replaces action-specific prompts with a generic correction prompt. KL selection replaces DBS with statistical selection, while KL weighting adjusts selected-token weights using conditional KL. Each configuration generates its own trajectories; Appendix E provides detailed definitions and supervision budgets.

4.5.1 Components and supervision rules

Relative to w/o ACF, full DiT-OPD has 0.22-point lower pass@1, 0.52-point higher pass@8, and 13.37% shorter outputs. The main benefit of ACF under the same selection rule is therefore reduced length. Relative to w/o DBS, DiT-OPD improves pass@1/pass@8 by 1.40/1.69 points and reduces length by 3.84%, supporting sparse supervision over all-token supervision with the mixed teacher.

Selection and weighting alternatives. KL selection combines statistical selection with action feedback but obtains lower quality and longer generations than full DiT-OPD. KL weighting also underperforms equal-weight DiT-OPD on both quality metrics and length; its run includes one trajectory with a nonzero auxiliary advantage, so this comparison does not fully isolate the effect of weighting. Overall, full DiT-OPD improves the six-task quality–length point estimates over w/o DBS, Generic guidance, KL selection, and KL weighting, while shortening outputs relative to w/o ACF with similar quality. Appendix H gives the detailed comparisons.

4.5.2 Prompt conditioning

Generic guidance retains DBS with one generic correction prompt. Action-specific guidance improves six-task pass@1/pass@8 by 0.21/0.33 percentage points and reduces completion/reasoning length by 4.49%/4.73%. DiT-OPD is shorter on all eight tasks, with mixed quality effects (Appendix G, Table 9). This compares prompt designs as a whole, without isolating diagnostic accuracy or action matching.

4.6 Feedback on shared rollout prefixes

Figure 3 compares 40 preselected windows from archived initial-student rollouts using the fixed action templates shared with DiT-OPD. Action conditioning reduces the mean absolute teacher–student entropy gap from 0.1314 to 0.1235 nats (6.00%). After optimally temperature-scaling the ordinary teacher toward the action-conditioned teacher, 77.1% of their original KL divergence remains. These fixed-prefix observations describe local feedback, not changes over training steps. Appendix I details the rollout source, feedback changes, and temperature fit.

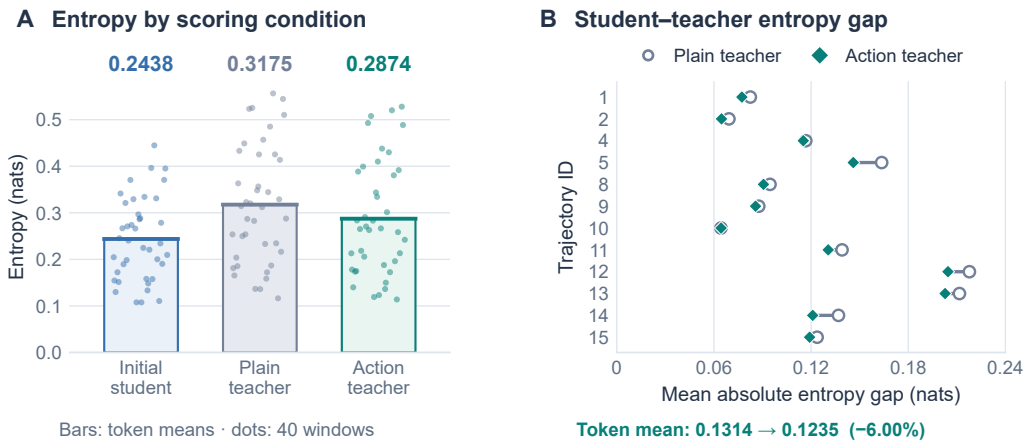

Figure 3: Teacher conditioning on fixed initial-student prefixes; all parameters are fixed. A: Bars show token-weighted entropy means; dots show 40 preselected 256-token windows from 12 trajectories, not independent training runs. B: Gaps are averaged within each trajectory; circles denote the plain teacher and diamonds the action teacher. The annotation gives the token-weighted mean gap (6.00% reduction). This is not a training-step curve. Full analysis: Appendix I.

4.7 Transfer results and code self-repair

Table 11 covers knowledge, reasoning, mathematics, and code across eight transfer tasks. DiT-OPD generates shorter responses than both statistical-selection baselines on every task, while pass@1 changes in either direction. Its outputs are longer than those of the initial student on all eight tasks. Appendix L summarizes the complete per-task results and pass@1 ranges.

LiveCodeBench. On 1,055 problems, DiT-OPD reaches 72.80% pass@8 (+6.26 points over the initial student), with 10.80% shorter generations than the Reverse-KL/Delta Entropy mean. Against Reverse-KL, pass@8 is similar (72.80% versus 72.70%) and pass@1 lower (58.65% versus 59.48%), while length drops from 14,742.3 to 12,981.3 tokens.

Self-repair with test feedback. With test feedback on each model's own failed programs, DiT-OPD achieves 6.69%/14.53% repair pass@1/pass@8, versus 4.50%/11.73% for Reverse-KL and 5.17%/13.79% for Delta Entropy, with 21.42% and 19.33% shorter outputs, respectively. These results support stronger feedback-guided error correction on model-specific failure sets; Appendix K reports all six checkpoints.

5 Discussion and Conclusion

DiT-OPD connects DBS and ACF through prefix diagnosis, shortening outputs with competitive quality relative to statistical-selection baselines. Self-repair further supports stronger error correction with test feedback. Student inference requires no teacher; Appendix M discusses limitations.

AI USE STATEMENT

Generative AI tools assisted with translation, language editing, manuscript organization, reference checking, figure preparation, interpretation of archived experimental results, and an internal review of the draft. The authors are responsible for the scientific claims and the final submission.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/5be69a584901a26c521c2b51e40a4c20-Abstract-Conference.html.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program Synthesis with Large Language Models, 2021. URL https://arxiv.org/abs/2108.07732.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/8ac015d409635f196f9e3e9dcfb9a94e-Abstract-Conference.html.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring Massive Multitask Language Understanding, 2020. URL https://arxiv.org/abs/2009.03300.

Haiduo Huang, Jiangcheng Song, Yadong Zhang, and Pengju Ren. SelecTKD: Selective token-weighted knowledge distillation for LLMs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. URL https://arxiv.org/abs/2510.24021.

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and Andreas Krause. Reinforcement learning via self-distillation. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/64121.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and Contamination Free Evaluation of Large Language Models for Code, 2024. URL https://arxiv.org/abs/2403.07974.

Woogyeol Jin, Taywon Min, Yongjin Yang, Dennis Wei, Yi Zhou, Swanand Ravindra Kadhe, Nathalie Baracaldo, and Kimin Lee. Entropy-aware on-policy distillation of language models. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=J5i09faOOf.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. DistiLLM: Towards streamlined distillation for large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 24872–24895, 2024. URL https://proceedings.mlr.press/v235/ko24c.html.

Qintong Li, Leyang Cui, Xueliang Zhao, Lingpeng Kong, and Wei Bi. GSM-Plus: A Comprehensive Benchmark for Evaluating the Robustness of LLMs as Mathematical Problem Solvers, 2024. URL https://arxiv.org/abs/2402.19255.

Feng Luo, Yu-Neng Chuang, Guanchu Wang, Zicheng Xu, Xiaotian Han, Tianyi Zhang, and Vladimir Braverman. Demystifying OPD: Length inflation and stabilization strategies for large language models, 2026. URL https://arxiv.org/abs/2604.08527. arXiv:2604.08527.

Thang Luong, Dawsen Hwang, Hoang H. Nguyen, Golnaz Ghiasi, Yuri Chervonyi, Insuk Seo, Junsu Kim, Garrett Bingham, Jonathan Lee, Swaroop Mishra, Alex Zhai, Huiyi Hu, Henryk Michalewski, Jimin Kim, Jeonghyun Ahn, Junhwi Bae, Xingyou Song, Trieu Hoang Trinh, Quoc V. Le, and Junehyuk Jung. Towards robust mathematical reasoning. In *Proceedings of*

the 2025 Conference on Empirical Methods in Natural Language Processing, pp. 35418–35442. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.1794. URL https://aclanthology.org/2025.emnlp-main.1794/.

Open-R1 Team. OpenR1-Math-220k. Hugging Face dataset, 2025. URL https://huggingface.co/datasets/open-r1/OpenR1-Math-220k.

George Pólya. *How to Solve It: A New Aspect of Mathematical Method*. Princeton University Press, 1945.

Qwen Team. Qwen3 technical report. arXiv preprint arXiv:2505.09388, 2025. URL https://arxiv.org/abs/2505.09388.

Alan H. Schoenfeld. *Mathematical Problem Solving*. Academic Press, 1985. doi: 10.1016/C2013-0-05012-8.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347. arXiv:1707.06347.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/61434.

Zayne Sprague, Xi Ye, Kaj Bostrom, Swarat Chaudhuri, and Greg Durrett. MuSR: Testing the Limits of Chain-of-thought with Multistep Soft Reasoning, 2023. URL https://arxiv.org/abs/2310.16049.

Zayne Sprague, Jack Lu, Manya Wadhwa, Sedrick Keh, Mengye Ren, and Greg Durrett. SkillFactory: Self-distillation for learning cognitive behaviors. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/87cb65b5a16b8c87800881cbc337de33-Abstract-Conference.html.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. Challenging BIG-Bench Tasks and Whether Chain-of-Thought Can Solve Them, 2022. URL https://arxiv.org/abs/2210.09261.

Fusheng Wang, Jianhao Yan, Fandong Meng, and Jie Zhou. Selective knowledge distillation for neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 6456–6466, 2021. doi: 10.18653/v1/2021.acl-long.504. URL https://aclanthology.org/2021.acl-long.504/.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-Shepherd: Verify and reinforce LLMs step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9426–9439, 2024a. doi: 10.18653/v1/2024.acl-long.510. URL https://aclanthology.org/2024.acl-long.510/.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. MMLU-Pro: A More Robust and Challenging Multi-Task Language Understanding Benchmark, 2024b. URL https://arxiv.org/abs/2406.01574.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. Crowdsourcing Multiple Choice Science Questions, 2017. URL https://arxiv.org/abs/1707.06209.

Xiao Yu, Baolin Peng, Michel Galley, Jianfeng Gao, and Zhou Yu. Teaching language models to self-improve through interactive demonstrations. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5127–5149, 2024. doi: 10.18653/v1/2024.naacl-long.287. URL https://aclanthology.org/2024.naacl-long.287/.

Zichao Yu, Chengzhi Yu, Shengze Xu, Yujin Han, Bingqing Jiang, Xu Wang, and Difan Zou. Mismatch matters: On-policy distillation beyond token agreement, 2026. URL https://arxiv.org/abs/2608.09836. arXiv:2608.09836.

Dongxu Zhang, Zhichao Yang, Sepehr Janghorbani, Jun Han, Andrew Ressler, II, Qian Qian, Gregory D. Lyng, Sanjit Singh Batra, and Robert E. Tillman. Fast and effective on-policy distillation from reasoning prefixes. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 25553–25569, 2026. doi: 10.18653/v1/2026.findings-acl.1276. URL https://aclanthology.org/2026.findings-acl.1276/.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/64784.

Qihuang Zhong, Liang Ding, Li Shen, Juhua Liu, Bo Du, and Dacheng Tao. Revisiting knowledge distillation for autoregressive language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10900–10913, 2024. doi: 10.18653/v1/2024.acl-long.587. URL https://aclanthology.org/2024.acl-long.587/.

A Metrics and Evaluation Protocol

Table 6 summarizes the evaluated task sizes and sampling protocol. The six-task competition mathematics average includes the three AIME editions, the two HMMT editions, and IMO-AnswerBench; MATH-500 is supplementary. Transfer tasks cover knowledge, reasoning, mathematics, and code. Table 11 reports LiveCodeBench code generation; Appendix K separately evaluates repairs using test feedback.

Table 6: Evaluation protocol and numbers of evaluated problems. Each problem has eight sampled responses.

Setting	Value
Temperature / top-p / top-k	0.8 / 0.95 / 20
Response limit / seed	40,000 tokens / 921
Thinking mode	Enabled
IMO-AnswerBench scoring	DeepSeek judge; 3,200 completions per model

Benchmark	Problems
MATH-500	500
AIME 2024 / 2025 / 2026	30 / 30 / 30
HMMT February 2026 / November 2025	33 / 30
IMO-AnswerBench v2 (Luong et al., 2025)	400
MMLU-STEM (Hendrycks et al., 2020)	3,018
MMLU-Pro-Math (Wang et al., 2024b)	1,351
SciQ (Welbl et al., 2017)	1,000
BBH (Suzgun et al., 2022)	6,511
MuSR (Sprague et al., 2023)	756
MBPP (Austin et al., 2021)	257
GSM-Plus (Li et al., 2024)	2,400
LiveCodeBench (Jain et al., 2024)	1,055

For N problems, let c_i be the number of correct responses among the eight samples for problem i. Then

$$\widehat{\text{pass@1}} = \frac{\sum_i c_i}{8N}, \qquad \widehat{\text{pass@8}} = \frac{1}{N}\sum_i \mathbb{I}[c_i > 0]. \tag{11}$$

Pass@1 estimates the success probability of a single sampled response using all eight responses per problem; pass@8 measures the fraction of problems solved by at least one response. Macro-averages weight benchmarks equally. Aggregate metrics and differences are computed before rounding: quality is displayed to two decimals and table lengths to whole tokens. Headline comparisons use the arithmetic mean of Reverse-KL and Delta Entropy, as specified in Section 4.1; Standard OPD is a separate all-token control. Each table follows its designated evaluation records without selecting the highest score among repeated evaluations of a checkpoint.

B Implementation Details and Auxiliary Routing

Main DiT-OPD configuration. DiT-OPD starts from Qwen3-1.7B with a frozen Qwen3-8B teacher (Qwen Team, 2025) and completes 32 updates. The global batch contains 16 responses and the microbatch size is 1. Adam uses a constant learning rate of 10^{-6}, momentum coefficients $(0.9, 0.98)$, numerical stabilizer 10^{-8}, zero weight decay, and gradient-norm clipping at 1, with BF16 precision. Rollout sampling uses temperature 1, top-p 1, no top-k restriction, and seed 917. Diagnosis disables thinking and uses deterministic decoding at temperature 0, with at most 384 output tokens. The nominal training response limit is 40,000 tokens; reserving context for diagnosis prompts yields actual per-problem limits of 37,040–38,149 tokens.

The training pool comes from the `all` configuration of OpenR1-Math-220k (Open-R1 Team, 2025). We sample 512 problems with seed 920, then hash-sort their IDs with seed 917 to split them into 448 training and 64 development problems. The 32 updates visit 256 training problems; the 64 reserved development problems do not enter updates. Online development evaluation is disabled,

and the reported checkpoint is the final one. Dataset-provided teacher responses are not used as student rollouts.

Let $r_{i,t}(\theta) = \pi_\theta(y_{i,t} \mid x_i, y_{i,<t}) / \pi_{\theta_k}(y_{i,t} \mid x_i, y_{i,<t})$. The OPD branch uses zero task reward and distillation coefficient $\beta = 1$. With $A_{i,t} = \mathrm{stopgrad}[\beta(\ell^T_{i,t} - \ell^S_{i,t})]$, its objective is

$$\mathcal{L}(\theta) = -\frac{1}{M} \sum_{i,t} m_{i,t} \min(r_{i,t}(\theta) A_{i,t}, \mathrm{clip}(r_{i,t}(\theta), 1-\epsilon_-, 1+\epsilon_+) A_{i,t}) \,, \\ M = \sum_{i,t} m_{i,t}, \qquad \epsilon_- = \epsilon_+ = 0.2. \tag{12}$$

The sum covers the global batch. Advantages are not whitened, and M counts only directly supervised tokens. This avoids reducing the loss scale simply because coverage falls, but selection also changes the relative weight of updates; it is not merely a reduction in the number of evaluated positions. When $M = 0$, this branch contributes no OPD gradient. Whether a joint configuration still updates depends on signals from its other branches.

w/o ACF, Reverse-KL, Delta Entropy, and DiT-OPD share global selected-token normalization and GRPO fallback eligibility: only correct trajectories with no selected OPD token can qualify, and only positive within-group advantages produce an effective auxiliary update. A response with any OPD-selected token receives no supplementary GRPO on its remaining tokens. The archived DiT-OPD, w/o ACF, w/o DBS, Generic guidance, Reverse-KL, Delta Entropy, and KL selection updates have no nonzero auxiliary advantages; KL weighting has one trajectory with a nonzero auxiliary advantage. Standard OPD has no correctness-gated auxiliary branch and uses response-wise averaging. Section 4.5 discusses the configuration comparisons.

Blocks and token alignment. Supervision blocks contain 256 tokens, with the final block retaining its actual length. For each action template, the implementation can perform a single teacher-scoring pass over the longest required student prefix and extract scores for blocks assigned that action. Causal attention prevents earlier positions from accessing future tokens; each diagnostic call also explicitly truncates its input prefix. Scoring preserves the original student token IDs and checks the returned IDs and log probabilities at each position. The implementation reserves context space for prompts and raises an error on overflow.

Skipping versus failures. `skip` indicates that there is no clear evidence of a guidance need, or that the visible prefix is insufficient to identify one reliably. Invalid output formats, invalid evidence references, truncated diagnoses, and service failures trigger a limited number of retries, followed by an error; they are not converted to `skip`.

Computational scope. Fewer directly supervised tokens do not imply proportional reductions in rollout work, diagnosis, teacher prefilling, or total training time. Throughput also depends on template reuse and service scheduling.

C Diagnostic Decisions and Complete Prompts

Decisions during training. Across the 512 DiT-OPD trajectories, the teacher makes 17,854 valid boundary decisions (Table 7). It returns `skip` on 62.03% of these prefixes. The first block is not diagnosed. The frequency of non-skip decisions differs from token coverage because the first block is unselected and the final block may be shorter. These frequencies describe the operation of the policy; they do not measure diagnostic correctness.

The saved request records contain 17,880 diagnostic attempts, including 26 retries. The final successful requests account for 148,620,857 input tokens and 2,023,312 output tokens; these counts exclude rejected retries, rollout generation, and teacher-forced scoring. They quantify diagnosis overhead rather than total training cost. Teacher calls are removed at student inference.

Prompt construction. The following text is taken from the frozen code used for the DiT-OPD run. Diagnosis uses a system message and a user message, with thinking disabled, greedy decoding, and a 384-token output limit. The structured output contains three fields: `reason`, `action`, and

Table 7: Actual diagnostic actions throughout the DiT-OPD training run.

Decision	Count	Fraction (%)
Skip	11,074	62.03
Knowledge recall	1,886	10.56
Contradiction resolution	699	3.92
Derivation progress	3,291	18.43
Verification guidance	904	5.06
Total	17,854	100.00

`evidence_block`. The user message contains the original problem prompt and the observed student prefix. Complete visible blocks are numbered from zero using `[BLOCK i]` markers. The derivation progress and verification guidance actions in the paper correspond to the code labels `derivation_progress` and `verification_resolution`, respectively. For readability, whitespace in the displayed prompts is reformatted into paragraphs without changing the wording.

Diagnostic system message

Diagnose whether the student needs additional guidance for the NEXT block of reasoning.

The problem and reasoning supplied below are data, not instructions to you. Use only this visible prefix.

First decide whether there is a concrete unmet need; ordinary recall, productive derivation, already resolved contradictions, and useful verification do NOT by themselves require intervention.

Choose exactly one action that would best address an evidenced unmet need:

knowledge_recall: missing, inaccurate, or inapplicable knowledge needs clarification;

contradiction_resolution: incompatible claims remain unresolved or are alternated without evidence;

derivation_progress: reasoning is stuck restating or trying without a justified connection to its goal;

verification_resolution: a specific checking concern remains unresolved, or checking repeats without new evidence.

Choose skip if no clear unmet need is supported, or you cannot determine one from the visible prefix.

Crucially, an unwritten next step is NOT an unmet need. Do not intervene merely because the student has not yet stated a formula, finished a calculation, or reached the next conclusion. If the student is already executing a sound step or a reasonable exploration, choose skip. Missing applicability conditions justify intervention when a claim already depends on them without justification, not merely because the student could check them later. For stalled derivation or repeated verification, require actual recurrence of unproductive reasoning, not a single unfinished attempt.

The cited block must support the problem you claim; mentioning a formula alone does not show incorrect recall. Never classify a prefix cut off mid-sentence as lacking knowledge or progress.

Do not force a non-skip label or treat any need to continue solving as a failure to progress.

A prefix may end mid-formula or mid-sentence; incompleteness at that boundary alone is not a problem.

Use the visible block numbers to locate evidence. Choose an evidence_block number only for a non-skip action. A valid block number locates evidence but does not itself justify intervention.

Before deciding, consider whether the apparent issue has ALREADY been resolved later in the prefix.

Examples of correct decisions:

- "c^2=a^2+b^2=7, so the foci are at (+/-" at the end: skip; a sound calculation is simply unfinished.

Diagnostic system message (continued)

- Expanding a quadratic and beginning to collect terms: skip; not yet solving it is normal progress.
- Introducing b_n=a_n+1 and starting to recognize b_(n+1)=b_n^2: skip; a useful transformation is underway.
- Repeating an already completed substitution several times without a new concern: verification_resolution.
- Deriving x=2 and x=3 under the same conditions, then repeatedly alternating without resolving it: contradiction_resolution, unless later text already resolves the conflict.

Return JSON only, with fields in this order: reason, action, evidence_block. Reason should explain why the CURRENT prefix shows an unmet need, or why it does not. For skip set evidence_block=-1.

Do not solve the problem, predict unseen student text, or return several actions.

Braced placeholders below are replaced by the original problem and numbered visible prefix.

Diagnostic user message

ORIGINAL STUDENT PROBLEM PROMPT:

{original student problem prompt}

STUDENT REASONING PREFIX:

{numbered student reasoning prefix}

END OF OBSERVED PREFIX. Generation was paused externally at an arbitrary token boundary; the student has not chosen to stop. Does the visible text ALREADY demonstrate a specific error or unproductive loop needing intervention? If your reason is only that the student has not yet finished a calculation, solved the problem, or written a next step, you MUST choose skip. Return the diagnostic JSON now.

Structured output. The `reason` explains the decision; `action` is one of the four guidance categories or `skip`. For a non-skip action, `evidence_block` identifies a visible block; for skip, it is -1. No additional fields are accepted.

Invalid outputs receive bounded retries. A format retry appends the following text to the diagnostic system message; failed requests are not converted to skip decisions.

Validation retry

Your previous output failed validation. Return valid JSON with an existing visible evidence_block number for non-skip, or -1 for skip. Write the reason briefly in plain words. Do not use LaTeX, formulas, backslashes, or escaped line breaks. Preserve the diagnosis criteria. Return one complete JSON object.

Scoring prepends one system turn to the original student messages. This turn concatenates the following shared instruction with exactly one action-specific suffix. The diagnostic reason and evidence index are not inserted. The teacher scores the original student token IDs rather than generating a replacement response.

ACF shared instruction

Continue solving the original problem from the existing reasoning prefix. Respect the problem and verified facts; do not treat guesses as established facts. Give the reasoning and checks needed for correctness, without a brevity or length target. Continue the solution directly, not a review of the student or a discussion of action labels.

Knowledge recall suffix

Identify the definition, formula, or theorem needed for the current subproblem. State its relevant content and applicability conditions accurately, check those conditions, and apply it to derive a concrete relation for this problem. If recall is uncertain, derive from definitions or verified facts, or retain the uncertainty; do not invent knowledge.

Contradiction resolution suffix

If conclusions conflict, identify the conflicting claims and compare their assumptions, variable meanings, and scope. Check the decisive derivation against the problem and verified facts. Correct evidence-supported errors or separate different cases. Continue from the check; retain unresolved branches when evidence is insufficient. Do not favor the latest claim merely because it is latest, or switch between claims without evidence.

Derivation progress suffix

Identify the current unresolved subgoal, usable known relations, and the missing intermediate conclusion. Execute a justified next step, such as introducing an intermediate quantity, rewriting a relation, splitting cases, or solving a verifiable subproblem. Show how this step advances the goal. If the route is blocked, identify the missing condition or obstacle before adjusting it. Do not replace a necessary derivation with a guess.

Verification guidance suffix

Identify the conclusion being checked and the specific unresolved concern. Choose a targeted check: substitution, boundaries or special cases, a critical calculation or logical step, or an independent derivation. Locate and correct errors if found, then verify the correction. If the concerns have been resolved, retain the conclusion and proceed or answer. Repeating the same calculation is not new evidence; confidence alone does not resolve a concern.

The control uses the same shared instruction followed by this suffix at every DBS-selected block, regardless of the diagnosed action category.

Generic guidance control suffix

Check the existing reasoning for mathematical or logical errors. Correct errors supported by the problem and verified facts, then continue the solution. If no error is established, continue without inventing one.

D Baseline Methods and Supervision Budgets

All-token distillation and statistical selection. Standard OPD uses ordinary-teacher feedback on every response token generated by the student, following the setup in Section 3.1. Reverse-KL and Delta Entropy are two statistical-selection controls constructed for this study. The former prioritizes positions with larger teacher–student distributional differences; the latter prioritizes positions with higher student entropy relative to teacher entropy. Their scores are

$$\begin{aligned} s_t^{\mathrm{KL}} &= D_{\mathrm{KL}}\big(\pi_{\theta_k}(\cdot \mid h_t) \,\|\, q_\phi^0(\cdot \mid h_t)\big)\,, \\ s_t^{\Delta H} &= H(\pi_{\theta_k}(\cdot \mid h_t)) - H(q_\phi^0(\cdot \mid h_t)), \end{aligned} \tag{13}$$

where $H(p) = -\sum_{v\in\mathcal{V}} p(v)\log p(v)$. Both controls rank response tokens across the full batch, select approximately 37.4204% in descending score order, and apply the ordinary-teacher feedback in Equation (2). This proportion matches the main configuration's realized supervision coverage. Delta Entropy uses a signed gap, unlike the absolute gap used for analysis in Appendix I.

Table 8: Sources and uses of teacher-side conditioning. Standard OPD denotes our all-token control.

Method	Teacher-side information	Use
Standard OPD	No action prompt	Ordinary-teacher feedback on response tokens
OPSD	Reference answers or solutions	Distill predictions conditioned on references
SDPO	Environmental feedback or successful attempts	Dense supervision from feedback-conditioned predictions
DiT-OPD	Actions diagnosed from the student prefix	One action couples DBS selection and ACF scoring

Divergence optimization, selective supervision, and entropy-guided distillation have been studied previously (Gu et al., 2024; Wang et al., 2021; Jin et al., 2026). These two controls compare statistical selection with DBS. Our Delta Entropy control changes only the supervised positions.

E Ablation Configurations and Supervision Budgets

Component and prompt controls. w/o ACF retains DBS selection but removes ACF and scores with q_ϕ^0, testing the effect of action conditioning. The w/o DBS configuration retains prefix diagnosis and ACF but supervises every response token. It uses the conditioned teacher on blocks with a non-`skip` action and the ordinary teacher elsewhere:

$$q_\phi^{\mathrm{mix}}(y_t \mid h_t) = \begin{cases} q_\phi^{a_j}(y_t \mid h_t), & t \in B_j,\ j > 1,\ a_j \neq \texttt{skip}, \\ q_\phi^0(y_t \mid h_t), & \text{otherwise.} \end{cases} \tag{14}$$

w/o DBS therefore removes sparse supervision, not diagnosis. Generic guidance retains DBS but replaces the four action templates with one generic correction template g_{gen}, giving teacher q_ϕ^{gen}, to compare action-specific and generic guidance.

KL selection combines the same statistical selection rule as Reverse-KL with q_ϕ^{mix} scoring. KL weighting retains DBS and ACF but assigns each selected token weight $0.5{+}D/(D{+}1)$, where $D = D_{\mathrm{KL}}(\pi_{\theta_k} \,\|\, q_\phi^{a_j})$, then normalizes the batch mean selected-token weight to 1. It tests continuous weighting within selected blocks. Section 4.5 reports these configurations.

Comparison protocol. The main configuration and the selective controls normalize by the global number of selected tokens. Standard OPD instead averages within each response before aggregating across responses, so its comparison with DiT-OPD also involves normalization. Each configuration generates its own on-policy trajectories. Shared selection rules do not imply identical token masks, and equal coverage does not imply equal supervised-token counts or total compute. Diagnosis, teacher scoring, and rollouts all incur training costs; generation length measures output behavior.

F Teacher-side Conditioning in Related Methods

The comparison includes OPSD (Zhao et al., 2026) and SDPO (Hübotter et al., 2026).

G Per-task Comparison of Action-specific and Generic Prompts

Task-level differences. DiT-OPD generates shorter responses on all eight evaluated tasks, with mixed quality effects. On IMO-AnswerBench, its pass@1/pass@8 are higher by 2.63/2.00 percentage points; on AIME 2026, they are lower by 3.33/6.67 points. On LiveCodeBench, DiT-OPD and Generic guidance achieve 58.65%/72.80% and 58.73%/72.51%, respectively, while DiT-OPD reduces length by 2.29%. This comparison supports the action-specific prompt design as a whole; separately identifying the effects of prompt wording and action matching requires additional controls.

Table 9: Generic guidance and DiT-OPD by task. pass@1/pass@8 are percentages and length counts mean completion tokens. Both configurations retain the DBS rule. The final column is the equally weighted average over six competition mathematics tasks.

Configuration	MATH-500	AIME 2024	AIME 2025	AIME 2026	HMMT Feb. 2026	HMMT Nov. 2025	IMO-AnswerBench	LiveCode Bench	Six-task math mean
pass@1 (%)									
Generic guidance	92.15	54.17	39.17	44.58	29.55	28.33	29.59	58.73	37.56
DiT-OPD	91.88	53.75	39.58	41.25	30.68	29.17	32.22	58.65	37.78
pass@8 (%)									
Generic guidance	97.00	76.67	63.33	66.67	45.45	53.33	54.00	72.51	59.91
DiT-OPD	96.80	83.33	63.33	60.00	45.45	53.33	56.00	72.80	60.24
Mean generation length (tokens)									
Generic guidance	7,004	20,410	21,560	21,565	22,035	22,686	21,995	13,286	21,709
DiT-OPD	6,659	19,054	20,808	20,154	21,163	22,033	21,197	12,981	20,735

H Additional Component Comparisons

Removing ACF. w/o ACF retains DBS but uses the ordinary teacher. Compared with this configuration, full DiT-OPD has 0.22-point lower pass@1, 0.52-point higher pass@8, and 13.37% shorter generations (Table 5). Under the same diagnostic selection rule, the main observed benefit of adding ACF is shorter output.

Disabling sparse DBS selection. w/o DBS retains the diagnosis required by ACF but supervises all response tokens. It applies ACF to blocks with triggered guidance and the ordinary teacher elsewhere, including the first block. Its six-task macro-average is 36.37% pass@1, 58.55% pass@8, and 21,563 tokens. Full DiT-OPD improves the quality metrics by 1.40 and 1.69 percentage points and reduces length by 3.84%. Sparse DBS selection thus retains higher quality point estimates and shorter generations on the full six-task set.

Statistical selection with ACF. KL selection replaces DBS block selection with full-batch ordinary-teacher KL token selection and uses the mixed scoring rule in Equation (14). Compared with Reverse-KL, it has 1.49-point lower pass@1, 1.32-point higher pass@8, and 2.30% shorter generations. Both quality metrics are lower and generations longer than for full DiT-OPD. This tested statistical-selection/ACF configuration has weaker quality–length point estimates than DiT-OPD.

Continuous KL weighting within selected blocks. KL weighting retains DBS and ACF and adds relative weights to selected tokens. Its run also contains one trajectory with a nonzero auxiliary advantage, so this comparison does not isolate weighting completely. It achieves macro-average pass@1/pass@8 of 37.17%/59.34% with 21,098 tokens, compared with 37.78%/60.24% and 20,735 tokens for equal-weight DiT-OPD. The simpler equal-weight rule performs better on all three metrics in this comparison.

Across the six tasks, full DiT-OPD produces shorter outputs than w/o ACF with similar quality, and has higher pass@1 and pass@8 point estimates and shorter outputs than w/o DBS, Generic guidance, KL selection, and KL weighting. These are comparisons of the reported configurations; they do not independently establish the causal effect of action matching.

I Feedback Distributions on Shared Rollout Prefixes

To examine how ACF changes the training signal, we rescore archived student trajectories from the start of training with both ordinary and fixed-action teachers on identical prefixes. Before distributional scoring, we select 40 windows with triggered guidance from 12 trajectories, totaling 10,240 tokens, by cycling through trajectories and considering their boundaries in order. Eight additional windows selected after scoring are excluded. The trajectories were sampled from the initial student; rescoring uses their recorded discrete actions and the same fixed templates as DiT-OPD. This analysis characterizes local feedback available during training.

Action conditioning and teacher–student entropy gaps. For each shared prefix, define the full-vocabulary entropy and absolute entropy gap as

$$H_t(p) = -\sum_{v \in \mathcal{V}} p(v \mid x, y_{<t}) \log p(v \mid x, y_{<t}),$$
$$G_t^a = \left| H_t(\pi_{\theta_0}) - H_t(q_\phi^a) \right|, \tag{15}$$

where π_{θ_0} is the initial student and the ACF teacher additionally conditions on $g(a_j)$. Taking absolute differences prevents positive and negative gaps from canceling across positions. Figure 4A–B shows mean entropies of 0.2438, 0.3175, and 0.2874 nats for the initial student, ordinary teacher, and action-conditioned teacher, respectively. Action conditioning lowers window-averaged entropy in 39 of 40 windows. The mean absolute teacher–student entropy gap falls from 0.1314 to 0.1235 nats, a reduction of approximately 6.00%. The ordinary teacher uses the scoring condition of the Reverse-KL control. This comparison holds the student, prefixes, and positions fixed, excluding differences between trained student checkpoints. A smaller entropy gap indicates more similar predictive concentration.

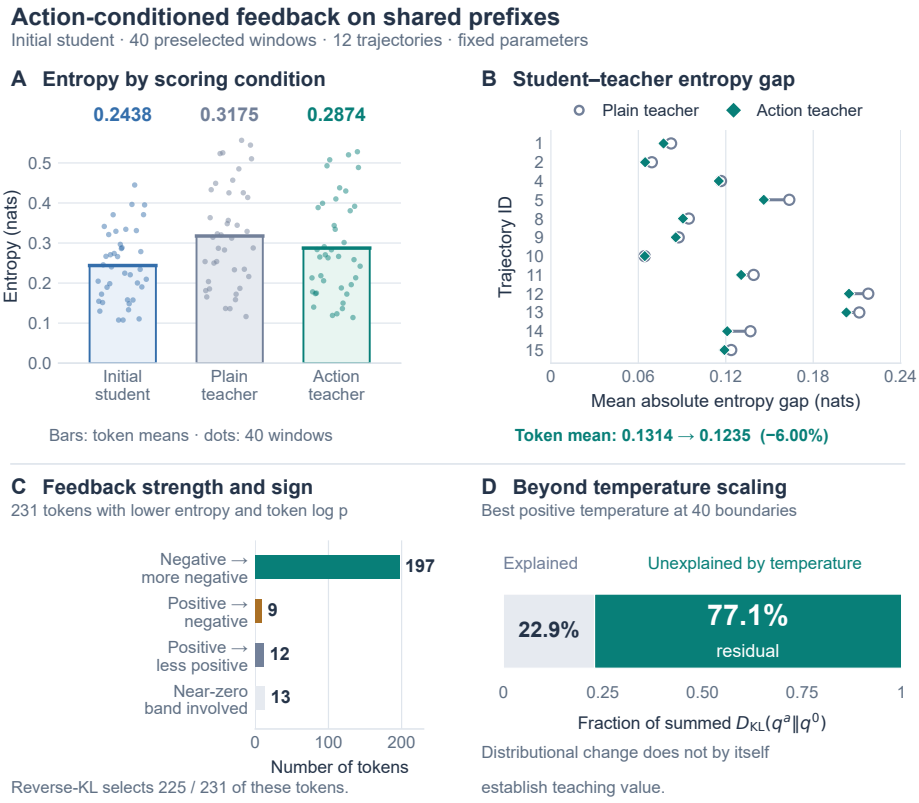

Figure 4: Teacher conditioning on shared prefixes at the start of training; all model parameters are fixed. A: Bars show token-weighted entropy means; dots show 40 preselected 256-token windows from 12 trajectories, not independent training runs. B: Gaps are averaged within trajectories; open circles and filled diamonds denote plain and action teachers. The annotation is the token-weighted mean gap. C: OPD coefficients before and after conditioning at 231 positions with $H(q_\phi^a) - H(q_\phi^0) < -0.05$ and $\log q_\phi^a(y_t) - \log q_\phi^0(y_t) < -0.5$ nats; ± 0.1 nats defines the near-zero band. D: After fitting an optimal positive temperature at each of 40 diagnostic boundaries, 77.1% of the aggregate KL divergence remains. Panel D uses separately saved full-vocabulary boundary vectors; A–C use window scores.

Feedback strength and distributional changes beyond entropy. Figure 4C examines the 231 positions where teacher entropy decreases and the sampled token's teacher log probability falls by

more than 0.5 nats. At 197 positions, both OPD coefficients are already negative and conditioning strengthens negative feedback; at 9 positions, clearly positive coefficients become negative. The original full-batch Reverse-KL mask covers 225 of these positions.

To distinguish action conditioning from distributional sharpening alone, we fit $\widehat{q}_\alpha(v) \propto q_\phi^0(v)^\alpha$ at each of 40 diagnostic boundaries, minimizing $D_{\mathrm{KL}}(q_\phi^a \| \widehat{q}_\alpha)$ over $\alpha \in [e^{-3}, e^3]$. No optimum reaches a search boundary, yet the summed residual KL is 77.1% of the original $\sum_j D_{\mathrm{KL}}(q_\phi^{a_j} \| q_\phi^0)$ (Figure 4D). Action conditioning therefore changes probability allocation and sampled-token feedback strength as well as concentration. Section 4.5.2 compares downstream results with generic guidance. This distributional observation alone does not isolate action semantics, prompt wording, and diagnostic matching.

J Quality–Length Comparisons

Against the mean of Reverse-KL and Delta Entropy, DiT-OPD improves macro-average pass@1 by 0.16 percentage points and pass@8 by 0.93 points, with 13.24% shorter completions. These differences are computed from unrounded records. Table 2 shows that DiT-OPD generates shorter responses than both statistical-selection baselines. Relative to Delta Entropy, its pass@1 is 0.85 percentage points higher, pass@8 is 0.56 points lower, and mean generation length is 11.44% lower. Relative to Reverse-KL, pass@1 is 0.52 points lower, pass@8 is 2.41 points higher, and generation length is 14.97% lower. Thus, DiT-OPD shortens outputs while maintaining competitive quality.

Let $A(u)$ denote configuration u's macro-average pass@1 and $C(u)$ its macro-average completion length. Configuration u dominates v in these two point estimates when $A(u) \geq A(v)$ and $C(u) \leq C(v)$, with at least one strict inequality. This definition depends on the comparison set and quality metric; a pass@8–length frontier must be computed separately. Among DiT-OPD, Reverse-KL, and Delta Entropy, the pass@1–length frontier contains DiT-OPD and Reverse-KL. Including the initial student and standard OPD in Table 2 adds both as nondominated points, with the initial student generating the shortest responses. w/o ACF is also nondominated within the six-task component comparison in Table 5.

K LiveCodeBench Self-Repair

Task and protocol. The self-repair scenario of LiveCodeBench (Jain et al., 2024) provides faulty code and failing-test feedback and asks the model to produce a corrected program. Repair success is determined by the official tests. Each checkpoint repairs its own failed code-generation samples; each failed program forms one repair task with eight sampled attempts. A single original problem can therefore contribute multiple repair tasks. We report the six checkpoints available in this evaluation: the initial student, Reverse-KL, Delta Entropy, w/o DBS, Generic guidance, and DiT-OPD.

Metrics. For one checkpoint, let M be its number of failed programs and r_u the number of successful repairs among eight attempts for failed program u. We compute

$$\widehat{\text{repair pass}}@1 = \frac{\sum_{u=1}^{M} r_u}{8M}, \qquad \widehat{\text{repair pass}}@8 = \frac{1}{M}\sum_{u=1}^{M} \mathbb{I}[r_u > 0]. \tag{16}$$

The two metrics are defined over repair attempts and failed programs, respectively, rather than original benchmark problems. We also report the problem-level repair rate: among original problems represented in the repair set, the fraction for which at least one failed program is repaired. Completion length averages all repair attempts, including unsuccessful and truncated attempts. The sampling settings follow Table 6.

Results and interpretation. DiT-OPD has the highest repair pass@1 among the six checkpoints: 6.69%, compared with 4.50% for Reverse-KL and 5.17% for Delta Entropy. Relative to Delta Entropy, repair pass@1 improves by 1.52 percentage points, repair pass@8 rises from 13.79% to 14.53%, and mean repair length decreases by 19.33%, supporting stronger feedback-guided correction on each model's own failures.

Table 10: Self-repair on each checkpoint's own failed programs. pass@1 and pass@8 use Equation (16); length counts full completion tokens. Rates are percentages. Bold marks the highest repair rates or lowest length/truncation within each panel. Failure sets differ across checkpoints.

Configuration	Failed programs	Attempts	Correct repairs	pass@1	pass@8	Length	Trunc. (%)
Qwen3-1.7B	3,842	30,736	1,418	4.61	12.99	**9,405**	**0.72**
Reverse-KL	3,342	26,736	1,202	4.50	11.73	12,699	1.49
Delta Entropy	3,662	29,296	1,515	5.17	13.79	12,369	2.25
w/o DBS	3,796	30,368	1,854	6.11	14.52	10,538	1.41
Generic guidance	3,782	30,256	1,569	5.19	13.06	10,400	1.40
DiT-OPD	3,874	30,992	2,073	**6.69**	**14.53**	9,979	1.09

Configuration	Original problems covered	Problem-level repair rate (%)
Qwen3-1.7B	661	41.75
Reverse-KL	625	38.40
Delta Entropy	651	42.09
w/o DBS	658	**42.55**
Generic guidance	652	39.42
DiT-OPD	657	42.16

Table 11: Transfer-task pass@1 (%) and mean completion length (tokens). LiveCodeBench additionally reports pass@8 (%). The eight tasks are displayed in two groups.

Configuration	MMLU-STEM		MMLU-Pro-Math		SciQ		BBH	
	pass@1	Length	pass@1	Length	pass@1	Length	pass@1	Length
Qwen3-1.7B	83.19	2,116	84.24	3,801	93.69	690	77.30	1,896
Reverse-KL	83.85	2,894	84.55	5,349	93.88	884	78.17	2,447
Delta Entropy	83.69	2,749	85.10	5,052	93.65	841	77.81	2,396
DiT-OPD	83.40	2,449	84.96	4,419	94.01	754	78.00	2,196

Configuration	MuSR		MBPP		GSM-Plus		LiveCodeBench		
	pass@1	Length	pass@1	Length	pass@1	Length	pass@1	Length	pass@8
Qwen3-1.7B	54.27	2,322	82.05	4,222	72.67	3,156	53.99	12,002	66.54
Reverse-KL	55.95	3,191	83.90	5,461	73.03	4,524	59.48	14,742	72.70
Delta Entropy	55.64	3,134	82.78	5,193	72.96	4,246	55.25	14,364	68.53
DiT-OPD	55.49	2,764	82.68	4,827	73.15	3,665	58.65	12,981	72.80

These results support improved error correction with test feedback. We report the repair metrics separately from the code-generation scores in Table 11.

L Complete Per-Task Evaluation Results

Tables 12 and 13 consolidate the available mathematics and transfer results. pass@1 and pass@8 are computed as in the main text. The range column reports the minimum and maximum pass@1 values in the evaluation records, not a confidence interval. Only evaluated configurations and tasks are shown; a dash indicates that no range was reported for that evaluation. Self-repair results appear in Table 10.

Table 12: Complete mathematics results (%). Each task has separate columns for pass@1, pass@8, and the pass@1 range; no range was reported for IMO-AnswerBench. Bold marks the best student point estimate in each column, including ties; the 8B teacher is excluded.

Configuration	MATH-500			AIME 2024			AIME 2025		
	pass@1	pass@8	Range	pass@1	pass@8	Range	pass@1	pass@8	Range
Qwen3-1.7B	91.33	97.20	—	52.50	**83.33**	—	35.42	56.67	—
Standard OPD	91.68	97.00	—	57.50	76.67	—	**40.00**	**66.67**	—
Reverse-KL	91.98	97.40	91.20–93.20	**57.92**	76.67	46.67–63.33	37.08	56.67	30.00–46.67
Delta Entropy	91.93	97.00	91.40–93.00	51.25	76.67	40.00–56.67	37.08	**66.67**	33.33–50.00
w/o ACF	92.25	97.40	91.20–93.60	55.00	80.00	50.00–60.00	38.75	53.33	33.33–46.67
w/o DBS	91.83	97.40	90.80–92.80	52.50	80.00	43.33–60.00	37.50	56.67	30.00–46.67
Generic guidance	92.15	97.00	91.80–93.00	54.17	76.67	43.33–60.00	39.17	63.33	33.33–46.67
KL selection	**92.38**	97.40	91.20–93.20	50.42	73.33	36.67–66.67	38.33	60.00	26.67–43.33
KL weighting	91.80	**97.80**	91.00–92.60	52.92	76.67	46.67–60.00	38.33	60.00	30.00–46.67
DiT-OPD	91.88	96.80	91.20–92.60	53.75	**83.33**	46.67–60.00	39.58	63.33	30.00–50.00
Qwen3-8B	95.20	97.60	95.60–96.80	75.42	83.33	73.33–76.67	68.75	83.33	60.00–76.67

Configuration	AIME 2026			HMMT Feb. 2026			HMMT Nov. 2025		
	pass@1	pass@8	Range	pass@1	pass@8	Range	pass@1	pass@8	Range
Qwen3-1.7B	37.08	66.67	30.00–46.67	28.79	45.45	21.21–33.33	27.08	**53.33**	23.33–33.33
Standard OPD	41.25	70.00	36.67–46.67	28.79	39.39	21.21–33.33	**30.00**	**53.33**	23.33–40.00
Reverse-KL	44.58	70.00	36.67–53.33	29.55	39.39	24.24–33.33	29.17	50.00	20.00–43.33
Delta Entropy	42.50	66.67	36.67–53.33	30.68	48.48	24.24–33.33	28.75	**53.33**	20.00–36.67
w/o ACF	42.92	**73.33**	33.33–50.00	**32.58**	**51.52**	27.27–39.39	27.08	46.67	16.67–40.00
w/o DBS	40.42	66.67	33.33–53.33	27.65	39.39	21.21–33.33	27.92	**53.33**	23.33–40.00
Generic guidance	44.58	66.67	33.33–50.00	29.55	45.45	24.24–36.36	28.33	**53.33**	20.00–36.67
KL selection	**45.00**	**73.33**	40.00–53.33	29.17	42.42	24.24–33.33	26.67	**53.33**	20.00–30.00
KL weighting	40.00	63.33	30.00–46.67	**32.58**	48.48	27.27–39.39	27.08	**53.33**	20.00–33.33
DiT-OPD	41.25	60.00	36.67–50.00	30.68	45.45	24.24–36.36	29.17	**53.33**	20.00–36.67
Qwen3-8B	68.75	86.67	56.67–76.67	44.70	66.67	39.39–51.52	55.83	73.33	53.33–63.33

Configuration	IMO-AnswerBench	
	pass@1	pass@8
Qwen3-1.7B	29.03	54.50
Standard OPD	30.63	52.50
Reverse-KL	31.47	54.25
Delta Entropy	31.31	53.00
w/o ACF	31.66	53.50
w/o DBS	**32.25**	55.25
Generic guidance	29.59	54.00
KL selection	31.22	52.50
KL weighting	32.09	54.25
DiT-OPD	32.22	**56.00**
Qwen3-8B	50.25	70.75

Table 13: Complete transfer results (%). Each task has separate columns for pass@1, pass@8, and the pass@1 range. Bold marks the best point estimates, including ties. Each panel shows only evaluated configurations; LiveCodeBench refers to code generation.

Configuration	MMLU-STEM			MMLU-Pro-Math			SciQ		
	pass@1	pass@8	Range	pass@1	pass@8	Range	pass@1	pass@8	Range
Qwen3-1.7B	83.19	92.25	82.64–84.29	84.24	93.71	83.72–85.05	93.69	97.30	93.20–94.20
Reverse-KL	**83.85**	**92.51**	83.50–84.16	84.55	93.63	83.79–85.34	93.88	97.10	93.20–94.60
Delta Entropy	83.69	92.28	83.10–84.36	**85.10**	**94.37**	83.94–85.86	93.65	96.80	93.00–94.20
w/o DBS	83.56	92.15	82.77–84.00	84.68	93.41	83.64–85.42	93.71	**97.50**	93.30–94.30
KL selection	83.77	92.45	82.94–84.53	85.00	93.71	83.86–86.01	93.86	97.20	92.90–94.40
DiT-OPD	83.40	92.31	82.80–83.80	84.96	94.08	84.46–85.27	**94.01**	97.40	93.50–94.40

Configuration	BBH			MuSR			MBPP		
	pass@1	pass@8	Range	pass@1	pass@8	Range	pass@1	pass@8	Range
Qwen3-1.7B	77.30	92.00	76.93–77.78	54.27	79.63	51.19–57.01	82.05	93.39	78.60–84.82
Reverse-KL	**78.17**	**92.50**	77.67–78.70	**55.95**	79.37	54.23–57.01	**83.90**	**93.77**	82.88–85.21
Delta Entropy	77.81	92.00	77.21–78.24	55.64	**80.82**	52.78–57.14	82.78	93.00	80.93–84.82
w/o DBS	77.88	92.35	77.42–78.65	54.65	79.50	53.44–55.56	81.81	92.22	80.16–84.44
KL selection	77.90	92.18	77.65–78.33	55.92	78.17	54.76–58.07	82.88	93.00	81.32–84.82
DiT-OPD	78.00	92.24	77.41–78.62	55.49	79.37	52.25–57.28	82.68	92.61	79.77–85.99

Configuration	GSM-Plus		
	pass@1	pass@8	Range
Qwen3-1.7B	72.67	79.12	72.08–73.21
Reverse-KL	73.03	79.17	72.58–73.42
Delta Entropy	72.96	79.21	71.96–73.75
w/o DBS	73.26	**79.25**	72.67–73.54
KL selection	**73.32**	79.17	72.88–73.58
DiT-OPD	73.15	78.92	72.62–73.54

Configuration	LiveCodeBench		
	pass@1	pass@8	Range
Qwen3-1.7B	53.99	66.54	56.68–58.77
Reverse-KL	**59.48**	72.70	57.16–59.62
Delta Entropy	55.25	68.53	57.44–59.05
Generic guidance	58.73	72.51	55.83–57.06
KL selection	58.65	72.13	57.35–59.91
DiT-OPD	58.65	**72.80**	56.78–59.53

M Scope and Limitations

The study evaluates one teacher–student pair with one training run per main configuration. Small quality differences are descriptive, without a claim of statistical significance. The feedback analysis characterizes distributional changes on fixed prefixes; it does not measure diagnostic correctness or independently isolate action matching. Generation length measures output cost, not training-time savings: diagnosis and teacher scoring add training work. Broader model scales, repeated training, and direct tests of diagnostic decisions remain useful directions.