PREFERENCE-BASED SELF-DISTILLATION WITH ONLINE CONTEXTUAL COMPARISONS

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy self-distillation (OPSD) uses privileged context to create a teacher from the student's base model, then transfers its guidance through token-level distribution matching. We study whether this contextual supervision can be used more effectively through sequence-level comparisons of generated responses. Preference-Based Self-Distillation (PBSD) retains the contextual self-teacher and adopts teacher-referenced preference distillation: teacher responses are paired with current student responses and scored relative to the frozen contextual teacher. The student receives only the original problem, and no separate preference judge is required. Across three Qwen3 scales, PBSD improves over OPSD in Math Avg. and tool-call accuracy, achieving the highest scores on these aggregate metrics among the evaluated methods. On Qwen3-4B, it reaches 64.5% Math Avg. versus 63.5% for OPSD. With the same contextual-teacher configuration, imitation, fixed-negative training, and an unconditioned reference achieve 60.7%, 60.8%, and 59.3%, respectively. Among the tested contexts, reference reasoning with the correct answer yields the highest scores. These results motivate studying teacher context together with the procedure used for distillation.

1 INTRODUCTION

Reference solutions, demonstrations, and task-specific context contain useful guidance for training language models. On-policy self-distillation makes this information available to a teacher instantiated from the same base model, while the student learns from its own generated responses (Zhao et al., 2026a; Shenfeld et al., 2026). This retains the use of student-generated trajectories in on-policy distillation (Agarwal et al., 2024; Lu & Thinking Machines Lab, 2025) without requiring a larger external teacher. It is particularly attractive when verified training examples are available but repeated reward-based rollouts are costly (Shao et al., 2024; Song & Zheng, 2026; Li et al., 2026a).

The teacher's advantage in OPSD comes from what it sees. A reference solution can help it produce a useful answer, but the student must respond without that information. OPSD transfers this guidance by matching token distributions on student trajectories. This raises a practical question: does distribution matching make effective use of the available contextual supervision? Kim et al. (Kim et al., 2026) report reasoning degradation associated with suppressed uncertainty expression in some self-distillation settings. Other analyses identify unreliable teacher guidance on student prefixes and distribution-matching failure modes (Fu et al., 2026; Li et al., 2026b). These observations motivate evaluating a different use of the same contextual-teacher construction: comparing teacher- and student-generated sequences. We evaluate its effects on task performance and output behavior.

We study this alternative through Preference-Based Self-Distillation (PBSD). PBSD retains OPSD's information asymmetry—the teacher receives privileged context and the student does not—while transferring guidance through response comparisons. It uses teacher-referenced preference learning to transfer this guidance. A teacher response supplies the positive example, an online student response supplies the negative, and the contextual teacher provides reference scores for both. The teacher stays fixed as the student and its responses evolve. Our focus is whether this use of contextual supervision improves on OPSD and which choices affect the resulting gains. Figure 1 illustrates the information flow; Section 3 specifies the update.

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

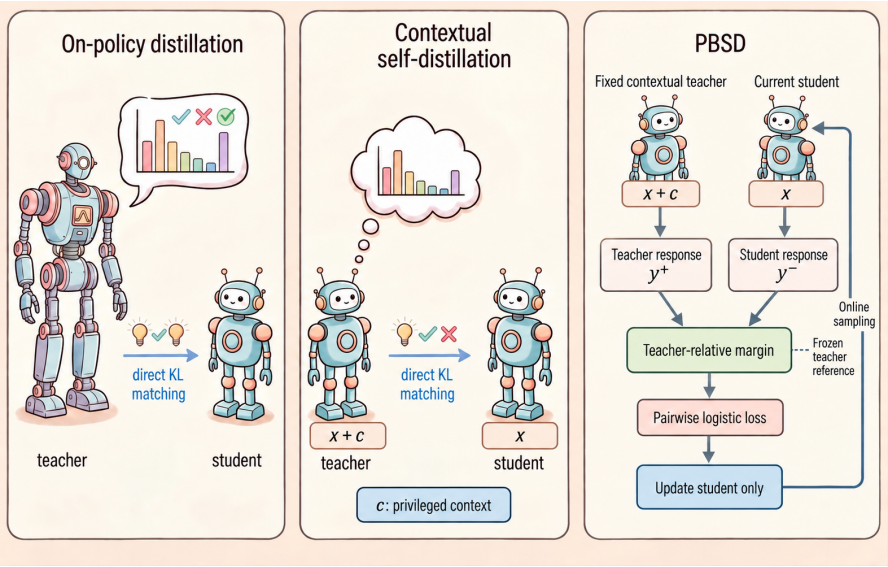


Figure 1: Selected distillation configurations. Left: distribution matching to an external teacher. Middle: distribution matching to a contextual self-teacher. Right: PBSD pairs a response from the fixed contextual teacher with an online student response, scores their teacher-relative likelihood margin, and updates only the student using a pairwise logistic loss. The teacher remains frozen and also supplies the reference probabilities. Labels y^+ and y^- indicate roles, not correctness.

Table 1: Training properties of the representative configurations discussed here. "Online" means that student responses are generated during training; "external teacher" means a separate pretrained teacher model. Outcome scoring denotes a task reward or verifier. Teacher-side context is privileged information unavailable to the student.

Method	Online student responses	External teacher	Outcome scoring	Pairwise objective	Teacher-side context
SFT	✗	✗	✗	✗	✗
GRPO	✓	✗	✓	✗	✗
OPD	✓	✓	✗	✗	✗
OPSD	✓	✗	✗	✗	✓
DPKD (Li et al., 2024)	✓	✓	✗	✓	✗
PBSD (Ours)	✓	✗	✗	✓	✓

Each PBSD comparison uses one teacher response and one on-policy student response, with no separate preference annotator. The labels reflect the teacher and student generation roles; responses are not individually verified. Table 1 summarizes the training properties.

Building on OPSD, our experiments examine both the effectiveness of response comparisons and what makes contextual guidance useful for self-distillation. PBSD improves Math Avg. and tool-use accuracy over OPSD at three model scales. On Qwen3-4B, variants using the same contextual-teacher configuration score lower with imitation, fixed negatives, or an unconditioned reference. Context ablations further show that providing the correct answer alone does not recover the scores obtained with reference reasoning. These results examine both teacher context and alternative uses of its supervision. Response and pair-correctness diagnostics characterize the resulting behavior and supervision.

The contributions of this work are:

- **Supervision use matters alongside teacher construction.** PBSD retains OPSD's contextual self-teacher construction and applies teacher-referenced response comparisons, improving Math Avg. and tool-use accuracy over OPSD at three model scales.

- **Training variants yield distinct empirical outcomes.** With the same contextual-teacher configuration on Qwen3-4B, PBSD exceeds imitation, fixed-negative training, and unconditioned reference scoring by 3.8, 3.7, and 5.2 Math Avg. points.
- **The information supplied to the self-teacher affects the resulting supervision.** Reference reasoning plus the correct answer scores highest among the tested contexts. Pair diagnostics separately identify where generation-role labels agree with, conflict with, or remain unresolved by answer correctness.

2 Related Work

On-policy distillation. On-policy distillation (OPD) trains the student on its own sampled trajectories while querying a stronger teacher on those visited states, thereby preserving the on-policy nature of RL-style optimization while replacing sparse rewards with dense token-level supervision (Agarwal et al., 2024; Song & Zheng, 2026; Fu et al., 2026). Recent work extends this paradigm to reasoning, RLVR, and domain-specific agent settings (Zhao et al., 2026a; Yang et al., 2026; Li et al., 2026a; Afsharrad et al., 2026). External-teacher OPD can encounter teacher–student mismatch in vocabulary, output space, and distributional support (Li et al., 2026b; Song & Zheng, 2026).

Self-distillation. Self-distillation takes several forms: reference-solution or demonstration conditioning (Zhao et al., 2026a; Shenfeld et al., 2026), concise-instruction conditioning (Sang et al., 2026), environment feedback (Hübotter et al., 2026), and privileged reasoning keywords for search (Chen et al., 2026). Separately, SSD improves code generation by supervised training on raw self-generated samples (Zhang et al., 2026). OPSD learns through token-level distribution matching (Zhao et al., 2026a). AntiSD (Shen et al., 2026) reverses the teacher–student divergence signal within GRPO, using grouped student rollouts. PBSD uses the contextual teacher as a preference reference, requiring only one teacher and one student response per prompt. RLSD (Yang et al., 2026) and SRPO (Li et al., 2026a) use explicit verifier rewards for updates or sample routing, whereas PBSD constructs comparison labels from teacher–student generation roles without a separate reward scorer.

Preference optimization. Preference-learning analyses study reward estimation from comparisons (Zhu et al., 2023), while DPO connects a KL-regularized policy objective to pairwise optimization (Rafailov et al., 2023). Iterative and online approaches refresh the response distribution or preference feedback during training (Pang et al., 2024; Guo et al., 2024; Qi et al., 2024; Shi et al., 2025; Tu et al., 2025). Direct Preference Knowledge Distillation (DPKD) (Li et al., 2024) uses teacher positives, online student negatives, and teacher reference scores. We apply this formulation to contextual self-distillation, studying its effectiveness relative to OPSD's token-level matching. See Appendix C for details.

3 Methodology

PBSD replaces OPSD's token-level matching with teacher-referenced sequence-level comparisons of generated responses. It retains the contextual self-teacher as both a response source and a reference for scoring. Training has three operations: generating a teacher response, pairing it with an online student response, and scoring their margin relative to the contextual teacher. We first define this response construction, then the objective and its update.

3.1 Contextual Teacher and Online Comparisons

Let x denote an input prompt and $\pi_\theta(y \mid x)$ the student policy, where $y = (y_1, \ldots, y_T)$ is an output sequence of length T. Under the standard autoregressive factorization,

$$\pi_\theta(y \mid x) = \prod_{t=1}^{T} \pi_\theta(y_t \mid x, y_{<t}). \tag{1}$$

Each training example pairs an input x with privileged context $\boldsymbol{c}$, such as its reference solution. Following contextual self-distillation (Zhao et al., 2026a), the teacher uses the same initial model as the student but receives this additional context:

$$\pi^{\text{teach}}(y \mid x) := \pi_{\theta_0}(y \mid x, \boldsymbol{c}), \tag{2}$$

where θ_0 denotes the initial model parameters, held fixed during PBSD training. The student is initialized at θ_0 and updated to θ, while receiving only x. We suppress $\boldsymbol{c}$ in the teacher notation; expectations over the training data include its associated context.

In the default fixed-teacher procedure, generate one teacher response per training question and reuse it when that question recurs within the run. Sample a new negative response from the current student at each occurrence. Both generations are capped at 1,024 new tokens. Responses that reach this cap are retained in the training loss; the compared sequences need not be complete solutions. The labels identify generation roles, using privileged context as supervision; they are not per-response correctness judgments. As the student changes, its sampled responses change the comparisons used for learning.

A fixed source of guidance, an evolving comparison distribution. At a training iteration, the current student acts as the behavior policy for collecting negative responses. The resulting pairs are then held fixed for the gradient update; subsequent rollouts use the updated student. Thus the negative distribution follows the student's current behavior, while the teacher's parameters and conditioning rule remain fixed. This differs from retaining negatives sampled only at initialization: it exposes the loss to responses the student currently produces, without assuming that later negatives are necessarily harder or more informative. The static-pair comparison in Section 4.4 evaluates this choice.

3.2 Teacher-Relative Comparison Objective

We instantiate the teacher-referenced logistic objective of DPKD (Li et al., 2024), built on DPO (Rafailov et al., 2023), with the contextual teacher as reference. The length-normalized loss below uses $p(y \mid x) = \pi_{\theta_0}(y \mid x, \boldsymbol{c})$ in DPKD Eq. (9). We use mean response-token log-probabilities, $s_\theta(y \mid x) = |y|^{-1} \log \pi_\theta(y \mid x)$ and $s_{\text{teach}}(y \mid x) = |y|^{-1} \log \pi^{\text{teach}}(y \mid x)$. For teacher and student responses y^+, y^-, define

$$m_\theta(x, y^+, y^-) = \beta \left[s_\theta(y^+ \mid x) - s_{\text{teach}}(y^+ \mid x) - s_\theta(y^- \mid x) + s_{\text{teach}}(y^- \mid x) \right]. \tag{3}$$

We use the Bradley–Terry logistic form to model the assigned ordering:

$$P_\theta(y^+ \succ y^- \mid x) = \sigma\big(m_\theta(x, y^+, y^-)\big). \tag{4}$$

Here $\sigma(z) := 1/(1 + \exp(-z))$ denotes the logistic sigmoid function. Operationally, each PBSD update uses teacher–student comparisons collected from the current student rollout and then optimizes the pairwise likelihood on that sampled batch. Writing π^{old} for the behavior policy used to generate the current student responses, we maximize the corresponding sampled pairwise log-likelihood,

$$\max_\theta \; \mathbb{E}_{(x,\boldsymbol{c})\sim\mathcal{D}} \mathbb{E}_{y^+\sim\pi^{\text{teach}}(\cdot|x), y^-\sim\pi^{\text{old}}(\cdot|x)} \left[\log \sigma\big(m_\theta(x, y^+, y^-)\big) \right]. \tag{5}$$

The teacher draw denotes generation of the stored positive response; repeated questions reuse this draw within a run rather than resample it. Training minimizes the negative of Eq. (5), with no auxiliary SFT or language-model loss. Here y^+ is generated by the contextual teacher and y^- is generated on policy by the current student behavior policy π^{old}. During the gradient step, the sampled pair is treated as fixed while gradients flow only through the optimized student policy π_θ. For a fixed pair, Eq. (5) favors increasing the student's mean log-probability difference between the teacher-generated and student-generated responses. The comparison distribution adapts to the current student, while the teacher remains fixed. Student samples are negatives by construction, including when they are correct; teacher samples are not individually verified. PBSD uses the logistic structure of DPO as a contextual self-distillation update. The generation roles supply the training labels, so each update requires one teacher response and one student response without a separate judging step.

Two uses of contextual supervision. The contextual teacher generates the positive response and evaluates the reference likelihoods of both responses. To see the second role, write $a_\theta = s_\theta(y^+ \mid x) - s_\theta(y^- \mid x)$ and $a_{\text{teach}} = s_{\text{teach}}(y^+ \mid x) - s_{\text{teach}}(y^- \mid x)$. Then $m_\theta = \beta(a_\theta - a_{\text{teach}})$: the margin measures the student's mean log-probability difference relative to the teacher's. The reference is evaluated on each generated response using its own prefixes. It remains fixed during optimization, but its pair-dependent offset changes the logistic weight and hence the strength of the student update. For a fixed pair, replacing the contextual reference with an unconditioned reference changes this weighting. Section 4.4 compares the two reference choices in runs with independently generated teacher responses.

Algorithm 1 PBSD: Preference-Based Online Self-Distillation

Input: Training pairs $\mathcal{D} = \{(x, \boldsymbol{c})\}$; initial parameters θ_0; preference scale β.
Parameters: Learning rate η; batch size b; training steps K; generation cap $L = 1{,}024$.
Output: Trained student parameters θ.

Initialize $\theta \leftarrow \theta_0$; freeze teacher parameters θ_0.
for $k = 1, \ldots, K$ **do**
  Sample $\mathcal{B} = \{(x_i, \boldsymbol{c}_i)\}_{i=1}^{b}$ from $\mathcal{D}$.
  for each $(x_i, \boldsymbol{c}_i) \in \mathcal{B}$ **do**
    Retrieve the stored teacher response for this question; if absent, generate $y_i^+ \sim \pi_{\theta_0}(\cdot \mid x_i, \boldsymbol{c}_i)$ and store it for reuse.
    Generate current-student response $y_i^- \sim \pi_\theta(\cdot \mid x_i)$.
    Cap both generations at L new tokens; retain truncated responses for the loss.
    Compute mean token log-probabilities s_θ and s_{teach} for both responses.
    Compute teacher-relative margin $m_i = m_\theta(x_i, y_i^+, y_i^-)$ via Eq. (3).
    Compute $\mathcal{L}_i = -\log \sigma(m_i)$.
  end for
  Average $\mathcal{L}_{\text{PBSD}} = b^{-1} \sum_{i=1}^{b} \mathcal{L}_i$.
  Take an AdamW step with gradient $\nabla_\theta \mathcal{L}_{\text{PBSD}}$ and learning rate η.
end for

3.3 Update and Interpretation

For a sampled triple, let $\ell_i(\theta) = -\log \sigma(m_i)$, where $m_i = m_\theta(x_i, y_i^+, y_i^-)$. With the sampled responses held fixed,

$$\nabla_\theta \ell_i = -\beta \sigma(-m_i) \left[\nabla_\theta s_\theta(y_i^+ \mid x_i) - \nabla_\theta s_\theta(y_i^- \mid x_i) \right]. \tag{6}$$

The logistic factor assigns greater weight to smaller teacher-relative margins. It is a margin-dependent weight, not a correctness score. When the student's relative score falls below the reference's, $m_i < 0$ and $\sigma(-m_i) > 1/2$; the weight decreases as the margin grows. The update therefore emphasizes pairs whose assigned ordering is less strongly expressed relative to the reference. This weighting does not verify either response, and an incorrect role ordering can still receive a large weight. The pair-composition diagnostic in Section 4.6 examines answer correctness separately from this optimization mechanism. Relative to distribution matching, PBSD changes the supervision into a contrast between two generated responses, with the negative response tracking the current student. Whether this use of contextual supervision improves task performance is evaluated experimentally. Shared parameters mean that an update need not monotonically change each individual response probability. Appendix E.2 expands the token-level gradient; Algorithm 1 summarizes training. Appendices D and E provide an optional reward interpretation and conditional local analysis.

Training and inference requirements. Each comparison uses one teacher response and one student response, with student and frozen-teacher likelihoods evaluated on both. Gradients pass only through the student likelihoods, not through response sampling or teacher scoring. These generation and scoring requirements are part of the training cost. At inference, only the trained student $\pi_\theta(\cdot \mid x)$ is used: neither privileged context nor a teacher call is required.

4 Experiments

Starting from OPSD's contextual-teacher setting, we examine task performance, training variants, teacher context, and pair composition. Main-table comparisons evaluate complete training procedures. Component comparisons share training questions, the contextual-teacher configuration, and the specified optimization settings; teacher responses are generated independently across variants. Context ablations and response diagnostics then characterize the setting in which the gains occur. Additional experiments assess transfer and retention outside the training domain.

4.1 Experiment Setup

Models and Tasks. We experiment with instruct-tuned Qwen models (Yang et al., 2025) at three scales: Qwen3-1.7B, Qwen3-4B, and Qwen3-8B. We study two task domains: **mathematical reasoning** and **tool use**. Unless otherwise noted, these two domains use the same training pipeline, model initialization, LoRA configuration, optimizer, teacher–student construction, rollout configuration, and checkpoint-selection protocol; their training data, evaluation benchmarks, and metrics are task-specific. For mathematical reasoning, we use the math subset of OpenThoughts (Guha et al., 2025), sampling up to 30K problem–solution pairs with chain-of-thought reasoning, and report results on AIME 2024, AIME 2025, and HMMT 2025. For tool use, we follow the ToolAlpaca-based setup of Shenfeld et al. (2026), using a 4046-example training split and a 94-example test split. This small test set measures generated tool-call accuracy, not broad autonomous-agent capability. Mathematical reasoning evaluation uses thinking mode (`thinking=True`). Dataset details are in Appendix F.1.

Baselines and Metrics. We compare PBSD with six baselines: SFT, GRPO (Shao et al., 2024), DAPO (Yu et al., 2025a), OPSD (Zhao et al., 2026a), SDFT (Shenfeld et al., 2026), and SRPO (Li et al., 2026a). DAPO is a large-scale RLVR recipe that strengthens GRPO-style training through decoupled clipping and dynamic sampling. SDFT is a demonstration-conditioned self-distillation method that enables on-policy learning directly from demonstrations. SRPO is a hybrid on-policy objective that routes correct samples to GRPO-style reward optimization and failed samples to self-distillation-based correction. The training setup is shared across the two task domains, but the evaluation protocol is task-specific. For mathematical reasoning, we report ***Avg@12***, namely the average accuracy over 12 sampled responses per question. For tool use, we report top-1 accuracy on the test split. Detailed evaluation protocols for both tasks are provided in Appendix F.4.

Implementation Details. Across all methods and both task domains, we fine-tune Qwen3 instruct models with LoRA (rank 64 and $\alpha = 128$) on 8 H100 GPUs using AdamW, bfloat16 precision, gradient checkpointing, and FlashAttention 2. The PBSD teacher remains fixed at the initial checkpoint throughout training. In the main-table comparison, all trainable methods are trained for 500 steps and evaluated every 50 steps. We report the latest checkpoint at step 500 for all trainable methods. Appendix F.10 reports designated longer-horizon PBSD diagnostics that continue training through step 1,000. Appendix F.3 describes the shared setup and training configurations.

4.2 Task Performance and Transfer

Response comparisons improve on OPSD across scales. Table 2 reports gains over OPSD of 1.0/1.0/2.3 percentage points in Math Avg. and 6.4/6.7/7.5 points in tool-use accuracy at 8B/4B/1.7B, respectively. PBSD also achieves the highest Math Avg. and tool-use accuracy among all compared methods at each scale. These results support its effectiveness across the tested model scales and task domains. The gains are not uniform across individual math benchmarks: GRPO scores higher on 8B HMMT25 and OPSD on 4B HMMT25. This comparison evaluates complete training procedures at step 500, including their different losses and response construction; Section 4.4 examines specific design choices.

PBSD also performs competitively when trained separately on coding and instruction-following data (Appendix F.9). We next examine behavior over training, then investigate the comparison design, teacher context, and correctness composition of the response pairs.

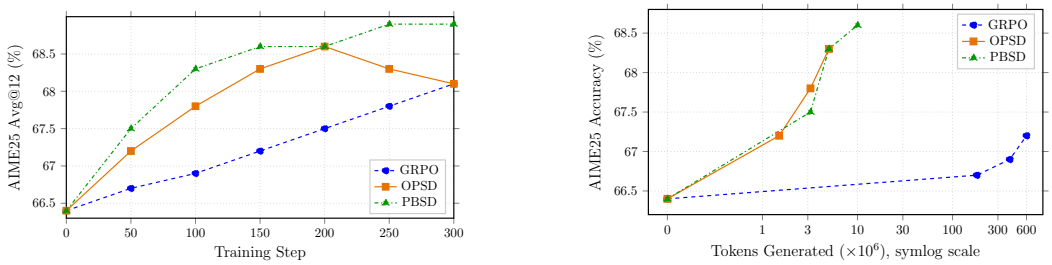
4.3 Training Dynamics and Behavioral Diagnostics

Accuracy over training. In Figure 2(A), OPSD declines after step 200, while PBSD maintains its accuracy and GRPO continues to improve. Appendix F.10 reports PBSD checkpoint scores through 1,000 steps; later checkpoints fall below step 500 for 4B and 1.7B, while 8B improves.

Computational scope. Default PBSD generates one teacher response per distinct training question and reuses it on later occurrences; student responses are generated online. Teacher generation and policy-scoring passes mean that matching the number of updates does not match total computation. We report the sampling setup but do not infer an end-to-end compute or wall-clock advantage from update counts.

Table 2: Performance comparison across mathematical reasoning and tool use for Qwen3 models. Under **Math**, we report Avg@12 using our reported sampling configuration (temperature 1.0, maximum generation length 38,912 tokens). Under **Tool Use**, we report top-1 accuracy on the tool-use test set. All scores are percentages. Scores are means over three evaluation-sampling repeats at the same checkpoint; parenthesized values denote standard deviations across these repeats. All trainable methods in the main-table comparison are trained for 500 steps and evaluated every 50 steps. This table reports the latest checkpoint at step 500 for each trainable method; Appendix F.10 reports designated longer-horizon PBSD diagnostics through step 1,000. Within each model scale, the best result in each column is shown in **bold**, and the second-best result is underlined.

Model	Method	Math				Tool Use
		AIME24	AIME25	HMMT25	Math Avg.	Acc.
Qwen3-8B	Base (Instruct)	75.6 (0.5)	65.3 (0.4)	43.6 (0.2)	61.5	61.3 (0.5)
	+ SFT	71.9 (1.5)	63.9 (0.2)	42.4 (0.3)	59.4	61.7 (1.7)
	+ GRPO	76.1 (0.6)	69.0 (0.1)	**46.3** (0.5)	63.8	68.8 (2.7)
	+ DAPO	76.0 (0.9)	68.7 (1.0)	46.0 (0.7)	63.6	67.7 (0.5)
	+ OPSD	77.3 (0.3)	70.1 (0.6)	45.1 (0.6)	64.2	65.6 (1.8)
	+ SDFT	75.9 (0.7)	69.1 (0.7)	44.2 (0.6)	63.1	62.8 (1.7)
	+ SRPO	77.1 (0.3)	69.6 (1.5)	43.9 (0.7)	63.5	62.1 (2.5)
	+ PBSD	**78.4** (0.3)	**71.0** (0.1)	46.1 (0.2)	**65.2**	**72.0** (1.3)
Qwen3-4B	Base (Instruct)	74.5 (0.3)	66.1 (0.2)	42.0 (0.3)	60.9	45.7 (0.9)
	+ SFT	71.3 (0.6)	64.3 (0.8)	43.4 (0.1)	59.7	51.1 (1.7)
	+ GRPO	75.7 (0.1)	67.8 (0.6)	44.4 (0.8)	62.6	58.9 (1.3)
	+ DAPO	76.1 (0.6)	67.7 (0.3)	44.3 (0.7)	62.7	48.2 (2.2)
	+ OPSD	76.3 (0.8)	68.1 (0.3)	**46.0** (0.6)	63.5	53.9 (1.0)
	+ SDFT	75.6 (0.3)	67.1 (0.3)	44.7 (0.2)	62.5	41.8 (1.3)
	+ SRPO	76.3 (0.3)	67.8 (0.4)	45.5 (0.3)	63.2	46.1 (2.0)
	+ PBSD	**78.3** (0.2)	**69.7** (0.2)	45.6 (0.3)	**64.5**	**60.6** (1.7)
Qwen3-1.7B	Base (Instruct)	51.4 (0.2)	37.2 (0.5)	23.1 (0.1)	37.2	36.9 (1.0)
	+ SFT	49.9 (0.3)	36.6 (0.5)	22.8 (0.2)	36.4	39.7 (0.5)
	+ GRPO	50.6 (0.5)	38.4 (0.1)	23.5 (0.3)	37.5	42.9 (0.5)
	+ DAPO	51.5 (0.7)	38.6 (1.4)	23.5 (0.3)	37.9	43.6 (0.9)
	+ OPSD	57.1 (0.8)	43.3 (0.6)	29.3 (0.3)	43.2	37.2 (1.7)
	+ SDFT	57.0 (0.3)	43.0 (0.1)	28.7 (0.3)	42.9	38.3 (1.7)
	+ SRPO	58.1 (0.2)	44.1 (0.3)	29.4 (0.5)	43.9	36.9 (1.3)
	+ PBSD	**59.2** (0.4)	**46.4** (0.2)	**31.0** (0.2)	**45.5**	**44.7** (1.7)



(A) (B)

Figure 2: AIME25 training diagnostics for GRPO, OPSD, and PBSD. (A) Avg@12 over steps 0–300. (B) Accuracy versus the reported cumulative generated-token count on a symlog axis. These panels show diagnostic checkpoints, not the step-500 results in Table 2. Generated-token counts are not a measure of total training computation.

Output behavior and capability retention. Appendix F.14 reports mean response lengths and self-checking expression frequencies. In the separate length diagnostic, PBSD maintains mean response lengths near 1,350 tokens under a 2,048-token generation cap. It also shows less suppression of self-checking expressions, a lexical measure that complements accuracy without directly measuring reasoning quality. Separately, the math-trained model stays within 0.8 percentage points of the base on five OOD metrics but loses 4.73 points on MBPP+ (Appendix F.11).

4.4 Controlled Component Comparisons

Table 3 compares four Qwen3-4B variants using the same training questions, contextual-teacher configuration, and optimization settings (Appendix G). Each variant independently generates teacher responses and reuses its own response when a question recurs. Teacher-response SFT imitates these responses; static-pair DPO uses negatives sampled from the initial student; the unconditioned-reference variant uses online negatives but scores the reference without context. All variants use 500 steps and share batch size, learning rate, and LoRA configuration.

Full PBSD scores highest among the tested variants. Full PBSD achieves 64.5% Math Avg., compared with 60.7% for teacher-response SFT, 60.8% for static-pair DPO, and 59.3% for the unconditioned-reference variant. The gaps of 3.8, 3.7, and 5.2 points favor the full procedure in these runs. Full PBSD also exceeds the initial model (60.9%) by 3.6 points, while the three alternative variants score below it. Because teacher texts are not controlled across variants, the comparisons do not isolate the contribution of each design choice from differences in generated supervision. The gaps are not additive component effects.

Table 3: Component comparisons on Qwen3-4B. Scores are Avg@12 accuracy (%); trainable variants use the latest checkpoint at step 500. Math Avg. is the unweighted mean of the three benchmarks. Initial-model and full-PBSD scores are reused from Table 2.

Method	AIME24	AIME25	HMMT25	Math Avg.
Initial model	74.5	66.1	42.0	60.9
Teacher-response SFT	72.4	65.9	43.9	60.7
Static-pair DPO	72.3	66.2	43.8	60.8
PBSD, unconditioned reference	72.1	64.3	41.4	59.3
Full PBSD	**78.3**	**69.7**	**45.6**	**64.5**

4.5 Sensitivity to Teacher Context

We next vary the information provided to the contextual teacher on Qwen3-4B. Table 4 compares the default reference reasoning and answer against answer-only context and synthetic reasoning with the same ground-truth answer. The synthetic trace is generated by Qwen3-32B conditioned on the question and answer; its reasoning is not guaranteed to be correct. The student receives only the question in every condition. Appendix F.8 provides further details.

Table 4: Privileged-context sensitivity on Qwen3-4B mathematical reasoning. The GT reasoning + GT answer row reuses the default PBSD mean scores from Table 2 at step 500.

Teacher Privileged Information	AIME24	AIME25	HMMT25
GT reasoning + GT answer	78.3	69.7	45.6
GT answer only	76.2	67.9	45.1
Synthetic reasoning + GT answer	76.5	68.5	45.3

The correct answer alone does not recover the full effect of context. Reference reasoning with the correct answer scores highest among the three tested configurations. Answer-only context lowers AIME24/AIME25/HMMT25 accuracy by 2.1/1.8/0.5 points; synthetic reasoning with the answer produces smaller gaps of 1.8/1.2/0.3 points. Since every variant contains the correct answer, answer availability alone does not explain this ordering. The form of teacher-side information therefore matters for downstream self-distillation. Context affects both response generation and reference scoring; this experiment does not isolate reasoning correctness from length, style, or content, or test robustness to incorrect answers.

Additional sensitivity analyses. The β sweep examines the preference scale (Appendix F.12), while teacher refreshes provide no clear benefit in the tested setup (Appendix F.7). GKD and distribution-gap comparisons provide further context for the teacher configuration (Appendix F.13).

4.6 Correctness Composition of Teacher–Student Pairs

To complement the task-level comparisons, we examine answer correctness within the role-labeled training pairs. We examine 1,536 teacher–student pairs from Qwen3-4B at step 500. After excluding 60 truncated pairs (3.91%), 1,476 pairs remain for answer-correctness classification. Table 5 reports their composition; Appendix H gives the diagnostic protocol and supplementary accounting.

Table 5: Answer-correctness composition of Qwen3-4B teacher–student pairs at step 500. Percentages use the 1,476 classified pairs; 60 of the original 1,536 pairs are excluded for truncation.

Pair category	**Count**	**Share (%)**
Teacher correct / Student incorrect	440	29.81
Teacher incorrect / Student correct	7	0.47
Both correct	801	54.27
Both incorrect	228	15.45

Role labels have asymmetric correctness support, not per-pair verification. Among classified pairs, answer correctness supports the teacher-first ordering in 29.81% and opposes it in 0.47%. Correctness-aligned pairs therefore substantially outnumber reversed pairs at this checkpoint. Most pairs, however, have equal final-answer correctness, which neither establishes equal response quality nor validates their assigned ordering. These counts describe pair composition, not gradient contributions or which pairs drive the gains.

5 Limitations

The Qwen3-4B component comparisons use independently generated teacher responses, so they do not isolate design effects at fixed texts. Context ablations retain correct answers, while generation-role labels do not verify individual responses; pair counts and lexical diagnostics therefore characterize supervision and behavior rather than identify the updates responsible for performance gains. The local theory concerns a fixed-distribution surrogate, not adaptive training convergence or downstream accuracy. Empirically, the primary comparison uses 500 training steps and thinking-enabled mathematical evaluation, and reported standard deviations measure evaluation-sampling variability rather than independent training variability. The small tool-use test and selected context configurations also limit generalization to broader agent tasks and other forms of privileged information. See Appendix A.

6 Conclusion

PBSD improves Math Avg. and tool-use accuracy over OPSD at three model scales by using teacher-referenced response comparisons. Among the tested Qwen3-4B variants, full PBSD scores highest. Reference reasoning with the correct answer also scores higher than answer-only context, while pair diagnostics characterize the limits of generation-role labels. These findings highlight two considerations for contextual self-distillation: what information the self-teacher receives, and how the training procedure uses that information.

AI USE STATEMENT

During preparation of this submission, OpenAI Codex was used to assist with migration of the existing manuscript to the ICLR 2027 LaTeX template, formatting checks, revising the manuscript narrative and shortening prose to satisfy the page limit, drafting the submission statements, integrating author-provided experimental results and clarifications, assisting with revision of the local theoretical argument and proof, and providing critical feedback on the methodology and interpretation of existing results. OpenAI image generation was used to edit Figure 1, preserving the original illustration style while replacing the PBSD panel with the implemented response-pairing and student-update flow. For the synthetic-context experiment, Qwen3-32B generated reasoning conditioned on training questions and their ground-truth answers (Appendix F.8). The model-based generation of teacher and student responses used by PBSD is described in the methodology and experimental setup.

REPRODUCIBILITY STATEMENT

The implemented preference objective, gradient, and training procedure are specified in the methodology. Appendix D separately presents the reward-regularized interpretation and its derivations. The assumptions and proofs supporting the local statistical analysis are provided in Appendix E. Appendix F.1 describes the datasets, while Appendices F.3 and F.4 document the training and evaluation configurations, including model initialization, optimizer, LoRA settings, hardware, rollout generation, and checkpoint selection. Additional ablations and longer-horizon checkpoint comparisons are provided in Appendix F.

REFERENCES

Amirhossein Afsharrad, Amirhesam Abedsoltan, Ahmadreza Moradipari, and Sanjay Lall. On-policy distillation of language models for autonomous vehicle motion planning. *arXiv preprint arXiv:2604.07944*, 2026.

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 *The Twelfth International Conference on Learning Representations*, 2024.

Jiamu Bai, Xin Yu, Meilong Xu, Weitao Lu, Xin Pan, Kiwan Maeng, Daniel Kifer, Jian Wang, and Yu Wang. Towards better optimization for listwise preference in diffusion models. *arXiv preprint arXiv:2510.01540*, 2025.

Ben Chen, Siyuan Wang, Yufei Ma, Zihan Liang, Xuxin Zhang, Yue Lv, Ying Yang, Huangyu Dai, Lingtao Mao, Tong Zhao, et al. Onesearch-v2: The latent reasoning enhanced self-distillation generative search framework. *arXiv preprint arXiv:2603.24422*, 2026.

Yuqian Fu, Haohuan Huang, Kaiwen Jiang, Jiacai Liu, Zhuo Jiang, Yuanheng Zhu, and Dongbin Zhao. Revisiting on-policy distillation: Empirical failure modes and simple fixes. *arXiv preprint arXiv:2603.25562*, 2026.

Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, et al. Openthoughts: Data recipes for reasoning models. *arXiv preprint arXiv:2506.04178*, 2025.

Shangmin Guo, Biao Zhang, Tianlin Liu, Tianqi Liu, Misha Khalman, Felipe Llinares, Alexandre Rame, Thomas Mesnard, Yao Zhao, Bilal Piot, Johan Ferret, and Mathieu Blondel. Direct language model alignment from online AI feedback. *arXiv preprint arXiv:2402.04792*, 2024.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, et al. Reinforcement learning via self-distillation. *arXiv preprint arXiv:2601.20802*, 2026.

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. *arXiv preprint arXiv:2403.07974*, 2024.

Jeonghye Kim, Xufang Luo, Minbeom Kim, Sangmook Lee, Dohyung Kim, Jiwon Jeon, Dongsheng Li, and Yuqing Yang. Why does self-distillation (sometimes) degrade the reasoning capability of llms? *arXiv preprint arXiv:2603.24472*, 2026.

Gengsheng Li, Tianyu Yang, Junfeng Fang, Mingyang Song, Mao Zheng, Haiyun Guo, Dan Zhang, Jinqiao Wang, and Tat-Seng Chua. Unifying group-relative and self-distillation policy optimization via sample routing. *arXiv preprint arXiv:2604.02288*, 2026a.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huanang Gao, Wenkai Yang, Zhiyuan Liu, and Ning Ding. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*, 2026b.

Yixing Li, Yuxian Gu, Li Dong, Dequan Wang, Yu Cheng, and Furu Wei. Direct preference knowledge distillation for large language models. *arXiv preprint arXiv:2406.19774*, 2024. URL https://arxiv.org/abs/2406.19774v2.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by ChatGPT really correct? rigorous evaluation of large language models for code generation. *arXiv preprint arXiv:2305.01210*, 2023. URL https://arxiv.org/abs/2305.01210.

Kevin Lu and Thinking Machines Lab. On-policy distillation. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20251026. https://thinkingmachines.ai/blog/on-policy-distillation.

Richard Yuanzhe Pang, Weizhe Yuan, Kyunghyun Cho, He He, Sainbayar Sukhbaatar, and Jason Weston. Iterative reasoning preference optimization. *arXiv preprint arXiv:2404.19733*, 2024.

Biqing Qi, Pengfei Li, Fangyuan Li, Junqi Gao, Kaiyan Zhang, and Bowen Zhou. Online dpo: Online direct preference optimization with fast-slow chasing. *arXiv preprint arXiv:2406.05534*, 2024.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, 2023.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof Q&A benchmark. *arXiv preprint arXiv:2311.12022*, 2023. URL https://arxiv.org/abs/2311.12022.

Hejian Sang, Yuanda Xu, Zhengze Zhou, Ran He, Zhipeng Wang, and Jiachen Sun. On-policy self-distillation for reasoning compression. *arXiv preprint arXiv:2603.05433v1*, 2026.

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

Guobin Shen, Xiang Cheng, Chenxiao Zhao, Lei Huang, Jindong Li, Dongcheng Zhao, and Xing Yu. Anti-self-distillation for reasoning RL via pointwise mutual information. *arXiv preprint arXiv:2605.11609*, 2026.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning. *arXiv preprint arXiv:2601.19897*, 2026.

Ruizhe Shi, Runlong Zhou, and Simon Shaolei Du. The crucial role of samplers in online direct preference optimization. In *International Conference on Learning Representations*, 2025.

Mingyang Song and Mao Zheng. A survey of on-policy distillation for large language models. *arXiv preprint arXiv:2604.00626*, 2026.

Joel A. Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12:389–434, 2012. doi: 10.1007/s10208-011-9099-z.

Songjun Tu, Jiahao Lin, Xiangyu Tian, Qichao Zhang, Linjing Li, Yuqian Fu, Nan Xu, Wei He, Xiangyuan Lan, Dongmei Jiang, and Dongbin Zhao. Enhancing llm reasoning with iterative dpo: A comprehensive empirical investigation. In *Conference on Language Modeling*, 2025.

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. *arXiv preprint arXiv:2406.01574*, 2024. URL https://arxiv.org/abs/2406.01574.

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

Chenxu Yang, Chuanyu Qin, Qingyi Si, Minghui Chen, Naibin Gu, Dingyu Yao, Zheng Lin, Weiping Wang, Jiaqi Wang, and Nan Duan. Self-distilled rlvr. *arXiv preprint arXiv:2604.03128*, 2026.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025a.

Xin Yu, Yujia Wang, Jinghui Chen, and Lingzhou Xue. Altlora: Towards better gradient approximation in low-rank adaptation with alternating projections. *arXiv preprint arXiv:2505.12455*, 2025b.

Xin Yu, Cong Xie, Ziyu Zhao, Tiantian Fan, Lingzhou Xue, and Zhi Zhang. Prunedlora: Robust gradient-based structured pruning for low-rank adaptation in fine-tuning. *arXiv preprint arXiv:2510.00192v1*, 2025c.

Xin Yu, Hanwen Xing, and Lingzhou Xue. Exact: Explicit attribute-guided decoding-time personalization. *arXiv preprint arXiv:2602.17695*, 2026.

Ruixiang Zhang, Richard He Bai, Huangjie Zheng, Navdeep Jaitly, Ronan Collobert, and Yizhe Zhang. Embarrassingly simple self-distillation improves code generation. *arXiv preprint arXiv:2604.01193*, 2026.

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. *arXiv preprint arXiv:2601.18734*, 2026a.

Ziyu Zhao, Yixiao Zhou, Xin Yu, Zhi Zhang, Didi Zhu, Tao Shen, Zexi Li, Jinluan Yang, Xuwu Wang, Jing Su, et al. Each rank could be an expert: Single-ranked mixture of experts lora for multi-task learning. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 1998–2009, 2026b.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023. URL https://arxiv.org/abs/2311.07911.

Banghua Zhu, Michael I. Jordan, and Jiantao Jiao. Principled reinforcement learning with human feedback from pairwise or k-wise comparisons. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 43037–43067. PMLR, 2023.

Banghua Zhu, Michael I. Jordan, and Jiantao Jiao. Iterative data smoothing: Mitigating reward overfitting and overoptimization in rlhf. *arXiv preprint arXiv:2401.16335*, 2024.

A Additional Discussion of Limitations

What the component comparisons identify. The component comparisons share training questions, teacher configuration, and optimization settings, but each variant generates and reuses its own teacher responses. Their gaps compare training procedures without isolating effects at fixed texts or exhausting component interactions. The OPSD comparison also changes both the objective and response construction.

Context quality and generation-role labels. Context ablations retain correct answers and do not test incorrect or missing solutions. Reasoning traces can differ in correctness, length, style, and content. Pair correctness at one checkpoint characterizes generation-role labels but does not identify gradient contributions; equal final-answer correctness need not imply equal response quality. Length and lexical frequencies likewise measure behavior rather than reasoning quality or calibration.

Training variability and evaluation coverage. The reported main-table standard deviations measure evaluation-sampling variability at fixed checkpoints. They do not quantify reproducibility across independently trained models. The primary comparison reports the latest checkpoint at step 500; the longer-horizon diagnostics concern PBSD and do not compare all baselines beyond that budget. Mathematical evaluation uses thinking mode, and performance under non-thinking evaluation remains untested here. The 94-example tool-use test measures generated tool-call accuracy rather than broad autonomous-agent capability. The math-trained checkpoint also loses 4.73 percentage points on MBPP+, so the OOD results do not establish uniformly preserved capabilities.

Teacher comparisons and computation. The larger-teacher comparison changes both teacher configuration and distillation objective. It cannot establish that smaller teachers are generally preferable or that context matters more than model scale. Teacher–student KL measurements likewise do not establish an ordering of downstream teaching quality. PBSD additionally generates teacher responses and scores both responses under both policies; a shared update budget is not a matched computation budget. Generated-token curves alone do not establish an end-to-end compute or wall-clock advantage.

Scope of the theoretical interpretation. The sequence-level reward interpretation and closed-form optimum describe an idealized objective, not a derivation of the implemented length-normalized training loss. The statistical results require a fixed comparison distribution, bounded scores, and positive full local curvature; they characterize local estimation rather than adaptive training convergence or accuracy gains. If teacher and student distributions coincide in an idealized realizable setting, independent samples have the same distribution and the expected comparison gradient cancels. Thus the objective does not itself guarantee departing from or outperforming the teacher. The performance claims rest on the experiments.

B Appendix Overview

This appendix is organized to mirror the logic of the main paper from background, to derivation, to theory, and finally to experimental detail. Appendix C expands the literature discussion that is abbreviated in the main text. It is divided into three thematic parts: on-policy distillation, self-distillation, and RLHF/DPO-style preference optimization. The goal of this section is to situate PBSD relative to dense teacher-based post-training, contextual self-distillation, and role-labeled comparisons.

Appendix D provides the derivations that support the methodology section. Section D.2 contains two components: first, the derivation of the optimal reward-tilted teacher distribution used in Proposition 1; second, the proof that this reward-tilted solution is no worse than directly using the teacher under the same reward-regularized objective, corresponding to Proposition 2. These identities support the reward-regularized interpretation; they do not establish equivalence to the implemented objective.

Appendix E develops the supplementary local statistical analysis of the comparison objective. Section E.3 derives the sample-level gradient and the local Hessian form of the online PBSD objective, which are the ingredients needed to interpret PBSD as a pairwise logistic-style estimation problem. Section E.4 then states the local assumptions used in the analysis, explains the role and scope of these assumptions, discusses what happens when they are violated, and finally proves the conditional local estimation bound for the induced learning problem.

Appendix F collects the detailed experimental material that supports Section 4 of the main paper. Section F.1 describes the two task domains, namely mathematical reasoning and tool use, together with the corresponding training and evaluation splits. The data-visualization subsection then gives concrete examples of the processed mathematical reasoning and tool-use instances so that the structure of the training data is explicit. Section F.3 documents the shared setup and training configurations. Section F.4 specifies the decoding and evaluation protocols for both mathematical reasoning and tool use. The case-by-case study subsection explains the problem-level analysis on 30 selected AIME25 questions. The output-diagnostics subsection describes the scope of behavioral proxies. The teacher-update-frequency subsection reports the fixed-teacher versus refreshed-teacher ablation. We then include additional analyses on privileged-context sensitivity, broader-domain transfer, longer-horizon checkpoint behavior, OOD performance, β sensitivity, larger-teacher comparisons, teacher–student KL and verbalized uncertainty. Appendix F.15 discusses possible explanations of the tool-use gains.

C Appendix for Related Work

On-policy distillation. OPD trains the student on its own sampled trajectories while querying a stronger teacher on those visited states, reducing the mismatch of off-policy distillation (Agarwal et al., 2024). This paradigm has become an important post-training recipe for LLMs because it keeps the on-policy nature of RL-style optimization while replacing sparse outcome rewards with dense token-level supervision (Song & Zheng, 2026; Fu et al., 2026). Recent extensions apply this idea to reasoning and RLVR settings, including contextual on-policy supervision (Zhao et al., 2026a), self-distilled RLVR (Yang et al., 2026), sample-routing formulations that connect distillation and group-relative optimization (Li et al., 2026a), and domain-specific adaptations such as autonomous driving (Afsharrad et al., 2026). However, most OPD methods still depend on an additional large teacher model throughout training, which increases online inference cost and system complexity. They also suffer from teacher–student matching issues, including incompatible vocabularies, output spaces, and large distribution gaps that can make token-level imitation brittle (Li et al., 2026b; Song & Zheng, 2026). These limitations motivate self-distillation methods that seek to preserve dense on-policy supervision without relying on an external teacher.

Self-distillation. Contextual self-distillation uses additional information to create a teacher from the student's base model. OPSD applies this construction to reasoning (Zhao et al., 2026a), while related methods use expert demonstrations for continual learning (Shenfeld et al., 2026), concise instructions for reasoning compression (Sang et al., 2026), or environment feedback for reinforcement learning (Hübotter et al., 2026). OneSearch-V2 internalizes privileged reasoning keywords for generative search (Chen et al., 2026). A distinct approach, SSD, uses ordinary supervised fine-tuning on self-generated code without a privileged teacher (Zhang et al., 2026). Analyses of self-distillation report that direct matching can sometimes harm reasoning (Kim et al., 2026); other approaches combine distillation with reward-based updates or sample routing (Yang et al., 2026; Li et al., 2026a). PBSD studies online response comparisons within this contextual-teacher setting, using a fixed teacher to supply both positive responses and the reference distribution.

RLHF and DPO-style preference optimization. A parallel line of work studies preference optimization more broadly through RLHF and DPO-style objectives. Zhu et al. (Zhu et al., 2023) analyze reward estimation from pairwise and K-wise comparisons, while DPO connects a KL-regularized policy objective to a preference loss (Rafailov et al., 2023). Iterative Data Smoothing instead addresses reward-model overfitting through model-dependent soft labels (Zhu et al., 2024). On the algorithmic side, recent work has explored iterative and online variants of DPO as a lower-cost alternative to RL. Iterative Reasoning Preference Optimization (Pang et al., 2024) shows that repeatedly regenerating reasoning trajectories and re-optimizing preference pairs can substantially improve reasoning, especially when combined with an auxiliary NLL term. Tu et al. (Tu et al., 2025) further provide a comprehensive empirical study of iterative DPO for reasoning, arguing that multi-round DPO together with iterative reward-model refinement can approach RL-level performance at lower computational cost. In the RLVR setting, DAPO (Yu et al., 2025a) scales GRPO-style training with decoupled clipping and dynamic sampling, while SRPO (Li et al., 2026a) explicitly combines group-relative reward optimization with self-distillation through sample routing. In a more explicitly online setting, OFS-DPO and COFS-DPO (Qi et al., 2024) study online DPO under streaming or cross-domain preference updates, emphasizing continual adaptation and regret-based analysis. Complementarily, Shi et al. (Shi et al., 2025) analyze the optimization behavior of online DPO and show that sampler design

has a decisive effect on convergence rates, with stronger online samplers yielding faster convergence both theoretically and empirically. Compared with these lines of work, our focus is different: rather than studying DPO primarily as a general preference-optimization alternative to RLHF, we study role-labeled comparisons for contextual self-distillation and use the reward-regularized formulation as an interpretation of the teacher reference.

Relationship to DPKD. DPKD (Li et al., 2024) studies preference distillation from a separate teacher. PBSD uses its length-normalized teacher-referenced preference term in a contextual self-distillation setting: the teacher is a frozen copy of the initial student with privileged context. Teacher-positive/student-negative pairing and teacher reference scoring are shared. DPKD's full recipe also includes an auxiliary language-model loss on a pretraining corpus; PBSD uses the preference term alone. Our experiments compare training procedures within contextual self-distillation, rather than isolate the effect of teacher construction relative to the full DPKD recipe.

AntiSD and reward-assisted self-distillation. AntiSD (Shen et al., 2026) addresses reasoning degradation by ascending teacher–student Jensen–Shannon divergence within a GRPO-based training procedure, with an entropy-triggered gate for the distillation signal. It therefore retains a contextual teacher while changing how its signal is used. Its reported configuration samples eight student rollouts per prompt. PBSD uses the contextual teacher as both a positive-response generator and a preference reference, constructing each pair from one teacher response and one on-policy student response. Each PBSD pair uses one teacher response and one online student response, with teacher responses reused on repeated questions in the default procedure; no group-based reward estimation is used. RLSD (Yang et al., 2026) and SRPO (Li et al., 2026a) also incorporate explicit verifier rewards for updates or sample routing, whereas PBSD assigns preferences through teacher–student generation roles without an external scorer.

D Appendix for Methodology

This section collects the proofs and derivations corresponding to the methodology section in the main text. Section D.2 derives the reward-tilted optimal policy under the reward-regularized objective and compares its objective value with that of the teacher.

D.1 Interpreting the Teacher Reference

Let the teacher policy be denoted by $\pi^{\mathrm{teach}}(\cdot \mid x)$, omitting parameters because the teacher is fixed during optimization. Classical on-policy distillation often trains the student to match the teacher distribution through KL divergence. A common formulation minimizes the reverse KL from the student to the teacher:

$$\min_{\pi_\theta} \ \mathbb{E}_{x\sim\mathcal{D}}\left[D_{\mathrm{KL}}\big(\pi_\theta(\cdot \mid x) \,\|\, \pi^{\mathrm{teach}}(\cdot \mid x)\big)\right], \tag{7}$$

which provides an idealized sequence-level reference objective. Practical OPD methods can instead use token-level divergence surrogates (Agarwal et al., 2024; Lu & Thinking Machines Lab, 2025; Fu et al., 2026); OPSD adopts forward KL in its reported main experiments (Zhao et al., 2026a). Under this objective, the teacher distribution itself is treated as the target to be learned. The following sequence-level reward-regularized objective gives an optional interpretation of the teacher as a reference measure. It uses unnormalized sequence log-probabilities and does not derive the implemented length-normalized preference loss. It introduces a latent reward for this interpretation; PBSD does not observe that reward or use it to label training pairs. For this objective-level interpretation, associate each input with a fixed context and regard the resulting teacher distribution as given. Let $r(x, y)$ denote a bounded latent target reward. Over unrestricted conditional distributions π on the teacher support, consider

$$\max_{\pi(\cdot|x)} \ \mathbb{E}_{x\sim\mathcal{D}}\big[\mathbb{E}_{y\sim\pi(\cdot|x)}[r(x,y)] - \beta D_{\mathrm{KL}}\big(\pi(\cdot \mid x) \,\|\, \pi^{\mathrm{teach}}(\cdot \mid x)\big)\big], \tag{8}$$

where $\beta > 0$ weights the teacher-reference KL penalty. Here, reward reweights teacher-supported responses, while larger β penalizes teacher divergence more strongly.

The following statements concern this unrestricted distributional optimum; they do not assume that it is realizable by the parameterized student or its LoRA adaptation. Context is held fixed in these statements, rather than optimized or averaged across different contexts for the same input.

Proposition 1 (Optimal reward-tilted policy)**.** *For each fixed input and associated context, with $0 < Z(x) < \infty$, the unrestricted optimizer of Eq. (8) is the reward-tilted teacher distribution*

$$\pi^\star(y \mid x) = \frac{\pi^{\text{teach}}(y \mid x)\exp(r(x,y)/\beta)}{Z(x)}, \qquad Z(x) = \sum_y \pi^{\text{teach}}(y \mid x)\exp(r(x,y)/\beta). \tag{9}$$

Equivalently, the latent reward can be written in terms of the optimal policy and the teacher up to an x-dependent additive constant:

$$r(x,y) = \beta \log \frac{\pi^\star(y \mid x)}{\pi^{\text{teach}}(y \mid x)} + \beta \log Z(x). \tag{10}$$

The reward–policy reparameterization (Rafailov et al., 2023) makes the teacher's role explicit: it provides a reference measure, while the reward changes the target within its support. Appendix D.2 gives the derivation. The next proposition states the corresponding objective-level comparison.

Proposition 2 (Optimal objective value relative to the teacher)**.** *Let F denote the population version of Eq. (8) aggregated over $x \sim \mathcal{D}$. Then the optimizer $\pi^\star$ in Eq. (9) satisfies*

$$F(\pi^\star) \geq F(\pi^{\text{teach}}), \tag{11}$$

with strict inequality whenever $r(x,y)$ is non-constant over teacher-supported responses on a set of inputs with positive measure.

The proof is in Appendix D.2. This is an objective-value comparison for the unrestricted optimum. Performance of the trained PBSD student is evaluated empirically.

D.2 Details for the Reward-Regularized Distillation Objective

Derivation of Proposition 1. For notational simplicity, fix x and its associated context, and write $\pi_y = \pi(y \mid x)$, $\pi_y^{\text{teach}} = \pi^{\text{teach}}(y \mid x)$, and $r_y = r(x,y)$. The single-input objective in Eq. (8) becomes

$$\max_{\pi \in \Delta} \left\{ \sum_y \pi_y r_y - \beta \sum_y \pi_y \log \frac{\pi_y}{\pi_y^{\text{teach}}} \right\}, \tag{12}$$

where Δ denotes the unrestricted probability simplex on the teacher support, not the parameterized student family. Since the objective is strictly concave in π whenever $\beta > 0$, the optimizer is unique and can be obtained from the first-order optimality conditions. Introducing a Lagrange multiplier λ for the normalization constraint $\sum_y \pi_y = 1$, the Lagrangian is

$$\mathcal{J}(\pi, \lambda) = \sum_y \pi_y r_y - \beta \sum_y \pi_y \log \frac{\pi_y}{\pi_y^{\text{teach}}} + \lambda \left(\sum_y \pi_y - 1 \right). \tag{13}$$

Differentiating with respect to each coordinate π_y gives

$$\frac{\partial \mathcal{J}}{\partial \pi_y} = r_y - \beta \left(\log \frac{\pi_y}{\pi_y^{\text{teach}}} + 1 \right) + \lambda = 0. \tag{14}$$

Here we used the identity

$$\frac{\partial}{\partial \pi_y} \left(\pi_y \log \frac{\pi_y}{\pi_y^{\text{teach}}} \right) = \log \frac{\pi_y}{\pi_y^{\text{teach}}} + 1.$$

Rearranging Eq. (14) yields

$$\log \frac{\pi_y}{\pi_y^{\text{teach}}} = \frac{r_y + \lambda - \beta}{\beta}, \tag{15}$$

and exponentiating both sides gives

$$\pi_y = \pi_y^{\text{teach}} \exp(r_y/\beta) \exp((\lambda - \beta)/\beta). \tag{16}$$

The last exponential factor is independent of y, so all coordinates share the same proportionality constant. To determine it, impose the normalization constraint:

$$1 = \sum_y \pi_y = \exp((\lambda - \beta)/\beta) \sum_y \pi_y^{\text{teach}} \exp(r_y/\beta).$$

Therefore,

$$\exp((\lambda - \beta)/\beta) = \left(\sum_y \pi_y^{\text{teach}} \exp(r_y/\beta) \right)^{-1} = \frac{1}{Z(x)}.$$

Substituting this back gives

$$\pi_y = \frac{\pi_y^{\text{teach}} \exp(r_y/\beta)}{Z(x)},$$

which is exactly the reward-tilted teacher distribution in Eq. (9). This derivation makes explicit that the teacher policy serves only as a reference measure, while the reward term changes the final target by exponentially reweighting the teacher support according to r_y.

Proof of Proposition 2. We first compute the value of the reward-regularized objective at the optimizer $\pi^\star$. For a fixed x, substituting

$$\pi^\star(y \mid x) = \frac{\pi^{\text{teach}}(y \mid x) \exp(r(x,y)/\beta)}{Z(x)}$$

into the KL term gives

$$\log \frac{\pi^\star(y \mid x)}{\pi^{\text{teach}}(y \mid x)} = \frac{r(x,y)}{\beta} - \log Z(x).$$

Hence

$$\begin{aligned}
&\mathbb{E}_{y \sim \pi^\star(\cdot|x)}[r(x,y)] - \beta D_{\text{KL}}\big(\pi^\star(\cdot \mid x) \,\|\, \pi^{\text{teach}}(\cdot \mid x)\big) \\
&= \sum_y \pi^\star(y \mid x) r(x,y) - \beta \sum_y \pi^\star(y \mid x) \left(\frac{r(x,y)}{\beta} - \log Z(x) \right) \\
&= \sum_y \pi^\star(y \mid x) r(x,y) - \sum_y \pi^\star(y \mid x) r(x,y) + \beta \log Z(x) \sum_y \pi^\star(y \mid x) \\
&= \beta \log Z(x).
\end{aligned}$$

Aggregating over $x \sim \mathcal{D}$ gives

$$F(\pi^\star) = \mathbb{E}_{x \sim \mathcal{D}} \left[\beta \log Z(x) \right] = \mathbb{E}_{x \sim \mathcal{D}} \left[\beta \log \sum_y \pi^{\text{teach}}(y \mid x) \exp(r(x,y)/\beta) \right]. \tag{17}$$

In contrast, evaluating the teacher itself under the same objective removes the KL term because $D_{\text{KL}}(\pi^{\text{teach}} \| \pi^{\text{teach}}) = 0$. Therefore,

$$F(\pi^{\text{teach}}) = \mathbb{E}_{x \sim \mathcal{D}} \mathbb{E}_{y \sim \pi^{\text{teach}}(\cdot|x)} \left[r(x,y) \right]. \tag{18}$$

To compare the two values, fix x and write

$$Z(x) = \mathbb{E}_{y \sim \pi^{\text{teach}}(\cdot|x)} \left[\exp(r(x,y)/\beta) \right].$$

Since $\log(\cdot)$ is concave, Jensen's inequality implies

$$\beta \log \mathbb{E}_{y \sim \pi^{\text{teach}}(\cdot|x)} \left[\exp(r(x,y)/\beta) \right] \geq \mathbb{E}_{y \sim \pi^{\text{teach}}(\cdot|x)} \left[r(x,y) \right], \tag{19}$$

where equality holds only when $r(x,y)$ is constant over the support of $\pi^{\text{teach}}(\cdot \mid x)$. Combining Eq. (17), Eq. (18), and Eq. (19) yields

$$F(\pi^\star) \geq F(\pi^{\text{teach}}).$$

Thus the unrestricted optimum has objective value at least that of the teacher. The inequality is strict when the reward varies over teacher-supported responses on a set of inputs with positive measure. This comparison concerns the stated reward-regularized objective and does not establish an accuracy ordering for the trained student.

E Appendix for Theoretical Analysis

This appendix collects supplementary mathematical properties of the comparison objective. Section E.3 derives the sample-level gradient and the local Hessian form of the online PBSD objective. Section E.4 proves the conditional local estimation bound and formalizes the statistical interpretation based on informative comparison pairs.

This appendix derives the sample gradient and the complete Hessian, then analyzes estimation under a fixed comparison distribution. The local bound requires control of the full Hessian, including the nonlinear-margin remainder. We distinguish these mathematical conditions from empirical observations about teacher–student distributions.

E.1 Local Analysis of the Preference Objective

We characterize the local geometry of the comparison loss and estimation error under a frozen comparison distribution. The result relates empirical and population stationary points under explicit full-curvature conditions. It concerns the surrogate risk with role-assigned labels; the adaptive optimization trajectory and downstream accuracy are outside its scope.

E.1.1 Curvature and Local Estimation

Freeze the teacher and the student behavior policy, and let $Z_i = (x_i, y_i^+, y_i^-)$, $i = 1, \dots, n$, be i.i.d. comparisons from this fixed distribution Q. All derivatives below are in a fixed d-dimensional local parameterization. Define

$$\widehat{\mathcal{L}}_n(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell(\theta; Z_i), \qquad \mathcal{L}(\theta) = \mathbb{E}_Q[\ell(\theta; Z)], \tag{20}$$

where $\ell(\theta; Z) = -\log \sigma(m_\theta(Z))$. Write $d_i(\theta) = \nabla_\theta s_\theta(y_i^+ \mid x_i) - \nabla_\theta s_\theta(y_i^- \mid x_i)$, so that $\nabla m_\theta(Z_i) = \beta d_i(\theta)$. The Gauss–Newton contribution is

$$G_i(\theta) = \beta^2 \sigma(m_i)\sigma(-m_i) d_i(\theta) d_i(\theta)^\top, \qquad m_i = m_\theta(Z_i), \tag{21}$$

but the full empirical Hessian also contains the nonlinear-margin term:

$$\begin{aligned} \widehat{J}_n(\theta) &= \nabla^2 \widehat{\mathcal{L}}_n(\theta) = \widehat{H}_n(\theta) + \widehat{R}_n(\theta), \\ \widehat{H}_n(\theta) &= \frac{1}{n}\sum_i G_i(\theta), \qquad \widehat{R}_n(\theta) = -\frac{1}{n}\sum_i \sigma(-m_i)\nabla^2 m_\theta(Z_i). \end{aligned} \tag{22}$$

The remainder need not be positive semidefinite. Hence positive Gauss–Newton curvature alone is insufficient for the following local estimation argument.

Theorem 1 (Local estimation under full curvature)**.** *Let $\theta^\star$ be a fixed interior stationary point of $\mathcal{L}$ in a convex neighborhood $\mathcal{U}$, with $\mathbb{E}_Q \nabla \ell(\theta^\star; Z) = 0$ and $\|\nabla \ell(\theta^\star; Z)\|_2 \le B$ almost surely. Assume the sample losses are twice continuously differentiable on $\mathcal{U}$. For any $\delta \in (0, 1)$, with probability at least $1 - \delta$, every interior empirical stationary point $\widehat{\theta}_n \in \mathcal{U}$ for which*

$$\overline{J}_n := \int_0^1 \widehat{J}_n(\theta^\star + t(\widehat{\theta}_n - \theta^\star))\, dt \succeq \kappa I_d, \quad \kappa > 0, \tag{23}$$

satisfies

$$\|\widehat{\theta}_n - \theta^\star\|_2 \le \frac{2B}{\kappa}\sqrt{\frac{2\{d \log 5 + \log(2/\delta)\}}{n}}. \tag{24}$$

The probability controls the empirical score; the curvature condition and existence of a local stationary point are additional requirements, not conclusions. Bounded score gaps $\|d_i(\theta^\star)\|_2 \le G$ give $B = \beta G$. Appendix E.4 proves the result using the integrated *full* Hessian and gives explicit sufficient conditions involving population Gauss–Newton information and a bound on the remainder. For fixed positive curvature the bound decreases as $n^{-1/2}$; its curvature dependence is $1/\kappa$.

Interpreting informative comparisons. The Gauss–Newton term reflects both coverage of score-gap directions and the logistic weight $\sigma(m)(1-\sigma(m))$, which vanishes for large $|m|$. These describe potentially useful comparison geometry, but do not imply full curvature without controlling $\widehat{R}_n$. Also, $\mathrm{rank}(\widehat{H}_n) \le n$, so positive definiteness is not automatic in a high-dimensional neural parameterization. Applying the bound to a particular model requires an identifiable local parameterization and verification of the curvature conditions. Mean teacher–student KL alone does not establish them.

E.2 Token-Level Gradient

The token-level effect of this objective is explicit from its gradient. For a sampled triple (x_i, y_i^+, y_i^-), define

$$\ell_i(\theta) = -\log \sigma\big(m_\theta(x_i, y_i^+, y_i^-)\big). \tag{25}$$

Since the teacher policy is fixed, differentiating only through the student policy gives

$$\begin{aligned}\nabla_\theta \ell_i(\theta) &= -\beta\sigma\big(-m_\theta(x_i, y_i^+, y_i^-)\big)\left[\nabla_\theta s_\theta(y_i^+ \mid x_i) - \nabla_\theta s_\theta(y_i^- \mid x_i)\right] \\ &= -\beta\sigma\big(-m_\theta(x_i, y_i^+, y_i^-)\big)\left[\frac{1}{|y_i^+|}\sum_{k=1}^{|y_i^+|}\nabla_\theta \log \pi_\theta(y_{i,k}^+ \mid x_i, y_{i,<k}^+)\right. \\ &\qquad \left. -\frac{1}{|y_i^-|}\sum_{k=1}^{|y_i^-|}\nabla_\theta \log \pi_\theta(y_{i,k}^- \mid x_i, y_{i,<k}^-)\right].\end{aligned} \tag{26}$$

The negative gradient contrasts the positive- and negative-response score vectors, weighted by $\sigma(-m_\theta(x_i, y_i^+, y_i^-))$. This weight increases as the preference margin decreases. Shared model parameters mean that an update need not raise every positive-token probability or lower every negative-token probability.

E.3 Details for Online PBSD Objective

Gradient derivation for Eq. (26). Since the teacher policy $\pi^{\mathrm{teach}}(\cdot \mid x)$ is fixed, differentiating the PBSD margin in Eq. (3) gives

$$\nabla_\theta m_\theta(x_i, y_i^+, y_i^-) = \beta\left(\nabla_\theta s_\theta(y_i^+ \mid x_i) - \nabla_\theta s_\theta(y_i^- \mid x_i)\right) = \beta d_i(\theta). \tag{27}$$

By the chain rule,

$$\nabla_\theta \ell_i(\theta) = \frac{d}{dm}\big[-\log\sigma(m)\big]\Big|_{m=m_\theta(x_i,y_i^+,y_i^-)} \nabla_\theta m_\theta(x_i, y_i^+, y_i^-). \tag{28}$$

Using $\sigma'(m) = \sigma(m)(1-\sigma(m))$, we obtain

$$\frac{d}{dm}\big[-\log\sigma(m)\big] = -\frac{\sigma'(m)}{\sigma(m)} = -(1-\sigma(m)) = -\sigma(-m). \tag{29}$$

Substituting Eq. (27) and Eq. (29) into Eq. (28) yields

$$\nabla_\theta \ell_i(\theta) = -\beta\sigma\big(-m_\theta(x_i, y_i^+, y_i^-)\big)\, d_i(\theta),$$

which is exactly the sample-level form underlying Eq. (26).

Local Hessian derivation for Eq. (22). Write

$$\phi(m) := -\log\sigma(m) = \log(1+\exp(-m)).$$

Its derivatives are

$$\phi'(m) = -\sigma(-m), \qquad \phi''(m) = \sigma(m)\sigma(-m) = \sigma(m)(1-\sigma(m)).$$

By the chain rule,

$$\begin{aligned}\nabla_\theta^2 \ell_i(\theta) = \phi''\big(m_\theta(x_i, y_i^+, y_i^-)\big)\,\nabla_\theta m_\theta(x_i, y_i^+, y_i^-)\nabla_\theta m_\theta(x_i, y_i^+, y_i^-)^\top \\ +\phi'\big(m_\theta(x_i, y_i^+, y_i^-)\big)\,\nabla_\theta^2 m_\theta(x_i, y_i^+, y_i^-).\end{aligned} \tag{30}$$

The first term is the Gauss–Newton component $G_i(\theta)$. The full identity is

$$\nabla_\theta^2 \ell_i(\theta) = G_i(\theta) - \sigma(-m_\theta(Z_i))\nabla_\theta^2 m_\theta(Z_i).$$

The second term vanishes for an affine margin but generally not for a neural policy. Averaging both terms gives Eq. 22; the proof below retains the remainder.

E.4 Details for Local Estimation and Comparison Geometry

Statistical setting and scope. Fix a distribution Q of ordered teacher–student comparisons before observing the sample. One possible construction draws x from a prompt distribution, y^+ from a frozen contextual teacher, and y^- from a fixed student behavior policy. The loss is $\ell(\theta; Z) = -\log \sigma(m_\theta(Z))$. We study its population risk $\mathcal{L}(\theta) = \mathbb{E}_Q \ell(\theta; Z)$ and assume a fixed interior stationary point $\theta^\star$ exists. Differentiation and expectation must be interchangeable there, so $\mathbb{E}_Q \nabla \ell(\theta^\star; Z) = 0$. These assumptions do not follow from assigning the teacher response the positive label.

The d-dimensional parameterization is fixed independently of the sample. It can represent a specified local subspace, but we do not identify such a subspace for the empirical LoRA runs or claim positive curvature in their full raw parameter space. The theorem concerns a local empirical stationary point within a convex neighborhood; it does not establish that training reaches that point. No correct specification of latent preference labels is needed to bound estimation of this surrogate stationary point, and such estimation is not a guarantee of improved true reward.

A bounded-score concentration event. Let $\psi_i = \nabla \ell(\theta^\star; Z_i)$ and $g_n = n^{-1} \sum_i \psi_i$. The assumptions give independent mean-zero vectors with $\|\psi_i\|_2 \le B$. In particular, $\psi_i = -\beta \sigma(-m_{\theta^\star}(Z_i)) d_i(\theta^\star)$, so bounded score gaps give $B = \beta G$.

Take a Euclidean $1/2$-net $\mathcal{N}$ of the unit sphere with $|\mathcal{N}| \le 5^d$. For a fixed $u \in \mathcal{N}$, scalar Hoeffding concentration yields

$$\Pr\{|u^\top g_n| > t\} \le 2 \exp\left(-\frac{n t^2}{2B^2} \right).$$

A union bound over the net, together with $\|g_n\|_2 \le 2 \max_{u \in \mathcal{N}} |u^\top g_n|$, gives the event

$$\|g_n\|_2 \le \varepsilon_n(\delta) := 2B \sqrt{\frac{2\{d \log 5 + \log(2/\delta)\}}{n}} \tag{31}$$

with probability at least $1 - \delta$. This event depends on the score at fixed $\theta^\star$, independently of the chosen empirical stationary point.

Proof of Theorem 1. Let $\Delta = \widehat{\theta}_n - \theta^\star$. Twice continuous differentiability and convexity of the neighborhood give the exact integral identity

$$0 = \nabla \widehat{\mathcal{L}}_n(\widehat{\theta}_n) = g_n + \left[\int_0^1 \nabla^2 \widehat{\mathcal{L}}_n(\theta^\star + t\Delta)\, dt \right] \Delta = g_n + \overline{J}_n \Delta.$$

This uses the full Hessian and does not require a single intermediate point to work for all gradient coordinates. On a realization where $\overline{J}_n \succeq \kappa I_d$, its inverse has operator norm at most $1/\kappa$. Consequently,

$$\|\Delta\|_2 = \|\overline{J}_n^{-1} g_n\|_2 \le \kappa^{-1} \|g_n\|_2 \le \frac{\varepsilon_n(\delta)}{\kappa},$$

which is the stated bound on the score event. If the local-stationarity and curvature requirements hold with probability at least $1 - \delta_{\mathrm{loc}}$, a union bound gives the complete conclusion with probability at least $1 - \delta - \delta_{\mathrm{loc}}$. We do not assert that this local event occurs automatically.

Connecting Gauss–Newton information to full curvature. At $\theta^\star$, define

$$G^\star(Z) = \beta^2 \sigma(m_{\theta^\star}(Z)) \sigma(-m_{\theta^\star}(Z)) d(\theta^\star; Z) d(\theta^\star; Z)^\top,$$

$$H^\star = \mathbb{E}_Q G^\star(Z), \qquad \widehat{H}_n^\star = \frac{1}{n} \sum_i G^\star(Z_i).$$

Assume $\|d(\theta^\star; Z)\|_2 \le G$ almost surely and $\lambda_{\min}(H^\star) \ge \mu > 0$. These assumptions imply $0 \preceq G^\star(Z) \preceq K I_d$, with $K = \beta^2 G^2/4$.

Lemma 1 (Empirical information under a frozen comparison distribution)**.** *Suppose* $n \ge 8(K/\mu) \log(d/\delta_H)$. *Then the following holds with probability at least* $1 - \delta_H$*:*

$$\widehat{H}_n^\star \succeq \tfrac{1}{2} H^\star, \qquad \lambda_{\min}(\widehat{H}_n^\star) \ge \mu/2.$$

Proof. Set $Y_i = (H^\star)^{-1/2} G^\star(Z_i)(H^\star)^{-1/2}$. The matrices are independent and positive semidefinite, $\mathbb{E}Y_i = I_d$, and $\lambda_{\max}(Y_i) \le K/\mu$. The matrix Chernoff lower-tail bound (Tropp, 2012) gives

$$\Pr\left\{\lambda_{\min}\left(\frac{1}{n}\sum_i Y_i\right) < \frac{1}{2}\right\} \le d\exp\Big(-\frac{n\mu}{8K}\Big) \le \delta_H.$$

Conjugating by $(H^\star)^{1/2}$ proves the claim.

This lemma controls $\widehat{H}_n$ at one point only. For a local estimation conclusion, an additional uniform condition is needed. Suppose that, on a local event, an interior empirical stationary point lies in $\mathcal{U}$, and for all $\theta \in \mathcal{U}$,

$$\widehat{H}_n(\theta) \succeq c_0 \widehat{H}_n^\star, \qquad \|\widehat{R}_n(\theta)\|_{\mathrm{op}} \le c_0\mu/4, \qquad c_0 > 0. \tag{32}$$

On the intersection with the lemma's event,

$$\widehat{J}_n(\theta) = \widehat{H}_n(\theta) + \widehat{R}_n(\theta) \succeq (c_0\mu/2 - c_0\mu/4) I_d = (c_0\mu/4) I_d.$$

Thus the integrated full Hessian has $\kappa = c_0\mu/4$.

Theorem 2 (Conditional vanishing local bound)**.** *Under the bounded-score assumptions above and the sample-size condition of Lemma 1, with probability at least $1 - \delta_S - \delta_H$, every realization also satisfying the local event in Eq. 32 obeys*

$$\|\widehat{\theta}_n - \theta^\star\|_2 \le \frac{4\varepsilon_n(\delta_S)}{c_0\mu}.$$

If the local event has probability at least $1 - \delta_{\mathrm{loc}}$, the unconditional success probability is at least $1 - \delta_S - \delta_H - \delta_{\mathrm{loc}}$.

Proof. Intersect the score event with the matrix-concentration event, apply the full-curvature bound just derived, and use Theorem 1. The union bound does not require these events to be independent.

For fixed d, B, c_0, μ and failure levels, the right-hand side is of order $n^{-1/2}$, with a $1/\mu$ curvature dependence. A consistency claim additionally requires the local event to occur with probability tending to one. Neither lower mean teacher–student KL nor bounded response lengths establishes Eq. 32. Each G_i has rank at most one, so $\mathrm{rank}(\widehat{H}_n) \le \min(n, d)$; the positive-information condition cannot be assumed for an arbitrary overparameterized neural model. These results describe a conditional frozen-distribution surrogate problem, not the full adaptive PBSD algorithm.

F Appendix for Experiment

This section provides the detailed experimental design underlying the results in Section 4. Our implementation follows the OPSD protocol of Zhao et al. (2026a) whenever applicable to keep applicable settings comparable; the comparison evaluates the combined training procedures rather than isolating an individual component. The appendix is organized as follows. Appendix F.1 summarizes the datasets used in the mathematical reasoning and tool-use experiments. Appendix F.2 visualizes representative examples from the processed training data. Appendix F.3 describes the shared setup and training configurations. Appendix F.4 describes the evaluation protocols for math and tool use. Appendix F.5 presents the case-by-case study design. Appendix F.6 discusses the scope of output-behavior diagnostics. Appendix F.7 reports the teacher update frequency ablation.

F.1 Datasets

We experiment with Qwen3 instruct models at three scales: Qwen3-1.7B, Qwen3-4B, and Qwen3-8B. Across the main paper, we study two task domains: mathematical reasoning and tool use.

Mathematical reasoning data. Following the setup described in Section 4.1, we train on the mathematical reasoning subset of OpenThoughts (Guha et al., 2025), from which we sample up to 30K problem–solution pairs that contain chain-of-thought reasoning traces. This domain shares the same training pipeline as the tool-use experiments; the training data, evaluation benchmarks, and metrics are task-specific. We evaluate on three competition-level mathematical benchmarks: AIME 2024, AIME 2025, and HMMT 2025. In the main paper, all reported mathematical reasoning numbers use the same Avg@12 evaluation protocol.

Tool-use data. For the additional tool-use study, we follow the setup in Shenfeld et al. (2026), which uses ToolAlpaca as the underlying domain. Each example consists of a user query together with tool or API information, and the model must generate the correct tool call. As in mathematical reasoning, we use the same general training pipeline with task-specific training data, evaluation split, and metric. In our processed split, the training set contains 4046 examples and the test set contains 94 examples. This benchmark is used to evaluate whether the benefit of demonstration-conditioned self-distillation extends beyond long-form reasoning to structured action generation.

F.2 Illustrative Data Records

The examples below illustrate the fields of mathematical reasoning and tool-use records. They show task content and reference outputs, rather than serialized chat prompts or the teacher–student input split.

Mathematical reasoning example 1

Problem. Let a, b, c be positive integers such that $a + b + c = 12$. How many ordered triples (a, b, c) satisfy $abc = 27$?

Reference solution. Since $abc = 27 = 3^3$, each variable must be a power of 3. The positive triples with product 27 are permutations of $(1, 1, 27)$, $(1, 3, 9)$, and $(3, 3, 3)$. Their sums are 29, 13, and 9, respectively. None satisfies $a + b + c = 12$, so there are no such ordered triples.

Final answer. 0.

Mathematical reasoning example 2

Problem. Find the remainder when 7^{100} is divided by 13.

Reference solution. By Fermat's little theorem, $7^{12} \equiv 1 \pmod{13}$. Since $100 = 8 \cdot 12 + 4$, we have $7^{100} \equiv 7^4 \pmod{13}$. Moreover, $7^2 \equiv 10 \pmod{13}$ and $7^4 \equiv 10^2 \equiv 9 \pmod{13}$.

Final answer. 9.

Tool-use example 1

User request. Book a flight from San Francisco to Seattle next Tuesday after 6pm.

Tool specification.
`search_flights`: Search available flights between two cities on a date.
Arguments (all strings): `origin`, `destination`, `date`, `departure_time_after`.

Reference tool. `search_flights`
Reference arguments.

```
{
  "origin": "San Francisco",
  "destination": "Seattle",
  "date": "next Tuesday",
  "departure_time_after": "18:00"
}
```

Tool-use example 2

User request. Set a reminder to call Alex tomorrow at 9am.

Tool specification.
`create_reminder`: Create a reminder with a title and scheduled time.
Arguments (both strings): `title`, `time`.

Reference tool. `create_reminder`
Reference arguments.

```
{
  "title": "Call Alex",
  "time": "tomorrow 09:00"
}
```

F.3 Detailed Training Configuration

LoRA Hu et al. (2022) is a powerful tool for parameter efficient fine-tuning and has been widely investigated recently (Zhao et al., 2026b; Yu et al., 2025b;c). Here, all methods are fine-tuned with LoRA on 8 H100 GPUs. This training configuration is shared across both task domains; unless explicitly stated otherwise, mathematical reasoning and tool use use the same model initialization, optimizer, LoRA setup, rollout configuration, and checkpoint-selection protocol. Across all runs, we use AdamW, bfloat16 precision, gradient checkpointing, and FlashAttention 2. The shared LoRA configuration is rank $r = 64$, LoRA alpha $\alpha = 128$, and target modules `q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, and `down_proj`. To accelerate rollout generation for on-policy methods, we use vLLM for inference. We keep the optimization hyperparameters of the baselines aligned with OPSD (Zhao et al., 2026a) as closely as possible where applicable, while explicitly retaining the methods' different generation requirements. All trainable methods in the main-table comparison are trained for 500 steps and evaluated every 50 steps. The main comparison uses the latest checkpoint at step 500; Appendix F.10 reports designated longer-horizon PBSD diagnostics through step 1,000.

SFT. SFT is trained on the reasoning traces in the training set and can be viewed as an off-policy distillation baseline. We use learning rate 5×10^{-6}, effective batch size 32, LoRA rank 64, LoRA alpha 128, and maximum sequence length 16,000, following the OPSD paper.

GRPO. For GRPO, we follow the OPSD implementation settings: learning rate 5×10^{-6}, effective batch size 32, maximum completion length 16,000, 8 generations per prompt, sampling temperature 1.2, and KL coefficient 0.0. Because GRPO uses group-based rollouts, its token cost scales with the number of sampled responses per prompt.

OPSD. For OPSD, the following hyperparameters follow Zhao et al. (2026a), with the training horizon set separately for our comparison: learning rate 5×10^{-6}, effective batch size 32, maximum completion length 1024, a single on-policy rollout per prompt, and sampling temperature 1.1. Unless otherwise stated, OPSD uses full-vocabulary logit distillation. The source paper selects forward KL and reports 100 OPSD steps; our reported comparison uses 500 steps. As in the original paper, the teacher is the same base model instantiated with privileged reasoning context, and gradients are backpropagated only through the student branch.

DAPO, SDFT, and SRPO. These baselines use our common optimization configuration: learning rate 5×10^{-6}, effective batch size 32, 500 training steps, and the shared LoRA configuration. Method-specific settings follow their respective original papers: DAPO (Yu et al., 2025a), SDFT (Shenfeld et al., 2026), and SRPO (Li et al., 2026a). This convention preserves each method's distinct training mechanism while using the common optimization budget and reporting the latest checkpoint at step 500.

PBSD. PBSD shares the base model, LoRA setup, optimizer, batch size, student rollout count, maximum completion length, and sampling temperature of OPSD; PBSD additionally generates one teacher response per distinct question and reuses it within the run. Concretely, we use learning rate 5×10^{-6}, effective batch size 32, maximum completion length 1024, one online student response paired with a stored teacher response per prompt, and sampling temperature 1.1. We set the preference scale to $\beta = 0.1$ throughout the default PBSD runs; this is distinct from the rollout sampling temperature. Relative to OPSD, PBSD replaces token-level KL matching with a length-normalized pairwise loss and adds a generated teacher response; no auxiliary SFT or language-model loss is used. Teacher and student responses truncated at the 1,024-token cap are retained and scored over the available generated sequence, without dropping the pair or resampling because of truncation. In addition, we fix the teacher policy to the initial checkpoint throughout training to avoid teacher drift and to keep the teacher signal stationary while the student evolves on-policy.

For OPSD and PBSD, the teacher and student are instantiated from the same underlying model but receive different contexts. The student conditions only on the original problem, while the teacher is augmented with privileged information derived from the training example. In all on-policy methods, the student generates the rollout that defines the training state distribution. For OPSD, this rollout is used for token-level KL matching against the teacher distribution. For PBSD, the same on-policy

student rollout serves as the negative sample, while the contextual teacher provides the positive sample used in the pairwise objective.

Preference modeling also appears in adjacent settings, including listwise optimization for diffusion models (Bai et al., 2025) and attribute-guided decoding-time personalization (Yu et al., 2026). These are distinct applications, not evidence for the effectiveness of the PBSD training construction.

The comparison evaluates complete training procedures. PBSD and OPSD share the optimizer and student-sampling parameters listed above; PBSD additionally generates teacher responses. Thus, their comparison changes both the loss and response construction. All trainable methods in the main-table comparison are trained for 500 steps and evaluated every 50 steps. The main comparison reports the latest checkpoint at step 500 for each trainable method. Intermediate evaluations track training progress. Appendix F.10 reports designated PBSD continuations from step 500 to 1,000.

F.4 Detailed Evaluation Configuration

Mathematical reasoning evaluation. For the main-table results, we repeat evaluation with three sampling seeds at each fixed checkpoint and report the mean and standard deviation across these evaluation repeats. These repeats quantify evaluation-sampling variability, rather than variability across independently trained models.

The numerical sampling parameters below follow Table 8 of Zhao et al. (2026a). Mathematical reasoning evaluation uses thinking mode (`thinking=True`) for both the base model and all trained methods. We sample 12 responses per prompt with temperature 1.0. For a benchmark with N questions, Avg@12 is $100(12N)^{-1}\sum_{i=1}^{N}\sum_{j=1}^{12} a_{ij}$, where $a_{ij} \in \{0, 1\}$ indicates whether response j to question i is correct. The Math Avg. column is the unweighted arithmetic mean of AIME24, AIME25, and HMMT25 scores; tool-use accuracy is reported separately. The maximum number of generated tokens per response is 38,912. We use top-$p = 0.95$, top-$k = -1$, min-$p = 0.0$, and zero presence penalty. All methods share these mathematical decoding settings.

Enhanced coding tests. For HumanEval and MBPP, we report results on the EvalPlus enhanced test suites (Liu et al., 2023), denoted HumanEval+ and MBPP+, respectively. These names refer to the enhanced-test scores in both the domain-specific training results (Table 8) and the OOD evaluation of the math-trained checkpoint (Table 10).

Tool-use evaluation. For tool use, evaluation is performed on the 94-example test split described in Appendix F.1. We report top-1 accuracy on the generated tool call. The general training configuration is shared with mathematical reasoning, while the training data and evaluation protocol are task-specific: tool use is evaluated on its own test split and metric rather than Avg@12 on math benchmarks. The comparison protocol mirrors our main mathematical reasoning experiments: we evaluate the base model and compare SFT, GRPO, DAPO, OPSD, SDFT, SRPO, and PBSD under matched optimization settings whenever possible, as a comparison of the full training procedures. As in the mathematical reasoning experiments, PBSD uses the same base model as both teacher and student, with the teacher instantiated through additional contextual information and the student trained on-policy. This setup allows us to test whether the complete comparison procedure remains beneficial when the desired output is a correct tool-use action sequence rather than long-form reasoning.

F.5 Case-by-Case Study

In addition to aggregate benchmark accuracy, we conduct a case-by-case study on AIME25. For 30 AIME25 questions, we record whether each method's majority-vote prediction over sampled responses is correct. This problem-level majority-vote diagnostic is distinct from the sample-average accuracy (Avg@12) in Table 2.

The study is organized as a grid-style comparison. Each row corresponds to one evaluation setting and each column corresponds to one selected AIME25 problem. The rows include: (1) the base student model prompted only with the original question, (2) the same base model prompted with a reference solution as an expert demonstration, (3) OPSD, and (4) PBSD. Each cell is marked with a check if the majority-vote answer is correct and a cross otherwise.

The teacher row is an oracle-context diagnostic: it receives the evaluated problem's reference solution. Its perfect score illustrates access to that solution, not independently validated preference-label quality. The rows compare different information conditions and should not be read as a fair teacher-versus-student capability ranking. Table 6 reports the full case-by-case study.

Table 6: Case-by-case analysis on 30 selected AIME25 problems. "Base (Student)" denotes the base model prompted only with the original question, while "Base (Teacher)" denotes the same base model prompted with the reference solution as an expert demonstration. Each cell is marked with ✓ if the majority-vote answer is correct and ✗ otherwise.

Method	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16	Q17	Q18	Q19	Q20	Q21	Q22	Q23	Q24	Q25	Q26	Q27	Q28	Q29	Q30
Base (Student)	✓	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✓	✗	✗	✓	✓	✗	✓	✓	✓	✓	✓	✓	✗	✗
Base (Teacher)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
OPSD	✓	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗
PBSD	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✗

F.6 Scope of Output-Behavior Diagnostics

Output length and lexical self-checking frequency describe different aspects of generated text. Neither alone measures reasoning quality or calibrated uncertainty. We use the reported keyword-frequency comparison in Appendix F.14 only as an exploratory behavioral diagnostic, alongside task accuracy. The case-by-case results above concern a selected set of problems rather than an independent estimate of overall performance.

F.7 Teacher Update Frequency Study

In the main experiments, we keep the teacher fixed to the initial checkpoint throughout PBSD training. To test whether a more adaptive teacher could further improve learning, we also test refreshing the teacher from the current student every 5 training steps.

This comparison evaluates two teacher update rules on Qwen3-4B using the mathematical training and evaluation setup. The default row reuses the updated main-table results; the periodic-update row retains its separately reported results. The fixed-teacher variant uses the initial context-augmented model as the teacher for the entire run, while the refreshed-teacher variant periodically replaces the teacher with the latest student checkpoint and then re-instantiates the contextual teacher from that checkpoint. Table 7 reports the benchmark-wise comparison on AIME24, AIME25, and HMMT25, together with the average across the three tasks.

Table 7: Teacher update frequency ablation on Qwen3-4B. "Fixed" denotes the default PBSD setting in which the teacher remains equal to the initial checkpoint throughout training. "Update every 5 steps" denotes a variant in which the teacher is refreshed from the student every 5 gradient-update steps. The fixed-teacher row reuses the default PBSD mean scores from Table 2 at step 500.

Teacher Update Rule	AIME24	AIME25	HMMT25	Math Avg.
Fixed teacher	78.3	69.7	45.6	64.5
Update every 5 steps	77.2	68.6	44.4	63.4

F.8 Sensitivity to Privileged Context

Because PBSD relies on a contextual teacher, it is important to understand how sensitive the method is to the exact form of privileged information. We therefore compare three variants of the teacher context on Qwen3-4B mathematical reasoning: (1) ground-truth reasoning together with the answer, which is the default setting in the main paper; (2) ground-truth answer only; and (3) noisy reasoning together with the ground-truth answer, where Qwen3-32B generates a reasoning trace conditioned on the question and its ground-truth answer. This synthetic trace is not guaranteed to be correct. The default privileged information comes from the reference solutions already present in the training dataset, following the OPSD protocol; it is not an extra annotation constructed only for PBSD. The student receives only the original question, including at evaluation time.

Table 4 in the main text reports these results.

Among the three evaluated context configurations, reference reasoning with the correct answer scores highest on all three benchmarks. Relative to this default context, answer-only conditioning scores 2.1, 1.8, and 0.5 points lower on AIME24, AIME25, and HMMT25; synthetic reasoning with the answer scores 1.8, 1.2, and 0.3 points lower. All three conditions retain the ground-truth answer. This comparison does not test incorrect answers or verify individual teacher–student preferences.

F.9 Training on Coding and Instruction-Following Tasks

To test whether PBSD extends beyond mathematical reasoning and tool use, we train separate Qwen3-4B models for coding and instruction following and compare SFT, GRPO, OPSD, and PBSD under matched settings. Coding training uses `TIGER-Lab/AceCode-87K`, with evaluation on the coding benchmarks in Table 8, including LiveCodeBench (Jain et al., 2024). Instruction-following training uses `allenai/RLVR-IFEval`, with evaluation on `google/IFEval` (Zhou et al., 2023). These are domain-specific training experiments, distinct from the OOD evaluation of a math-trained checkpoint in Appendix F.11.

Table 8: Coding and instruction-following results for separately trained Qwen3-4B models. HumanEval+ and MBPP+ use the EvalPlus enhanced test suites.

Method	HumanEval+	MBPP+	LiveCodeBench	IFEval
Base (Instruct)	68.29	77.22	53.88	83.18
+ SFT	69.42	78.84	57.34	86.14
+ GRPO	70.43	**81.13**	62.82	87.06
+ OPSD	70.23	80.63	61.20	87.34
+ PBSD	**71.83**	80.87	**63.40**	**88.54**

The reported scores provide evidence in two additional training domains. PBSD achieves the strongest performance on HumanEval+, LiveCodeBench, and IFEval, and is close to the best MBPP+ score.

F.10 Longer-Horizon Training and Checkpoint Selection

Table 9 uses the step-500 AIME25 results from Table 2 and reports the available scores at steps 600–1,000. This diagnostic extends PBSD to 1,000 steps; the main comparison uses 500.

Table 9: AIME25 checkpoint results for PBSD. Step-500 scores match Table 2.

Model	Step 500	Step 600	Step 700	Step 800	Step 900	Step 1000
Qwen3-8B	71.0	71.1	71.4	71.8	71.3	71.6
Qwen3-4B	69.7	68.5	68.9	69.1	68.3	69.5
Qwen3-1.7B	46.4	44.1	44.6	44.5	43.8	44.3

Extending training beyond step 500 does not improve all model scales. Qwen3-8B reaches 71.8 at step 800, 0.8 points above its step-500 score. Qwen3-4B scores 68.3–69.5 at steps 600–1,000, below its step-500 score of 69.7; Qwen3-1.7B scores 43.8–44.6, below 46.4 at step 500. Longer training therefore does not improve performance uniformly across scales.

F.11 OOD Performance and KL-to-Base Diagnostics

To check whether improving mathematical reasoning causes catastrophic forgetting, we evaluate the math-trained Qwen3-4B PBSD checkpoint on a set of out-of-domain benchmarks and compare it against the corresponding base model. Scientific-reasoning evaluations use GPQA-Diamond (Rein et al., 2023) and MMLU-Pro-STEM (Wang et al., 2024).

On five of the six OOD metrics, PBSD and base scores differ by at most 0.8 percentage points. On MBPP+, accuracy decreases from 77.22 to 72.49, a decline of 4.73 percentage points. Thus,

Table 10: OOD performance of the math-trained Qwen3-4B PBSD checkpoint and base model. HumanEval+ and MBPP+ use the EvalPlus enhanced test suites.

Field	Benchmark	Metric	Base	PBSD
Scientific reasoning	GPQA-Diamond	Avg@4	47.73	47.22
	MMLU-Pro-STEM	Acc	63.03	63.63
Coding	HumanEval+	Pass@1	68.29	68.90
	MBPP+	Pass@1	77.22	72.49
	LiveCodeBench	Pass@1	53.88	53.12
Instruction-following	IFEval	Strict/Loose avg.	83.18	83.32

mathematical training retains similar scores on most evaluated tasks but reduces performance on this coding benchmark. As an additional diagnostic, the token-level KL divergence between the math-trained PBSD checkpoint and the base checkpoint on OOD data is 0.08, reported as an auxiliary diagnostic rather than a guarantee of OOD capability preservation.

F.12 β Sensitivity

In the implemented pairwise loss, β scales the log-ratio margin. In the reward-regularized interpretation, it is the coefficient of the teacher-reference KL term. Table 11 reports Qwen3-4B sensitivity.

Table 11: β sensitivity on Qwen3-4B mathematical reasoning.

β	AIME24	AIME25	HMMT25
0.02	77.1	68.9	46.4
0.05	77.4	68.2	46.4
0.1 (default)	78.3	69.7	45.6
0.3	76.3	68.5	45.1
0.5	76.3	68.2	44.8

In this Qwen3-4B comparison, the default $\beta = 0.1$ has the highest reported AIME24 and AIME25 scores, while $\beta = 0.02$ and 0.05 improve HMMT25 by 0.8 points over the default. We use $\beta = 0.1$ as a fixed, task-independent default rather than tune it separately on each benchmark. We assess sensitivity through comparisons within this sweep, which reuses the default PBSD scores of 78.3/69.7/45.6 from Table 2. In the idealized reward objective, β weights the KL penalty; in the implemented loss it scales the margin and affects both gradient magnitude and sigmoid saturation. Thus the reward interpretation does not imply monotonic teacher proximity as β varies during PBSD training. The benchmark-wise tradeoff does not establish a universally optimal preference scale.

F.13 Comparison to Larger Teachers and Teacher–Student KL

We compare PBSD with a Qwen3-4B contextual self-teacher against generalized knowledge distillation (GKD) (Agarwal et al., 2024) using a Qwen3-14B external teacher, with a Qwen3-4B student in both cases. This comparison changes both the distillation objective and teacher configuration, so it is not an isolated teacher-size ablation.

Table 12: Contextual self-teacher versus larger external teacher on Qwen3-4B. The PBSD row reuses the default mean scores from Table 2 at step 500.

Teacher Setting	AIME24	AIME25	HMMT25
PBSD: Qwen3-4B + context c	78.3	69.7	45.6
GKD: Qwen3-14B teacher	76.4	69.2	45.3

PBSD with a Qwen3-4B contextual self-teacher obtains higher reported scores than GKD with a Qwen3-14B teacher on the three evaluated mathematical benchmarks. Because both the teacher configuration and training objective differ, this comparison evaluates the complete training setups

rather than teacher size alone. It does not establish that a smaller teacher is generally preferable or that context is more important than model scale.

Teacher–student distribution gap. We separately compare a Qwen3-8B student without privileged context to teachers from the same model family. Table 13 reports the token-level KL measurements. The contextual self-teacher is closer than both larger teachers with the same context and larger teachers without context. Under the reported calculation, the self-teacher has the smallest value. These scalars do not establish the full-curvature conditions in Appendix E.1 or universally rank teacher quality.

Table 13: Teacher–student KL for a Qwen3-8B student without privileged context.

Teacher	Teacher context	Token-level KL
Qwen3-8B (self-teacher)	Privileged context c	0.188
Qwen3-14B	Privileged context c	0.274
Qwen3-32B	Privileged context c	0.315
Qwen3-14B	No privileged context	0.258
Qwen3-32B	No privileged context	0.247

F.14 Response Length and Verbalized Uncertainty Diagnostics

We track mean response length and lexical self-checking in Qwen3-4B mathematical reasoning across training checkpoints. For the length diagnostic, we generate responses from the saved student checkpoints with a maximum of 2,048 new tokens for both PBSD and OPSD. This diagnostic is separate from training, where completions are capped at 1,024 tokens. Table 14 reports mean generated-response length, excluding prompt tokens.

Table 14: Mean generated-response length for Qwen3-4B mathematical reasoning under a shared 2,048-token completion limit. Prompt tokens are excluded.

Method	Step 0	Step 200	Step 400	Step 600
PBSD	1388	1369	1333	1352
OPSD	1398	1222	1214	1234

PBSD's mean response length ranges from 1,333 to 1,388 tokens across the measured checkpoints, ending at 1,352 tokens at step 600. OPSD decreases from 1,398 to 1,234 tokens over the same interval. Thus, PBSD exhibits less response shortening in this diagnostic. Length characterizes output behavior and is interpreted alongside accuracy and lexical measurements.

We compare hedging and self-checking expression frequencies in Qwen3-4B mathematical-reasoning outputs at selected training checkpoints. Table 15 reports occurrences per 1,000 generated tokens using a 22-expression lexicon that includes "maybe", "wait", and "let me check". This exploratory lexical diagnostic uses thinking-mode outputs; its scope is discussed in Appendix A.

Table 15: Hedging/self-checking expression frequency per 1,000 generated tokens.

Method	Step 0	200	400	600	800	1000
PBSD	1.58	1.66	1.46	1.51	1.44	1.47
OPSD	1.58	1.66	0.96	0.75	0.78	0.74

PBSD's reported expression frequency remains between 1.44 and 1.66 per 1,000 tokens; OPSD declines to 0.74 at step 1,000. These lexical measurements show less suppression of hedging and self-checking in this setting. Privileged context is provided only to the teacher; excluding it from student evaluation avoids direct test-time hints but does not by itself establish absence of all forms of information leakage.

F.15 Interpreting the Tool-Use Gains

A possible explanation for the tool-use gains is that correctness depends on discrete sequence-level decisions, including tool selection, argument construction, and call ordering, while multiple surface forms can be valid. Token-level KL matching may penalize benign variation, whereas PBSD compares teacher- and student-generated sequences. Privileged tool/API information may also provide particularly useful guidance for these decisions. These explanations are hypotheses consistent with the results, not causal findings established by a component ablation.

G Component-Comparison Details

All four variants use the same training questions and contextual-teacher configuration, including the full PBSD run whose results are reused from Table 2. Teacher responses are generated independently in each run. Within a run, a repeated question reuses the same teacher response; responses are not shared across the four variants. Each variant is trained for 500 steps with effective batch size 32 and learning rate 5×10^{-6}. They share the LoRA configuration in Appendix F.3: rank 64, alpha 128, and the same target modules.

Table 3 compares four Qwen3-4B variants with the shared configuration described above. Teacher-response SFT trains the student to imitate the contextual teacher responses using the original task input. Static-pair DPO uses its own teacher positives and fixed negatives generated by the initial student, with the contextual teacher as reference. PBSD with an unconditioned reference retains contextual teacher positives and online student negatives but uses $q_0(y \mid x) = \pi_{\theta_0}(y \mid x)$ for reference scoring instead of $q_c(y \mid x) = \pi_{\theta_0}(y \mid x, c)$. Full PBSD retains both online negatives and contextual reference scoring. Thus removing context from the reference does not remove it from positive-response generation.

Full PBSD exceeds teacher-response SFT by 5.9, 3.8, and 1.7 percentage points on the respective benchmarks; its corresponding advantages over static-pair DPO are 6.0, 3.5, and 1.8 points, and over PBSD with an unconditioned reference are 6.2, 5.4, and 4.2 points. The comparison supports the complete procedure over these alternatives in this setting. It does not constitute an exhaustive factorial study or establish that either component is necessary across all regimes and scales.

H Teacher–Student Pair Composition at Step 500

This diagnostic characterizes the answer-correctness composition of the role-assigned pairs summarized in Section 4.6. It uses the step-500 Qwen3-4B PBSD student and its frozen contextual teacher.

Sampling and classification. We use 512 training questions with three sampling replicates per question, yielding 1,536 teacher–student pairs. The teacher receives the privileged context and the student receives the original input. The three replicates are response-sampling repetitions, not independent training seeds. Pairs are classified by the correctness of the teacher and student answers. Sixty pairs are excluded because of truncation, leaving 1,476 classified pairs. This exclusion applies only to this answer-correctness diagnostic: training retains truncated responses in the loss. The reported exclusion count is pair-level; it does not specify which response was truncated.

Counts and denominators. Among the classified pairs, 440 have a correct teacher answer and an incorrect student answer, 7 have the opposite ordering, 801 have two correct answers, and 228 have two incorrect answers. The four category proportions in Table 5 use 1,476 as their denominator. Eligibility and truncation rates use all 1,536 pairs. Teacher and student accuracies below are conditional on the same 1,476 classified pairs:

$$\mathrm{Acc}_T = 100\frac{440 + 801}{1476} = 84.08\%, \qquad \mathrm{Acc}_S = 100\frac{7 + 801}{1476} = 54.74\%.$$

These conditional diagnostic accuracies are distinct from benchmark Avg@12.

Interpretation. Answer correctness supports the teacher-first ordering more often than it opposes it in this sample. Both-correct and both-incorrect pairs do not determine preference quality: equal

Table 16: Step-500 pair accounting for Qwen3-4B. Counts, denominators, and rates are shown explicitly. Correct-answer counts are restricted to the classified pairs.

Quantity	Count	Denominator	Rate (%)
Classified pairs	1,476	1,536	96.09
Pairs excluded for truncation	60	1,536	3.91
Correct teacher answers	1,241	1,476	84.08
Correct student answers	808	1,476	54.74

answer correctness need not imply equal response quality. The counts describe pair composition, not logistic-weight allocation or gradient contribution. In particular, a category's share of logistic weight would require summing its observed $w_i = \sigma(-m_i)$ values and dividing by the weight sum over the specified comparison set; it cannot be inferred from counts. The reported diagnostic does not establish that any category causes the benchmark gains.