IP-OPD: LEARNING FROM INTERVENTION PREFERENCES FOR ON-POLICY DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy distillation (OPD) reduces student-state mismatch by supervising students on trajectories they actually visit. Yet state alignment does not guarantee that supervision at a visited state is useful: even a stronger source can provide locally plausible reasoning that worsens the student's downstream outcome. We make this distinction explicit through intervention advantage, defined as the change in expected reward caused by inserting a reasoning block at a student-visited state and then returning control to the student. We introduce Intervention-Preference On-Policy Distillation (IP-OPD), which compares source-proposed and native blocks at sparsely selected on-policy states. Both branches are completed by the same student, and discordant terminal outcomes induce bidirectional block-level preferences. Our analysis connects intervention advantage to the return of an explicit intervention-mixture policy and, for binary verifiable outcomes, shows that the population preference optimum ranks blocks consistently with downstream continuation values. Because supervision is determined by downstream student utility rather than source identity, IP-OPD requires neither privileged source logits nor a learned critic or router and naturally supports external, self-, and heterogeneous proposals. Experiments across five mathematical reasoning benchmarks show consistent performance gains, stronger diagnostic signals of intervention utility than several distributional proxies, and improvements under outcome-information-matched and compute-matched controls.

1 INTRODUCTION

Knowledge distillation (Gou et al., 2021) transfers reasoning through intermediate supervision such as generated sequences or reasoning steps (Wei et al., 2022; Hsieh et al., 2023; Ho et al., 2023; Li et al., 2023). However, teacher-generated trajectories may visit states different from those encountered by the student at inference time (Kim & Rush, 2016; Ross et al., 2011). On-policy distillation (OPD) reduces this mismatch by supervising student-induced states (Agarwal et al., 2024; Gu et al., 2024), with recent extensions refining when or how supervision is applied (Jin et al., 2026; Zheng et al., 2026; Jiang & Ferraro, 2026; Yu et al., 2026). Yet state alignment alone does not determine whether a supplied block helps the student: a stronger teacher can still provide locally unhelpful guidance (Mirzadeh et al., 2020), and a plausible step need not improve the final answer (Uesato et al., 2022; Lightman et al., 2024). Thus, OPD addresses *where* supervision is applied, but not by itself *whether* a concrete intervention improves downstream reasoning from that state.

We capture this state-specific supervision utility through *intervention advantage*: the change in expected reward caused by inserting one reasoning block at a student-visited state and then returning control to the student. Algebraically, this is an advantage over block-level actions; here it serves as a student-conditioned criterion for deciding which local behavior should be distilled. The student remains the reference policy, so external teachers, stronger-inference self-proposals, and heterogeneous models can be evaluated by the same downstream criterion rather than by source identity or nominal strength.

Based on this view, we introduce *Intervention-Preference On-Policy Distillation* (IP-OPD; Figure 1). At sparsely selected on-policy states, IP-OPD compares a source-proposed block with the student's native block. Both branches share the same prefix and are completed by the same frozen student; discordant outcomes induce a sampled block-level preference. Crucially, the source is not assumed

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.

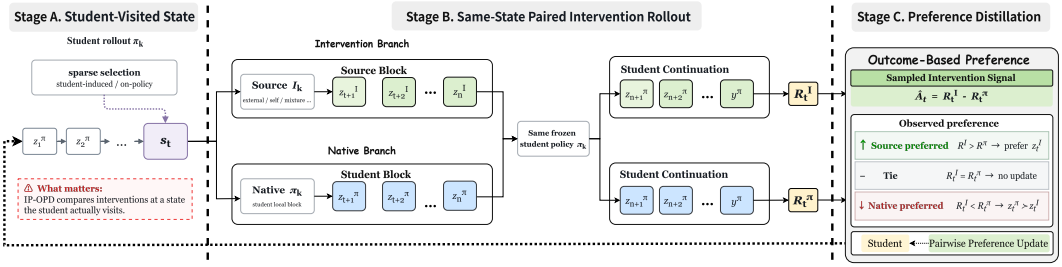


Figure 1: IP-OPD compares a native student block with a source-proposed intervention at the same on-policy state. Both branches are continued by the same student, and their sampled terminal outcomes provide a local block preference.

superior: when the native block yields the better outcome, the preference is reversed. The source supplies candidates, while downstream student utility determines which local behavior is reinforced.

Our analysis separates intervention utility from the learning surrogate. First, we connect local intervention advantage to the change in return under an explicit intervention-mixture policy. Second, for binary verifiable outcomes, we show that the statewise population optimum of the IP-OPD preference objective orders blocks consistently with downstream continuation values; filtering equal-outcome pairs preserves this optimum under the stated conditions. IP-OPD therefore requires no auxiliary critic, source logits, or learned router.

Across five mathematical reasoning benchmarks, IP-OPD improves Macro Avg@8 from 31.4 for EOPD to 35.4, while a compute-matched run reaches 34.0. Under matched outcome information and collection rules, IP-OPD exceeds winner-only imitation by 2.0 macro points after full training and 1.7 at common compute. A separate diagnostic finds that a single paired outcome is more strongly associated with independently estimated intervention utility than several distributional proxies (AUROC 0.79 versus 0.57–0.64), while using additional rollout and reward information. Further experiments examine reversed preferences, native–native comparisons, heterogeneous sources, and computational cost.

Our main contributions are:

- We formulate local OPD supervision as a *student-conditioned intervention problem*, using intervention advantage as a source-identity-agnostic supervision criterion and linking it to an explicit intervention-mixture policy.
- We propose IP-OPD, which compares source-proposed and native blocks at the same student state, completes both with the same student, and learns from bidirectional outcome-induced preferences.
- We characterize the resulting population preference objective and show empirically that IP-OPD improves reasoning under outcome-information-matched and compute-matched controls and across external, self-, mixture, and cross-architecture proposal sources.

2 Related Work

On-policy reasoning distillation. Knowledge and sequence-level distillation transfer teacher predictions or generated sequences (Hinton et al., 2015; Kim & Rush, 2016), while OPD supervises student trajectories to reduce state mismatch (Agarwal et al., 2024). Recent extensions consider privileged self-supervision and multiple teachers (Zhao et al., 2026; Ma et al., 2026). IP-OPD asks a complementary question: for a concrete block proposed at a student-visited state, does inserting it improve the current student's downstream continuation relative to native behavior? The student, rather than the proposal source, is therefore the reference policy for evaluating local supervision.

Selective and utility-aware supervision. Recent OPD methods select, weight, route, or filter supervision using uncertainty, trajectory structure, teacher–student discrepancy, advantages, behavioral differences, or counterfactual relevance (Jin et al., 2026; Zheng et al., 2026; Jiang & Ferraro,

2026; Xie et al., 2026b; Yu et al., 2026; Heo et al., 2026; Li et al., 2026; Jiang et al., 2026); concurrent work also shows that teacher signals can be harmful, misaligned with reasoning progress, or noisy (Armandpour et al., 2026; Yang et al., 2026; Ding & Zhang, 2026). IP-OPD instead directly compares a source block with a native block at the same student-induced state and returns both branches to the same frozen student. The resulting preference is bidirectional, so supervision is determined by downstream student utility rather than source identity, logits, or nominal strength.

Outcome supervision and preference learning. Prior work studies process- and outcome-based reasoning feedback (Uesato et al., 2022; Lightman et al., 2024), while DPO, Step-DPO, and SimPO optimize sequence- or step-level preferences (Rafailov et al., 2023; Lai et al., 2024; Meng et al., 2024). IP-OPD uses a standard logistic preference objective but constructs pairs online: source and native blocks share an on-policy prefix, are completed by the same student, and may be ordered in either direction by terminal outcomes. Under the stated binary-outcome and support conditions, the resulting statewise population objective preserves downstream block-value ordering.

3 Intervention Advantage for On-Policy Distillation

For multi-step reasoning, supervision utility depends on the state and on the student's ability to continue from a supplied block. We formulate local *reasoning interventions* independently of their source's identity and interface.

3.1 Reasoning Intervention

Let π be the current reasoning policy and $x \sim \mathcal{D}$ an input. Reasoning is a finite-horizon process over bounded token spans,

$$\tau = (z_1, \ldots, z_T), \quad T \le H, \qquad s_t = (x, z_{<t}). \tag{1}$$

Each block is a macro-action under a common stopping rule, not necessarily a semantically complete step. A generic source proposes $z_t^I \sim q(\cdot \mid s_t)$ and controls only this block; reasoning then immediately returns to π: $s_t \longrightarrow z_t^I \longrightarrow \pi$. The source may be an external model, the current model under stronger inference, or one member of a heterogeneous source mixture. Its identity is intentionally excluded from the definition below, although its quality continues to matter through the induced proposal distribution $q(\cdot \mid s_t)$.

3.2 Intervention Advantage

Let $R(\tau)$ denote the terminal task reward and

$$J(\pi) = \mathbb{E}_{\tau \sim \pi}[R(\tau)] \tag{2}$$

the expected return of π. The continuation value of state s is

$$V^\pi(s) = \mathbb{E}_{\tau \sim \pi(\cdot|s)}\left[R(\tau)\right], \tag{3}$$

while the value of inserting block z once and then returning to π is

$$Q^\pi(s, z) = \mathbb{E}\left[R(\tau) \mid s,\ z,\ \text{then follow } \pi\right]. \tag{4}$$

Definition 3.1 (Intervention Advantage). *The intervention advantage of block z at state s is*

$$A^\pi(s, z) = Q^\pi(s, z) - V^\pi(s). \tag{5}$$

Thus, $A^\pi(s, z) > 0$ indicates that the intervention improves the expected downstream outcome relative to the policy's native continuation, whereas $A^\pi(s, z) < 0$ identifies a harmful intervention regardless of its source. For a stochastic source q, define

$$\bar{A}^\pi(s; q) = \mathbb{E}_{z \sim q(\cdot|s)}\left[A^\pi(s, z)\right]. \tag{6}$$

The supervision criterion is therefore state-conditional intervention utility rather than source quality.

3.3 Intervention as Local Policy Improvement

Consider a policy that follows q with state-dependent probability $\eta(s) \in [0, 1]$ and otherwise follows π:

$$\pi_{\eta,q}(z \mid s) = \big(1 - \eta(s)\big)\pi(z \mid s) + \eta(s)q(z \mid s). \tag{7}$$

Although $A^\pi(s, z)$ evaluates a single local deviation followed by π, the policy $\pi_{\eta,q}$ may intervene repeatedly; the following result aggregates these local advantages along its induced state distribution.

Theorem 3.2 (**Performance Difference under Interventions**). *For the finite-horizon block-level process above,*

$$J(\pi_{\eta,q}) - J(\pi) = \sum\nolimits_{t=1}^{H} \mathbb{E}_{s_t \sim d_t^{\pi_{\eta,q}}} \left[\eta(s_t)\, \bar{A}^\pi(s_t; q)\right], \tag{8}$$

where $d_t^{\pi_{\eta,q}}$ is the state distribution induced by $\pi_{\eta,q}$ at step t.

This result specializes the standard performance-difference identity (Schulman et al., 2015) to the intervention-mixture policy $\pi_{\eta,q}$. It characterizes the return change under this explicit policy; its relation to the learned surrogate is discussed in Section 4.2. The proof is provided in Appendix A.1.

Corollary 3.3 (**Positive-Advantage Improvement**). *If $\bar{A}^\pi(s; q) \geq 0$ at every state where $\eta(s) > 0$, then*

$$J(\pi_{\eta,q}) \geq J(\pi). \tag{9}$$

The inequality is strict whenever $\eta(s)\bar{A}^\pi(s; q) > 0$ on a set of states with non-zero occupancy under $\pi_{\eta,q}$.

For multiple sources $\{q_m\}_{m=1}^M$, let $q_\alpha(\cdot \mid s) = \sum_{m=1}^M \alpha_m(s) q_m(\cdot \mid s)$, where $\alpha_m(s) \geq 0$ and $\sum_{m=1}^M \alpha_m(s) = 1$. By linearity of expectation, the expected intervention advantage of this state-dependent mixture is

$$\bar{A}^\pi(s; q_\alpha) = \sum\nolimits_{m=1}^{M} \alpha_m(s)\, \bar{A}^\pi(s; q_m). \tag{10}$$

Thus, external, self-, and multi-source proposals share the same definition; Appendix A.2 gives the derivation and the source-identity invariance result.

This analysis formalizes the downstream utility of local supervision and its role in the return of an explicit intervention policy. Section 4 uses paired on-policy outcomes to obtain a noisy estimate of this utility and construct block-level preferences.

4 IP-OPD: Intervention-Preference On-Policy Distillation

IP-OPD converts intervention advantage (Section 3) into block preferences without an auxiliary critic. "On-policy distillation" here denotes student-induced training states with source-proposed local behavior. At each round, sparse source interventions and native blocks are followed by the same student; their outcomes supply fresh pairs for a logistic preference loss.

4.1 On-Policy Paired Interventions

At training round k, we freeze the current model as behavior policy π_k and generate

$$\tau^\pi = (z_1^\pi, \ldots, z_T^\pi) \sim \pi_k(\cdot \mid x), \qquad x \sim \mathcal{D}. \tag{11}$$

As the native trajectory is generated, a small number of non-initial states $s_t = (x, z_{<t}^\pi)$ are selected *online*, before sampling the native block z_t^π or observing any later suffix. Selection is measurable with respect to the current prefix and past selection randomness and is capped by a per-trajectory budget N_s. It never depends on the realized trajectory length or terminal reward. The fixed-horizon hazard used by default is given in Appendix B.2.

For a selected state s_t, the original trajectory itself supplies the *native branch*: z_t^π is the block produced by π_k, and R_t^π is the terminal reward of its native continuation. An intervention source receives only s_t and proposes one bounded block

$$z_t^I \sim q_k(\cdot \mid s_t). \tag{12}$$

The intervention is then handed back to the same behavior policy,

$$\tau_t^I \sim \pi_k(\cdot \mid s_t, z_t^I), \qquad R_t^I = R(\tau_t^I). \tag{13}$$

We estimate the intervention advantage with

$$\hat{A}_t = R_t^I - R_t^\pi. \tag{14}$$

The two branches share the same on-policy prefix and use the same policy for downstream continuation; their suffixes may differ because of sampling stochasticity. For a realized pair, $R_t^I - R_t^\pi$ is therefore a noisy comparison between one concrete source block and one sampled native continuation. Averaging over the native branch recovers

$$\mathbb{E}_{z_t^\pi \sim \pi_k(\cdot|s_t)}\left[Q^{\pi_k}(s_t, z_t^I) - Q^{\pi_k}(s_t, z_t^\pi)\right] = Q^{\pi_k}(s_t, z_t^I) - V^{\pi_k}(s_t) = A^{\pi_k}(s_t, z_t^I).$$

Under the prefix-measurable selection and conditional-independence conditions stated in Appendix B.1, $\hat{A}_t$ is thus an unbiased single-sample estimator of intervention advantage even after conditioning on the state being selected, although any individual preference label remains subject to continuation noise.

4.2 Advantage-Induced Distillation

IP-OPD requires an evaluable terminal task reward in addition to source proposals. For mathematical reasoning, a reference-answer verifier gives $R \in \{0, 1\}$, so $\hat{A}_t \in \{-1, 0, 1\}$. This reward access is additional to distribution-only OPD feedback. Only outcome-discordant pairs, $\hat{A}_t \neq 0$, supply directional supervision.

Because the theory treats a bounded reasoning block as an action, the default objective uses its actual block probability rather than a length-normalized score; a length-normalized robustness check is reported in Appendix E. Under the same bounded block-generation rule used for data collection,

$$\log \pi_\theta(z \mid s) = \sum\nolimits_{j=1}^{|z|} \log \pi_\theta\left(z_j \mid s, z_{<j}\right), \tag{15}$$

including the terminating delimiter when it is part of the generation rule. We define the block-level log-probability margin

$$\Delta_\theta = \log \pi_\theta(z_t^I \mid s_t) - \log \pi_\theta(z_t^\pi \mid s_t), \qquad y_t = \operatorname{sign}(\hat{A}_t). \tag{16}$$

The IP-OPD objective is

$$\mathcal{L}_{\mathrm{IP}}(\theta) = \mathbb{E}_{\mathcal{B}_k}\left[\operatorname{softplus}\left(-\beta y_t \Delta_\theta\right)\right], \tag{17}$$

where $\mathcal{B}_k$ contains the outcome-discordant intervention pairs collected with π_k and $\beta > 0$ controls the pairwise margin scale. If $\hat{A}_t = +1$, IP-OPD favors the intervention block; if $\hat{A}_t = -1$, it instead favors the native block. Equal-outcome pairs are ignored because a single binary rollout does not establish a directional preference.

The logistic objective provides a smooth loss over relative block probabilities, treating either branch as preferred according to its outcome. We next characterize its statewise population optimum.

Statewise global population alignment. The pairwise objective has a globally coherent statewise population solution. Fix a selected state s and a finite comparison support $\mathcal{Z}_s$. Let G_s connect blocks that are compared with positive probability; the formal population risk and graph construction are given in Appendix A.3.

Theorem 4.1 (Global Q-Consistent Population Optimum)**.** *Assume binary terminal rewards $R \in \{0, 1\}$. Suppose that $Q^\pi(s, z) \in (0, 1)$ for all $z \in \mathcal{Z}_s$, branch outcomes are conditionally independent given each compared pair, and G_s is connected. Then the statewise population IP-OPD objective has the unique full-support minimizer*

$$\rho^\star(z \mid s) = \frac{\exp\left(\beta^{-1} \operatorname{logit} Q^\pi(s, z)\right)}{\sum_{u \in \mathcal{Z}_s} \exp\left(\beta^{-1} \operatorname{logit} Q^\pi(s, u)\right)}. \tag{18}$$

Equivalently,

$$\beta \log \frac{\rho^\star(z \mid s)}{\rho^\star(z' \mid s)} = \operatorname{logit} Q^\pi(s, z) - \operatorname{logit} Q^\pi(s, z'), \tag{19}$$

so

$$Q^\pi(s, z) > Q^\pi(s, z') \quad \Longleftrightarrow \quad \rho^\star(z \mid s) > \rho^\star(z' \mid s). \tag{20}$$

Thus all edge-wise population optima are induced by the same latent utility $\operatorname{logit} Q^\pi(s, z)$ rather than conflicting across pairs. The connection to intervention advantage follows by averaging the concrete source–native Q gap over native blocks:

$$\mathbb{E}_{z^\pi \sim \pi(\cdot|s)} \left[Q^\pi(s, z^I) - Q^\pi(s, z^\pi) \right] = Q^\pi(s, z^I) - V^\pi(s) = A^\pi(s, z^I). \tag{21}$$

Thus, IP-OPD learns from concrete source–native value comparisons, while their expectation over native behavior recovers intervention advantage.

Corollary 4.2 (Invariance to Discordant-Pair Filtering)**.** *Under the assumptions of Theorem 4.1, conditioning on discordant outcomes $R \neq R'$ does not change the population minimizer. More generally, strictly positive reweighting of observed comparison edges preserves the same minimizer as long as the comparison graph is unchanged.*

Appendix A.3 gives the population-risk derivation, proofs, and finite-support boundary cases. These are population statements, not a monotonic-return guarantee for a finite-data parametric update.

4.3 Compatibility with Heterogeneous Sources

At the algorithmic interface, IP-OPD uses the source only through a proposed text block z^I. The same paired estimator and objective therefore apply to external teachers, stronger-inference self-proposals, mixtures of sources, and cross-architecture sources. Proposal quality and generation cost may differ, but IP-OPD requires no source logits, learned critic, routing network, or access to the source architecture. The proposed text is encoded with the student tokenizer before continuation and optimization. Concrete source instantiations and their costs are given in Appendices G and H.

5 Experiments

We evaluate IP-OPD along five axes: downstream reasoning performance, intervention-utility identification, the contribution of paired supervision, compatibility across proposal sources, and computational cost. Appendices C.6 and H.2 document the outcome-information-matched control and measured cost accounting from the completed runs.

5.1 Experimental Setup

Models, data, and evaluation. The primary student is `Qwen3-1.7B-Base`; the frozen teacher and intervention source are the same `Qwen3-8B` model, with thinking disabled (Yang et al., 2025). We used MATH's 7,500 training problems (Hendrycks et al., 2021). A subject–difficulty-stratified 500-problem development split (seed 20260915) was used for configuration selection. After configuration selection, all final runs restarted from the base checkpoint and trained on the full 7,500-problem MATH training set for three epochs. We evaluated the complete MATH500 (500), AMC23 (40), Minerva Math (272), AIME24 (30), and AIME25 (30) sets (Lightman et al., 2023; Lewkowycz et al., 2022; Zhang & Math-AI, 2025). No test set was used to select checkpoints. Eight samples per problem were generated with temperature 1.0, top-$p = 0.8$, and an 8,192-token completion cap. Avg@8 is mean sample correctness, not majority voting; Macro Avg. weights the five benchmarks equally. Pass@8 and uncertainty are reported in Appendix D.

Training and comparisons. We ran Sequence KD (Kim & Rush, 2016), GRPO (Shao et al., 2024), Vanilla OPD (Agarwal et al., 2024), EOPD (Jin et al., 2026), and common-setting implementations of DOPD, IW-OPD, and Counterfactual Recoverability (Yu et al., 2026; Xie et al., 2026b; Jiang et al., 2026). For methods whose original settings do not exactly match ours, we report setting-specific adaptations—DOPD-adapt, IW-OPD-adapt, and CR-style—rather than claiming official numerical reproduction; Appendix C.4 specifies the implemented scope and differences. Each online

Table 1: Main results: Avg@8 (%). Qwen3-8B → Qwen3-1.7B-Base, MATH training. Rows follow the common evaluation setting. CM denotes the compute-matched IP-OPD run stopped at the EOPD training budget. Citations identify method sources. Training-seed scope is given in Appendix D.3.

Method	MATH500	AMC23	Minerva	AIME24	AIME25	Avg.
Base student (Yang et al., 2025)	54.2	26.3	20.5	5.0	2.1	21.6
Teacher (reference) (Yang et al., 2025)	86.5	69.4	48.2	26.7	22.5	50.7
Sequence KD (Kim & Rush, 2016)	63.1	37.2	27.8	9.6	3.3	28.2
GRPO (Shao et al., 2024)	68.8	40.6	29.5	9.2	4.6	30.5
Vanilla OPD (Agarwal et al., 2024)	67.8	39.3	29.8	8.2	6.1	30.2
EOPD (Jin et al., 2026)	68.7	41.9	30.2	10.4	5.8	31.4
DOPD-adapt (Yu et al., 2026)	70.0	43.8	31.5	11.7	7.1	32.8
IW-OPD-adapt (Xie et al., 2026b)	69.4	42.5	31.0	11.3	6.7	32.2
CR-style (Jiang et al., 2026)	69.9	44.1	31.2	11.3	7.5	32.8
IP-OPD (CM)	71.2	45.5	32.5	12.5	8.3	34.0
IP-OPD	**72.4**	**48.0**	**33.5**	**13.5**	**9.4**	**35.4**

round contained 128 prompts; IP-OPD collected one native trajectory, at most $N_s = 2$ online-selected states, one source block per state, and $K = 1$ intervention continuation. Blocks were capped at $L_{\max} = 128$ student tokens; $H = 32$ was the block horizon, giving a 4,096-token training cap, with the fixed-horizon selector specified in the method appendix. Full-parameter AdamW used a learning rate of 3×10^{-6}, cosine decay, four 32-pair updates per nonempty buffer, and preference scale $\beta = 0.1$. Training-seed scope is detailed in Appendix D.3; single-seed ablations are marked separately. Appendix C records the prompts, batch semantics, reward parsing, adaptations, and all remaining settings used in the runs.

Outcome-information-matched control. Winner-only CE and IP pairwise received the same branch rewards and preference directions, with matched initialization, source, prompt order, selector, and rollout caps. Each method collected states from its own evolving policy. Full and compute-matched results are reported in Section 5.4, with details in Appendix C.6.

5.2 Main Results

Table 1 reports a Macro Avg@8 of 35.4 for full IP-OPD and 34.0 for the compute-matched run, improving over EOPD by 4.0 and 2.6 points, respectively. The corresponding paired 95% bootstrap intervals are [2.1, 5.4] and [0.8, 4.2] percentage points (Appendix D.3). Because reward access and collection budgets differ across baseline families, Table 1 establishes end-to-end performance rather than isolating the contribution of the IP objective; Section 5.4 provides the outcome-information-matched attribution. The teacher is reported only as a reference.

5.3 Does Intervention Advantage Identify Useful Supervision?

Independent diagnostic evaluation. We next ask whether a single paired outcome is informative about downstream intervention utility; this is a signal-quality question rather than an equal-information prediction comparison against distributional proxies. We reserved 500 MATH test problems outside MATH500 for diagnostics only. At an epoch-one student checkpoint, selected states and proposals were fixed. An independent bank of $K_{\text{ref}} = 16$ continuations per branch was used to estimate $A^{\pi}(s, z^I)$; native reference continuations resampled their first block. None of these outcomes was reused in the single-pair score. We compared teacher entropy, student entropy, teacher–student KL, source block likelihood, and the signed paired outcome. AUROC used the sign of the independent estimated advantage; ties, reference uncertainty, and score orientation are reported in Appendix F. IP observed outcomes, whereas the proxies did not, so this was a signal-quality comparison rather than an equal-information prediction contest.

Figure 2(a) reports AUROC 0.79 for the paired score, versus 0.57–0.64 for proxies. Panel (b) shows negative observed pairs across prefix positions; panel (c) shows APR decreasing from 41.0% to 29.2%. The higher AUROC should therefore be interpreted as evidence that downstream outcomes

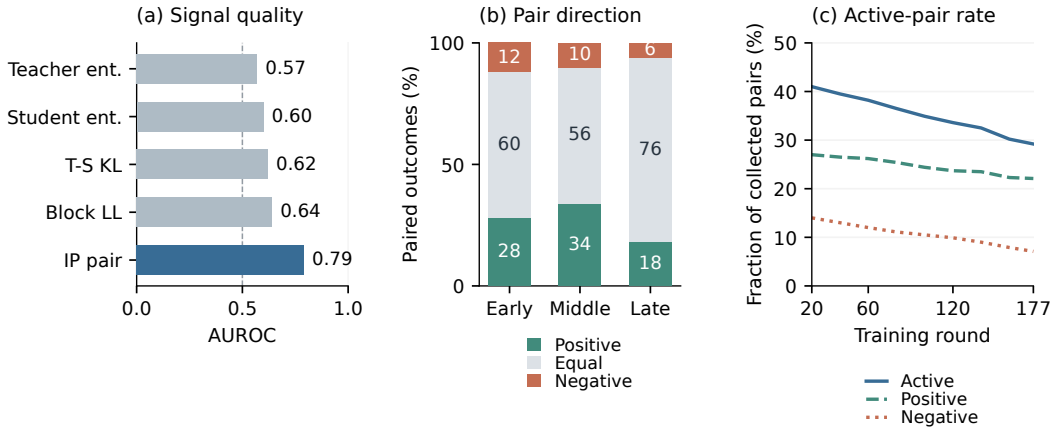


Figure 2: Intervention signal analysis. (a) Association with independently estimated intervention utility. (b) Positive, equal, and negative observed paired outcomes by prefix position. (c) Active-pair rate across training. A negative observed pair does not establish negative expected advantage. See Appendix F for diagnostic details.

contain useful intervention information, not as a cost- or information-matched superiority claim over these proxies. This information advantage comes with additional rollout and reward access. A separate fixed-block check in Appendix F.3 compares preference labels with the concrete Q gap, not with a resampled-native state-value baseline. Repeated-rollout and same-state, multiple-intervention diagnostics distinguish negative outcomes from harmful expected interventions (Appendix F).

5.4 What Makes IP-OPD Effective?

Outcome-matched winner imitation versus pairwise learning. To test whether access to outcome rewards alone explains the benefit, we ran a Full-MATH comparison against winner-only CE. Both objectives used all discordant pairs, including native winners, and neither used a critic or extra regularizer. Table 2 reports the three-seed results. Appendix D.3 specifies seed aggregation and uncertainty. This experiment evaluated the learning rule together with its induced data distribution.

Table 2: Full-MATH outcome-information-matched control. Measured three-seed Avg@8 (%); Macro is mean ± sample SD across the executed runs. Full denotes three epochs; CM uses the EOPD budget. CM GPU-h entries denote budget caps; the checkpoint rule and measured cost accounting are in Appendix H.2.

Budget	Objective	MATH	AMC	Minerva	AIME24	AIME25	Macro	GPU-h
Full	Winner CE	70.7	44.7	31.9	11.7	7.5	33.3 ± 0.4	94.2
Full	IP pairwise	72.4	48.0	33.5	13.5	9.4	35.4 ± 0.1	99.0
CM	Winner CE	69.7	43.3	31.1	10.8	6.7	32.3 ± 0.4	≤ 77.4
CM	IP pairwise	71.2	45.5	32.5	12.5	8.3	34.0 ± 0.4	≤ 77.4

IP pairwise gains 2.0/1.7 macro points at Full/CM budgets, with paired 95% bootstrap CIs of [0.6, 3.1]/[0.4, 2.9], respectively (Appendix D.3). With matched branch rewards and preference directions, this provides controlled evidence for pairwise learning beyond winner imitation, while comparing complete online procedures rather than identical fixed trajectories.

Contribution of reversed preferences. We compared full IP-OPD with always imitating source blocks, imitating only source-successful blocks, and keeping only positive outcome-discordant pairs. Each variant used full MATH training, seed 42, identical candidate-generation budgets, and four updates per nonempty buffer; its own evolving student induced its states. The first two used block cross-entropy, whereas positive-only retained the pairwise loss. Thus, positive-only provided the cleanest test of negative-pair use; the other two also changed the objective. Exact definitions and matched-yield checks are in Appendix E.

Table 3a compares outcome-aware supervision rules, culminating in the full bidirectional native-relative preference objective. In this single-seed ablation, full IP-OPD reaches 60.0% Avg.2 ver-

Table 3: Mechanism and source-interface checks. (a) Full MATH, seed 42; Avg.2 averages MATH500 and AMC23. (b) MATH-2K, one epoch, seed 42; both columns are Avg.2.

(a) Core mechanism.

Variant	MATH	AMC	Avg.2
Always imitate	68.8	41.3	55.0
Success-only	69.7	43.1	56.4
Positive-only	71.1	45.0	58.1
Full IP-OPD	**72.4**	**47.5**	**60.0**

(b) Source compatibility.

Source	Control	IP-OPD
External	48.9	52.5
Self	45.3	48.3
Multi	49.6	53.0
Cross-arch.	47.7	51.2

Table 4: Computational cost. One run on four RTX 4090 GPUs; GPU-h includes reserved-device time; CM reports a cap. Source columns distinguish generated and scored tokens (millions); student tokens include native and additional suffix generation. CM stops at the EOPD budget. See Appendix H.

Method	Source tokens (M)		Student	GPU-h	Peak	Avg.
	Gen.	Scored	tokens (M)		GiB	
Vanilla OPD	0.0	28.8	28.8	72.0	21.2	30.2
EOPD	0.0	28.8	28.8	77.4	21.8	31.4
IP-OPD	1.5	0.0	39.6	99.0	22.1	35.4
IP-OPD (CM)	1.1	0.0	30.7	≤ 77.4	22.1	34.0

sus 58.1% for positive-only, consistent with a benefit from retaining native-preferred comparisons. Length normalization and parameter sensitivity are reported as supporting checks.

Source–native versus native–native preferences. To separate the value of source proposals from preference learning itself, we also compare source–native IP with native–native preferences under the same overall framework. Source–native IP is higher by 2.5/1.8 macro points at the Full/CM budgets; definitions and matched controls are given in Appendix E.3.

5.5 Source Compatibility and Efficiency

The controlled extension used external Qwen3-8B, a majority-of-four self-proposal, a uniform Qwen3-4B/8B mixture, and DeepSeek-R1-Distill-Llama-8B (Guo et al., 2025); source prompts and budgets are specified in Appendix G. Self-proposal selection never saw a gold answer, and its search outcomes were not reused for preference labels. All settings retained the same student and outcome-pair objective. Table 3b shows that the same IP-OPD interface improves over source imitation across external, self, mixture, and cross-architecture proposal settings. These comparisons assess source-interface compatibility, not equal source quality or compute across sources. Table 4 shows the cost–accuracy tradeoff: the compute-matched IP run reaches 34.0% Macro Avg. Deployed model remains the student alone, with no source calls or paired rollouts at test time.

The MATH-2K source experiments assess interface compatibility rather than equal-cost source quality. Appendix D.4 further shows the same trend in a supporting 0.6B-student run (23.6 vs. 21.2 Macro Avg@8 for IP-OPD and EOPD); detailed pair-yield and cost accounting are provided in Appendices H.3 and H.2.

6 Conclusion

For outcome-verifiable reasoning, state alignment alone does not ensure useful supervision. IP-OPD compares source-proposed and native blocks through the same student continuation and converts discordant outcomes into bidirectional preferences. Our analysis links intervention advantage to an explicit intervention policy and characterizes the population objective; experiments show gains across heterogeneous sources, while outcome-information- and compute-matched controls support the learning and efficiency comparisons. IP-OPD requires evaluable outcomes and extra training rollouts, and population consistency does not imply monotonic finite-update improvement. Overall, the results support student-conditioned, source-identity-agnostic distillation based on downstream effect rather than source identity.

AI USE STATEMENT

Generative AI assisted with figure preparation and minor language polishing to improve the clarity and readability of the manuscript.

ETHICS STATEMENT

This work concerns mathematical reasoning using public benchmarks and existing language models. The experiments involved neither human participants nor collection of personal data. Use and redistribution of models and datasets are subject to their respective licenses and terms.

REPRODUCIBILITY STATEMENT

Sections 3–4 specify the formulation and objective. Appendices A–C provide proofs, pseudocode, data splits, prompts, hyperparameters, and baseline adaptations. Evaluation, diagnostics, source configurations, and compute accounting are detailed in Appendices D–H.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024.

Mohammadreza Armandpour, Fatih Ilhan, David Harrison, Ajay Jaiswal, Duc NM Hoang, Fartash Faghri, Yizhe Zhang, Minsik Cho, and Mehrdad Farajtabar. Unmasking on-policy distillation: Where it helps, where it hurts, and why. *arXiv preprint arXiv:2605.10889*, 2026.

Yi Ding and Ruqi Zhang. Does on-policy distillation really distill? from noisy teacher to self-improvement. *arXiv preprint arXiv:2608.31046*, 2026.

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

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *International Conference on Learning Representations*, volume 2024, pp. 32694–32717, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

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

Byeongho Heo, Jaehui Hwang, Sangdoo Yun, and Dongyoon Han. On-policy delta distillation. *arXiv preprint arXiv:2607.15161*, 2026.

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

Namgyu Ho, Laura Schmid, and Se-Young Yun. Large language models are reasoning teachers. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 14852–14882, 2023.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the association for computational linguistics: ACL 2023*, pp. 8003–8017, 2023.

De Jiang, Zhengyang Zhang, Kehong Yuan, and Shaohua Ma. Not every divergence should be suppressed: Counterfactual recoverability in on-policy distillation. *arXiv preprint arXiv:2608.04408*, 2026.

Yuxuan Jiang and Francis Ferraro. Bridging reasoning trajectories in on-policy distillation via near-future guidance. *arXiv preprint arXiv:2606.00305*, 2026.

Woogyeol Jin, Taywon Min, Yongjin Yang, Dennis Wei, Yi Zhou, Swanand Ravindra Kadhe, Nathalie Baracaldo, and Kimin Lee. Entropy-aware on-policy distillation of language models. *arXiv preprint arXiv:2603.07079*, 2026.

Yoon Kim and Alexander M Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 1317–1327, 2016.

Xin Lai, Zhuotao Tian, Yukang Chen, Senqiao Yang, Xiangru Peng, and Jiaya Jia. Step-dpo: Step-wise preference optimization for long-chain reasoning of llms. *arXiv preprint arXiv:2406.18629*, 2024.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857, 2022.

Enhan Li, Junhao He, and Hongyang Du. Crop: Task relevance via counterfactuals for selective on-policy distillation. *arXiv preprint arXiv:2608.13387*, 2026.

Liunian Harold Li, Jack Hessel, Youngjae Yu, Xiang Ren, Kai-Wei Chang, and Yejin Choi. Symbolic chain-of-thought distillation: Small models can also "think" step-by-step. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2665–2679, 2023.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024.

Wenhan Ma, Jianyu Wei, Liang Zhao, Hailin Zhang, Bangjun Xiao, Lei Li, Qibin Yang, Bofei Gao, Yudong Wang, Rang Li, et al. Mopd: Multi-teacher on-policy distillation for capability integration in llm post-training. *arXiv preprint arXiv:2606.30406*, 2026.

Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37:124198–124235, 2024.

Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 5191–5198, 2020.

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. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. Pmlr, 2015.

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

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process-and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022.

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

Yan Xie, Sijie Zhu, Tiansheng Wen, Bo Chen, and Yifei Wang. On the position bias of on-policy distillation. *arXiv preprint arXiv:2606.22600*, 2026a.

Yan Xie, Sijie Zhu, Tiansheng Wen, Bo Chen, and Yifei Wang. On the position bias of on-policy distillation. *arXiv preprint arXiv:2606.22600*, 2026b.

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.

Chen Yang, Haiyuan Wan, Rengrong Xiong, Yize Chen, and Danny HK Tsang. Beyond imitation: Filtering on-policy distillation by reasoning progress. *arXiv preprint arXiv:2608.19408*, 2026.

Xinlei Yu, Gen Li, Qingyi Si, Guibin Zhang, Yuqi Xu, Congcong Wang, Shuai Dong, Kaiwen Tuo, Xiangyu Zeng, Kaituo Feng, et al. Dopd: Dual on-policy distillation. *arXiv preprint arXiv:2606.30626*, 2026.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime). *Mathematical Association of America (MAA)*, 2025.

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*, 2026.

Binbin Zheng, Xing Ma, Yiheng Liang, Jingqing Ruan, Xiaoliang Fu, Kepeng Lin, Benchang Zhu, Ke Zeng, and Xunliang Cai. Scope: Signal-calibrated on-policy distillation enhancement with dual-path adaptive weighting. *arXiv preprint arXiv:2604.10688*, 2026.

A Theory of Intervention Advantage

We use the finite-horizon block process of Section 3, with a common stopping rule, absorbing terminal states, and time included in the state.

A.1 Performance Difference and Positive Interventions

Proof of Theorem 3.2. For $\pi' = \pi_{\eta,q}$, the finite-horizon performance-difference identity gives

$$J(\pi') - J(\pi) = \sum_{t=1}^{H} \mathbb{E}_{s_t \sim d_t^{\pi'}, z_t \sim \pi'(\cdot|s_t)} A^{\pi}(s_t, z_t). \tag{22}$$

Since $\mathbb{E}_{z\sim\pi(\cdot|s)} A^{\pi}(s,z) = V^{\pi}(s) - V^{\pi}(s) = 0$, linearity of the mixture implies

$$\mathbb{E}_{z\sim\pi'(\cdot|s)} A^{\pi}(s,z) = \eta(s)\bar{A}^{\pi}(s;q). \tag{23}$$

Substitution proves Eq. (8). □

Although A^{π} evaluates one deviation followed by π, the occupancy here is induced by π', on whose trajectories interventions can occur repeatedly. Corollary 3.3 follows because every term is nonnegative; strict improvement requires a strictly positive contribution on a set with positive π' occupancy. More generally, if $\bar{A}^{\pi}(s;q) \geq -\epsilon$ at all intervention states, then

$$J(\pi') - J(\pi) \geq -\epsilon \sum_{t=1}^{H} \mathbb{E}_{d_t^{\pi'}} \eta(s_t) \geq -H\epsilon. \tag{24}$$

Thresholding a noisy sample estimate alone does not establish this premise.

A.2 Source Identity and Mixture Sources

Only the induced proposal law matters: $q_1 = q_2$ implies equal source-average advantage. For $q = \sum_m \alpha_m q_m$, linearity gives $\bar{A}^{\pi}(s;q) = \sum_m \alpha_m \bar{A}^{\pi}(s;q_m)$. In particular $\bar{A}^{\pi}(s;\pi) = 0$. This unfiltered average does not remove the learning signal from outcome-discordant native–native pairs (Appendix E.3). Concrete source implementations are specified once in Appendix G.

A.3 Global Alignment of the IP-OPD Surrogate

Fix a selected state s, frozen policy π, and finite support $\mathcal{Z}_s$. Let $\mu_s(z,z')$ be the pre-filter comparison law and let G_s connect distinct blocks compared with positive probability. Write

$$p(z) = Q^{\pi}(s,z), \qquad u(z) = \operatorname{logit} p(z). \tag{25}$$

Under conditional independence and $0 < p(z) < 1$, the discordant-outcome probabilities are

$$a(z,z') = p(z)(1-p(z')), \qquad b(z,z') = (1-p(z))p(z') > 0. \tag{26}$$

Both a and b are positive. For a log-probability margin Δ, define

$$\phi_{z,z'}(\Delta) = a\,\mathrm{softplus}(-\beta\Delta) + b\,\mathrm{softplus}(\beta\Delta). \tag{27}$$

Proof of Theorem 4.1. With $\sigma(v) = (1+e^{-v})^{-1}$, $\phi' = \beta[b\sigma(\beta\Delta) - a\sigma(-\beta\Delta)]$ and $\phi'' = \beta^2(a+b)\sigma(\beta\Delta)\sigma(-\beta\Delta) > 0$. Thus each comparison has the unique optimum

$$\beta\Delta^{\star} = \log(a/b) = u(z) - u(z'). \tag{28}$$

All such margins are simultaneously realized by the full-support distribution

$$\rho^{\star}(z \mid s) = \frac{e^{u(z)/\beta}}{Z_s}, \qquad Z_s = \sum_{v\in\mathcal{Z}_s} e^{u(v)/\beta} < \infty. \tag{29}$$

Consequently it minimizes every positive-weight term of

$$\mathcal{R}_s(\rho) = \sum_{z,z'} \mu_s(z,z')\phi_{z,z'}\big(\log\rho(z \mid s) - \log\rho(z' \mid s)\big). \tag{30}$$

Any other minimizer $\widetilde{\rho}$ must attain each edge's unique optimal margin: no term can improve beyond its individual minimum. Hence $g(z) = \log \widetilde{\rho}(z \mid s) - u(z)/\beta$ is constant along every edge. Connectivity makes g constant on $\mathcal{Z}_s$, and normalization forces $\widetilde{\rho} = \rho^\star$. Finally, strict monotonicity of logit gives $Q^\pi(s,z) > Q^\pi(s,z') \iff \rho^\star(z \mid s) > \rho^\star(z' \mid s)$. □

Proof of Corollary 4.2. Let $d(z,z') = a + b$ and $D_s = \mathbb{E}_{\mu_s} d > 0$. Filtering yields pair law $\mu_s^{\mathrm{act}} = \mu_s d / D_s$ and conditional direction probabilities $a/d, b/d$. Therefore

$$\mathcal{R}_s^{\mathrm{act}}(\rho) = \mathcal{R}_s(\rho)/D_s, \tag{31}$$

which has the same minimizer. Strictly positive, outcome-independent edge weights likewise preserve the simultaneously attainable edge optima and, when the graph is unchanged, uniqueness. □

Boundary and support scope. At $Q = 0$ or 1, logits may diverge and a finite full-support minimizer need not exist. Disconnected graphs leave relative component masses undetermined. Uniqueness is over distributions normalized on $\mathcal{Z}_s$; ratios on this support do not determine mass outside it. These results do not imply monotonic return improvement for a finite-data, shared-parameter student.

A.4 Actual Parameter Updates: Utility, Calibration, and Interference

We now analyze shared parameters at their current margins, without assuming simultaneous realizability of all statewise optima.

Frozen collection law. During one collection round fix the behavior π_k, source q_k, selected-state law, and remaining rollout horizon. Let $e = (s, z^I, z^N) \sim \mu$ denote a candidate pair before outcome filtering. Write $p_I = Q^{\pi_k}(s, z^I)$, $p_N = Q^{\pi_k}(s, z^N)$, $a_e = p_I(1 - p_N)$, $b_e = (1 - p_I)p_N$, $d_e = a_e + b_e$, and $D = \mathbb{E}_\mu d_e > 0$. Outcomes are conditionally independent given e. For the trainable policy define

$$\Delta_e(\theta) = \log \pi_\theta(z^I \mid s) - \log \pi_\theta(z^N \mid s), \quad h_e = \nabla_\theta \Delta_e. \tag{32}$$

Assume positive block probabilities, differentiability, and integrable derivatives in the neighborhood considered. All collection quantities are stop-gradient. The globally pooled active-pair population loss for this law is

$$L_k(\theta) = \frac{1}{D} \mathbb{E}_\mu \left[a_e \operatorname{softplus}(-\beta \Delta_e) + b_e \operatorname{softplus}(\beta \Delta_e) \right]. \tag{33}$$

A finite buffer normalized by its realized active count is a random empirical objective; it is not assumed to have an exactly unbiased gradient for this pooled population loss. In particular, shared native suffixes create dependence between pairs from the same trajectory.

Proposition A.1 (Exact parametric gradient decomposition)**.** *For the frozen law above,*

$$-\nabla L_k(\theta) = \frac{\beta}{2D} \big(U_\theta - C_\theta \big), \tag{34}$$

$$U_\theta = \mathbb{E}_\mu[(p_I - p_N) h_e], \quad C_\theta = \mathbb{E}_\mu[d_e \tanh(\beta \Delta_e / 2) h_e]. \tag{35}$$

Here U_θ transports the concrete downstream-value gap into parameter space, while C_θ accounts for the current probability margin.

Proof. The negative scalar derivative of one unnormalized risk is $\beta[a_e \sigma(-\beta\Delta_e) - b_e \sigma(\beta \Delta_e)]$. Using $a_e - b_e = p_I - p_N$, $a_e + b_e = d_e$, and $2\sigma(x) - 1 = \tanh(x/2)$ yields $\beta[(p_I - p_N) - d_e \tanh(\beta \Delta_e / 2)]/2$. Apply the chain rule and average, keeping D fixed. □

If $\|h_e\| \le G$ and $|\Delta_e| \le M$, then $\|C_\theta\| \le DG \tanh(\beta M/2)$. Thus an already large likelihood margin can reverse the update coefficient despite a positive value gap.

Finite updates and shared parameters. For an active minibatch of size n, let $c_i = \beta y_i \sigma(-\beta y_i \Delta_i)$ and $\widehat{v} = n^{-1}\sum_i c_i h_i = -\nabla \widehat{L}$. If the margin Hessian has norm at most B_e along the actual displacement δ, Taylor's theorem gives

$$|\Delta_e(\theta+\delta) - \Delta_e(\theta) - h_e^\top \delta| \le B_e \|\delta\|^2/2. \tag{36}$$

For SGD, $\delta = \alpha\widehat{v}$, so the first-order change is $\alpha n^{-1}\sum_i c_i h_e^\top h_i$: cross-pair inner products can be negative, even though the Gram matrix is positive semidefinite. AdamW must be analyzed with its actual displacement, including momentum and weight decay, rather than substituted into the SGD expression.

If task return J has an L_J-Lipschitz gradient locally, then

$$J(\theta+\delta) - J(\theta) \ge \nabla J(\theta)^\top \delta - L_J \|\delta\|^2/2. \tag{37}$$

Thus sufficient alignment and a sufficiently small step imply improvement; the preference objective alone does not establish that alignment. For the special case $q = \pi_k$ with normalized finite-horizon occupancy sampling and identical scored/sampling policies, the score-function identity yields $U_{\theta_k} = 2\nabla J(\theta_k)/H$: expanding the two independent native terms cancels the zero-mean score terms. Hence $-\nabla L_k = \beta \nabla J/(HD) - \beta C/(2D)$. The correction C, non-occupancy state selection, external proposals, finite buffers, and shared trajectories prevent treating this special identity as a general policy-improvement guarantee.

Why an unconditional improvement theorem would be false. Consider a one-state, two-action problem with success probabilities 0.6 and 0.5 and $\beta = 0.1$. Parameterize the better-action probability by $\sigma(\theta)$ and take any connected, positive-weight frozen comparison law. The optimum log-probability margin is $\theta^\star = 10\log(1.5)$, corresponding to probability approximately 0.983. At an initial probability 0.99, $\theta = \log(99) > \theta^\star$. A sufficiently small descent step on the exact population preference risk therefore decreases θ and also decreases the true return $J(\theta) = 0.5 + 0.1\sigma(\theta)$, despite correct population ordering and exact parameter realizability. This counterexample isolates the calibration term and rules out inferring monotonic return improvement from loss descent.

A.5 Independent Symmetric Reward Corruption

Fix (s, z^I, z^π) and condition on the frozen behavior policy. Let the two binary rewards be conditionally independent with success probabilities p_I, p_N. Independently flip each reward with probability $\epsilon \in [0, 1/2)$; the flips are also independent of the clean rewards and blocks. For branch $b \in \{I, N\}$,

$$\widetilde{p}_b = \epsilon + (1-2\epsilon)p_b, \qquad \widetilde{p}_I - \widetilde{p}_N = (1-2\epsilon)(p_I - p_N). \tag{38}$$

Thus the corrupted mean reward difference has the same sign as the clean concrete Q gap. If $\widetilde{p}_I, \widetilde{p}_N \in (0,1)$, the population logistic risk has the free-margin optimum

$$\beta\widetilde{\Delta}^\star = \mathrm{logit}(\widetilde{p}_I) - \mathrm{logit}(\widetilde{p}_N), \qquad \mathrm{sign}(\widetilde{\Delta}^\star) = \mathrm{sign}(p_I - p_N). \tag{39}$$

This follows from $\widetilde{p}_b = (1-\epsilon)p_b + \epsilon(1-p_b)$: independent flips preserve branch independence, so the population proof applies with $\widetilde{p}_b$. At zero margin the unnormalized derivative is $\beta(1-2\epsilon)(p_N - p_I)/2$; active-pair normalization only rescales it positively. This is an ordering result, not a sample-efficiency or finite-update return guarantee. At $\epsilon = 1/2$ the signal vanishes; asymmetric or correlated errors are outside the result.

B IP-OPD Estimation and Training Details

B.1 Paired-Rollout Estimation

Let I_t indicate whether reached state s_t is selected. Unbiasedness in Section 4.1 requires selection and proposals to exclude the future native continuation.

Assumption B.1 (Prefix-Measurable Selection and Intervention)**.** *The selection decision I_t is made before sampling z_t^π and is measurable with respect to the current prefix and past selector randomness. The intervention block is sampled according to*

$$z^I \sim q(\cdot \mid s) \tag{40}$$

using only the observed state s. The selector and source use randomness independent of the native and intervention continuation randomness. In particular,

$$(I_t, z_t^I) \perp (z_{\geq t}^\pi, R_t^\pi) \mid s_t. \tag{41}$$

The selector is drawn before the native block, and the source sees only the prefix. Thus $\mathbb{E}[R^I \mid s, z^I, I = 1] = Q^\pi(s, z^I)$ and $\mathbb{E}[R^\pi \mid s, z^I, I = 1] = V^\pi(s)$, giving

$$\mathbb{E}[\hat{A} \mid s, z^I, I = 1] = Q^\pi(s, z^I) - V^\pi(s) = A^\pi(s, z^I). \tag{42}$$

Multiple selections can share one native terminal reward. This dependence does not invalidate conditional pairwise unbiasedness, but can increase minibatch variance.

B.2 Sparse State and Block Sampling

Post-hoc selection can depend on future trajectory length. Instead, at each reached non-terminal s_t, before sampling z_t^π, let $C_{t-1} = \sum_{u=2}^{t-1} I_u$ and draw

$$I_t = \mathbf{1}(C_{t-1} < N_s)\widetilde{I}_t, \qquad \widetilde{I}_t \sim \text{Bernoulli}(\rho_t(s_t, C_{t-1})). \tag{43}$$

The hazard may use only the prefix, step index, and past selection randomness. Our default is $N_s = 2$ and

$$\rho_t(s_t, C_{t-1}) = \min\{1, (N_s - C_{t-1})/(H - t + 1)\}, \quad 2 \leq t \leq H. \tag{44}$$

This uniformly preselects indices from $\{2, \ldots, H\}$; unreached indices are not replaced, so N_s is a cap. The prompt-only state is excluded, but the last reached non-terminal state remains eligible. Both branches use the same bounded block rule; exact stopping and handoff settings appear in Appendix C.2. No learned selector is used.

B.3 Repeated Suffixes

The default uses one binary reward per branch. The $K > 1$ ablation fixes both concrete blocks and averages independent suffix rewards; its target is $Q^\pi(s, z^I) - Q^\pi(s, z^N)$. Estimating $A^\pi(s, z^I)$ instead requires resampling native first blocks in the reference bank. The implemented ablation uses only the sign, discards ties, and adds no magnitude weighting or threshold.

B.4 Complete Training Algorithm

Algorithm 1 summarizes IP-OPD. The behavior policy is frozen within each data-collection round, so all paired trajectories in $\mathcal{B}_k$ are associated with the same base policy π_k.

After each update, the next round collects fresh states from the updated policy. Buffer reuse and empty-buffer handling are specified in Appendix C.2.

Source implementations and cost. Source-specific settings are given in Appendix G. For m realized selections, added generation is $\sum_{j=1}^m (|z_j^I| + |\tau_j^I|)$; N_s is a cap, not a realized count. All suffix repetitions and self-search are charged under Appendix H.

C Full Experimental Setup

Unless stated otherwise, $H = 32$, $L_{\max} = 128$ student tokens, and each round contains 128 prompts. Shared settings below apply to all subsequent experiments.

C.1 Datasets, Splits, and Contamination Checks

Table 5 lists data and evaluation sizes.

Algorithm 1 Intervention-Preference On-Policy Distillation (IP-OPD)

Require: Dataset $\mathcal{D}$; trainable policy π_θ; intervention source q_k; state budget N_s; horizon H; scale β

for training round $k = 0, 1, \ldots$ **do**
  Freeze the current model as behavior policy π_k
  Initialize paired buffer $\mathcal{B}_k \leftarrow \varnothing$
  for each sampled problem $x \sim \mathcal{D}$ **do**
    Initialize $s_1 \leftarrow (x)$, $C \leftarrow 0$, and $\mathcal{C}_x \leftarrow \varnothing$
    for $t = 1, \ldots, H$ while s_t is non-terminal **do**
      if $t > 1$ and $C < N_s$ **then**
        Draw I_t using the prefix-measurable hazard in Eq. (43)
      else
        $I_t \leftarrow 0$
      end if
      Sample native block $z_t^\pi \sim \pi_k(\cdot \mid s_t)$
      if $I_t = 1$ **then**
        Add (s_t, z_t^π) to $\mathcal{C}_x$; $C \leftarrow C + 1$
      end if
      Append z_t^π to the native trajectory and obtain s_{t+1}
    end for
    Evaluate the native terminal reward R^π
    for each $(s_t, z_t^\pi) \in \mathcal{C}_x$ **do**
      Sample one intervention block $z_t^I \sim q_k(\cdot \mid s_t)$
      Continue from (s_t, z_t^I) using π_k to obtain R_t^I
      Compute $\hat{A}_t \leftarrow R_t^I - R^\pi$
      if $\hat{A}_t \neq 0$ **then**
        Add $(s_t, z_t^I, z_t^\pi, \hat{A}_t)$ to $\mathcal{B}_k$
      end if
    end for
  end for
  Update θ on $\mathcal{B}_k$ by minimizing Eq. (17)
end for

Table 5: Dataset manifest. Split sizes and evaluation sampling. Official test membership is fixed.

Dataset / split	Problems	Samples	Purpose / fixed source
MATH train	7,500	1	Final full-data runs; Hendrycks et al. (2021)
MATH pilot train	7,000	1	Training after excluding development IDs
MATH-dev-500	500	8	Hyperparameters only; seed 20260915
MATH-2K	2,000	1	Nested pilot subset; seed 20260916
MATH-4K	4,000	1	Extension-size confirmation
MATH-Diag-500	500	$16 + 16$	Reference branches; MATH test excluding MATH500
MATH500	500	8	Final test; Lightman et al. (2023)
AMC23	40	8	Final test; Qwen math-evaluation snapshot
Minerva Math	272	8	Final test; Lewkowycz et al. (2022)
AIME24	30	8	Final test; both contest parts
AIME25	30	8	Final test; both contest parts

Splits and overlap audit. We allocated 500 development items by subject–level quotas $500n_g/7500$, using floors then largest fractional remainders (lexicographic ties). Within strata, IDs were sorted by SHA-256 of `20260915|problem_id`. Nested MATH-2K/4K subsets used the remaining 7,000 items, the same rule, and seed 20260916. Final runs restarted on all 7,500 items for three epochs; development data did not select final checkpoints.

MATH500 membership was resolved against MATH test by IDs or normalized hashes. MATH-Diag-500 used the remaining test items, the same stratification, and seed 20260917, excluding all

primary tests. Pinned Qwen snapshots supplied AMC23, Minerva and AIME; Minerva parent-problem IDs were preserved. Before training, we applied NFKC normalization, normalized line endings, and collapsed repeated whitespace, flagged exact matches and word-13-gram Jaccard similarity above 0.80, and adjudicated overlaps. Confirmed duplicates were removed from training consistently across methods; official tests were unchanged. Split IDs, hashes, and post-audit dataset/round counts were recorded.

C.2 Training, Generation, and Buffer Semantics

Table 6: Common configuration. Method-specific exceptions are stated explicitly.

Parameter	Default / exceptions
Student / source	Qwen/Qwen3-1.7B-Base / frozen non-thinking Qwen/Qwen3-8B
Precision / update	BF16, full parameter; all trainable methods
Epochs / collection batch	3 / 128 distinct prompts; MATH-2K: 1 epoch; retain incomplete batches
Native samples / prompt	1; GRPO: 8; DOPD-adapt: 4
AdamW	lr 3×10^{-6} (Sequence KD: 10^{-5}); $(b_1, b_2) = (0.9, 0.999)$, $\epsilon = 10^{-8}$
Weight decay / clipping	0.0 / global norm 1.0
Scheduler / warmup	Cosine / 6 rounds (pilot: 1); advances on empty rounds
Updates / batch	4 per nonempty buffer / 32 pairs (OPD: prompts; GRPO: groups)
Microbatch / reuse	1 pair per GPU, 8 accumulations on 4 GPUs; discard buffer after round
Training decoding	All temperatures 1.0; top-$p = 1$, top-k disabled; repetition penalty 1.0; no minimum length
Prompt / completion cap	2,048 / 4,096 tokens; no silent question truncation
$H, L_{\max}, N_s, K, \beta$	$32, 128, 2, 1, 0.1$
Evaluation	8 samples; temperature 1.0, top-$p = 0.8$, 8,192 new tokens
Seeds	Training: 42–44 where repeated (see Appendix D.3); evaluation: 1000–1007; diagnostic: 20260917; bootstrap: 20260918
Hardware	4×RTX 4090, 24 GiB/card; profiled peak memory

Rounds and updates. Shuffling once per epoch and retaining incomplete batches gave 177 rounds and at most 708 updates. Each frozen-policy round sampled 32 active pairs uniformly per update, with replacement if necessary. Empty buffers caused no update, refill, or replay; all buffers were then discarded. Loss was masked to the two blocks and used raw joint likelihood, with no auxiliary KL, entropy, critic, router, or SFT term. OPD-like methods split each prompt batch into four groups with their within-prompt rollouts: prompt exposure, not examples or compute, was matched. Optimizer, scheduler, RNG and prompt order were preserved on resumption; unique pairs, repeated exposures and realized updates were logged.

Stopping, selection and handoff. Blocks ended at 128 student tokens or EOS, not newlines; native token IDs were preserved. Generation stopped at EOS, 32 blocks or 4,096 completion tokens, then the answer was parsed. Selection followed Eq. (44). After insertion at t, at most $H - t$ further blocks were allowed under the same global cap. Source EOS was terminal, including empty text followed by EOS. Independent role/replicate seeds were digests of run seed, round, problem ID, state, role and replicate. Infrastructure retries reused the same seed.

Development. Development experiments used MATH-2K for one epoch with seed 42 and varied one factor around the default: $\beta \in \{0.05, 0.1, 0.2\}$, $N_s, K \in \{1, 2, 4\}$, $L_{\max} \in \{64, 128, 256\}$, and $T_q \in \{0.7, 1.0, 1.3\}$. Configuration selection used development Avg@8; when scores were within 0.2 points, the lower-cost setting was preferred, with the default as the final tie-breaker. Settings were frozen before final training.

C.3 Exact Prompts and Answer Verification

We used the following serialization for the Base student and same-family source. The closing thinking tag was unscored context. Only the current prefix was inserted; later native tokens and gold

answers were never exposed. Source requests were stopped after one block, whereas teacher evaluation used the full evaluation cap.

```
<|im_start|>user
{problem}
Please reason step by step and put the final answer in \boxed{}.
<|im_end|>
<|im_start|>assistant
<think>

</think>

{student_prefix}
```

The heterogeneous source used its documented chat template with the following user message:

```
Problem:
{problem}

Existing reasoning prefix:
{student_prefix}

Continue this reasoning from its current point. Output only the
↪  continuation;
do not repeat the prefix or add meta-commentary. Put a final answer in
↪  \boxed{}
only if this continuation finishes the solution.
```

The student scored returned text under its original prefix rather than the source prompt. We archived token IDs, chat templates, and model revisions.

We used the pinned Qwen parser and equivalence checker: the final balanced `\boxed{}` expression was extracted first, followed by its documented fallback. Benchmark-specific symbolic, unit, and tolerance conventions were applied with a one-second symbolic timeout and without executing generated code. Unparseable, timed-out, and incorrect answers received zero and remained in the denominators. We audited a preselected 100-output sample and recorded parser failures and disagreements.

C.4 Baseline Definitions and Adaptation Boundaries

Table 7: Baseline implementations and information access. Adapted rows use our common experimental setting while preserving the cited method's defining objective and information access.

Method	Objective / method-specific settings
Sequence KD (Kim & Rush, 2016)	One unfiltered cached teacher completion per problem; response CE; lr 10^{-5}, batch 128, three epochs.
GRPO (Shao et al., 2024)	Eight samples; standardized binary rewards, SD floor 10^{-6}; clipping 0.2, reference KL 0.001; equal-reward groups have zero advantage.
Vanilla OPD (Agarwal et al., 2024)	Detached sampled reverse-KL log-ratio advantage; policy-ratio clipping 0.2; one rollout.
EOPD (Jin et al., 2026)	OPD plus entropy-selected FKL; threshold 0.8 nats, coefficient 1.0, teacher top-16 support.
IW-OPD-adapt (Xie et al., 2026b;a)	Released prefix weighting enabled, clipped at 1.5; common model/data and development split.
DOPD-adapt (Yu et al., 2026)	Four-regime privileged distillation; four samples, top-128 light branches, weak/light weights 0.3/0.6; cached local hints.
CR-style (Jiang et al., 2026)	Teacher-continuation/rollback control; two budget-matched branches; adaptation below.

DOPD and IW-OPD adaptations. Our DOPD adaptation retained the four losses and routing conditions in Eqs. (6)–(11) of Yu et al. (2026), with detached privileged-student targets. We cached one Qwen3-8B hint per training problem at temperature zero, capped at 256 tokens: "Give at most three high-level mathematical subgoals; omit the answer and worked calculations." Both privileged policies received the hint; the deployed policy did not. Hint generation was included in compute accounting, and upstream top-k support and normalization were pinned. We did not import numerical results from the original setting. For IW-OPD, we retained the released cumulative-prefix weighting and stop-gradient implementation with maximum 1.5 (Xie et al., 2026a), replaced the model/data paths, and used the common development split for checkpoint selection. The exact source revision and commit were archived.

CR-style control. Our CR-style control used the same online candidate indices and restricted extra branch analysis to native failures. We compared teacher completion from s_t with student completion from s_{t-1} after deleting the preceding block. Both received $4096 - |z_{<t}|$ new tokens, temperature 1, and top-$p = 1$. Teacher-only success retained s_t and applied teacher OPD to its next native block; rollback-only success used the first rollback block as context for teacher OPD. Equal outcomes used ordinary OPD on the original block, as did native-successful trajectories. We averaged the local replacement with unchanged token losses, counted each block once, and used the earlier index when candidates affected the same block. This was our free-form-math adaptation of the original error localization procedure. Both extra branches were included in compute accounting.

C.5 Robustness-Control Protocols

The supporting robustness checks use the default student, external source, verifier, raw joint block likelihood, $L_{\max} = 128$, and $K = 1$ unless stated otherwise. They are intended to test qualitative stability rather than define additional headline comparisons.

Reward corruption. Reward corruption used MATH-2K, one epoch, seed 42 and independent branch flips at $\epsilon \in \{0, 0.05, 0.10, 0.20\}$ before tie removal. A hash of seed, round, trajectory, state and branch defined the corruption stream. Corrupted labels were fixed across replay; clean rewards were retained and used for evaluation. All hyperparameters stayed fixed. This models symmetric noise, not a learned judge.

Continuation boundary. The boundary control stopped at the first newline after 32 student tokens, or EOS/128 tokens if earlier. Both rules used a separate 4,096-token horizon without a 32-block limit, with suffixes receiving the remaining tokens. At non-initial boundaries, an online $2/31$ hazard selected at most two states without refill. Positions and yields were logged: segmentation changes opportunity counts, so states were not matched. The default block-horizon run remained a separate reference.

C.6 Full-MATH Outcome-Information-Matched Winner-CE Control

Both arms used the default Full-MATH configuration (Table 6), three epochs, and seeds 42–44. Both retained all discordant pairs, including native winners. Each arm collected from its own evolving policy; matched initialization, prompt order, and reward access therefore did not imply identical realized trajectories.

The only objective change. We set $(z^+, z^-) = (z^I, z^\pi)$ for $(R^I, R^\pi) = (1, 0)$ and reversed the blocks for $(0, 1)$. For a frozen round buffer $\mathcal{B}_k$, the two objectives were

$$\mathcal{L}_{\text{winner}}(\theta) = -\mathbb{E}_{\mathcal{B}_k}[\log \pi_\theta(z^+ \mid s)], \tag{45}$$

$$\mathcal{L}_{\text{pair}}(\theta) = \mathbb{E}_{\mathcal{B}_k}\left[\text{softplus}\big(-0.1\left[\log \pi_\theta(z^+ \mid s) - \log \pi_\theta(z^- \mid s)\right]\big)\right]. \tag{46}$$

Both objectives summed token log probabilities and averaged over pairs, retained native winners, and used no auxiliary regularizer. CE backpropagated through the winner only. Both received the same learning-rate grid $\{10^{-6}, 3 \times 10^{-6}, 10^{-5}\}$ on MATH-dev-500 and the same tie rule. Full training ended at epoch three; CM followed Appendix H with the full-horizon schedule unchanged. Tuning/evaluation costs were separate; CE received no padding computation.

C.7 Systems and Release Manifest

All runs used four RTX 4090s, at least 128 GiB host RAM, BF16 full-parameter FSDP, activation checkpointing and a microbatch size of one. Collection alternated with updates; optimizer allocation was released before loading the behavior snapshot. Inference used two-way tensor parallelism where applicable; heterogeneous source loading was serialized. Reserved devices and loading were charged. The pinned `verl`/HybridFlow stack (Sheng et al., 2025) manifest recorded software/driver/backend and model/tokenizer revisions, data hashes/split IDs, prompts/configs, sampler/verifier tests, outputs, trajectory metadata, training logs, profiles and the lockfile/container digest.

D Complete Results and Statistical Analysis

D.1 Metric Definitions and Reproduced Baseline Results

For n questions and eight evaluation outputs per question,

$$\mathrm{Avg@8} = \frac{100}{8n}\sum_{i,j} R_{ij}, \qquad \mathrm{Pass@8} = \frac{100}{n}\sum_{i} \mathbf{1}\left(\sum_{j} R_{ij} > 0\right). \tag{47}$$

Pass@8 counts questions solved at least once. Macro Avg. weights the five benchmarks equally, and Avg.2 uses only MATH500 and AMC23. All reported values were rounded after aggregation. The IP row reaches 56.9% Macro Pass@8. Our separate FKL+CE KD run gives Avg@8 scores of 63.8/38.1/28.1/10.0/3.3 (Macro 28.7%), distinct from the CE-only Sequence KD run.

Table 8: Pass@8 (%). Results from our executed evaluations; citations identify the originating methods.

Method	MATH500	AMC23	Minerva	AIME24	AIME25	Avg.
KD (FKL + CE)	84.2	70.0	44.1	20.0	16.7	47.0
GRPO	84.6	70.0	48.2	20.0	16.7	47.9
Vanilla OPD	84.8	70.0	47.1	20.0	16.7	47.7
EOPD	87.6	75.0	50.7	23.3	16.7	50.7
IP-OPD	89.2	80.0	55.1	33.3	26.7	56.9

D.2 Seed Variability and Inference

Table 9: Across-seed summary. Per-benchmark Avg@8 is mean $\pm$ sample SD across training seeds, computed from unrounded run scores; Avg. is the equally weighted macro mean.

Method	MATH500	AMC23	Minerva	AIME24	AIME25	Avg.
Vanilla OPD	67.8 ± 0.2	39.3 ± 0.7	29.8 ± 0.1	8.2 ± 0.6	6.1 ± 0.5	30.2
EOPD	68.7 ± 0.2	41.9 ± 0.3	30.2 ± 0.1	10.4 ± 0.4	5.8 ± 0.7	31.4
IP-OPD	72.4 ± 0.1	48.0 ± 0.5	33.5 ± 0.0	13.5 ± 0.5	9.4 ± 0.2	35.4

Training-seed SD is distinct from evaluation-sampling uncertainty. The displayed SDs summarize the observed variability across the executed training seeds.

D.3 Aggregation and Statistical Scope

Sequence KD, GRPO, and IW-OPD-adapt were run with one training seed; Vanilla OPD, EOPD, DOPD-adapt, CR-style, Winner CE, NN, and IP were run with three. Base and teacher have no training-seed SD. The run ledger records the underlying numerators and denominators, with half-up rounding applied only after aggregation; means and SDs were computed from the unrounded run-level results.

For paired uncertainty, we used a 2,000-draw problem-cluster bootstrap (seed 20260918) on the saved per-question rewards. Within each benchmark we sampled problems while retaining all eight answers, used the same sampled clusters across methods and paired training seeds, preserved shared-stem Minerva clusters, averaged seeds within each draw, and then averaged the five benchmarks. Percentile intervals are conditional on the three trained models; training-seed SD is a separate uncertainty summary and is reported separately.

Table 10: Paired-gap uncertainty. Gaps are reported in percentage points, and the 95% intervals are paired problem-cluster bootstrap confidence intervals computed from the saved problem-level records.

Budget	Comparison	Gap	Paired 95% CI
Full	IP − EOPD	+4.0	[2.1, 5.4]
CM	IP − EOPD	+2.6	[0.8, 4.2]
Full	IP − Winner CE	+2.0	[0.6, 3.1]
CM	IP − Winner CE	+1.7	[0.4, 2.9]
Full	IP − NN	+2.5	[1.0, 3.8]
CM	IP − NN	+1.8	[0.5, 3.2]

All six paired intervals remain above zero. In particular, the Full and compute-matched IP–EOPD gaps have 95% intervals of [2.1, 5.4] and [0.8, 4.2] percentage points, respectively.

D.4 Student-Scale Confirmation

Table 11: Qwen3-8B → Qwen3-0.6B-Base (Avg@8, %). Results from our MATH training runs with the 0.6B student.

Method	MATH500	AMC23	Minerva	AIME24	AIME25	Avg.
OPD	50.1	24.7	20.0	2.5	1.3	19.7
EOPD	52.0	27.8	20.8	4.2	1.3	21.2
IP-OPD	55.6	31.3	23.0	5.4	2.9	23.6

The 0.6B run tests a larger source–student capacity gap under MATH training.

E Additional Ablations and Sensitivity

E.1 Core Ablations and Sample-Yield Controls

All variants used the same type and budget of candidates, although trajectories diverged as policies changed. Always-imitate minimized $-\log \pi(z^I \mid s)$ on all candidates; success-only used that loss when $R^I = 1$. Positive-only retained the pairwise objective only when $(R^I, R^\pi) = (1, 0)$; full IP also reversed the preference for $(0, 1)$. Branch generation, joint likelihood, and four updates per nonempty buffer were preserved across variants. Accepted-pair and supervised-token counts were recorded. The endpoint comparison is reported once in Table 3a; we do not repeat the same absolute scores in the appendix. The definitions above and the exposure control below specify the additional implementation details. For the exposure control, we subsampled a fixed full-method buffer to the positive-only pair count using outcome-independent indices. This isolated optimization exposure without adding rollouts; it was not treated as a full on-policy training comparison.

E.2 Small-Scale Parameter Study

Repeated suffixes. The $K = 2, 4$ ablations follow Appendix B.3, retaining the original outcome as sample one. Both blocks remain fixed; advantage-reference banks instead resample native first blocks.

Length, selector and objective. For $L_{\max} = 64, 128, 256$, we used $H = 64, 32, 16$ to preserve the 4,096-token budget and recorded realized selected states and prefix-token positions. The alter-

Table 12: MATH-2K development sensitivity. One epoch, seed 42. To keep this supporting sweep separate from final test evidence, we report only the MATH-dev-500 selection score and relative GPU-hour cost.

Factor	Value	Dev Avg@8	Cost ratio
Default (1-epoch endpoint)	—	63.4	1.00
K	2	63.5	1.45
K	4	63.6	2.34
N_s	1	62.7	0.78
N_s	4	63.6	1.49
β	0.05	63.1	1.00
β	0.2	62.9	1.00
$L_{\max}$	64	62.8	0.94
$L_{\max}$	256	63.0	1.11
T_q	0.7	63.0	0.99
T_q	1.3	62.8	1.03

native constant-hazard run used $2/31$, capped at two states with no outcome-based refill. Length-normalized scoring used $\log \pi(z^I \mid s)/|z^I| - \log \pi(z^\pi \mid s)/|z^\pi|$ and a scale-matched development grid $\beta \in \{6.4, 12.8, 25.6\}$ with the same number of trials. The sign-permutation control shuffled labels within each buffer while preserving counts; one-sign buffers remained unchanged and were logged. Representative outcomes from these checks are summarized together with reward and boundary perturbations in Table 14.

E.3 Is a Source Proposal Needed? Native–Native Outcome Preferences

NN changes only the proposal to $q_k = \pi_k$, keeping the Full-MATH configuration, tuning grid, selector, joint likelihood, loss masks and update rule of Appendix C.6. At each selected prefix, branch 1 reuses the native block/suffix and branch 2 samples independently:

$$z_1, z_2 \overset{\text{iid}}{\sim} \pi_k(\cdot \mid s), \qquad R_i = R(s, z_i, \text{then follow } \pi_k). \tag{48}$$

Both receive the same remaining horizon. Retain $R_1 \neq R_2$, set $y = \operatorname{sign}(R_2 - R_1)$ and minimize

$$\mathcal{L}_{\text{NN}} = \mathbb{E}_{\mathcal{B}_k^{\text{NN}}} \operatorname{softplus}(-0.1y[\log \pi_\theta(z_2 \mid s) - \log \pi_\theta(z_1 \mid s)]) . \tag{49}$$

Identical blocks are not resampled; inactive pairs are not refilled. NN uses no teacher, hints or search, unlike the stronger-inference self-source in Appendix G. Although mean native advantage is zero, sampled blocks can differ in value and yield informative preferences. Second-branch generation counts as student compute. Each arm collects its own evolving states; Full/CM endpoints follow the common accounting rule.

Table 13: Native–native control. Measured three-seed Avg@8 (%); Macro includes sample SD across the executed runs. CM columns give the budget caps used for checkpoint selection.

Budget	Pair	MATH	AMC	Minerva	AIME24	AIME25	Macro	GPU-h
Full	NN	70.1	43.9	31.4	11.3	7.5	32.8 ± 0.4	90.6
Full	SN (IP)	72.4	48.0	33.5	13.5	9.4	35.4 ± 0.1	99.0
CM	NN	69.4	42.8	30.9	10.8	7.1	32.2 ± 0.4	≤ 77.4
CM	SN (IP)	71.2	45.5	32.5	12.5	8.3	34.0 ± 0.4	≤ 77.4

Paired uncertainty for the source–native versus native–native comparison appears in Table 10; seed aggregation conventions are given in Appendix D.3.

E.4 Compact Robustness Summary

The checks below are supporting MATH-2K, one-epoch, seed-42 perturbations. Rather than repeat full test vectors for each small variant, Table 14 reports only the change in Avg.2, the equal-weight

average of MATH500 and AMC23, relative to the default endpoint (52.5). This presentation is intended to show direction and scale rather than promote these single-seed checks to primary results.

Table 14: Representative robustness checks. ΔAvg.2 is in percentage points relative to the default MATH-2K endpoint. The sign-permutation row is a negative control; the remaining rows test scoring, selection, reward noise, and continuation boundaries.

Check	Setting	ΔAvg.2
Outcome sign	Permuted within buffer	-7.5
Score	Length-normalized	-0.9
Selector	Constant online hazard	-0.3
Reward corruption	$\epsilon = 0.05$	-0.3
Reward corruption	$\epsilon = 0.20$	-1.8
Boundary	Newline stopping rule	-0.1

The negative control produces the expected large degradation. The remaining perturbations are modest at this scale, while stronger symmetric reward noise reduces performance more noticeably. Appendix A.5 separately characterizes the population effect of independent symmetric reward corruption.

F Intervention Diagnostics

F.1 Independent Advantage Reference and Score Definitions

We fixed the epoch-one IP checkpoint from seed 42. For each of 500 diagnostic problems, we generated at most four native trajectories and kept the first state nominated online; unrepresented problems were retained as such without retrospective refill. We recorded one source proposal and its independent score $S_{\mathrm{IP}} = R^I - R^\pi$. Fresh 16-sample banks produced

$$\widetilde{A}_{16}(s, z^I) = \frac{1}{16}\sum_j \widetilde{R}^I_j - \frac{1}{16}\sum_j \widetilde{R}^N_j. \tag{50}$$

Native references resampled their first block under the same remaining horizon and training sampling; no reference outcome entered the score or training. Sign AUROC excluded zero reference differences. All item scores and eligible class labels were recorded. Proxies used negative teacher entropy, positive student entropy and $D_{\mathrm{KL}}(\pi_k \| q)$ at s, and mean teacher log likelihood of the proposal. Temperature was 1.0 over the full vocabulary; orientations were fixed independently of outcomes. Spearman used all references; Precision@20% used $\widetilde{A}_{16} > 0$ and fractional selection at tied scores. All scores shared states/proposals; logit proxies used only the same-tokenizer setting.

Table 15: Signal quality. Precision@20% is a percentage. IP observes one-pair outcomes; proxies do not.

Signal	AUROC	Spearman	Precision@20%
Teacher entropy	0.57	0.12	59.0
Student entropy	0.60	0.18	63.0
Teacher-student KL	0.62	0.23	66.0
Block likelihood	0.64	0.27	68.0
IP paired outcome	0.79	0.49	82.0

The paired score has AUROC 0.79 and Spearman 0.49. A preselected 100-problem subset repeated the reference with 64 samples per branch to assess near-zero reference uncertainty, without tuning the main method.

F.2 Negative Outcomes, Negative Advantage, and State Specificity

A negative observed pair need not imply negative expected advantage. For each repeated-reference item, we formed 97.5% Clopper–Pearson intervals $[l_I, u_I]$, $[l_N, u_N]$ and the at-least-95% Bonferroni

difference interval $[l_I - u_N, u_I - l_N]$. Its upper bound below zero indicates supported negative advantage; its lower bound above zero indicates supported positive advantage; otherwise the item is unresolved. Classifications are pointwise rather than multiple-testing-adjusted discoveries. The problem-level diagnostic records were retained together with these pointwise classifications; the table below reports the observed fractions. For 200 preselected problems, each of which yielded an

Table 16: Diagnostic counts. Observed counts from the executed diagnostic runs; supported signs use independent reference intervals, not single-pair outcomes.

Statistic	Numerator / denominator	Percent
One-pair negative	49/500	9.8
Supported negative advantage	23/500	4.6
Resolved advantage sign	190/500	38.0
Mixed-sign states (four proposals)	42/200	21.0

eligible state, four proposals received 32 fresh suffixes each against a shared independent 16-rollout native bank. Mixed-sign states contained both supported-positive and supported-negative proposals (42/200); a state-only score cannot distinguish these proposals. The 49/500 observed negatives and 23/500 supported negatives measure different events: one reward difference versus expected advantage.

Positions and training dynamics. We binned prefix tokens by the fixed 4,096-token cap: early $[0, 1/3)$, middle $[1/3, 2/3)$, and late $[2/3, 1]$, and recorded bin counts. APR included all generated pairs and was aggregated over fixed 20-round windows, with the final shorter window retained. Positive and negative fractions summed to APR. A separate fixed-block reference compared learned margin signs with Q gaps; this diagnostic evaluates the parametric learner rather than establishing monotonic policy improvement.

F.3 Reliability of Concrete-Block Preference Directions

We fixed both blocks and estimated $D = Q^{\pi_k}(s, z^I) - Q^{\pi_k}(s, z^\pi)$ using 64 fresh suffixes per branch, the same remaining horizon and independent reference RNGs. Pointwise intervals used the preceding Clopper–Pearson/Bonferroni construction; resolution required exclusion of zero. Separate $K = 1, 2, 4$ score banks supplied non-tied signs, whose agreement was measured separately on reference-positive/negative items. Native first blocks were not resampled: this tests concrete-block preferences, not state-baseline advantage.

Table 17: Concrete-block reliability. Observed correct/eligible counts (%). Of 500 items, 200 have a resolved reference (120 positive, 80 negative); 300 remain unresolved. Each K uses one recorded score bank per item. Ties plus the two agreement denominators equal 200.

K	Positive agreement	Negative agreement	Score ties
1	53/64 (82.8)	33/42 (78.6)	94/200 (47.0)
2	69/78 (88.5)	44/52 (84.6)	70/200 (35.0)
4	85/92 (92.4)	56/62 (90.3)	46/200 (23.0)

Agreement rises with K at the cost of additional rollouts. Error rates apply only to non-tied scores on resolved references, which themselves have finite uncertainty. The 40.0% resolution here and 38.0% for advantage use different estimands and budgets. Default training uses no confidence filter.

G Source Instantiations and Interface Compatibility

The source-interface experiments used MATH-2K, one epoch, seed 42, the 1.7B-Base student, and the complete MATH500/AMC23 evaluation sets. The control always imitated the same proposal procedure and candidate budget. External proposals used non-thinking Qwen3-8B. Multi-source proposals independently selected Qwen3-4B or Qwen3-8B with probability 0.5 per state, with one proposal and no learned router. Source identities were logged only for accounting.

Self proposals. Four independent complete student continuations were answer-normalized without gold labels. We chose the most frequent answer, then its earliest candidate, breaking vote ties by first index; if all failed parsing, candidate one was used. Only its first block was returned. A fresh suffix supplied R^I; all four search continuations were charged. Search does not guarantee a better proposal.

Cross-architecture source. We used `deepseek-ai/DeepSeek-R1-Distill-Llama-8B` with the prompt in Appendix C.3. We streamed at most 256 source tokens, decoded them, and retained the longest text prefix of at most 128 student tokens without repairing content. Discarded tokens were included in compute accounting. Source EOS was treated as a student terminal action; source chat-control tokens were not inserted as reasoning, and native token IDs were preserved for scoring. This implementation therefore used text proposals rather than cross-tokenizer logit alignment. Within-source gaps provide the interface-compatibility comparison. Self-search consumed

Table 18: Controlled source comparisons (Avg@8, %). MATH-2K, one epoch, seed 42. IP GPU-h includes source search and loading.

Source	Always imitate		IP-OPD		IP GPU-h
	MATH500	AMC23	MATH500	AMC23	
External	62.7	35.0	65.6	39.4	12.0
Self	59.9	30.6	62.3	34.4	20.0
Multi	62.9	36.3	65.9	40.0	11.5
Cross-arch.	61.7	33.8	64.5	37.8	14.0

20.0 GPU-h versus 12.0 for external proposals; different source settings were not compute matched.

H Empirical Compute Accounting

H.1 Accounting Boundaries

GPU work was synchronized at timing boundaries and peak-memory counters were reset. The measured runtime included loading, snapshot synchronization, prefill, generation, scoring, updates, checkpoint writing, and reward processing. Reserved idle devices were counted in GPU-hours; wall time was recorded separately without summing overlapping timers. Final evaluation cost was recorded separately from training.

Source/native generation, added suffixes, privileged/self search, teacher-scored positions, prefill and loss-bearing tokens were logged separately. Token counts are not treated as FLOPs across different model sizes.

Table 19: Measured stage allocation (GPU-hours). Suffix+ includes added suffix generation, loading, and miscellaneous processing. Stage times come from synchronized timing logs; wall time is the elapsed time with four reserved devices.

Method	Source gen.	Native gen.	Teacher scoring	Update	Suffix+ other	Total GPU-h	Wall hours
Vanilla OPD	0.0	43.2	12.0	13.6	3.2	72.0	18.0
EOPD	0.0	43.2	14.0	17.0	3.2	77.4	19.4
IP-OPD	6.0	43.2	0.0	24.0	25.8	99.0	24.8

Full IP-OPD consumed about $1.28\times$ the EOPD GPU-hours and $1.38\times$ the OPD GPU-hours. Removing dense teacher scoring did not remove the expense of additional student continuations.

H.2 Common-Budget Checkpoint Rule

For CM, we selected the latest complete-round checkpoint below the same-seed full EOPD GPU-hour budget, without test-based selection or shortened-schedule retuning. The seed-specific EOPD

caps were 77.1, 77.4, and 77.7 GPU-hours for seeds 42–44; thus the reported ≤ 77.4 entries denote the mean cap rather than an achieved cost. Existing full-training costs are 94.2 GPU-h for Winner CE, 90.6 for NN, and 99.0 for IP. For every CM run, the retained checkpoint cost was below its cap and the next complete round would have exceeded it. Prompt exposure and unique active pairs were logged, and no artificial delay or redundant backward passes were added to equalize costs.

H.3 Realized Supervision Yield

Table 20 reports pooled counts and ratios. Each active pair is counted once regardless of optimizer replay; ratios are not averages of per-round ratios.

Table 20: Realized supervision yield. Counts and pooled ratios from the full IP-OPD training run over three passes through 7,500 prompts. Training cost for this run was 99.0 GPU-hours.

Quantity	Value
Native trajectories	22,500
Selected states	13,950
Unique active pairs	4,743
Selected states / trajectory	0.62
Pooled active-pair rate (%)	34.0
Unique active pairs / trajectory	0.21
Unique active pairs / training GPU-hour	47.9

The 34.0% pooled rate lies between the observed early/late window rates of 41.0% and 29.2%; the MATH-2K noise-control rate is a different run-level quantity. These counts show that $N_s = 2$ is a cap rather than the realized supervision density. We additionally tracked cumulative unique active pairs and GPU-hours at fixed epoch endpoints for the learning-efficiency analysis.