VERIFY BEFORE YOU DISTILL: PROMPT-LEVEL TEACHER GATING FOR ON-POLICY DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy distillation (OPD) accelerates post-training by providing dense token-level supervision from a frozen teacher on the student's own rollouts. Vanilla OPD applies this supervision uniformly across prompts, without checking whether the teacher is reliable for each prompt. Because reverse KL is mode-seeking, a confidently wrong teacher can induce a strong yet misleading update. Distributional proxies, such as entropy or teacher–student likelihood agreement, measure uncertainty or agreement but do not directly verify outcome correctness. We introduce Teacher-Gated On-Policy Distillation (TGOPD), built on the principle that teacher reliability should be verified at the prompt level before dense supervision is admitted. TGOPD estimates reliability from a small set of verifier-scored teacher probes and routes each prompt exclusively to dense OPD when the reliability check passes or to verifier-grounded GRPO otherwise. Across 4B and 35B students in mathematics, code, and instruction following, TGOPD improves the within-domain average over Vanilla OPD in all six single-domain settings and achieves higher seven-benchmark averages at both scales under multi-domain training. By using otherwise-idle teacher capacity for reliability estimation, TGOPD also reduces teacher-side compute waste in asynchronous OPD, increasing teacher-node GPU utilization from 9.8% to 78.9% in the measured 4B single-domain run.

1 INTRODUCTION

On-policy distillation (OPD) is a compute-efficient approach to post-training language models (Agarwal et al., 2024; Gu et al., 2024; Lu & Thinking Machines Lab, 2025). Reinforcement learning with verifiable rewards (RLVR) assigns a single scalar reward to an entire trajectory. In contrast, OPD samples rollouts from the student and uses a stronger teacher to provide a reverse-KL learning signal at every token, transforming sparse outcome supervision into dense per-token guidance. This dense feedback can enable the student to reach teacher-level accuracy roughly an order of magnitude faster than RLVR (Lu & Thinking Machines Lab, 2025).

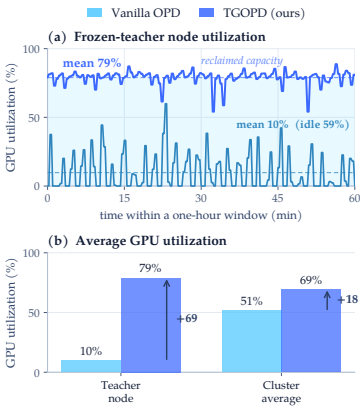

Figure 1: **Teacher-side utilization in asynchronous OPD. (a)** TGOPD fills idle intervals with reliability probes. **(b)** Mean teacher utilization rises from 9.8% to 78.9%.

Despite this training efficiency, asynchronous OPD leaves much of the teacher's compute idle. In a typical asynchronous deployment, the student generates rollouts on dedicated inference nodes, while a stronger, domain-specialized frozen teacher scores completed rollouts on a separate node. A scoring pass requires only a forward evaluation over tokens that have already been generated. It is therefore much cheaper than autoregressive decoding and cannot begin until a batch of student rollouts is ready, leaving the teacher's accelerators idle between scoring batches. Figure 1 quantifies this inefficiency in a 4B run: the teacher node averages 9.8% GPU utilization and remains below 5% utilization for 59%

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.

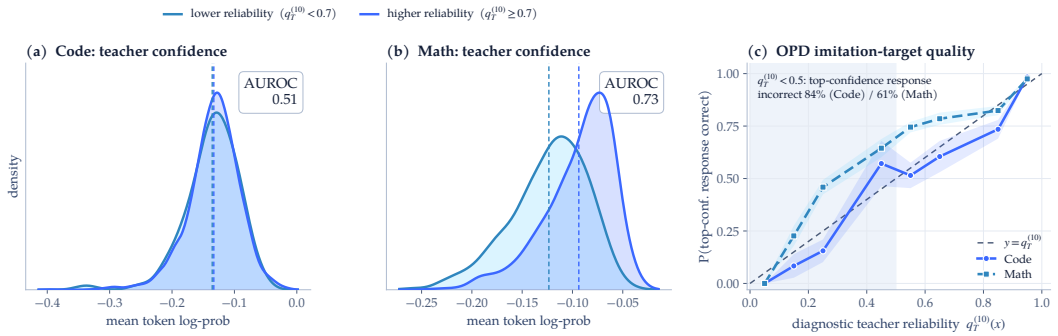

Figure 2: **Teacher confidence does not consistently indicate prompt-level reliability.** We use ten offline teacher samples per prompt only for this diagnostic; training uses $K_T = 3$. Confidence separates higher- and lower-reliability prompts at AUROC 0.51 on code and 0.73 on math. For $q_T^{(10)} < 0.5$, the highest-confidence response is incorrect on 84% of code and 61% of math prompts.

of the measured hour, while the rollout and training nodes remain busy. This observation motivates using the otherwise-idle teacher capacity to improve the training signal itself.

The idle capacity can be used to address a more fundamental weakness of Vanilla OPD: teacher supervision is admitted without a prompt-level reliability check. Reverse KL is mode-seeking and concentrates the student on the teacher's high-probability behavior (Gu et al., 2024; Lu & Thinking Machines Lab, 2025). This property makes OPD efficient when the teacher is reliable, but it can also amplify confident errors. Reliability varies by prompt: a globally informative teacher signal may not be locally exploitable (Li et al., 2026), and distillation from a much stronger reasoner may fail to benefit a capacity-limited student (Li et al., 2025b). Domain routing alone does not resolve this issue because existing multi-teacher frameworks use the selected teacher's dense reward without verifying its reliability on the particular prompt (Xiaomi LLM-Core Team, 2026).

Existing reliability-aware OPD methods regulate supervision using distributional signals such as entropy or teacher–student agreement (Jin et al., 2026; Xing et al., 2026; Ko et al., 2026), or verifier feedback at the trajectory or token level (Akhondzadeh et al., 2026; Yang et al., 2026a; Qu et al., 2026). These approaches improve local supervision but do not directly estimate whether the selected teacher is reliable for the current prompt. TGOPD instead uses repeated verifier-scored teacher outcomes to make a prompt-level admission decision before dense supervision is applied.

Figure 2 evaluates confidence against $q_T^{(10)}$, the verifier pass rate over ten offline teacher samples. Confidence is nearly uninformative on code (AUROC 0.51) and only moderately informative on math (0.73); when $q_T^{(10)} < 0.5$, even the highest-confidence response is wrong on 84% and 61% of code and math prompts, respectively. Reverse KL can therefore amplify precisely these confident errors.

Teacher-Gated On-Policy Distillation (TGOPD) uses the teacher's idle capacity to estimate prompt-level reliability before selecting a supervision signal. As shown in Figure 3, the teacher generates K_T probe rollouts for the same prompt; a verifier scores them; and their pass rate $q_T(x) = K_T^{-1} \sum_k r_k$ provides the online reliability estimate. If $q_T(x) \geq \tau$, TGOPD uses dense OPD alone. Otherwise, it rejects the teacher signal and uses verifier-grounded GRPO alone, provided that the student rollout group contains reward variation. The two branches are routed rather than blended. Because the probes run primarily while the teacher would otherwise be idle, teacher-node GPU utilization rises from 9.8% to 78.9% and cluster utilization from 51.5% to 69.5%, with modest measured end-to-end overhead.

Contributions. We make three contributions. **(1)** We formulate prompt-level teacher reliability as the admission criterion for dense OPD and estimate it online with verifier-scored probes. **(2)** We introduce a mutually exclusive router between dense OPD and verifier-grounded GRPO that reuses idle teacher capacity. **(3)** Across 4B/35B students in mathematics, code, and instruction following, TGOPD improves all six single-domain settings and the seven-benchmark MOPD averages.

2 PRELIMINARIES

Setup and notation. Let π_θ denote the trainable student, π_{old} its frozen rollout-policy snapshot, and π_T a frozen teacher; in a multi-teacher setting, $\pi_T = \pi_{T,\text{dom}(x)}$ is the teacher routed to the domain of prompt x. For each prompt $x \sim \mathcal{D}$, the student samples a group of G on-policy rollouts $\{y^i\}_{i=1}^G \sim \pi_{\text{old}}(\cdot \mid x)$, and a binary verifier $r(x, y) \in \{0, 1\}$ scores each rollout against a ground-truth outcome: normalized answer matching for CodeI/O output prediction, execution-based verification of predicted inputs against reference programs for CodeI/O input prediction, and rule-based verification for mathematics and instruction following. We write y_t^i for the t-th token of rollout i and $\text{sg}[\cdot]$ for the stop-gradient operator.

On-policy distillation (OPD). OPD supervises every token of the student's own rollouts by minimizing the mode-seeking reverse KL $\mathbb{KL}(\pi_\theta \,\|\, \pi_T)$. For each PPO update, the per-token advantage is computed from the teacher and rollout-policy log-probabilities (Gu et al., 2024; Lu & Thinking Machines Lab, 2025; Xiaomi LLM-Core Team, 2026):

$$\hat{A}_{i,t}^{\text{OPD}} = \beta \,\text{sg}\big[\log \pi_T\big(y_t^i \mid x, y_{<t}^i\big) - \log \pi_{\text{old}}\big(y_t^i \mid x, y_{<t}^i\big)\big], \tag{1}$$

where $\beta > 0$ sets the scale of the teacher signal. The advantage is computed once from the frozen rollout snapshot and treated as stop-gradient during optimization. Every token therefore receives its own dense, individually weighted learning signal, which contributes to OPD's sample efficiency relative to learning from a single trajectory-level reward. Being mode-seeking, the objective concentrates the student on the teacher's highest-probability behavior rather than spreading probability mass over alternatives. Note that equation 1 depends only on π_T and π_{old}, not on the verifier r: it measures how far the rollout policy is from the teacher but does not directly assess whether the teacher's output is correct. When π_T is unreliable on x, the resulting dense gradient can therefore be misleading.

GRPO with verifiable rewards. When teacher supervision is unavailable or withheld, the student can instead learn from the verifier alone. GRPO (Shao et al., 2024) centers the reward within each group; we omit the standard-deviation normalization (Liu et al., 2025):

$$\hat{A}_i^{\text{GRPO}} = r\big(x, y^i\big) - \frac{1}{G}\sum_{j=1}^{G} r\big(x, y^j\big), \qquad \hat{A}_{i,t}^{\text{GRPO}} = \hat{A}_i^{\text{GRPO}} \quad \forall\, t. \tag{2}$$

Unlike the token-specific OPD advantage in equation 1, this signal assigns the same scalar advantage to every token in a rollout. It is therefore trajectory-level rather than token-specific, but remains verifier-grounded: its sign is determined by the verifier outcome relative to the group mean.

The gap. OPD provides dense but unverified teacher supervision, whereas GRPO provides verifier-grounded but trajectory-level feedback. TGOPD chooses between these signals per prompt according to verified teacher reliability, rather than applying either rule uniformly.

3 TEACHER-GATED ON-POLICY DISTILLATION

TGOPD audits prompt-level teacher reliability and routes each prompt to dense OPD when the audit passes or verifier-grounded GRPO otherwise (Figure 3); the two signals are never blended.

3.1 TEACHER RELIABILITY PROBE AND GATE

The gate requires an estimate of teacher reliability for each prompt, which the vanilla OPD loop does not provide. We define this quantity below and estimate it with a small number of teacher samples.

Reliability and its estimator. We define the teacher's reliability on a prompt as its expected verifier reward,

$$R_T(x) \;=\; \mathbb{E}_{y \sim \pi_T(\cdot|x)}\big[r(x, y)\big] \;\in\; [0, 1], \tag{3}$$

that is, the probability that a sample from the teacher solves x. Since R_T is not available in closed form, we estimate it with a small teacher probe: the teacher independently generates K_T rollouts

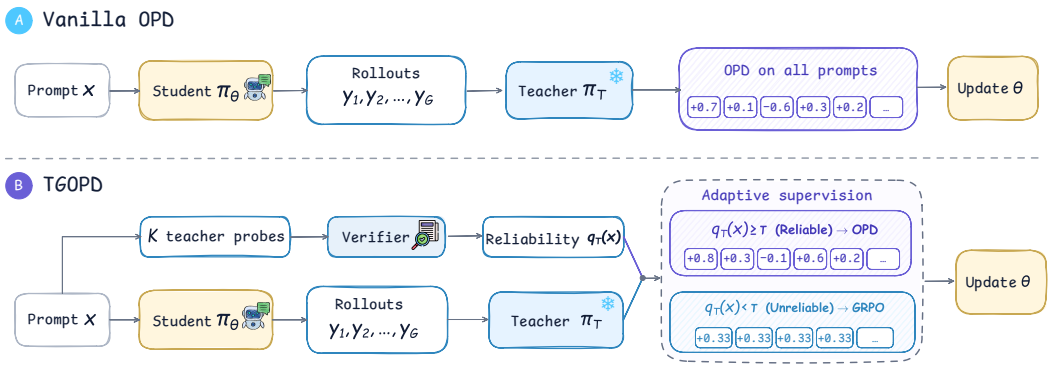

Figure 3: **TGOPD overview.** Vanilla OPD applies dense teacher supervision to every prompt. TGOPD uses otherwise-idle teacher capacity to generate K_T verifier-scored probes, estimates $q_T(x)$, and routes each prompt exclusively to OPD when $q_T(x) \geq \tau$ or GRPO otherwise.

$\{\hat{y}^k\}_{k=1}^{K_T} \sim \pi_T(\cdot \mid x)$ on the same prompt. The verifier used for student rollouts scores each teacher rollout, and their empirical pass rate is (Figure 3B)

$$q_T(x) \;=\; \frac{1}{K_T} \sum_{k=1}^{K_T} r\big(x, \hat{y}^k\big)\,. \tag{4}$$

The empirical pass rate is an unbiased estimator of $R_T(x)$. We use $K_T = 3$ in the main experiments and a larger budget for the threshold study in Section 4.5. Appendix A gives its finite-sample properties.

Verifier-grounded reliability signal. The estimator $q_T(x)$ uses task outcomes rather than the teacher's own confidence. Entropy and teacher–student likelihood agreement are functions of model distributions alone: they can quantify uncertainty or compatibility, but cannot by themselves distinguish equally confident correct and incorrect answers. Evaluating r on completed teacher outputs provides direct outcome evidence. Unlike verifier-aware token or trajectory gates, $q_T(x)$ aggregates repeated teacher outcomes into a prompt-level reliability estimate before the supervision branch is selected.

The reliability gate. A hard gate admits the teacher only when its estimated reliability reaches a threshold $\tau \in (0, 1]$ (Figure 3B, "reliability gate"):

$$g(x) \;=\; \mathbb{1}[\, q_T(x) \;\geq\; \tau \,] \;\in\; \{0, 1\}. \tag{5}$$

Since $q_T(x)$ takes values in $\{0, 1/K_T, \ldots, 1\}$, the gate opens exactly when at least $\lceil \tau K_T \rceil$ of the K_T probes pass; only the pair $\big(\lceil \tau K_T \rceil,\, K_T\big)$, and not τ in isolation, is operationally meaningful. We use $K_T = 3$ and $\tau = 2/3$ in all main experiments, i.e. a two-of-three majority. Section 4.5 sweeps the threshold at $K_T = 5$ and finds a broad optimum around a simple majority. This result suggests that a coarse reliability decision is sufficient; the gate need not rank prompts precisely.

3.2 Gate-Conditioned Supervision Routing

For each prompt, the gate selects one of the two supervision signals defined in §2. When it is open, the dense OPD signal controls the update and the trajectory-level verifier reward does not enter the gradient. When the gate is closed, teacher supervision is withheld and the verifier-grounded GRPO signal controls the update. Withholding teacher supervision therefore does not necessarily discard the training prompt. Whenever the student's rollout group contains reward variation, the group-relative fallback assigns positive advantage to above-average attempts and negative advantage to below-average ones (Figure 3B, lower signal box). If every rollout receives the same reward, however, the centered GRPO advantage is exactly zero and that prompt produces no update. For pipeline regularity, both candidate advantages are formed for every prompt before the update. This

eager computation does not imply additive supervision: the binary gate selects exactly one candidate, giving the per-token advantage

$$\hat{A}_{i,t}^{\mathrm{TGOPD}} \;=\; g(x)\,\hat{A}_{i,t}^{\mathrm{OPD}} \;+\; \big(1-g(x)\big)\,\hat{A}_{i,t}^{\mathrm{GRPO}}. \tag{6}$$

Because $g(x) \in \{0,1\}$, equation 6 is a selector rather than an interpolation: OPD and GRPO never jointly supervise the same prompt. Vanilla OPD and pure GRPO are its two degenerate endpoints, obtained by fixing $g(x)=1$ and $g(x)=0$, respectively, for every prompt.

The resulting advantage enters the standard clipped policy-gradient surrogate. Using the frozen rollout-policy snapshot π_{old}, define $\rho_{i,t}(\theta) = \pi_\theta(y_t^i \mid x, y_{<t}^i)/\pi_{\mathrm{old}}(y_t^i \mid x, y_{<t}^i)$. We optimize

$$\mathcal{L}_{\mathrm{TGOPD}}(\theta) = -\,\mathbb{E}_{\substack{x\sim\mathcal{D} \\ \{y^i\}_{i=1}^{G}\sim\pi_{\mathrm{old}}(\cdot|x)}}\left[\frac{1}{\sum_i |y^i|}\sum_{i=1}^{G}\sum_{t=1}^{|y^i|}\min\Big(\rho_{i,t}(\theta)\hat{A}_{i,t}^{\mathrm{TGOPD}},\; \mathrm{clip}\big(\rho_{i,t}(\theta),1-\epsilon,1+\epsilon\big)\hat{A}_{i,t}^{\mathrm{TGOPD}}\Big)\right]. \tag{7}$$

The routed advantage is computed once from the rollout batch and held fixed throughout the update. The asynchronous implementation additionally uses IcePop to suppress excessive train–inference mismatch; this system-level correction does not alter the supervision-routing objective above (§4.1). Appendix A gives the fallback-normalization convention.

Design choices. The gate acts at the prompt level: a prompt is supervised either entirely by OPD or entirely by the verifier-grounded fallback, never by a per-token or additive mixture. This avoids entangling two signals of different density and provenance within a single trajectory, where the effective supervision would depend on an arbitrary interpolation weight. The gate also withdraws teacher supervision only where the audit fails, so on the majority of prompts the full direction and magnitude of the dense OPD signal survives intact. Section 4.2 shows the cost of reducing the teacher's role on every prompt rather than only on those that fail the audit.

3.3 Asynchronous System Realization

Relative to vanilla OPD, TGOPD retains the same student rollout and teacher-scoring operations and adds the probe in equation 4. The teacher generates these probes concurrently with student decoding, so most audit work occupies capacity that would otherwise be idle. Teacher scoring remains unconditional, and the gate only selects which of the two already-formed advantages controls the update. Appendix A gives the complete rollout–update algorithm and system rationale; measured utilization and residual runtime overhead are reported in §4.4 and Appendix D.

4 Experiments

4.1 Experimental Setup

Models and teachers. We evaluate Qwen3.5-4B and the 3B-active Qwen3.6-35B-A3B on mathematics, code, and instruction following (IF). Each domain teacher is obtained by applying GRPO (Shao et al., 2024) to the corresponding base model and is then frozen for probing and rollout scoring. Its scores also provide the same-architecture GRPO reference in Table 1. All methods use slime (Zhu et al., 2025), the PPO objective in equation 7, and IcePop (Ling Team et al., 2025); Appendix B gives the remaining training and infrastructure settings.

Baselines. All comparisons share the initialization, teacher, data, and pipeline. We compare Vanilla OPD (Lu & Thinking Machines Lab, 2025), token-level trust-region TrOPD (Xing et al., 2026), trajectory-filtering RG-OPD (Akhondzadeh et al., 2026), and an RLSD-style baseline (Yang et al., 2026a) that uses verifier direction with teacher-scaled token magnitudes.

Training data and evaluation. Training prompts come from DAPO-Math-17K (Yu et al., 2025), CodeI/O (Li et al., 2025a), and Nemotron-Cascade 2 (Yang et al., 2026b). We evaluate on AIME 2025/2026 and HMMT-Feb 2025, LiveCodeBench and OJBench, and IFBench and IFEval; Appendix C gives the full protocols.

Table 1: In-domain distillation at two model scales. Avg. is the unweighted mean of the seven benchmarks. **Bold** marks the best distillation result within each scale; underlining marks scores above the corresponding domain teacher.

	MATH			CODE		IF		
Method	AIME 2025	AIME 2026	HMMT-Feb	LiveCodeBench	OJBench	IFBench	IFEval	Avg.
Qwen3.5-4B								
Base Model	47.8	58.1	40.5	39.4	14.4	35.9	85.3	45.91
Teacher	63.4	73.4	54.8	53.3	17.0	55.9	85.3	57.59
Vanilla OPD	61.1	71.2	54.3	45.0	18.1	47.8	84.5	54.57
TrOPD	62.6	70.4	**57.7**	**47.8**	19.4	53.3	84.4	56.51
RG-OPD	64.6	72.7	54.2	45.6	19.4	**53.5**	84.0	56.29
RLSD-style	46.8	57.0	42.4	38.8	13.8	33.4	79.8	44.57
TGOPD (ours)	**64.8**	**73.5**	55.7	47.1	**20.0**	50.4	**85.2**	**56.67**
Qwen3.6-35B-A3B								
Base Model	69.8	74.4	64.3	60.0	25.9	30.0	86.8	58.74
Teacher	76.3	79.8	65.3	62.7	27.6	52.1	90.8	64.94
Vanilla OPD	73.4	77.3	**64.1**	60.2	28.0	44.2	89.9	62.44
TrOPD	**75.0**	77.9	63.9	57.8	**28.7**	**44.5**	90.8	62.66
RG-OPD	74.8	78.1	62.8	57.0	26.9	44.2	90.6	62.06
RLSD-style	72.7	75.1	60.3	57.7	26.5	36.4	85.7	59.20
TGOPD (ours)	74.0	**78.7**	**64.1**	**64.0**	**28.7**	43.7	**91.5**	**63.53**

Table 2: TGOPD applied to multi-domain OPD (MOPD). Each run trains a single student on math, code, and IF simultaneously, routing prompts to domain-specialist teachers. Results are reported at 199 training steps with $K_T = 3$, $\tau = 2/3$. **Bold**: best per column at the displayed precision, including ties. Avg. is computed before display rounding.

		MATH			CODE		IF	
Method	Avg.	AIME 25	AIME 26	HMMT-Feb	LCB	OJBench	IFBench	IFEval
Qwen3.5-4B								
MOPD	53.40	**58.1**	63.8	50.7	**50.5**	18.1	48.8	83.9
MOPD + TGOPD	**54.54**	**58.1**	**67.6**	**51.3**	50.0	**19.0**	**51.7**	**84.1**
Qwen3.6-35B-A3B								
MOPD	60.99	**73.1**	77.3	60.7	57.4	23.3	**44.4**	90.7
MOPD + TGOPD	**61.94**	73.0	**78.0**	**62.4**	**57.8**	**27.6**	44.0	**90.8**

4.2 MAIN RESULTS

TGOPD has the highest seven-benchmark average at both scales (Table 1): 56.67 at 4B and 63.53 at 35B. These results exceed Vanilla OPD by 2.10 and 1.09 points, respectively, and the strongest baseline by average, TrOPD, by 0.16 and 0.87 points. TGOPD is best on nine of 14 benchmark–scale pairs, including ties, and exceeds the corresponding teacher on seven. Exceptions include TrOPD's stronger 4B HMMT-Feb and LiveCodeBench results; TGOPD ties TrOPD on 35B OJBench and Vanilla OPD on 35B HMMT-Feb.

The gains over Vanilla OPD hold in all six domain–scale settings. At 4B, the average improvements are 2.47 points on math, 2.00 on code, and 1.65 on IF; at 35B, they are 0.67, 2.25, and 0.55, respectively. Math gains most at 4B and code at 35B. The code gains agree with the weak confidence–reliability separation in Figure 2. Domain averages nevertheless mask variation: on 35B IFBench, TGOPD scores 43.7 versus 44.2 for Vanilla OPD, while improving IFEval from 89.9 to 91.5.

On 35B LiveCodeBench, Vanilla OPD improves only 0.2 points over the base and the other baselines fall below it, whereas TGOPD reaches 64.0: +4.0 over the base, +3.8 over Vanilla OPD, and +1.3 over the teacher. At 4B, TGOPD raises LiveCodeBench from 45.0 to 47.1.

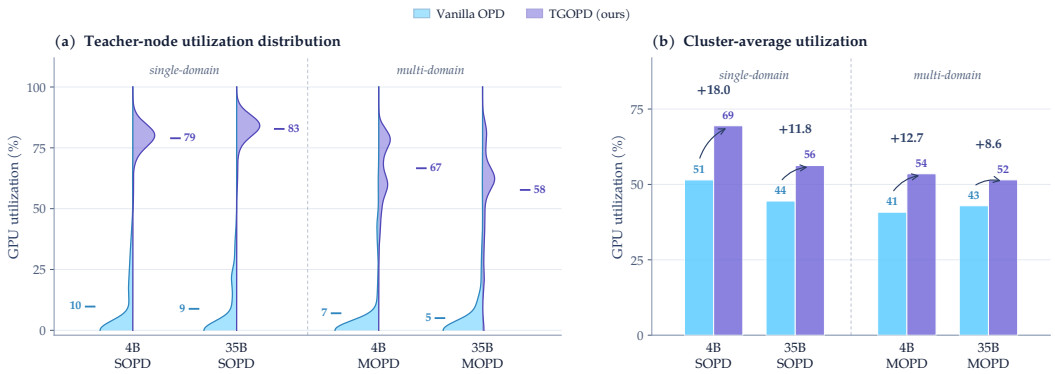

Figure 4: **GPU utilization before and after TGOPD, across four configurations. (a)** Teacher-node distributions for single-domain OPD (SOPD) and multi-domain OPD (MOPD). **(b)** Cluster-average utilization. Probing shifts teacher utilization upward in every setting; MOPD is bimodal because prompts activate different domain teachers.

The RLSD-style baseline, which reweights every prompt using verifier direction and teacher-scaled magnitudes, falls below the base on nine of 14 benchmark–scale pairs. Together, these results favor selectively retaining the full dense signal; Sections 4.5 and 4.6 examine the gate and closed-gate update directly.

4.3 Extension to Multi-Domain OPD

The experiments above evaluate TGOPD in the single-domain setting, where each run trains one student with one domain teacher. We also consider multi-domain OPD (MOPD) (Xiaomi LLM-Core Team, 2026; Ma et al., 2026), in which a single student is trained on math, code, and IF prompts simultaneously, with each prompt routed to a domain-specialist teacher. Because the gating decision is per-prompt and per-teacher, TGOPD applies to MOPD without modification: the routed teacher generates $K_T = 3$ probes on its assigned prompt, the domain verifier scores them, and the gate admits or withholds teacher supervision as before.

Table 2 reports results at 199 training steps. At the 4B scale, TGOPD improves the seven-benchmark average from 53.40 to 54.54 ($+1.14$); at the 35B scale, the average rises from 60.99 to 61.94 ($+0.95$). The largest single-benchmark gains appear on AIME 2026 ($+3.8$ at 4B), IFBench ($+2.9$ at 4B), and OJBench ($+0.9$ at 4B, $+4.3$ at 35B); the code result in particular is consistent with the single-domain finding, since confidence is least informative about teacher reliability on code. The few regressions (LCB at 4B, -0.48; AIME 2025, -0.15; IFBench, -0.41, at 35B) are all at most half a point. The same per-prompt gate therefore improves the overall benchmark average at both scales.

4.4 Reclaiming Idle Teacher Capacity

Figure 4 extends the utilization analysis to both model scales and to SOPD and MOPD. The idle pattern is structural rather than configuration-specific: teacher scoring is a single forward pass over already-generated tokens and cannot begin until the student finishes autoregressive decoding. Without probes, mean teacher utilization is only 5.0–9.8%, and 57–78% of the measured samples fall below 5% utilization. MOPD is especially idle because three domain-specialist engines share the teacher node and each runs only for prompts routed to its domain.

Adding probes raises mean teacher utilization to 78.9% and 82.8% under 4B and 35B SOPD, and to 66.6% and 57.7% under MOPD; the idle fraction falls to 0–2% in all four configurations. Student-rollout utilization changes by at most 2.1 points, showing that the increase primarily reclaims teacher capacity. Utilization alone does not determine end-to-end efficiency: in a matched 35B CodeI/O comparison, TGOPD increases mean step time by 5.9% over 500 aligned rollout cycles.

Cluster-wide utilization improves by $+18.0$ and $+11.8$ points under SOPD and by $+12.7$ and $+8.6$ under MOPD (4B and 35B, respectively). The ordering reflects both the teacher's cluster share ($1/5$

at 4B versus $1/7$ at 35B) and lower probe saturation when the MOPD teacher node is partitioned by domain. A role-wise decomposition attributes $13.8/18.0$, $10.6/11.8$, $11.9/12.7$, and $7.5/8.6$ points of the four cluster gains to the teacher, respectively. Appendix D reports the full table and one-hour traces.

4.5 Ablation: Gate Threshold

The main experiments fix $\tau = 2/3$ and $K_T = 3$. For this diagnostic ablation, we increase the probe budget to $K_T = 5$, yielding five nontrivial pass-count cutoffs and a finer-grained view of threshold sensitivity. We sweep $\tau \in \{1/5, 2/5, 3/5, 4/5, 5/5\}$ on Qwen3.5-4B mathematics runs for 99 training steps while holding all other hyperparameters fixed.

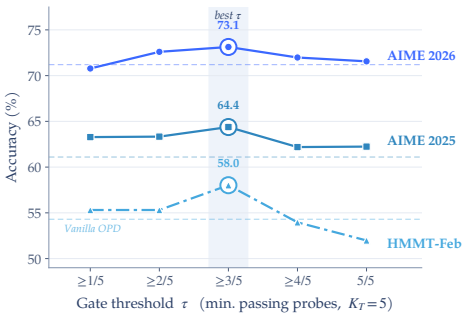

Figure 5: **Gate-threshold sweep on mathematics benchmarks** (Qwen3.5-4B, $K_T = 5$, 99 steps). Solid: TGOPD; dotted: Vanilla OPD. All three benchmarks peak at $\tau = 3/5$.

Figure 5 shows an inverted-U relationship between gate strictness and downstream accuracy. At low thresholds ($\tau \leq 2/5$), the gate admits more lower-reliability teacher signals, including confidently wrong ones. Performance still exceeds Vanilla OPD on AIME 2025 ($+2.2$) and HMMT-Feb ($+1.0$), but the gains are smaller because the gate provides little filtering. At $\tau = 3/5$, all three benchmarks peak: AIME 2026, AIME 2025, and HMMT-Feb reach 73.1, 64.4, and 58.0, exceeding Vanilla OPD by 1.9, 3.3, and 3.7 points. At high thresholds ($\tau \geq 4/5$), useful teacher signals are increasingly rejected; at $\tau = 5/5$, HMMT-Feb falls below Vanilla OPD (52.0 versus 54.3).

The peak at $\tau = 3/5$ (a 60% pass rate) is close to the default $\tau = 2/3$ ($\approx 67\%$) used in the main experiments. Under the $K_T = 5$ diagnostic setting, this suggests a broad optimum around a majority-vote criterion rather than a sharply tuned threshold.

4.6 Ablation: What the Gate Does After It Closes

The threshold ablation asks when teacher supervision should be admitted; we next ask whether the benefit of gating depends on the update used after rejection. Our main configuration uses verifier-grounded GRPO on audit-rejected prompts. We compare it with a mask-only variant that suppresses the teacher signal without applying a replacement update. Both variants use $K_T = 3$ and $\tau = 2/3$, isolating whether withholding unreliable supervision is sufficient and whether the GRPO fallback provides an additional benefit.

Table 3: Closed-gate policy averages over three SOPD mathematics benchmarks and all seven MOPD benchmarks.

	SOPD Math		MOPD	
Policy	4B	35B	4B	35B
Ungated OPD	62.20	71.60	53.40	60.99
Mask only	63.52	71.67	**54.97**	61.78
GRPO fallback	**64.67**	**72.27**	54.54	**61.94**

Both policies improve over ungated OPD in all four settings (Table 3). On single-domain mathematics, GRPO reaches 64.67 at 4B and 72.27 at 35B, compared with 63.52 and 71.67 for masking and 62.20 and 71.60 for ungated OPD. The fallback exceeds masking on all three 4B mathematics benchmarks and on both 35B AIME benchmarks, whereas masking retains a small advantage on 35B HMMT-Feb. Under multi-domain training, the ordering changes with scale: masking is stronger at 4B (54.97 versus 54.54), while GRPO is stronger at 35B (61.94 versus 61.78). The 4B mask-only variant leads on all three mathematics benchmarks and LiveCodeBench, whereas the fallback recovers the average lead at 35B. The clearest domain-level difference appears on OJBench, where GRPO exceeds masking at both scales (19.0 versus 18.5; 27.6 versus 27.2), consistent with execution-based verification providing a direct correctness signal.

Audit-rejected prompts can still support verifier-grounded learning. Mixed-outcome student groups, which yield non-zero group-relative advantages, account for 79.0% and 84.7% of rejected mathematics prompts at 4B and 35B. Relative to masking, the GRPO fallback changes the average by $+1.15$ and $+0.60$ points on SOPD mathematics at 4B and 35B, and by -0.43 and $+0.16$ points under MOPD. Thus, teacher admission is the consistent source of improvement, while the additional benefit of the replacement update is setting-dependent. We retain GRPO in the main configuration because rejected prompts often preserve a usable verifier-grounded learning signal, but the best closed-gate update remains a design choice. Appendix E reports the per-benchmark policy comparison, and Appendix F gives the reward-variation analysis.

5 Related Work

5.1 On-Policy Distillation and Multi-Teacher Frameworks

On-policy distillation (OPD) trains on student rollouts using token-level teacher supervision through mode-seeking reverse KL (Gu et al., 2024; Agarwal et al., 2024; Song & Zheng, 2026), improving sample efficiency over a sequence-level reward (Lu & Thinking Machines Lab, 2025). Multi-teacher OPD instead routes prompts to domain experts (Xiaomi LLM-Core Team, 2026; Ma et al., 2026; Sun et al., 2026b).

Teacher usefulness varies by instance: knowledge may not be locally exploitable (Li et al., 2026), student prefixes may drift from teacher-supported states (Fu et al., 2026), and capacity-limited students may not benefit from much stronger reasoners (Li et al., 2025b). Multi-teacher systems route by domain but do not verify the selected teacher per prompt. TGOPD instead estimates prompt reliability from repeated verifier-scored rollouts before admitting dense supervision.

5.2 Reliability-Aware Distillation and the Confidence–Correctness Gap

Distributional approaches regulate OPD using entropy (EOPD), teacher–student agreement (TrOPD), or reward clipping and sampling (REOPOLD) (Jin et al., 2026; Xing et al., 2026; Ko et al., 2026). REOPD extrapolates rewards from compatibility signals without an external verifier (Sun et al., 2026a); when no external grader is available, GATES uses agreement among sampled tutor traces as a trajectory-level reliability proxy (Stein et al., 2026). These methods address uncertainty, diversity, or distribution mismatch but do not directly verify outcome correctness. Teacher entropy can also conflate non-viable uncertainty with benign solution diversity (Liu et al., 2026), while KL or entropy does not fully explain position-dependent degradation (Zhou et al., 2026).

Outcome-aware methods operate more locally: RG-OPD conditions trajectory-level teacher supervision on verifier–likelihood agreement (Akhondzadeh et al., 2026), RLSD uses environmental correctness for update direction and self-distillation for magnitude (Yang et al., 2026a), and SPOT verifies teacher branches using verifier-scored student continuations (Qu et al., 2026). Related methods select or repair targets using reward-ranked responses, verifier-filtered self-generated solutions, or teacher-ranked token proposals (Dong et al., 2023; Singh et al., 2024; Xu et al., 2025). TGOPD instead verifies repeated complete teacher rollouts before dense supervision and routes each prompt to OPD or GRPO rather than calibrating trajectories, tokens, or branches.

6 Conclusion

This paper addresses Vanilla OPD's unconditional trust in teacher supervision with TGOPD, which uses verifier-scored teacher probes to make a prompt-level routing decision. TGOPD admits dense OPD when the reliability audit passes and otherwise uses verifier-grounded GRPO, selectively withholding unreliable teacher signals rather than weakening supervision on every prompt. Across 4B and 35B students in mathematics, code, and instruction following, TGOPD outperforms Vanilla OPD in all six single-domain settings and raises the corresponding seven-benchmark MOPD averages by 1.14 and 0.95 points. The probes reuse idle teacher capacity, raising utilization from 9.8% to 78.9% in the measured 4B single-domain run; a matched 35B code run incurs a 5.9% increase in mean step time. The current method assumes automatic verifiers and binary routing; future work should extend outcome-grounded reliability estimation to open-ended tasks and uncertainty-aware gates.

AI USE STATEMENT

In this work, generative AI tools were not used for research ideation, method or experiment design, theoretical derivations, data preparation, or the generation and interpretation of experimental results. Tasks involving human subjects or qualitative analysis were not applicable. We used generative AI tools only for manuscript polishing and formatting improvements. All AI-assisted changes were reviewed by the authors and verified to preserve the intended technical content. We take full responsibility for the final content of this work, including all text, claims, results, code, and artifacts.

REPRODUCIBILITY STATEMENT

An anonymized research-code snapshot is available at anonymous.4open.science/r/artifact-FB81/. It contains the core TGOPD implementation, generic launch configurations for the reported model scales and training regimes, and utilities for preparing and validating offline probe manifests. Appendix B reports the training configurations and infrastructure, while Appendix C specifies the evaluation protocols. The repository documents the external assets and environment variables that must be supplied; datasets, checkpoints, generated manifests, run logs, and credentials are excluded from the anonymous snapshot.

REFERENCES

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

Mohammad Sadegh Akhondzadeh, Vijay Lingam, Atula Tejaswi, Chanakya Ekbote, Sujay Sanghavi, and Aleksandar Bojchevski. Reward-gated on-policy distillation. *arXiv preprint arXiv:2607.04037*, 2026.

Hanze Dong et al. RAFT: Reward rAnked finetuning for generative foundation model alignment. *Transactions on Machine Learning Research*, 2023.

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.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

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.

Jongwoo Ko, Sara Abdali, Young Jin Kim, Tianyi Chen, and Pashmina Cameron. Scaling reasoning efficiently via relaxed on-policy distillation. *arXiv preprint arXiv:2603.11137*, 2026.

Junlong Li, Daya Guo, Dejian Yang, Runxin Xu, Yu Wu, and Junxian He. CodeIO: Condensing reasoning patterns via code input-output prediction. In *International Conference on Machine Learning (ICML)*, volume 267, pp. 34471–34489. PMLR, 2025a.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huan-ang 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*, 2026.

Yuetai Li et al. Small models struggle to learn from strong reasoners. *arXiv preprint arXiv:2502.12143*, 2025b.

Ling Team, Anqi Shen, Baihui Li, Bin Hu, Bin Jing, Cai Chen, Chao Huang, Chao Zhang, Chaokun Yang, Cheng Lin, et al. Every step evolves: Scaling reinforcement learning for trillion-scale thinking model. *arXiv preprint arXiv:2510.18855*, 2025.

Xiaogeng Liu, Xinyan Wang, Yingzi Ma, Yechao Zhang, and Chaowei Xiao. When are teacher tokens reliable? position-weighted on-policy self-distillation for reasoning. *arXiv preprint arXiv:2605.21606*, 2026.

Zichen Liu et al. Understanding R1-Zero-Like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.

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

Wenhan Ma, Jianyu Wei, Liang Zhao, Hailin Zhang, Bangjun Xiao, Lei Li, Qibin Yang, Bofei Gao, Yudong Wang, Rang Li, Jinhao Dong, Zhifang Sui, and Fuli Luo. MOPD: Multi-teacher on-policy distillation for capability integration in LLM post-training. *arXiv preprint arXiv:2606.30406*, 2026.

Zikun Qu, Min Zhang, Mingze Kong, Zhiwei Shang, Zhengyu Chen, Yikun Ban, Shuang Qiu, and Zhongxiang Dai. SPOT: Sparse probing and outcome calibration for on-policy distillation. *arXiv preprint arXiv:2608.04419*, 2026.

Zhihong Shao et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Avi Singh et al. Beyond human data: Scaling self-training for problem-solving with language models. *Transactions on Machine Learning Research (TMLR)*, 2024.

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

Alex Stein, Furong Huang, and Tom Goldstein. GATES: Self-distillation under privileged context with consensus gating. *arXiv preprint arXiv:2602.20574*, 2026.

Yang Sun, Lichao Ma, Houyuan Qin, Yuxin Liu, Hanyang Lu, Yao Zhu, Pinlong Cai, and Guohang Yan. REOPD: Reliability-adaptive reward extrapolation for on-policy distillation. *arXiv preprint arXiv:2608.11698*, 2026a.

Zechen Sun, Zhiwei Zhang, Fei Zhao, Juntao Li, Mu Chuan, Huayu Deng, Guojian Zhan, Wenliang Chen, Yao Hu, and Min Zhang. D^3-MOPD: Dynamic domain scheDuling for efficient multi-teacher distillation. *arXiv preprint arXiv:2608.24987*, 2026b.

Xiaomi LLM-Core Team. MiMo-V2-Flash technical report. *arXiv preprint arXiv:2601.02780*, 2026.

Xingrun Xing, Haoqing Wang, Boyan Gao, Ziheng Li, and Yehui Tang. Trust region on-policy distillation. *arXiv preprint arXiv:2606.01249*, 2026.

Wenda Xu et al. Speculative knowledge distillation: Bridging the teacher-student gap through interleaved sampling. In *International Conference on Learning Representations (ICLR)*, 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*, 2026a.

Zhuolin Yang et al. Nemotron-Cascade 2: Post-training LLMs with cascade RL and multi-domain on-policy distillation. *arXiv preprint arXiv:2603.19220*, 2026b.

Qiying Yu et al. DAPO: An open-source LLM reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38, 2025. doi: 10.52202/085713-3775.

Ziheng Zhou, Jiaqi Li, Huacong Tang, Ying Nian Wu, and Demetri Terzopoulos. Less is more: Early stopping rollout for on-policy distillation. *arXiv preprint arXiv:2605.27028*, 2026.

Zilin Zhu, Chengxing Xie, Xin Lv, and slime Contributors. slime: An LLM post-training framework for RL scaling. `https://github.com/THUDM/slime`, 2025.

A Additional Method and System Details

A.1 Finite-Sample Properties of the Reliability Estimate

Because the probes are i.i.d. draws from $\pi_T(\cdot \mid x)$ and r is binary, $K_T\, q_T(x) \sim \mathrm{Binomial}(K_T, R_T(x))$. The estimator is therefore unbiased, $\mathbb{E}[q_T(x)] = R_T(x)$, for any probe budget, with variance $R_T(x)\big(1 - R_T(x)\big)/K_T$. Unbiasedness alone does not imply that a particular finite probe budget yields reliable accept/reject decisions. We use $K_T = 3$ as a lightweight probe budget in the main experiments; Section 4.5 uses a larger budget to examine threshold sensitivity at finer resolution.

A.2 Fallback Normalization Convention

Our implementation omits the standard-deviation normalization of GRPO (Liu et al., 2025). This avoids rescaling the fallback by the within-group reward standard deviation. We use it as an implementation convention, not as a requirement of the gate. A controlled comparison with normalized variants is left to future work.

A.3 Asynchronous System Rationale

Overlapping probes with teacher idle time. In an asynchronous deployment, the teacher's standard task is a forward pass over already-generated tokens. This is far cheaper than the student's autoregressive decoding and cannot begin until a batch of student rollouts is ready, so the teacher node idles for most of each cycle (§4.4). TGOPD issues the K_T probe rollouts at the start of the cycle, concurrently with student generation, so most probe work occupies this window. If all K_T decodes finish before the student-side batch completes, the probe adds no wall-clock cost; in practice the overlap is substantial but not perfect (Appendix D).

Scoring stays unconditional. Teacher scoring runs on every prompt, exactly as under vanilla OPD, and the verifier scores the student's rollouts because that reward is the reinforcement-learning signal in any case. The scoring pass is a single forward evaluation over tokens the student has already produced. It is batched as rollouts arrive and runs on a node that is not the throughput bottleneck. Making this pass conditional for a minority of prompts would add a data-dependent pipeline branch while saving little computation. The additional work in TGOPD therefore consists of the K_T probe decodes, which is why teacher utilization can rise from 9.8% to 78.9% with only modest end-to-end overhead (§4.4): most audit work consumes capacity that was previously wasted, while the measured remainder slightly extends the cycle.

A.4 Rollout–Update Algorithm

Algorithm 1 summarizes one rollout-update cycle. Relative to vanilla OPD, TGOPD retains the same student rollout and teacher scoring operations and adds the probe in equation 4, which determines the gate in equation 5. Teacher probes run concurrently with student generation, using otherwise-idle teacher capacity; the measured residual overhead is summarized in Appendix D.

Algorithm 1 TGOPD: one asynchronous rollout–update cycle

Require: current student π_θ; frozen rollout snapshot π_{old}; frozen teacher π_T; verifier r
Require: prompt batch $\mathcal{B}$; group size G; probe budget K_T; threshold τ; PPO clip ϵ; step size η
Ensure: updated student parameters θ

for each prompt $x \in \mathcal{B}$ **do**
  concurrently: ▷ probe fills the teacher's idle window
    rollout nodes: $\{y^i\}_{i=1}^{G} \sim \pi_{\text{old}}(\cdot \mid x)$; record rollout log-probs ▷ nominal critical path
    teacher node: $\{\hat{y}^k\}_{k=1}^{K_T} \sim \pi_T(\cdot \mid x)$
  $q_T(x) \leftarrow \frac{1}{K_T} \sum_{k=1}^{K_T} r\big(x, \hat{y}^k\big)$ ▷ reliability estimate, equation 4
  // both candidate signals are formed for every prompt
  evaluate $\log \pi_T\big(y_t^i \mid x, y_{<t}^i\big)$ for all i, t ▷ teacher scoring pass
  score the student rollouts: $r\big(x, y^i\big)$ for $i = 1, \ldots, G$ ▷ the RLVR signal
  if $q_T(x) \geq \tau$ **then** ▷ teacher supervision admitted
    $\hat{A}_{i,t} \leftarrow \hat{A}_{i,t}^{\text{OPD}}$ for all i, t ▷ equation 1
  else ▷ teacher supervision withheld
    $\hat{A}_{i,t} \leftarrow \hat{A}_{i,t}^{\text{GRPO}}$ for all i, t ▷ equation 2
  end if
end for
form $\rho_{i,t}(\theta)$ from current and recorded rollout log-probs
$\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{\text{TGOPD}}(\theta)$ over $\mathcal{B}$ ▷ equation 7

B Training Hyperparameters and Infrastructure

Table 4 reports the settings retained in the run records at both model scales. Within each controlled comparison, Vanilla OPD and TGOPD share the student initialization, data, optimization settings, and system topology. IcePop/TIS is an implementation-level safeguard for the asynchronous engine; the method itself is defined by the gate and routed advantage in equation 6.

Infrastructure. The 4B runs use five 8-GPU nodes (40 GPUs): two trainer nodes, two student-rollout nodes, and one teacher node. The 35B runs use seven 8-GPU nodes (56 GPUs): four trainer nodes, two student-rollout nodes, and one teacher node. For the 4B 16K runs, we cap the SGLang static memory fraction at 0.60 and both the maximum running requests and teacher-server concurrency at 64.

Probe realization. At both model scales, the frozen teacher draws three fresh probe responses online for each prompt and the gate opens when at least two pass the verifier. Thus the 4B and 35B experiments instantiate the same reliability test described in §3.1; the resource topology changes with model scale, but the training rule does not.

Table 4: Training configuration by model scale. Both scales follow the same optimization and online-probing protocol; only model-dependent system settings differ.

Setting	Qwen3.5-4B	Qwen3.6-35B-A3B
Runtime	slime asynchronous stack	slime asynchronous stack
Execution	Asynchronous rollout and update	Asynchronous rollout and update
Rollout batch size	64 prompts per cycle	64 prompts per cycle
Responses per prompt (G)	4	4
Update batch size	128	128
Updates per rollout cycle	2	2
Optimizer	Adam ($\beta_1 = 0.9$, $\beta_2 = 0.98$, $\epsilon = 10^{-8}$)	Adam ($\beta_1 = 0.9$, $\beta_2 = 0.98$, $\epsilon = 10^{-8}$)
Learning rate	1×10^{-6}	1×10^{-6}
Learning-rate schedule	Constant	Constant
Weight decay	0.1	0.1
Advantage estimator	Group mean, without std normalization	Group mean, without std normalization
Policy surrogate	PPO clipping, $\epsilon = 0.2$	PPO clipping, $\epsilon = 0.2$
Asynchronous mismatch control	IcePop/TIS, ratio clip 2.0	IcePop/TIS, ratio clip 2.0
Probe realization	Three fresh online teacher rollouts	Three fresh online teacher rollouts
Teacher probes (K_T)	3	3
Default reliability threshold (τ)	$2/3$	$2/3$
Maximum response length	16,384 (math); 8,192 (code and IF)	16,384 (math); 8,192 (code and IF)

C Evaluation Benchmark Details

Table 5 lists the benchmarks used for in-domain evaluation, together with the number of independent generation runs and the metric reported in the main tables. For benchmarks evaluated with multiple runs, the reported score is the average over those runs. For LiveCodeBench we report only the code-generation subtask; for OJBench we report the overall score aggregated over the C++ and Python splits.

Table 5: Evaluation benchmark details. *Runs*: number of independent generations per checkpoint; the score reported in all main-text tables is the mean over these runs.

Domain	Benchmark	#Runs	Metric
Math	AIME 2025	64	Accuracy (avg64)
	AIME 2026	64	Accuracy (avg64)
	HMMT-Feb 2025	32	Accuracy (avg32)
Code	LiveCodeBench (code generation)	6	Pass@1 (avg6)
	OJBench (overall)	1	Aggregate (C++/Python)
IF	IFBench	1	Accuracy
	IFEval	1	Accuracy

D Per-Configuration GPU Utilization Traces

Table 6 reports the complete utilization statistics underlying Figure 4. Each measurement covers one hour; *idle* is the fraction of 15-second samples below 5% utilization. In MOPD, one teacher node hosts three frozen domain specialists (4 GPUs for code and 2 each for math and IF), and rollout utilization is averaged over the two rollout nodes.

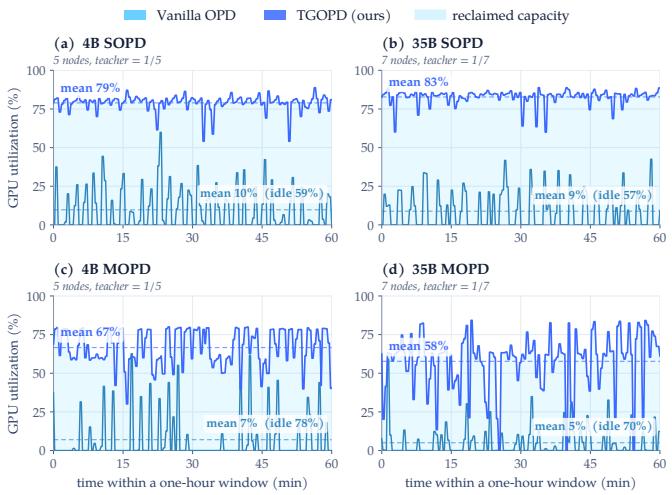

Figure 6: **Teacher-node GPU utilization over a one-hour window for each configuration.** Light blue: vanilla OPD. Dark blue: TGOPD. The shaded region indicates the teacher capacity reclaimed through probe generation. Dashed lines indicate the mean utilization of each run. **(a,b)** Single-domain OPD at the 4B and 35B scales. **(c,d)** Multi-domain OPD at the corresponding scales.

Table 6: GPU utilization under Vanilla OPD and TGOPD across four configurations.

	Teacher utilization		Teacher idle		Rollout nodes		Cluster average	
Configuration	OPD	TGOPD	OPD	TGOPD	OPD	TGOPD	OPD	TGOPD
Qwen3.5-4B (5 nodes, teacher = 1/5)								
SOPD	9.8%	**78.9%**	59.0%	0.0%	69.6%	70.5%	51.5%	**69.5%**
MOPD	7.0%	**66.6%**	78.0%	0.0%	64.2%	66.3%	40.8%	**53.5%**
Qwen3.6-35B-A3B (7 nodes, teacher = 1/7)								
SOPD	8.8%	**82.8%**	57.0%	0.0%	73.6%	73.8%	44.5%	**56.3%**
MOPD	5.0%	**57.7%**	70.0%	2.0%	65.6%	63.8%	42.9%	**51.5%**

Cluster-wide utilization improves by $+18.0$ and $+11.8$ points under SOPD and by $+12.7$ and $+8.6$ under MOPD (4B and 35B, respectively). The teacher contribution accounts for 13.8, 10.6, 11.9, and 7.5 points of these gains in the same order; the residual comes from utilization differences on the training nodes. The smaller gains at 35B reflect the teacher's lower cluster share ($1/7$ rather than $1/5$), while MOPD additionally partitions teacher capacity across domains.

Section 4.4 summarizes teacher-node utilization by its mean and idle fraction (Table 6) and its marginal distribution (Figure 4a). Neither summary preserves the time axis. Figure 6 therefore plots the raw one-hour trace for every configuration. All eight traces are sampled at 15-second resolution over a one-hour window taken from steady-state training, and the two curves within a panel are drawn on a common axis.

Matched end-to-end timing. To avoid inferring wall-clock neutrality from utilization alone, we additionally compare the completed 35B CodeIO TGOPD and OPD runs under identical settings. Across 500 aligned rollout cycles, TGOPD increases mean step time by 5.9%, while mean decode throughput changes by less than 0.1%. Thus most probe work overlaps with idle teacher capacity, but the audit is not strictly free.

The traces clarify three aspects of the utilization results in §4.4.

The baseline teacher is bursty, not uniformly slow. In all four panels the light-blue curve alternates between brief spikes and long stretches pinned at zero, rather than hovering at a low but

steady level. Each spike is one scoring pass; the flat intervals are the student's rollout phase, during which the teacher has no scoring work. The idle fraction in Table 6 measures these intervals. Their duration is set by the student rollout rather than teacher scoring speed, so shrinking the teacher or batching its scoring work differently would not remove the gap.

Probes occupy otherwise low-utilization intervals. The TGOPD traces show fewer near-zero utilization intervals than the OPD traces, consistent with the teacher performing additional probe generation between scoring workloads. The traces describe when the teacher is active, but do not isolate queueing delays or competition between probing and scoring. We therefore assess runtime impact using the matched timing comparison rather than inferring it from utilization peaks.

Multi-domain probing oscillates because it is domain-routed. The SOPD panels show a tight blue band around 79–83%, whereas the MOPD panels swing between roughly 40% and 85%. The MOPD teacher node is partitioned into three domain-specialist engines (code 4 GPU, math 2 GPU, IF 2 GPU). Each prompt activates only its corresponding teacher, while different prompts can use different teachers concurrently. Node occupancy therefore varies with the domain mix of the batch. This oscillation is what produces the bimodal marginal distribution visible in Figure 4a and the lower MOPD means in Table 6. It reflects how the node is divided rather than a failure of the probe to fill the window: the idle fraction still falls to 0–2%.

E Additional Ablation Results

E.1 Closed-Gate Policy: Full Results

Tables 7 and 8 give the per-benchmark results summarized in Table 3. The two gated variants use the same $K_T = 3$, $\tau = 2/3$ gate and differ only in the update applied after rejection.

Table 7: Closed-gate policy on single-domain mathematics runs. Avg. is the mean of the three benchmark scores before display rounding.

		Math		
Policy	Avg.	AIME 25	AIME 26	HMMT-Feb
Qwen3.5-4B				
Vanilla OPD	62.20	61.1	71.2	54.3
GRPO fallback	**64.67**	**64.8**	**73.5**	**55.7**
Mask only	63.52	62.3	72.8	55.5
Qwen3.6-35B-A3B				
Vanilla OPD	71.60	73.4	77.3	64.1
GRPO fallback	**72.27**	**74.0**	**78.7**	64.1
Mask only	71.67	73.5	77.2	**64.3**

On single-domain mathematics, both policies improve the average over Vanilla OPD at each scale. The GRPO fallback is stronger on all three 4B benchmarks and both 35B AIME benchmarks; masking has a small advantage on 35B HMMT-Feb (64.3 versus 64.1).

Table 8: Closed-gate policy on multi-domain runs at 199 training steps. Avg. is computed before display rounding.

		MATH			CODE		IF	
Closed-gate policy	Avg.	AIME 25	AIME 26	HMMT-Feb	LCB	OJBench	IFBench	IFEval
Qwen3.5-4B								
Vanilla MOPD (no gate)	53.40	58.1	63.8	50.7	50.5	18.1	48.8	83.9
GRPO fallback	54.54	58.1	67.6	51.3	50.0	**19.0**	**51.7**	**84.1**
Mask only	**54.97**	**60.0**	**69.8**	**52.0**	**50.8**	18.5	50.9	82.9
Qwen3.6-35B-A3B								
Vanilla MOPD (no gate)	60.99	73.1	77.3	60.7	57.4	23.3	44.4	90.7
GRPO fallback	**61.94**	73.0	**78.0**	62.4	**57.8**	**27.6**	44.0	**90.8**
Mask only	61.78	**74.2**	76.8	**62.5**	57.1	27.2	**44.6**	90.1

Under multi-domain training, masking is stronger at 4B ($+1.57$ over ungated MOPD versus $+1.14$ for GRPO), whereas the GRPO fallback is stronger at 35B ($+0.95$ versus $+0.79$). GRPO also outperforms masking on OJBench at both scales. These results show that gating is effective under either closed-gate policy, while the benefit of the replacement update is setting-dependent.

F REWARD VARIATION ON AUDIT-REJECTED MATHEMATICS PROMPTS

To complement Table 7, we examine student reward variation on mathematics prompts for which the teacher gate closes. The analysis follows the main training configuration, with three online teacher probes per prompt and four responses per student group.

We classify each recorded student group as mixed-outcome, all-correct, or all-incorrect, and aggregate group counts over the rejected prompts. Repeated groups from different training steps are included. Under binary rewards, mixed-outcome groups contain both correct and incorrect responses and therefore yield non-zero group-centered advantages, whereas all-correct and all-incorrect groups yield zero advantages.

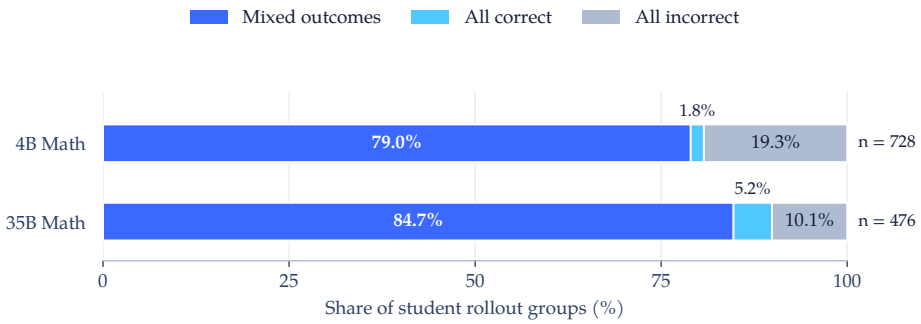

Figure 7: **Student reward variation on audit-rejected mathematics prompts.** Mixed-outcome groups yield non-zero GRPO advantages, whereas all-correct and all-incorrect groups yield zero advantages under binary rewards. Percentages are computed over recorded rollout groups; n is the number of rejected prompts. Each group contains four student responses, following the main training configuration.

Figure 7 shows that mixed-outcome groups account for 79.0% and 84.7% at 4B and 35B, respectively. All-incorrect groups account for 19.3% and 10.1%, while all-correct groups account for 1.8% and 5.2%. Thus, withholding teacher supervision does not necessarily eliminate within-group reward variation among student rollouts. This finding complements the ablation results reported in Section 4.6 by showing that audit-rejected prompts can still provide non-zero verifier-grounded advantages.