DUOOPD: LEARNING FROM JOINT TEACHER–STUDENT OUTCOMES FOR MULTI-TASK ON-POLICY DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy distillation (OPD) trains a student on its own responses with token-level feedback from a stronger teacher, yet the teacher can fail on questions the student already answers correctly, and how often each model succeeds varies across tasks. OPD ignores these outcomes and, on average, pushes down even the student's correct responses; gating feedback by student correctness fixes the direction but uses the teacher in the same way whether or not it succeeded. We introduce DuoOPD, in which the student's outcome sets the direction of feedback and the joint teacher–student outcome decides how the teacher supports it: when only the teacher succeeds, its verified answer becomes context for scoring the student's failed response, and when only the student succeeds, a weight shared within the task reinforces the whole response. A single rule covers all four outcome combinations without task-specific settings. Across Qwen3 and Llama, DuoOPD outperforms all five baselines in mean macro accuracy, improving over OPD by 2.58 and 5.98 percentage points, and it also leads on two further task mixtures spanning scientific calculation, instruction following, and code generation. Ablations show that outcome-based direction alone stays near the gated baseline, while the joint-outcome designs supply most of the gain.

1 INTRODUCTION

On-policy distillation (OPD) trains a student on its own responses with token-level feedback from a stronger teacher (Lin et al., 2020; Gu et al., 2024; Agarwal et al., 2024). When one teacher distills several tasks into one student, this feedback is not uniformly reliable: the teacher solves many questions the student misses, yet fails on others the student already answers correctly. Before distillation, our Qwen3 and Llama pairs show all four correctness combinations in task-dependent proportions (Initial bars in Figure 1); the Llama student, for example, succeeds where the teacher fails on 5.9% of biology responses but 9.8% of physics responses. Across more diverse tasks the spread widens: both models fail on 1% of chemistry-understanding training responses but 34% of physics-calculation ones. Multi-task distillation therefore needs one rule that learns from teacher successes while reinforcing student successes when the teacher fails.

Existing methods leave this rule open. Multi-task methods decide which teacher supervises each task (Ma et al., 2026a;b) or how strongly each task is trained (Ramesh et al., 2026), not how feedback on a response depends on which model succeeded. OPD itself ignores correctness, so it can suppress correct student responses or reinforce failures; student verification can constrain feedback direction (Lin et al., 2026), and teacher reliability probes can route prompts between OPD and reinforcement learning (Zhang et al., 2026). Direction alone is not enough: student prefixes can steer even a capable teacher toward an incorrect solution (Zhu et al., 2026), and teacher-dependent positive weights can leave parts of successful responses weakly reinforced. The teacher's scoring context and allocation of feedback must also depend on who succeeded.

Our key idea is to let student outcomes set the learning goal and joint outcomes determine how the teacher supports it. DuoOPD implements this principle as one four-outcome rule shared across tasks (Figure 2): when only the teacher succeeds, a teacher reference guides correction; when only the student succeeds, weight sharing reinforces its complete response; when both succeed or both fail,

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.

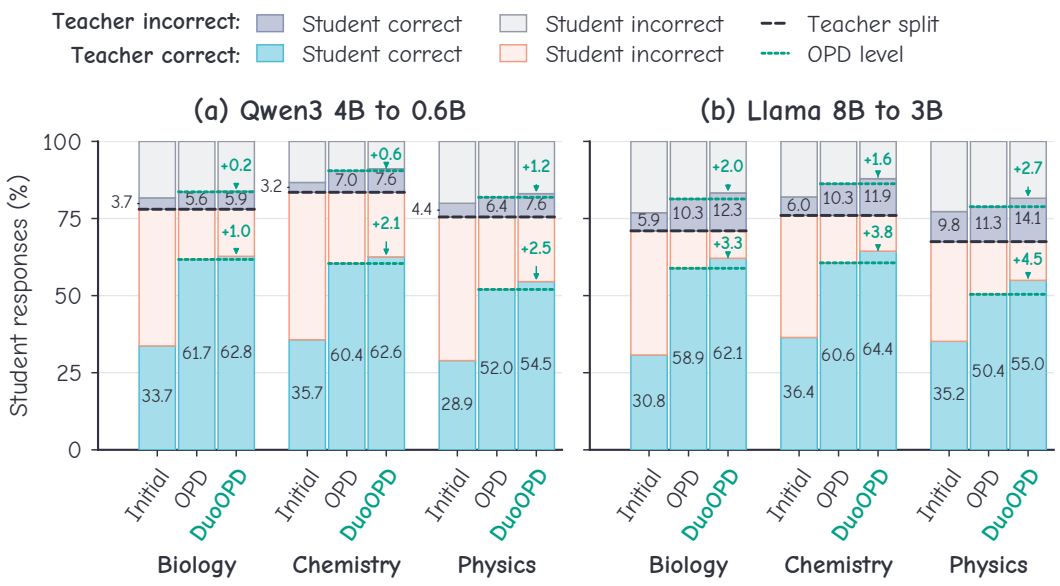

Figure 1: **DuoOPD improves student accuracy over OPD whether the teacher succeeds or fails:** (a) Qwen3; (b) Llama. Each bar splits student test responses by the fixed teacher response: below the dashed line are questions the teacher solves, above it those the teacher fails, so the line is identical for Initial, OPD, and DuoOPD. The lower segment of each part is student successes; on the DuoOPD bar, each green arrow gives the rise of the student-success segment below it above OPD's level (green dotted line, percentage points), positive in every domain for both parts. Each domain uses 200 shared questions with eight student responses per question; trained bars average three runs.

teacher preferences set the strength of reinforcement or suppression. When the models disagree, support comes from the model that succeeded. Because the rule conditions on each response's outcome rather than its task, it needs no task-specific settings.

We evaluate three task mixtures of increasing heterogeneity, up to physics answers, instruction following, and code generation with their own verifiers. In every domain for both model families, DuoOPD raises student accuracy above OPD, both on questions the teacher solves and on questions it fails (Figure 1). Our contributions are:

- A unified approach to multi-task on-policy distillation from a single teacher into a single student: one four-outcome rule, free of task-specific settings, in which student verification sets feedback direction and joint outcomes select teacher context and weight allocation.
- Consistent gains across model families and task mixtures. DuoOPD leads all five baselines for Qwen3 and Llama, improving mean macro accuracy over OPD by 2.58 and 5.98 percentage points, and also leads on two additional mixtures spanning scientific understanding and calculation, instruction following, and code generation.
- Evidence for where the gains come from: every outcome rule contributes, and while outcome-based direction is necessary because outcome-agnostic OPD pushes down correct student responses, it alone stays near OPDVR; teacher references and within-task weight sharing supply most of the gain over OPD.

2 Problem Setup and Background

Multi-task setting. For each task $k \in \{1, \ldots, K\}$, $\mathcal{D}_k$ contains questions q and ground-truth answers $a^\star$. We jointly train a student π_θ on a fixed task mixture with a frozen teacher π_T.

On-policy distillation. On-policy distillation obtains teacher feedback on student-generated prefixes. GKD matches full next-token distributions under a generalized divergence (Agarwal et al.,

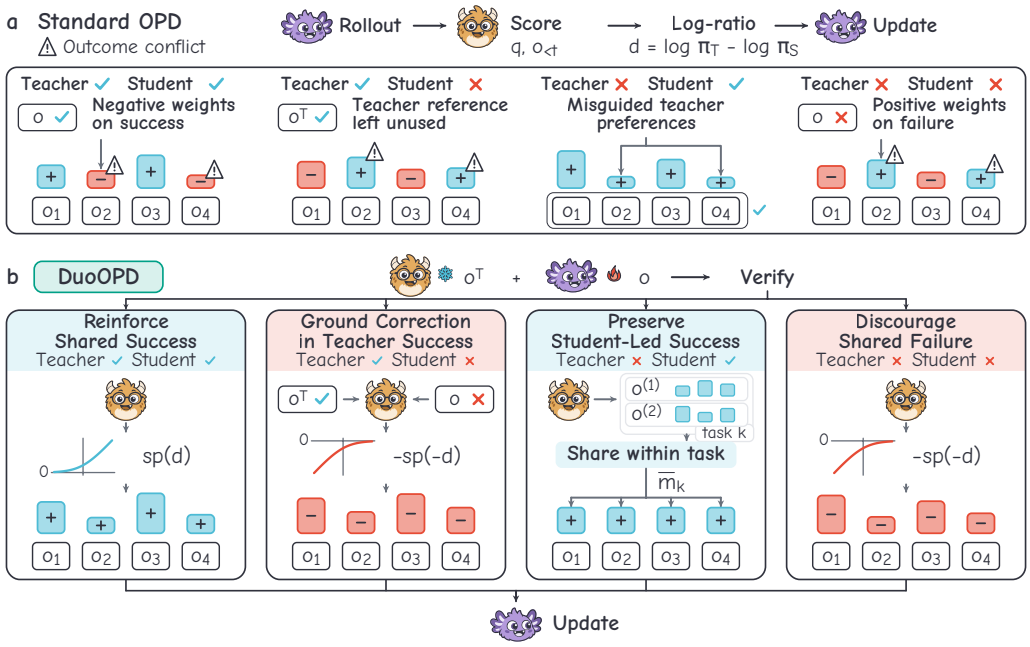

Figure 2: **Student outcomes set direction; joint outcomes select teacher use. a,** In OPD, triangles mark feedback opposing student outcomes, and a failing teacher's preferences weakly reinforce parts of correct responses. **b,** DuoOPD selects teacher use by joint outcome (Eq.(4)); only the teacher reads o^T, d is the teacher–student log-ratio, sp is Softplus, and shared weights follow Eq.(7). Snowflake/flame mark frozen/trainable models; weight heights are schematic.

2024), and MiniLLM optimizes sequence-level reverse KL with policy gradients (Gu et al., 2024). We use the per-token sampled form common in recent practice (Lu & Thinking Machines Lab, 2025): at each update, the student with parameters θ_{old} generates $o \sim \pi_{\theta_{\text{old}}}(\cdot \mid q)$. Teacher and student score each sampled token under its prefix, giving the feedback (Zhu et al., 2026):

$$d_{0,t} = \log \pi_T(o_t \mid q, o_{<t}) - \log \pi_{\theta_{\text{old}}}(o_t \mid q, o_{<t}). \tag{1}$$

Subscript 0 denotes scoring under q and $o_{<t}$. At a fixed prefix, weighting the negative student log-probability gradient by this log-ratio gives a sampled-token estimator of the reverse-KL gradient. Each sampled token thus receives one scalar weight, which DuoOPD reshapes according to joint outcomes. This formulation is our OPD baseline.

For a batch of B responses, index quantities by response i and let T_i count its valid tokens. The sampled OPD gradient averages within responses and then equally across them:

$$\widehat{g}_{\text{OPD}}(\theta) = -\frac{1}{B}\sum_{i=1}^{B}\frac{1}{T_i}\sum_{t=1}^{T_i} d_{0,i,t}\,\nabla_\theta \log \pi_\theta(o_{i,t} \mid q_i, o_{i,<t}). \tag{2}$$

Responses and log-ratio weights are computed before the update using θ_{old} and held fixed; gradients flow only through the student log probabilities. Under gradient descent, positive weights reinforce sampled tokens and negative weights suppress them.

Outcome verification and gated feedback. A task-specific verifier gives $r_S = V_k(o, a^\star) \in \{0, 1\}$, where 1 denotes correctness. OPDVR (Lin et al., 2026) masks feedback opposing this outcome and retains aligned weights, replacing $d_{0,i,t}$ in Eq.(2) with $A_{\text{gate}}(d_{0,i,t}, r_{S,i})$:

$$A_{\text{gate}}(d, r_S) = \begin{cases} \max\{d, 0\}, & r_S = 1, \\ \min\{d, 0\}, & r_S = 0 \end{cases}. \tag{3}$$

We also verify an independently generated teacher response $o^T \sim \pi_T(\cdot \mid q)$, obtaining $r_T = V_k(o^T, a^\star)$. The joint outcome (r_T, r_S) thus records whether a verified teacher reference is available and whether the student succeeds. Its distribution $\rho_k(a, b) = \Pr(r_T = a, r_S = b \mid k)$ differs across tasks and shifts as the student learns.

3 DUOOPD: DISTILLATION GUIDED BY JOINT OUTCOMES

Before training, we cache and verify one independent teacher response per training question; these references and outcomes stay fixed while the student generates fresh responses at each update. DuoOPD replaces each log-ratio weight in Eq.(2) with a weight $A_{i,t}$ determined by the joint outcome of its response (Figure 2).

3.1 PRINCIPLE: DIRECTION FROM THE STUDENT, SUPPORT FROM THE PAIR

DuoOPD separates two decisions for each student response: whether to reinforce or suppress it, and how to distribute that feedback over its tokens. The student outcome makes the first decision and the joint outcome makes the second:

$$A_{i,t} = \sigma_i \, m_{i,t}, \qquad \sigma_i = 2r_{S,i} - 1,$$
$$m_{i,t} = \begin{cases} \mathrm{sp}(\sigma_i \, d_{0,i,t}), & r_{T,i} = r_{S,i}, \\ \mathrm{sp}(-d_{T,i,t}), & r_{T,i} = 1,\ r_{S,i} = 0, \\ \bar{m}_{k(i)}, & r_{T,i} = 0,\ r_{S,i} = 1. \end{cases} \tag{4}$$

Here sp is Softplus (Section 3.2), d_T is the teacher–student log-ratio with the teacher's verified answer in its context (Eq.(6)), and $\bar{m}_k$ is a positive magnitude shared within task k (Eq.(7)). Every magnitude is positive, so the sign of each weight is the student's verified outcome: all valid tokens of a success are reinforced and all tokens of a failure are suppressed, whatever the teacher prefers. The joint outcome decides where the magnitude comes from. When the two models agree, teacher preferences in the original context set the strength of reinforcement or suppression. When they disagree, support comes from the model that succeeded: a successful teacher lends its verified answer as context for scoring the student's failed response, and a successful student receives a shared weight that the failing teacher's token preferences cannot reshape.

Eq.(4) is shared by all tasks and never estimates ρ_k: it applies to whatever mix of outcomes a task presents, and a task enters only through its verifier V_k and the scope of weight sharing. Section 3.2 explains the direction rule and the two agreements; Section 3.3 details the disagreement designs.

3.2 A SHARED FEEDBACK-DIRECTION RULE

The magnitude should follow the verified outcome, use the log-ratio to set strength, and stay nonzero for every valid token. Unit-temperature Softplus $\mathrm{sp}(x) = \log(1+\exp(x))$ meets these requirements, giving positive weights A^+ for student success and negative weights A^- for failure:

$$A^+(d) = \mathrm{sp}(d) = \max\{d, 0\} + \log(1 + \exp(-|d|)),$$
$$A^-(d) = -\,\mathrm{sp}(-d) = \min\{d, 0\} - \log(1 + \exp(-|d|)). \tag{5}$$

The first terms are OPDVR's gated weights at the same d (Eq.(3)); the residual follows the verified direction, with magnitude $\log 2$ at $d = 0$ and approaching zero as $|d|$ grows. Softplus thus preserves the gated contribution and supplies nonzero feedback where the gate returns zero. This matters in practice: the teacher–student log-ratio averages below zero even on verified successes, so gated weights on correct responses average only 0.08–0.09 on DuoOPD's training responses for both Qwen3 and Llama, versus 0.57–0.65 under Softplus (Appendix D.1). The gate would thus leave most successful responses almost unreinforced, and the within-task shared weight, an average of these magnitudes, would nearly vanish. Applying $A^{\pm}$ to d_0 for every outcome gives the "w/o both" variant in Table 4; retaining only the gated terms recovers OPDVR.

Reinforce Shared Success. When $r_{T,i} = r_{S,i} = 1$, the learning goal is to reinforce the verified response. We use $A^+(d_{0,i,t})$ under the original teacher context: every valid token receives positive feedback, even one the teacher assigns lower probability than the student, and teacher preferences set the strength of this positive feedback.

Discourage Shared Failure. When $r_{T,i} = r_{S,i} = 0$, no correct reference is available, and the learning goal is to discourage the failed response. We use $A^-(d_{0,i,t})$ under the original teacher context: tokens the teacher disfavors relative to the student receive stronger negative feedback, but verification fixes the direction, so these preferences adjust suppression strength without turning it into reinforcement. Appendix D.3 illustrates both agreements.

3.3 Support When Teacher and Student Disagree

Ground Correction in Teacher Success. When $r_{T,i} = 1$ and $r_{S,i} = 0$, the student response requires correction and a verified teacher reference is available. The student's erroneous prefix can bias teacher scoring despite the teacher's independent success, so we add the successful teacher reference o_i^T to the teacher's context:

$$d_{T,i,t} = \log \pi_T(o_{i,t} \mid q_i, o_i^T, o_{i,<t}) - \log \pi_{\theta_{\text{old}}}(o_{i,t} \mid q_i, o_{i,<t}). \tag{6}$$

$A^-(d_{T,i,t})$ keeps the negative direction set by the student's failure, while the reference-conditioned teacher distribution allocates correction across the unchanged student trajectory. As in OPSD and OPCD (Zhao et al., 2026; Ye et al., 2026), only the teacher sees the reference (Appendix B.1).

Preserve Student-Led Success. When $r_{T,i} = 0$ and $r_{S,i} = 1$, the student provides the verified success. Positive Softplus ensures the right direction, but the failing teacher can still assign weaker feedback to particular tokens or responses. We therefore give all valid tokens of these responses one positive weight shared within each task. Let $k(i)$ denote the task of response i, and let $\mathcal{I}_k = \{i : k(i) = k,\ r_{T,i} = 0,\ r_{S,i} = 1\}$ index them in the complete rollout batch. For each nonempty $\mathcal{I}_k$,

$$\bar{m}_k = \text{sg}\left[\frac{1}{|\mathcal{I}_k|} \sum_{i \in \mathcal{I}_k} \frac{1}{T_i} \sum_{t=1}^{T_i} \text{sp}(d_{0,i,t})\right]. \tag{7}$$

Here sg stops gradients, and the mean is computed once per rollout batch. It reinforces each complete response with the same coefficient while preserving the task's mean reinforcement scale; we share within tasks because log-ratio magnitudes differ across tasks (Section 4.3).

3.4 Student Update

Replacing $d_{0,i,t}$ in Eq.(2) with $A_{i,t}$ from Eq.(4) gives the gradient of the DuoOPD surrogate loss:

$$\mathcal{L}_{\text{DuoOPD}}(\theta) = -\frac{1}{B} \sum_{i=1}^{B} \frac{1}{T_i} \sum_{t=1}^{T_i} \text{sg}[A_{i,t}] \log \pi_\theta(o_{i,t} \mid q_i, o_{i,<t}). \tag{8}$$

Only student log probabilities receive gradients; teacher references affect $A_{i,t}$, not the student context. Appendix B.2 gives the correspondence between this surrogate and the implemented update.

4 Experiments

4.1 Experimental Setup

Data and Verification. Three task mixtures progressively widen task differences: (i) biology, chemistry, and physics knowledge differ in domain; (ii) materials knowledge, chemistry understanding, and physics calculation also differ in cognitive level; (iii) physics knowledge, instruction following, and code generation also differ in output format and verifier. The first two use fixed, disjoint training and test partitions built from SciKnowEval V2 (Feng et al., 2024). The third reuses the physics partition and adds RLVR-IFeval for training and official IFEval for testing (Zhou et al., 2023), and MBPP (Austin et al., 2021). Verification is binary: answer matching, instruction checks, and unit tests. All methods share data and verifiers within each mixture (Appendix A).

Models and Budget. We compare Qwen3-4B → Qwen3-0.6B (Yang et al., 2025) and Llama-3.1-8B-Instruct → Llama-3.2-3B-Instruct (Grattafiori et al., 2024) on the initial mixture; both additional mixtures use the Qwen3 pair. Within each mixture, all methods process the same fixed-order question stream for 60 student updates, with 48 questions per update (16 per task), one sampled student response per question, and a 1,024-token response limit. Smaller task pools cycle to fill this budget. Appendix B reports implementation settings.

Teacher Computation. DuoOPD caches one teacher response per distinct training prompt. This one-time cache takes 30% (Qwen) and 23% (Llama) of an OPD training loop in GPU-time and

Table 1: **Results across model families.** Test avg@8 accuracy (%); Llama models are Instruct variants. Bold and underlined scores mark the best and second-best distillation methods per column.

	Qwen3 4B → 0.6B				Llama 3.1-8B → 3.2-3B			
Method	Bio.	Chem.	Phys.	**Macro**	Bio.	Chem.	Phys.	**Macro**
Student (initial)	37.38	38.88	33.38	36.54	36.63	42.44	44.94	41.33
Teacher	79.56	81.38	76.88	79.27	73.25	75.06	69.88	72.73
OPD	67.38	67.42	58.38	64.39	69.19	70.90	61.75	67.28
EOPD	<u>68.06</u>	<u>68.44</u>	<u>60.92</u>	<u>65.81</u>	68.73	69.54	61.21	66.49
OPDVR	66.92	66.69	60.90	64.83	<u>69.75</u>	<u>72.40</u>	<u>63.90</u>	<u>68.68</u>
ExOPD	67.54	66.79	58.94	64.42	68.33	69.88	60.81	66.34
FiRe-OPD	67.00	67.23	59.50	64.58	69.50	71.81	61.60	67.64
DuoOPD (ours)	**68.60**	**70.19**	**62.10**	**66.97**	**74.42**	**76.33**	**69.02**	**73.26**

is reusable across students, seeds, and hyperparameters. During training, only responses where the teacher alone succeeds (13–23%) are scored with a reference. The scoring and verification stage, including communication, accounts for 5.9% and 4.7% of OPD's loop. DuoOPD's measured training-loop change is -0.7% (Qwen) and +7.3% (Llama), with student generation accounting for most of the Llama increase (Appendix B.3).

Comparison Methods. OPD and OPDVR use the original-context feedback weights defined in Eq.(1) and Eq.(3), respectively. Entropy-aware OPD (EOPD) adds an entropy-gated distillation term (Jin et al., 2026); ExOPD extrapolates teacher feedback relative to the initial student (Yang et al., 2026); and FiRe-OPD filters trajectories and reweights feedback using teacher confidence and student entropy (Li et al., 2026).

Evaluation. We evaluate each method's final checkpoint. For each test question, we sample eight responses and report *avg@8*, the mean binary success rate over responses and questions within each task. IFEval counts a response as correct only if it satisfies every instruction constraint (Zhou et al., 2023). Macro is the equal-weight mean across the three tasks. All results average three training runs per method (standard deviations in Appendix C.1); initial-student and teacher rows use the same protocol. Development trajectories for the scientific settings appear in Appendix C.2.

4.2 Main Results

4.2.1 Comparisons Across Model Families

DuoOPD leads all five baselines on biology, chemistry, and physics for both Qwen3 and Llama (Table 1), improving mean Macro over OPD by 2.58 and 5.98 percentage points. It also exceeds the strongest baselines, EOPD on Qwen3 and OPDVR on Llama, by 1.16 and 4.58 points. Physics shows the largest gains over OPD (3.73 and 7.27 points) and is the domain where only the teacher succeeds most often during training (24.8% and 14.9%), the outcome corrected with teacher references. Section 4.4 examines where these gains come from.

4.2.2 Results on Additional Task Mixtures

Materials knowledge, chemistry understanding, and physics calculation. Each additional mixture is trained separately with the same DuoOPD settings; tasks enter only through their verifiers. DuoOPD retains the highest mean Macro when joint training combines scientific knowledge, understanding, and calculation (Table 2, left). It reaches 63.48%, 0.86 points above ExOPD, the strongest baseline. Its largest task-level advantage, 3.25 points over the strongest baseline, is in physics calculation, where only the teacher succeeds on 26.8% of training responses and both models fail on 34%, whereas both succeed on 89% of chemistry-understanding responses.

Table 2: **Results on two additional task mixtures with Qwen3-4B → Qwen3-0.6B.** Left: materials knowledge, chemistry understanding, and physics calculation. Right: physics knowledge, IFEval (prompt-level strict), and MBPP. Protocol and marking follow Table 1.

	Science knowledge, understanding, calculation				**Heterogeneous** answers, instructions, code			
Method	Mat.	Chem.	Phys.	**Macro**	Phys.	IFEval	MBPP	**Macro**
Student (initial)	32.63	56.44	24.38	37.81	32.25	54.99	20.69	35.98
Teacher	67.44	96.94	56.00	73.46	77.25	78.70	57.69	71.21
OPD	**55.50**	91.71	39.77	62.33	59.29	50.95	31.41	47.22
EOPD	55.00	92.75	36.94	61.56	59.56	50.25	30.88	46.90
OPDVR	53.98	92.83	39.67	62.16	59.35	52.31	31.83	47.83
ExOPD	54.56	**93.21**	40.08	62.62	59.81	49.72	30.95	46.83
FiRe-OPD	54.67	92.17	38.31	61.72	58.96	52.89	30.74	47.53
DuoOPD (ours)	54.65	92.46	**43.33**	**63.48**	**61.75**	**53.30**	**32.69**	**49.24**

Table 3: **Replacing one outcome's feedback with OPD.** Share is the outcome's fraction of DuoOPD training responses (%). Accuracy in %; in Tables 3–5, bold and underlined values mark the best and second-best means. Δ is relative to DuoOPD.

Teacher	**Student**	**Feedback replacement**	Share	Bio.	Chem.	Phys.	**Macro**	Δ
DuoOPD		—	100	**68.60**	**70.19**	**62.10**	**66.97**	0.00
✓	✓	$\mathrm{sp}(d_0) \to d_0$	56.9	67.40	68.63	61.73	65.92	-1.05
✓	×	$-\mathrm{sp}(-d_T) \to d_0$	21.5	68.21	67.81	60.29	65.44	-1.53
×	✓	$\bar{m}_k \to d_0$	6.9	68.38	68.81	61.44	66.21	-0.76
×	×	$-\mathrm{sp}(-d_0) \to d_0$	14.7	68.27	68.79	60.67	65.91	-1.06

Physics knowledge, instruction following, and code generation. DuoOPD also leads all five baselines in mean Macro when the tasks require different output formats and verification procedures (Table 2, right). It reaches 49.24%, 1.41 points above OPDVR, and scores highest on every task. Instruction following shows how tasks interact under joint training: the initial student already satisfies most IFEval prompts, and DuoOPD best preserves this ability alongside physics and code (53.30% versus 50.95% for OPD), as IFEval has the mixture's highest share of responses where only the student succeeds (7.6%), which weight sharing reinforces.

Gains on the hardest task. Multi-task methods such as MT-GRPO explicitly target worst-task performance (Section 5). Without such an objective, DuoOPD attains the highest accuracy on the hardest task of every setting: physics for Qwen3 and Llama (62.10 and 69.02, versus at most 60.92 and 63.90), physics calculation (43.33 versus 40.08), and MBPP (32.69 versus 31.83).

4.3 Ablation Study

All ablations use three Qwen3 runs per condition. The four-rule and component ablations use biology, chemistry, and physics; the sharing-scope comparison covers all three task mixtures.

Contributions of the Four Guidance Rules. Replacing any one outcome's feedback with OPD's d_0 lowers mean Macro by 0.76–1.53 points (Table 3), so every rule contributes, including the rule for responses where only the student succeeds, which cover only 6.9% of training responses.

Teacher References and Weight Sharing. The two disagreement rules carry DuoOPD's distinctive designs (Table 4). Without the teacher reference, responses where only the teacher succeeds are scored in the original context, and mean Macro drops by 1.49 points; without weight sharing, responses where only the student succeeds keep token-level positive weights, and it drops by 0.71. Removing both lowers it by 2.41 points, to 64.56%, close to OPDVR (64.83%), which also sets feedback direction by student correctness. Outcome-based direction alone therefore explains lit-

Table 4: **Removing the two disagreement designs.** Feedback used for each disagreement outcome; the agreement rules are unchanged.

Variant	T ✓ S ×	T × S ✓	Macro	Δ
DuoOPD	$-\,\mathrm{sp}(-d_T)$	$\bar{m}_k$	**66.97**	0.00
w/o sharing	$-\,\mathrm{sp}(-d_T)$	$\mathrm{sp}(d_0)$	<u>66.26</u>	−0.71
w/o reference	$-\,\mathrm{sp}(-d_0)$	$\bar{m}_k$	65.48	−1.49
w/o both	$-\,\mathrm{sp}(-d_0)$	$\mathrm{sp}(d_0)$	64.56	−2.41

Table 5: **Sharing scope across the three mixtures.** Macro (%) with the student-success weight shared across or within tasks; B/C/P is Table 1, Sci./Het. are Table 2.

Shared	B/C/P	Sci.	Het.
Across tasks, $\bar{m}$	<u>66.74</u>	<u>63.22</u>	<u>48.89</u>
Within task	**66.97**	**63.48**	**49.24**

tle of DuoOPD's 2.58-point gain over OPD; the two joint-outcome designs supply the rest, with complementary contributions. Appendix D.2 shows both on fixed student responses.

Sharing Within Each Task. Weight sharing is where the multi-task setting enters the rule. Teacher log-ratios differ in scale across tasks: per-task shared weights average 0.56–0.57 on biology, chemistry, and physics, but 0.56–0.64 and 0.57–0.64 in the two additional mixtures, where physics calculation and code generation receive the largest weights. One weight shared across all tasks would impose a single scale; sharing within each task keeps every task's own reinforcement scale while removing the failing teacher's preferences inside that task. Within-task sharing outperforms sharing across all tasks in all three mixtures (Table 5), with larger margins in the two mixtures whose task scales differ more (0.26 and 0.35 versus 0.22 points).

4.4 Analysis by Joint Outcome

Gains on questions the teacher solves and on those it fails. Figure 1 splits DuoOPD's gain over OPD by teacher outcome. In every domain for both families, DuoOPD adds student successes both where the teacher succeeds (1.90 points on Qwen3, 3.86 on Llama) and where it fails (0.67 and 2.12): the student learns more of what the teacher knows and succeeds more often beyond it.

OPD pushes down correct responses. On DuoOPD's training responses, the original-context log-ratio that OPD would use as feedback is negative on average even when the student is correct: -1.31 on Qwen3 and -0.22 on Llama when both models succeed, and -1.24 and -0.25 when only the student succeeds. Correct responses make up 64–69% of training responses, so OPD suppresses much of what the student already gets right. Setting direction by the student outcome, as OPDVR and DuoOPD do, prevents this, but direction alone stays near OPDVR's accuracy (Table 4); the gain comes from how DuoOPD uses the teacher once the direction is fixed.

Teacher references sharpen correction. Teacher references guide the most frequent disagreement, where only the teacher succeeds; removing them costs 1.49 points (Table 4). In Qwen3 runs trained with references, this outcome's log-ratio and feedback weight average -1.80 and -2.35, compared with -1.35 and -1.91 on the trajectories of runs trained without references. Appendix D.2 holds a failed student response fixed and shows how adding the reference strengthens correction on its erroneous tokens.

5 Related Work

Multi-task learning and distillation. Multi-task tuning pools supervision across tasks (Sanh et al., 2022; Chung et al., 2024); distillation integrates task-specific ensembles (Liu et al., 2019), domain RL specialists in MOPD (Ma et al., 2026a), or soft-prompt teachers in PromptSD (Ma et al., 2026b), and MT-GRPO adapts task weights and sampling to improve worst-task performance (Ramesh et al., 2026). These methods act at the task level; DuoOPD acts within each response of a fixed mixture, so the two compose: r_T can verify a task-specific teacher, task weights can scale DuoOPD's per-task feedback, and ρ_k offers a signal for task scheduling. Even without a worst-task objective, DuoOPD leads on the hardest task of every setting (Section 4.2.2).

On-policy distillation and feedback calibration. On-policy distillation supervises student-generated trajectories through full-distribution divergences, sequence-level reverse KL, or per-token sampled feedback (Lin et al., 2020; Agarwal et al., 2024; Gu et al., 2024; Lu & Thinking Machines Lab, 2025). EOPD uses entropy to adjust distillation, ExOPD extrapolates teacher feedback relative to the initial student, and FiRe-OPD filters trajectories and reweights tokens by teacher confidence and student entropy (Jin et al., 2026; Yang et al., 2026; Li et al., 2026). DuoOPD instead uses verified teacher–student outcomes to set the direction and allocation of feedback.

Verified outcomes and supervision routing. Verification can route prompts between OPD and GRPO (Zhang et al., 2026) or calibrate token targets (Qu et al., 2026). OPDVR masks feedback that conflicts with student correctness (Lin et al., 2026); DuoOPD's Softplus direction rule keeps the reinforcement on verified successes that the gate nearly removes (Section 3.2). Direction alone stays near OPDVR (Section 4.3); DuoOPD's gain comes from also verifying the teacher, which selects the scoring context and weight sharing.

Privileged context and outcome-dependent teacher use. Privileged-context distillation gives teachers information unavailable to students (Lopez-Paz et al., 2016; Zhao et al., 2026; Ye et al., 2026); in thinking models, such context can suppress exploration and self-correction (Kaur et al., 2026). DuoOPD gives a verified teacher reference only when the teacher alone succeeds, and only the teacher reads it: the student's context is unchanged, and the reference only allocates negative feedback. H^2SD also separates student failures from successes, using hint-conditioned self-distillation and token-weighted reinforcement (Cai et al., 2026); DuoOPD additionally verifies an independent frozen teacher, whose success supplies the reference and whose failure triggers weight sharing on correct student responses.

6 Limitations and Future Work

DuoOPD relies on a binary verifier; open-ended tasks such as writing call for feedback rules that handle graded or uncertain judgments. Teacher computation (Section 4.1) could shrink further through shared caches, compressed references, and selective reference use, and longer reasoning trajectories warrant study of exploration and self-correction. Each mixture is also trained and evaluated on the same tasks; whether joint-outcome guidance improves transfer to held-out tasks, and whether task-level outcome statistics ρ_k can guide task scheduling, remain open.

7 Conclusion

DuoOPD lets the student's outcome set the direction of feedback and the joint outcome decide how the teacher supports it: teacher references correct failures the teacher can solve, and within-task weight sharing reinforces successes the teacher misses. Because the rule conditions on each response's outcome rather than its task, the same settings serve all three mixtures we test, which differ in domain, cognitive level, output format, and verifier. DuoOPD leads all five baselines across two model families and three task mixtures, including on the hardest task of every setting, and improves mean Macro over OPD by 2.58 points on Qwen3 and 5.98 on Llama. Our analyses show that setting direction by the student outcome is necessary, since outcome-agnostic OPD pushes down much of what the student already answers correctly, but not sufficient: in ablations, most of the gain comes from the two designs for disagreements.

Reproducibility statement

Sections 2 and 3 specify reference routing, token feedback, and loss aggregation. Appendices A and B provide data partitioning, prompts, model settings, optimization, and evaluation details. Section 4 defines the evaluation protocol. The anonymous supplementary package includes code and setup instructions, processed data splits and verifiers, cached teacher responses with generation settings, and five representative experiment configurations.

AI USE STATEMENT

Large language models assisted this work in four ways. They polished the writing and drafted parts of sections, which the authors then revised. They helped search for and summarize related work; the authors checked the cited sources. They also supported research ideation and execution, taking part in discussions of method and experiment design and helping write and edit code for training, evaluation, analysis, and figures. The authors made the research decisions, verified the reported results against the final experiment outputs, and take full responsibility for the content of this paper.

REFERENCES

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

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021. URL https://arxiv.org/abs/2108.07732.

Qiye Cai, Yichuan Ma, Peiji Li, Yongkang Chen, Qipeng Guo, Yicheng Zou, Linyang Li, Xiaocheng Feng, and Bing Qin. H^2SD: Hybrid hindsight self-distillation. *arXiv preprint arXiv:2607.18955*, 2026. URL https://arxiv.org/abs/2607.18955.

Hyung Won Chung, Le Hou, Shayne Longpre, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024. URL https://jmlr.org/papers/v25/23-0870.html.

Kehua Feng, Xinyi Shen, Weijie Wang, Xiang Zhuang, Yuqi Tang, Qiang Zhang, and Keyan Ding. SciKnowEval: Evaluating multi-level scientific knowledge of large language models. *arXiv preprint arXiv:2406.09098*, 2024. URL https://arxiv.org/abs/2406.09098.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://arxiv.org/abs/2407.21783.

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

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

Simran Kaur, Narutatsu Ri, Yinghui He, Liam Fowl, and Sanjeev Arora. Rethinking on-policy self-distillation for thinking models. *arXiv preprint arXiv:2607.05184*, 2026. URL https://arxiv.org/abs/2607.05184.

Yuying Li, Leqi Zheng, Yongzi Yu, Wenrui Zhou, Xuchang Zhong, Xing Hu, Jing Jin, Hangjie Yuan, and Tao Feng. Filter, then reweight: Rethinking optimization granularity in on-policy distillation. *arXiv preprint arXiv:2606.02684*, 2026. URL https://arxiv.org/abs/2606.02684.

Alexander Lin, Jeremy Wohlwend, Howard Chen, and Tao Lei. Autoregressive knowledge distillation through imitation learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 6121–6133, 2020. doi: 10.18653/v1/2020.emnlp-main.494. URL https://aclanthology.org/2020.emnlp-main.494/.

Wenze Lin, Jiale Zhao, Xitai Jiang, Songde Rao, Yining Li, Shenzhi Wang, Bingxiang He, and Gao Huang. On-policy distillation with verifiable reward. *arXiv preprint arXiv:2608.24696*, 2026. URL https://arxiv.org/abs/2608.24696.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. Improving multi-task deep neural networks via knowledge distillation for natural language understanding. *arXiv preprint arXiv:1904.09482*, 2019. URL https://arxiv.org/abs/1904.09482.

David Lopez-Paz, Léon Bottou, Bernhard Schölkopf, and Vladimir Vapnik. Unifying distillation and privileged information. In *International Conference on Learning Representations*, 2016. URL https://arxiv.org/abs/1511.03643.

Kevin Lu and Thinking Machines Lab. On-policy distillation. Thinking Machines Lab: Connectionism, 2025. URL https://thinkingmachines.ai/blog/on-policy-distillation.

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*, 2026a. URL https://arxiv.org/abs/2606.30406.

Yingzi Ma, Zichen Zhu, Ming Jiang, and Chaowei Xiao. One student, many teachers: Multi-task on-policy distillation via soft-prompt privileged context. *arXiv preprint arXiv:2607.18293*, 2026b. URL https://arxiv.org/abs/2607.18293.

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. URL https://arxiv.org/abs/2608.04419.

Shyam Sundhar Ramesh, Xiaotong Ji, Matthieu Zimmer, Sangwoong Yoon, Zhiyong Wang, Haitham Bou Ammar, Aurelien Lucchi, and Ilija Bogunovic. Multi-task GRPO: Reliable LLM reasoning across tasks. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=zA3KhRZ9A5.

Victor Sanh, Albert Webson, Colin Raffel, et al. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=9Vrb9D0WI4.

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. doi: 10.1145/3689031.3696075. URL https://doi.org/10.1145/3689031.3696075.

An Yang, Anfeng Li, Baosong Yang, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Wenkai Yang, Weijie Liu, Ruobing Xie, Kai Yang, Saiyong Yang, and Yankai Lin. Learning beyond teacher: Generalized on-policy distillation with reward extrapolation. *arXiv preprint arXiv:2602.12125*, 2026. URL https://arxiv.org/abs/2602.12125.

Tianzhu Ye, Li Dong, Xun Wu, Shaohan Huang, and Furu Wei. On-policy context distillation for language models. *arXiv preprint arXiv:2602.12275*, 2026. URL https://arxiv.org/abs/2602.12275.

Zhiwei Zhang, Zechen Sun, Fei Zhao, Kang Peng, Bin Liang, Huayu Deng, Yao Hu, Kam-Fai Wong, and Mu Chuan. Verify before you distill: Prompt-level teacher gating for on-policy distillation. *arXiv preprint arXiv:2609.02998*, 2026. URL https://arxiv.org/abs/2609.02998.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=Jpxfof0EaS.

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

Siqi Zhu, Xuyan Ye, Hongyu Lu, Weiye Shi, and Ge Liu. The many faces of on-policy distillation: Pitfalls, mechanisms, and fixes. *arXiv preprint arXiv:2605.11182*, 2026. URL https://arxiv.org/abs/2605.11182.

A Data and Evaluation Protocols

A.1 Biology, Chemistry, and Physics

We use the public SciKnowEval V2 question collection (`hicai-zju/SciKnowEval`) to construct controlled multi-task distillation tasks. The official release provides a single `test` collection and no training or development split.[1] We define our own fixed training, development, and test partitions and share them across all compared methods.

We retain the L1 biology and chemistry literature multiple-choice questions and four-option physics literature questions. We group duplicate questions across domains using Unicode NFKC normalization, case folding, and collapsed whitespace, and exclude groups with conflicting answers or domain assignments. Each domain contributes 960 training, 100 development, and 200 test questions, with partitions fixed before training.

A cross-partition audit of normalized question stems, excluding choices and the common instruction, finds no exact matches or pairs with character 5-gram Jaccard similarity at least 0.8.

The question and options are preserved, while the original instruction is replaced with:

> Explain the key reasoning briefly, then give only the final option letter (A, B, C, or D) in `\boxed{...}`.

The ground-truth answer and teacher reference are absent from student inputs. A response is scored as correct when its final boxed option matches the ground-truth letter. Missing or malformed boxes and invalid options receive zero credit.

A.2 Materials, Chemistry, and Physics

The second mixture uses Material L1 knowledge, Chemistry L2 understanding, and Physics L3 calculation from SciKnowEval V2, with the same response instruction and answer-matching rule as Appendix A.1. Each task has 100 development and 200 test questions, shared across all six methods. The fixed 60-update training stream visits 960 distinct materials questions, 326 chemistry questions, and 474 physics questions, with 16 questions per task at each update. Smaller pools are cycled and reshuffled when exhausted. Other settings follow Table 6.

A.3 Physics, Instruction Following, and Code Generation

Task sources and partitions. The physics task reuses the initial SciKnowEval V2 partition. Instruction-following training uses `allenai/RLVR-IFeval`, while evaluation uses the official IFEval prompts and checkers (Zhou et al., 2023).[2] MBPP uses the original Google Research release with its official training, validation, and test assignments (Austin et al., 2021).[3]

Instruction-following data and verification. Training uses eight constraint types: keyword inclusion, keyword frequency, bullet count, paragraph count, word count, JSON format, uppercase, and lowercase. We retain original user prompts, exclude choice and label-classification requests, and restrict keyword counts to 1–8, bullet and paragraph counts to 2–8, and word-count constraints to 10–150. The training pool contains 750 distinct prompts from 552 base questions; all variants of held-out base questions are excluded. Each update includes two prompts per constraint type, cycling smaller pools as needed.

Training responses are scored by whether they satisfy the specified constraint. We use the released RLVR checkers, with uppercase and lowercase checks requiring at least one cased character. Official IFEval evaluation uses prompt-level strict scoring: every constraint in the original prompt must be satisfied. Normalized prompt and base-question matching, including containment checks for base questions of at least 64 characters, finds no training–test overlap under these checks.

[1] Pinned SciKnowEval V2 data card.

[2] Pinned sources: RLVR-IFeval training data and official IFEval evaluation.

[3] Pinned MBPP data.

Code data and verification. We remove MBPP problems with duplicate descriptions within or across splits before model generation, leaving 371 training problems and 90 validation problems. All retained reference implementations pass the original tests. Prompts contain the task description, test setup, and first assertion to specify the interface, and request a Python code block. Verification executes generated code with the setup and all three original assertions in an isolated process; errors and timeouts count as failures.

Training and evaluation. Settings follow Table 6. Physics visits 960 distinct training questions, while the instruction and code pools cycle as needed. The test set contains 200 physics questions, all 541 official IFEval prompts, and 496 MBPP problems after duplicate removal.

B Implementation and Computational Cost

B.1 Teacher Cache and Reference Prompt

For DuoOPD, we cache one independently generated teacher response per distinct training prompt, using temperature one, top-p one, no top-k restriction, seed 42, and a 1,024-token response limit. Responses are labeled using the task's answer-matching, instruction-checking, or code-testing rule (Appendix A). Cached responses and labels remain fixed during training.

When the cached teacher response passes the task check and the student's response fails it, the complete teacher response is supplied as a reference. Before applying the chat template, we append the following text to the teacher's last user message:

```
Use the following verified reference to solve the question.
<reference>
[Full teacher response that passes the task check]
</reference>
```

Other outcomes use original-context scoring; the teacher prompt limit is 5,632 tokens.

Token alignment. Each teacher–student pair shares a tokenizer and vocabulary. Both models score the same student response token IDs at aligned response positions; only the teacher context can include a reference. The loss includes response tokens and actual EOS tokens, excluding prompts, references, and padding.

B.2 Optimization and Implementation

Setting	Main comparison
Teacher / initial student	Qwen3-4B / Qwen3-0.6B; Llama-3.1-8B-Instruct / Llama-3.2-3B-Instruct; teachers frozen
Optimizer	AdamW; $\beta = (0.9, 0.95)$, weight decay 0.1, $\epsilon = 10^{-8}$
Learning rate / schedule	3×10^{-6}; cosine decay; 6 warmup steps
Updates / global batch	60 / 48; 16 questions per domain
Student sampling	One response per question; temperature 1; top-$p = 1$; unrestricted top-k
Prompt / response limits	4,096 / 1,024 tokens for the student
Evaluation	Checkpoint after update 60; chat template; temperature 1; no top-p or top-k truncation; seed 42; Qwen3 thinking disabled
Aggregation	Mean over valid tokens per response, then mean over responses

Table 6: Settings for the initial biology, chemistry, and physics comparison. Additional-mixture differences are specified in Appendices A.2 and A.3.

OPD and DuoOPD share the verl v0.7.0 scoring and update implementation (Sheng et al., 2025), with one rollout batch and one optimizer update per step. Feedback weights are computed in FP32 without centering and held fixed during each update, as specified in Section 2. Additional entropy bonuses, reference-model KL penalties, and rollout correction are disabled. Task-level shared weights are computed over the complete batch before microbatch splitting; tasks with no response

where only the student succeeds need no shared weight, and task sampling proportions are unchanged. Losses use the response averaging specified in Table 6.

The actor uses verl's policy-ratio loss. With one minibatch and one epoch per rollout batch, its on-policy branch sets the denominator log probability to the detached current log probability. Writing $\ell = \log \pi_\theta(o_{i,t} \mid q_i, o_{i,<t})$, the ratio $u = \exp(\ell - \mathrm{sg}[\ell])$ has value one, so clipping is inactive, and $\nabla_\theta[-\mathrm{sg}[A_{i,t}]u] = -\mathrm{sg}[A_{i,t}]\nabla_\theta \ell$. The implemented loss therefore has the same gradient as Eq.(8), though their scalar values differ.

EOPD retains an additional teacher-entropy-gated KL loss on its top-16 teacher support, with entropy threshold 0.8 and auxiliary coefficient one. Its actor microbatch is one because the upstream auxiliary loss averages valid tokens within a microbatch. This makes that term average within a response before gradient accumulation averages responses; other methods use actor microbatch four.

ExOPD uses reward extrapolation factor $\lambda = 1.25$ and the frozen initial student as its reference model. FiRe-OPD uses a trajectory-filtering percentile of 20 and teacher-confidence and student-entropy coefficients $\alpha = \beta = 1$; its filtering threshold and weight normalization are computed within each actor microbatch of four responses. Both retain upstream synchronous rollouts and token-level rollout importance sampling with threshold five.

B.3 Teacher Computation and Runtime

On the initial biology, chemistry, and physics mixture, Table 7 reports DuoOPD's training and teacher-cache cost relative to OPD. Only responses where the teacher alone succeeds receive a reference, and the extra scoring input is measured against scoring the same student responses without references. Runtimes are mean training-loop times over the 60 updates, covering generation, scoring, and updates and excluding initialization, checkpoint saving, evaluation, and queue time. The scoring and verification stage also includes reference preparation and worker communication. Cache generation is timed on the same basis and converted to GPU-time. Student generation accounts for most of the Llama runtime increase, with DuoOPD responses averaging 177 tokens versus 152 for OPD.

Quantity	Qwen	Llama
Cache generated / student-sampled tokens per run	1.01–1.10×	0.76–0.90×
Responses scored with a reference per run	20.7–22.6%	13.3–14.6%
Extra teacher scoring input per run	14.16–15.45%	6.21–6.39%
Scoring/verification share of OPD training loop	5.9%	4.7%
Training-loop change, cache excluded	−0.7%	+7.3%
Cache generation / one OPD training loop	29.7%	23.0%
Training loop + full cache ($R = 1$)	+29.0%	+30.3%
Training loop + cache shared across $R = 3$	+9.2%	+15.0%

Table 7: DuoOPD training and teacher-cache cost relative to OPD. The first three rows give per-run ranges; rows with cache compare combined GPU-time against OPD's training-loop GPU-time, amortizing one cache over R runs.

C Additional Results

C.1 Variability Across Training Runs

Table 8 reports the Macro mean and sample standard deviation for Tables 1 and 2.

C.2 Development Trajectories

Figure A1 tracks one training run per method on 100 development questions per task, with four sampled answers per question, every six updates from the shared initial student. In the displayed Llama runs, DuoOPD leads all baselines at every checkpoint from update 24 onward and stays above 70%, while several baselines peak earlier and then decline. On Qwen3, DuoOPD has the highest development Macro at seven of the ten checkpoints, including the last three.

Method	Qwen3, B/C/P	Llama, B/C/P	Qwen3, science	Qwen3, heterogeneous
OPD	64.39 ± 0.45	67.28 ± 1.25	62.33 ± 0.39	47.22 ± 0.36
EOPD	65.81 ± 0.54	66.49 ± 0.52	61.56 ± 0.58	46.90 ± 0.36
OPDVR	64.83 ± 0.89	68.68 ± 1.37	62.16 ± 0.17	47.83 ± 0.35
ExOPD	64.42 ± 0.37	66.34 ± 0.58	62.62 ± 0.01	46.83 ± 0.08
FiRe-OPD	64.58 ± 0.39	67.64 ± 0.79	61.72 ± 0.22	47.53 ± 0.02
DuoOPD	66.97 ± 0.26	73.26 ± 1.04	63.48 ± 0.37	49.24 ± 0.64

Table 8: Macro test avg@8 (%), mean ± standard deviation.

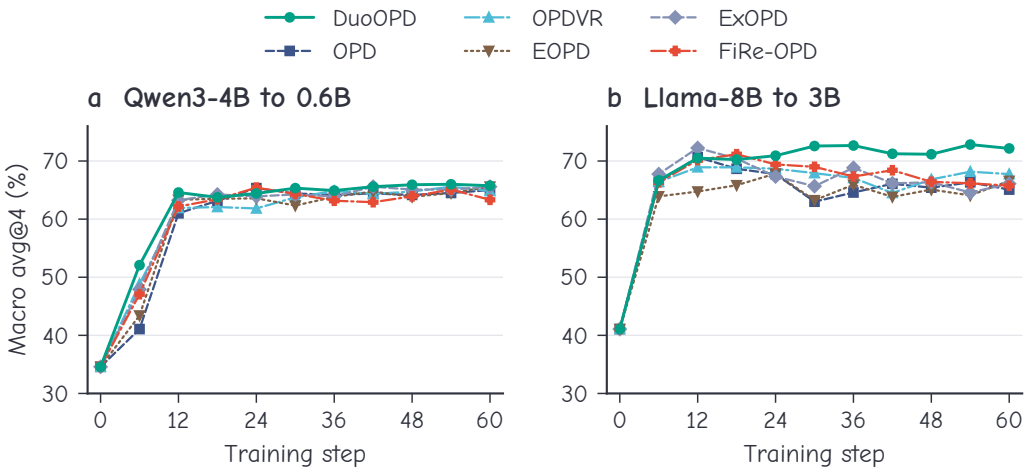

Figure A1: Development Macro avg@4 on biology, chemistry, and physics for Qwen3-4B → Qwen3-0.6B (left) and Llama-8B → Llama-3B (right). Checkpoints every six updates from the shared initial student; both panels share one vertical scale.

D How Joint Outcomes Change Token Feedback

We first compare feedback magnitudes under OPDVR's gate and DuoOPD's Softplus over all training responses, then illustrate on individual responses how teacher references and weight sharing allocate feedback and how student correctness determines its sign, each on a fixed student response.

D.1 Gated versus Softplus Feedback Magnitudes

Table 9 compares, on the same training responses, the mean weight that OPDVR's gate and DuoOPD's Softplus would assign from the original-context log-ratio d_0. Statistics pool three DuoOPD runs per family; since $\max\{d, 0\} = (d + |d|)/2$, the gated means follow from the recorded means of d_0 and $|d_0|$.

	Mean d_0		Gate		Softplus	
Joint outcome	Qwen3	Llama	Qwen3	Llama	Qwen3	Llama
Both succeed	-1.31	-0.22	$+0.09$	$+0.08$	$+0.57$	$+0.65$
Only the student succeeds	-1.24	-0.25	$+0.09$	$+0.09$	$+0.57$	$+0.64$
Both fail	-1.26	-0.24	-1.34	-0.33	-1.82	-0.88

Table 9: Mean token weight from d_0 under OPDVR's gate, $\sigma \max\{\sigma d_0, 0\}$, and under Softplus, $\sigma \operatorname{sp}(\sigma d_0)$, where $\sigma = 2r_S - 1$. The gate keeps little positive feedback on verified successes.

D.2 Reference-Guided Correction and Shared Reinforcement

Teacher correct, student incorrect: where to apply correction. In Figure A2 (top), the teacher selects ligand interactions, while the student selects superlattice formation. Adding the teacher reference strengthens negative feedback on "super" in the student's incorrect phrase. The student response and negative Softplus mapping stay fixed, isolating the reference's effect on correction.

Teacher incorrect, student correct: reinforce the correct selection. In Figure A2 (bottom), the student selects the correct NNP/MM answer despite the teacher's incorrect selection. Without sharing, the positive weight on "A" in "Only option A" is nearly zero; sharing gives it the task-level shared weight. This illustrates reinforcement of the correct selection: the student's earlier expansions of method names remain incorrect.

Teacher correct, Student incorrect

What controls nanoparticle interactions in nanocrystal self-assembly?

Teacher final: A. Ligand interactions (correct)
Student final: B. Superlattice formation (incorrect)

Student response excerpt

In nanocrystal self-assembly, the precise control over nanoparticle interactions in solution refers to **superlattice** **formation**. ...

Feedback on highlighted tokens

Token	super	l	attice	formation
No reference	-19.922	-0.693	-0.693	-0.693
Teacher reference	-26.795	-0.693	-0.693	-0.693

The reference strengthens suppression of the wrong selection.

Teacher incorrect, Student correct

Which method has lower MAE than ATM on most systems? PFKFB3 is the exception.

Teacher final: B. FEP+ (incorrect)
Student final: A. NNP/MM (correct)

Student response excerpt

... Only option **A** shows superiority with the best MAE for most systems. ...

Feedback on highlighted tokens

Token	A
No sharing	+0.00019
Weight sharing	+0.524

Such responses from this task in the batch share +0.524.
Correct option; earlier reasoning contains errors.

One column = one token · + reinforces · − suppresses
... marks omitted response text; token labels omit leading spaces.

Figure A2: **Feedback allocation when teacher and student disagree.** Top: a teacher reference changes negative token weights. Bottom: weight sharing reinforces a correct option that otherwise receives almost no positive feedback. Numeric columns are tokens of the fixed response.

D.3 Feedback Direction on Shared Success and Failure

Both correct: retain positive feedback. In Figure A3 (top), a token in the correct equipment name "Heating pads" receives OPD weight -3.83. Positive Softplus changes it to $+0.0215$, aligning feedback with the correct final answer.

Both incorrect: retain negative feedback. In Figure A3 (bottom), both models select a two-level qubit instead of the correct three-level system. The student's false assertion that the first law is violated receives mixed positive and negative OPD weights. Negative Softplus makes all displayed weights negative while preserving their ordering.

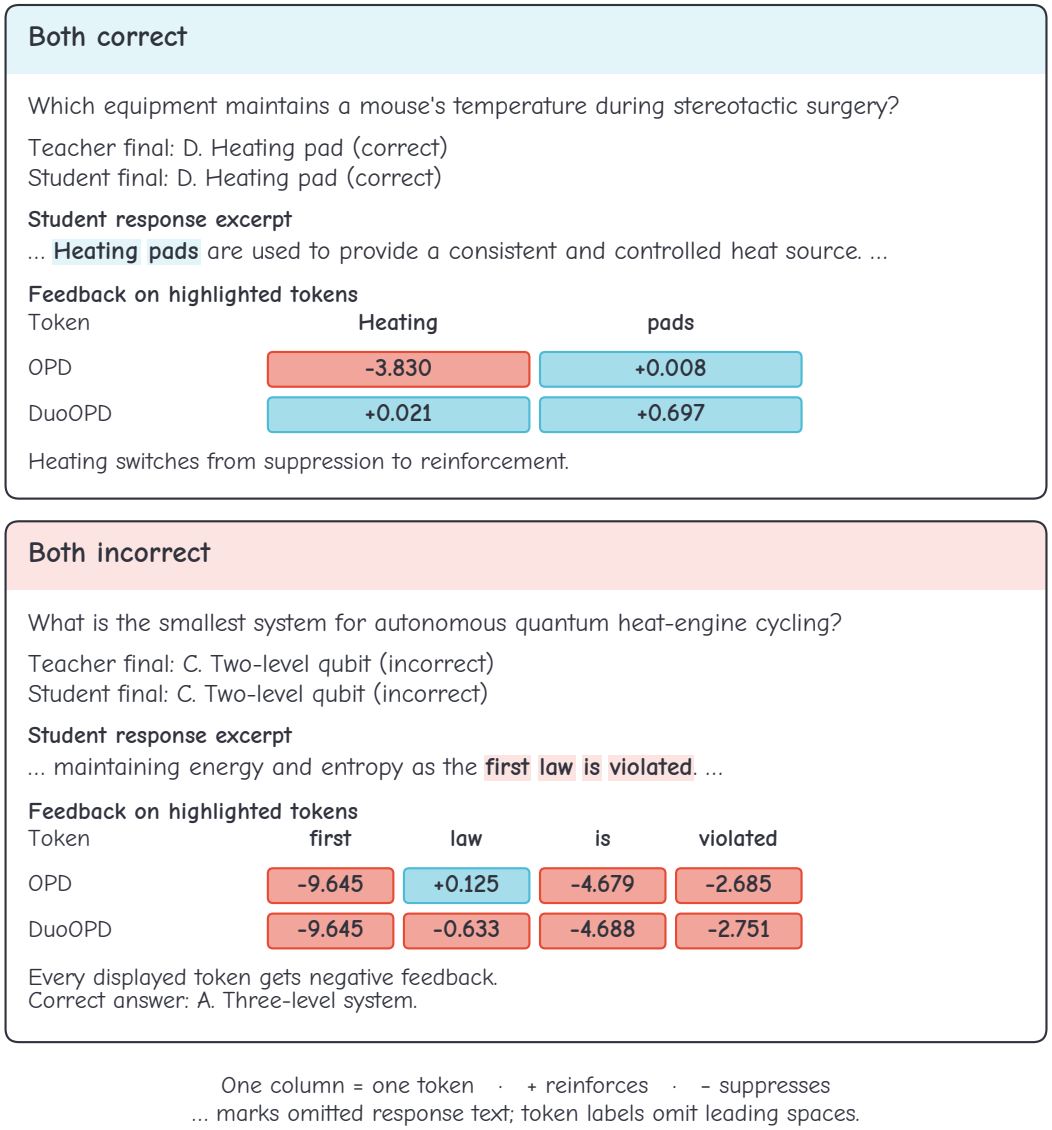

Figure A3: **Feedback direction when teacher and student agree.** Positive Softplus reinforces the successful response; negative Softplus suppresses the failed response. Numeric columns are tokens of the fixed response.

Case sources. Each case comes from the first rollout batch of Qwen3 training, when the student is still the initial model, and its token weights are computed with the initial teacher and student.