SELECTIVE FEEDBACK ATTENUATION FOR MULTI-TEACHER ON-POLICY DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-teacher on-policy distillation consolidates specialized capabilities into a single student by leveraging teacher feedback on student-generated responses. However, strictly matching teacher preferences can penalize valid alternative solutions, creating a fundamental misalignment between task correctness and imitation signals. To address this, we propose *Selective Attenuation for Multi-Teacher On-Policy Distillation (SA-MOPD)*. SA-MOPD uses verifier outcomes to attenuate negative feedback on successful student responses, while preserving positive feedback and standard supervision elsewhere. The rule modifies feedback coefficients within the existing distillation update. Our reverse-KL analysis identifies a mechanism by which disagreement among successful solutions can reduce student success despite a stronger teacher. A local analysis further characterizes when feedback selection improves task performance at a fixed attenuation budget. Our experiments demonstrate the value of selective supervision retention for capability integration: SA-MOPD raises recovery from Open-MOPD's 83.4% to 96.4% across mathematical reasoning, code generation, and instruction following. These gains extend to agentic tasks, where the single student surpasses the routed teacher ensemble in aggregate performance.

1 INTRODUCTION

Multi-teacher on-policy distillation (MOPD) consolidates specialist capabilities into a single language model by providing teacher feedback on student-generated responses (Ma et al., 2026; Gao et al., 2026). Supervision at the prefixes the student actually visits reduces the mismatch between teacher-generated training data and student behavior at inference time (Agarwal et al., 2024).

However, *matching a teacher's distribution is not equivalent to improving task success*. A task can admit multiple valid solutions to which a teacher assigns different probabilities. Teacher feedback can therefore penalize a successful student response that differs from the teacher's preferred solution, as shown in Figure 1. Yet a verified response can also carry positive teacher feedback, which is lost if its entire supervision is removed. Uniformly weakening negative feedback across responses can also reduce useful correction on failed attempts. Existing MOPD approaches route and balance supervision across specialists (Ma et al., 2026; Gao et al., 2026). Within this integration process, the student must balance protecting successful solutions with retaining supervision from each specialist. We ask: *can verified student success guide selective attenuation during multi-teacher integration, and does selection help beyond reducing feedback alone?*

To retain useful teacher supervision while protecting verified student solutions, we propose Selective Attenuation for Multi-Teacher On-Policy Distillation **(SA-MOPD)**, a novel selective attenuation rule for multi-teacher capability integration. For student responses that pass verification, SA-MOPD softly attenuates negative distillation coefficients through a retention factor, while preserving positive feedback and the full supervision on failed or unverifiable responses. This one-sided rule acts within the existing MOPD update, allowing verified outcomes to guide which feedback is relaxed.

To understand how selective attenuation affects learning, we analyze how disagreement among successful solutions can reduce student success under reverse-KL dynamics, even when the teacher succeeds more often. A local analysis then separates *where* feedback is attenuated from *how much* is removed. This motivates controls that match the attenuation budget globally or within each teacher domain, alongside a whole-response gating comparison. Together, these experiments assess the

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.

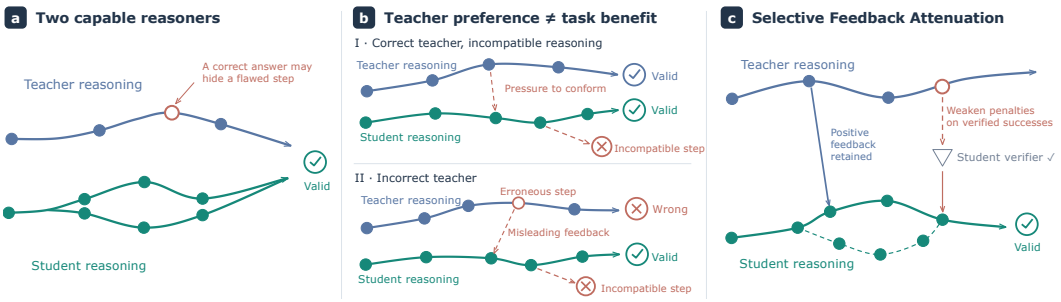

Figure 1: **Conceptual motivation: teacher consistency vs. task validity.** (a) Students can solve tasks via paths that differ from their teachers'. (b) MOPD derives feedback from teacher agreement; yet disagreement may reflect valid alternative solutions or teacher errors, not only student mistakes. (c) SA-MOPD attenuates negative feedback conditioned on verified student success while retaining positive signals. Observed examples and quantitative evaluations are provided in Section 5.5.

value of retaining supervision within verified responses and selecting where to relax teacher matching.

We evaluate capability integration in two settings: base capabilities with SmolLM3-3B and agentic tasks with Qwen-3.6-27B. Across six benchmarks covering mathematical reasoning, code generation, and instruction following, SA-MOPD improves all three domain averages over Open-MOPD and raises capability recovery from 83.4% to 96.4%. In the agentic setting, it outperforms mixed SFT and Open-MOPD on all four benchmarks and surpasses the routed teacher ensemble in aggregate performance.

Our contributions are summarized as follows:

- We introduce SA-MOPD, an outcome-guided rule for multi-teacher capability integration that relaxes negative feedback on verified successes while retaining positive feedback and supervision on other responses.
- We characterize a failure mechanism in teacher matching and derive a local criterion for beneficial feedback selection at equal attenuation mass.
- We demonstrate improvements in base and agentic capability integration and examine the role of supervision retention through whole-response gating and matched-attenuation controls.

2 Related Work

Model merging and capability integration. Capability integration seeks to combine specialized behaviors in one deployable model. Weight-space approaches combine trained checkpoints through parameter averaging (Wortsman et al., 2022) or task-vector arithmetic (Ilharco et al., 2023). Mixed-domain supervised fine-tuning and reinforcement learning instead train a shared model on pooled demonstrations or task rewards, with domain balance shaped by the training mixture (Gao et al., 2026). On-policy distillation transfers teacher knowledge through supervision on student-generated trajectories (Agarwal et al., 2024; Gu et al., 2024). By providing feedback at prefixes visited by the student, it reduces the mismatch between training sequences and the student's own generations. MOPD extends this approach to capability integration by routing student rollouts to domain specialists that provide dense token-level feedback (Ma et al., 2026). Combining these signals in a shared student requires allocating updates across domains with different response lengths and learning progress. Open-MOPD addresses this imbalance through token-share balancing, gap-following allocation, and reward refresh (Gao et al., 2026). Our work studies which feedback to retain within this integration process when the shared student already produces a verified solution.

Outcome-guided learning. Outcome feedback guides both policy optimization and the selection of training responses for language-model reasoning. DeepSeekMath studies outcome-supervised reinforcement learning with GRPO, using rewards normalized within groups of sampled responses

to construct policy-update advantages (Shao et al., 2024). DeepSeek-R1 uses verifiable correctness rewards for reasoning-oriented reinforcement learning and subsequently retains correct generated solutions through rejection sampling for supervised refinement (DeepSeek-AI, 2025). These uses connect response-level evaluation to both the strength of policy updates and the data used for further training. Very recent work has also brought outcome feedback into single-teacher on-policy distillation, using it to select trajectories or regulate token-level teacher supervision (Akhondzadeh et al., 2026; Xu et al., 2026; Lin et al., 2026). Related directions include verifying teacher reliability to route between OPD and GRPO (Zhang et al., 2026), and treating positive and non-positive feedback asymmetrically through reinforcement and localized divergence minimization (Jia et al., 2026). SA-MOPD uses verified outcomes to selectively soften negative teacher feedback on successful responses, while preserving positive feedback and full supervision elsewhere during multi-teacher capability integration.

3 SA-MOPD: Selective Feedback Attenuation

3.1 MOPD and Its Signed Feedback

Multi-teacher on-policy distillation trains a single student model on its own rollouts, delegating supervision for each prompt to a domain-specific teacher (Agarwal et al., 2024; Ma et al., 2026). For a prompt x_i from domain d_i, let $y_i \sim \pi_{\text{old}}(\cdot \mid x_i)$ denote the student rollout, $h_{it} = (x_i, y_{i,<t})$ its prefix at step t, and τ_{d_i} the corresponding teacher. Following the top-K learner of Open-MOPD (Gao et al., 2026), the feedback coefficients and baseline objective are defined as

$$\begin{aligned} A_{itk} &= \operatorname{sg}\left[q_{itk} \log \frac{\tau_{d_i}(k \mid h_{it})}{\pi_{\bar{\theta}}(k \mid h_{it})} \right], \qquad k \in S_{it}, \\ \mathcal{L}_{\text{MOPD}}(\theta) &= -\frac{1}{N} \sum_{i,t} m_{it} \sum_{k \in S_{it}} \psi(r_{itk}(\theta), w_i A_{itk}) \,. \end{aligned} \tag{1}$$

Here, S_{it} is the cached top-K support of the student, and q_{itk} is the student probability normalized over this support at the feedback-construction parameters $\bar{\theta}$. Open-MOPD refreshes these student-dependent feedback terms at each inner update while reusing cached teacher scores; $\operatorname{sg}[\cdot]$ blocks gradient flow through the resulting coefficients. The domain weight w_i combines token-share balancing and gap-following allocation (Gao et al., 2026). The importance ratio $r_{itk}(\theta) = \pi_\theta(k \mid h_{it})/\pi_{\text{old}}(k \mid h_{it})$ is processed by the clipped surrogate ψ, while m_{it} denotes the response mask and $N = \sum_{i,t} m_{it}$ the normalization constant. Appendix B details the normalization and clipping schemes (Schulman et al., 2017; Ye et al., 2020).

Diagnosis of Vanilla MOPD. Equation 1 assigns signed feedback from teacher–student probability ratios without conditioning on task success. A valid student choice can therefore receive negative feedback. As illustrated conceptually in Figure 1b, teacher disagreement may reflect an incompatible solution preference or erroneous guidance, rather than a student mistake. The learner needs a way to relax matching on valid responses while retaining correction elsewhere. SA-MOPD uses verified student success to select these contexts. Section 5.5 illustrates negative teacher feedback on verified student solutions.

3.2 Selecting Feedback via Verified Outcomes

To address this limitation, we propose SA-MOPD, which introduces outcome dependence directly into the baseline coefficient. Let $g_i = v_i c_i$, where $v_i \in \{0, 1\}$ indicates the availability of a verifier judgment and $c_i \in \{0, 1\}$ indicates success. Defining $A^+ = \max(A, 0)$ and $A^- = \min(A, 0)$, we replace A_{itk} in Eq. 1 with

$$\widetilde{A}_{itk} = A^+_{itk} + \big[1 - g_i(1 - \lambda_u)\big] A^-_{itk}, \qquad 0 \le \lambda_u \le 1. \tag{2}$$

Here, u denotes the global training step. In base-capability training, λ_u decreases linearly from 1 to 0 according to the schedule in Appendix B, Eq. 17. This formulation exclusively modulates the negative feedback for verified-successful responses, retaining only a fraction λ_u of its original magnitude. Positive feedback remains intact across all responses. For failed or unverifiable responses

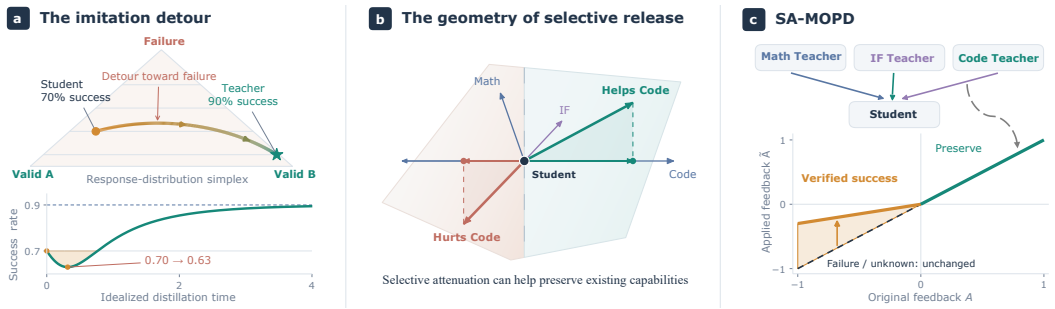

Figure 2: **From teacher matching to selective feedback attenuation.** (a) *The imitation detour.* Reverse-KL flow illustrates how matching a stronger teacher can reduce student success when within-outcome disagreement dominates its success advantage. (b) *Geometry of selective release.* Attenuating negative feedback changes the update direction and can help preserve existing capabilities when aligned with task improvement. (c) *SA-MOPD.* Relaxing penalties on successes while preserving supervision on failures, without explicit task-gradient estimation.

($g_i = 0$), the full baseline signal is preserved. Thus, the retention factor λ_u controls the strength of attenuation, while the gate g_i dictates its application scope. Standard MOPD is recovered when $\lambda_u = 1$, whereas setting $g_i = 1$ universally yields unconditional attenuation.

The response-level gate applies the same retention factor to negative feedback across all supervised candidates within a verified-successful response. SA-MOPD modifies only the coefficients within the existing learner, requiring neither a separate reward objective nor additional teacher rollouts. Implementation details are provided in Appendix B. The value of selection depends on whether the gate allocates attenuation to feedback whose release benefits the task (Figure 2b). Section 4 makes this connection precise by comparing selective and uniform attenuation at equal feedback mass.

4 From Teacher Matching to Task Influence

We analyze how teacher matching affects task success (Proposition 1) and how feedback selection changes the local task effect of an update at equal attenuation mass (Proposition 2).

4.1 Why Teacher Matching Need Not Improve Task Success

For a parameter-independent task score $r_d(x, y) \in [0, 1]$, define

$$J_d(\theta) = \mathbb{E}_{x\sim\mathcal{D}_d,\, y\sim\pi_\theta}[r_d(x, y)], \qquad h_d = \nabla_\theta J_d(\theta). \tag{3}$$

For smooth J_d, a small displacement ηv changes performance by

$$J_d(\theta + \eta v) - J_d(\theta) = \eta h_d^\top v + O(\eta^2). \tag{4}$$

Thus task improvement depends on alignment with h_d. Alignment among teacher-imitation gradients alone does not determine it (Appendix G.2). The following result makes this distinction explicit even for a stronger teacher.

Fix one prompt, a finite complete-response space, positive distributions p, τ, and a binary result $r(y)$. Let $v = \mathbb{E}_p r$, $u = \mathbb{E}_\tau r$ lie in $(0, 1)$, and write $K_+ = \mathrm{KL}(p(\cdot \mid r = 1)\|\tau(\cdot \mid r = 1))$, with K_- defined analogously for failure. The KL chain rule separates outcome probabilities from within-outcome matching:

$$\mathrm{KL}(p\|\tau) = \mathrm{KL}(\mathrm{Ber}(v)\|\,\mathrm{Ber}(u)) + vK_+ + (1 - v)K_-. \tag{5}$$

Proposition 1 (Task effect of ideal reverse-KL distillation) *Under Fisher natural-gradient descent on* $\mathrm{KL}(p\|\tau)$,

$$\dot{v} = v(1 - v)\big[\mathrm{logit}(u) - \mathrm{logit}(v) - K_+ + K_-\big]. \tag{6}$$

Even when the teacher succeeds more often ($u > v$), student success decreases if $K_+ - K_-$ exceeds the teacher's log-odds advantage. Thus within-outcome matching can oppose improved outcome probabilities. This establishes a failure mechanism for teacher matching (proof and counterexample in Appendix G.1). This motivates using verified outcomes to guide feedback attenuation. We next quantify how selective attenuation changes the local task effect of an update.

4.2 The value of selection at matched attenuation

Let $G_0 = -\nabla_\theta \mathcal{L}_{\mathrm{MOPD}}$ and $\beta = 1 - \lambda_u$. Removing response i's active negative feedback releases direction U_i with coefficient mass M_i, so $G_g = G_0 + \beta \sum_i g_i U_i$ (Appendix A.1). With update preconditioner B, define the mass-weighted distribution and local task benefit per unit released mass:

$$M = \sum_i M_i, \qquad \mu_i = \frac{M_i}{M}, \qquad e_{d,i} = \frac{h_d^\top B U_i}{M_i}, \qquad \bar{g} = \mathbb{E}_\mu g. \tag{7}$$

Zero-mass terms are omitted. Positive $e_{d,i}$ means that releasing this feedback locally benefits domain d.

Proposition 2 (Selection beyond matched attenuation) *Assume $M > 0$ and a smooth J_d with bounded local Hessian. From the same parameters, compare $\theta'_g = \theta + \eta B(G_0 + \beta \sum_i g_i U_i)$ with $\theta'_{\mathrm{uniform}}$, which replaces every g_i by $\bar{g}$. For fixed rollout data, detached coefficients, and gates, and a common fixed B, these updates release equal coefficient mass and satisfy*

$$J_d(\theta'_g) - J_d(\theta'_{\mathrm{uniform}}) = \eta\beta M \,\mathrm{Cov}_\mu(g, e_d) + O(\eta^2). \tag{8}$$

Selection has a positive first-order advantage when the gate favors responses with greater release benefit per unit mass. SA-MOPD selects responses using verified success. The proposition separates *where* feedback is attenuated from *how much* is removed, motivating two controls: Global matched fixes the total attenuation mass, while Domain matched also fixes its allocation across teachers. Section 5.4 evaluates both through actual updates and task outcomes.

The proof and within-domain decomposition are developed in Appendix A.2. A complementary optimizer audit finds 96.04%–100% sign agreement between the common-preconditioner model and actual Adam task projections for SA-MOPD versus Domain matched at an intermediate checkpoint (Appendix G.5).

5 Experiments

5.1 Experimental Setup

Models and baselines. We evaluate capability integration in two settings. *(i) Base capabilities:* We follow Open-MOPD's protocol (Gao et al., 2026) for data preparation, baseline implementations, and evaluation. The student is SmolLM3-3B, initialized from the four-epoch mixed-domain SFT model π_{mixsft}. Training data come from Open-MOPD-Data (BytedTsinghua-SIA, 2026), with initialization using its OpenR1-Math (Open R1 Team, 2025), OpenCodeReasoning (NVIDIA, 2025), and Instruction-Nemotron subsets. We compare against single-domain RL, single-domain OPD (Agarwal et al., 2024), RFT, mixed-domain RL, parameter averaging (Wortsman et al., 2022), task arithmetic (Ilharco et al., 2023), Vanilla MOPD (Ma et al., 2026), and Open-MOPD, the direct reference for our capability-integration setting. A separate comparison evaluates SA-MOPD against RG-OPD†, our multi-teacher adaptation of RG-OPD that retains its original trajectory-gating rule. RouteRL and RouteOPD provide multi-model references using ground-truth domain routing. SA-MOPD adds selective attenuation to Open-MOPD's top-K learner. *(ii) Agent capabilities:* The student and three teachers share a mixed-domain SFT initialization of Qwen-3.6-27B (Qwen Team, 2026). The teachers undergo subsequent specialist RL, and the student integrates their capabilities through multi-teacher distillation. We compare SA-MOPD with mixed SFT and Open-MOPD, using routed specialists as the multi-model reference. Teacher construction, reference routing, and the task-outcome gate are described in Appendix B.2.

Table 1: **Base capability integration on SmolLM3-3B.** SA-MOPD reports means over three independent training runs. Gray rows denote routed specialists. Bold and underlined values mark the best and second-best scores among Baselines and M-OPD methods.

	Math (avg@64)			Code (avg@10)			IF				
Method	AIME24	AIME25	avg	LCBv5	LCBv6	avg	IFEval	IFB_{test}	avg	Total	Recovery rate
SmolLM3-3B-Base	2.08	1.72	1.90	3.41	6.17	4.79	16.08	13.00	14.54	7.08	—
SFT Student	15.63	20.26	17.95	15.99	19.20	17.60	66.91	16.00	41.46	25.67	—
RL											
$\pi_{\phi_{\text{math}}}$	23.65	24.84	24.24	—	—	—	—	—	—	—	—
$\pi_{\phi_{\text{code}}}$	—	—	—	22.16	21.31	21.73	—	—	—	—	—
$\pi_{\phi_{\text{IF}}}$	—	—	—	—	—	—	74.49	27.67	51.08	—	—
RouteRL	23.65	24.84	24.24	22.16	21.31	21.73	74.49	27.67	51.08	32.35	100%
OPD											
Math-OPD	22.34	23.96	23.15	—	—	—	—	—	—	—	—
Code-OPD	—	—	—	22.28	21.14	21.71	—	—	—	—	—
IF-OPD	—	—	—	—	—	—	75.60	24.00	49.80	—	—
RouteOPD	22.34	23.96	23.15	22.28	21.14	21.71	75.60	24.00	49.80	31.55	88.0%
Baselines											
RFT	<u>22.97</u>	<u>23.91</u>	<u>23.44</u>	18.98	19.43	19.21	55.08	18.67	36.87	26.51	12.6%
π_{mixrl}	21.15	22.14	21.64	16.59	20.97	18.78	70.24	22.67	46.45	28.96	49.3%
ParamMerge-Avg	18.91	20.99	19.95	18.38	21.20	19.79	70.06	17.67	43.86	27.87	32.9%
ParamMerge-TA	21.93	22.76	22.34	<u>21.74</u>	**23.14**	<u>22.44</u>	71.53	21.67	46.60	30.46	71.7%
M-OPD											
Vanilla MOPD	20.92	21.60	21.26	17.53	20.99	19.26	68.61	18.67	43.64	28.05	35.6%
Open-MOPD	21.98	22.86	22.42	20.84	<u>22.63</u>	21.73	<u>74.49</u>	**24.67**	<u>49.58</u>	<u>31.24</u>	<u>83.4%</u>
SA-MOPD (ours)	**23.44**	**24.89**	**24.17**	**22.33**	<u>22.63</u>	**22.48**	**75.05**	<u>24.33</u>	**49.69**	**32.11**	**96.4%**

Benchmarks and metrics. *(i) Base capabilities:* We evaluate Math on AIME 2024/2025 (Mathematical Association of America, n.d.), Code on LiveCodeBench v5/v6 (Jain et al., 2024), and IF on IFEval and IFBench (Zhou et al., 2023; Pyatkin et al., 2025). Each domain score averages its two benchmarks. For Math and Code, we report avg@k, the mean correctness over k samples, with k=64 and k=10, respectively, rather than pass@k. *(ii) Agent capabilities:* We evaluate BFCL-v4 (Patil et al., 2025), TAU2 (τ^2-Bench; Barres et al., 2025), Claw-Eval (Ye et al., 2026), and Skills-Bench (Li et al., 2026). *Total* is the unweighted mean of the three domain scores in the base setting and the four benchmark scores in the agent setting. Following Open-MOPD (Gao et al., 2026), we also report *Recovery rate*, $100(S - S_{\text{SFT}})/(S_{\text{route}} - S_{\text{SFT}})$, where S is Total. This measures the fraction of the improvement from mixed SFT to routed specialists recovered by a single student. We use RouteRL for base capabilities and the routed teachers for agent capabilities, with fixed reference scores across methods within each setting. Recovery above 100% means that the student exceeds the routed reference in aggregate.

5.2 Main results: basic capability integration

We first evaluate how effectively SA-MOPD integrates specialist capabilities in Math, Code, and instruction following into a single student. The results are presented in Table 1, which reports SA-MOPD's mean scores over three training runs; per-benchmark training variability is provided in Appendix C.1. SA-MOPD reaches 32.11 Total, improving over Open-MOPD by 0.87 pp and increasing capability recovery from 83.4% to 96.4%. This aggregate advantage spans all three domains, yielding consistent gains over both Open-MOPD (Math $+1.75$, Code $+0.75$, IF $+0.11$ pp) and Vanilla MOPD. Benchmark-wise, SA-MOPD improves performance on both AIME variants, LiveCodeBench v5, and IFEval, ties LiveCodeBench v6 at 22.63, and trails IFBench by 0.34 pp (24.33 vs. 24.67). Compared against the routed RouteRL reference, SA-MOPD narrows the initial performance shortfall left by Open-MOPD by 78.4%, leaving a gap of 0.24 pp. The student exceeds the routed reference in Code, while IF accounts for the largest remaining domain deficit. Overall, under the Recovery metric (Gao et al., 2026), SA-MOPD demonstrates that a single student can recover 96.4% of the aggregate improvement achieved when moving from mixed SFT to routed

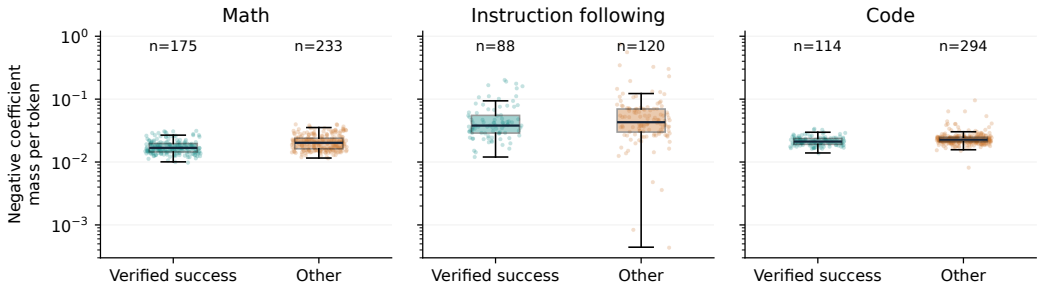

Figure 3: **Negative feedback across verification outcomes.** Per-token negative feedback mass summed over top-16 candidates (including unsampled) in a frozen batch of responses generated by an intermediate MOPD baseline checkpoint; "Other" denotes responses without verified success. The y-axis is base-10 log-scaled, with boxes showing medians and IQRs ($1.5\times$ whiskers). All 377 verified-successful responses contain negative feedback.

Table 2: **Agentic task performance on Qwen-3.6-27B.** Total averages four benchmarks. Teachers 1–3 are RL specialists derived from the shared mixed-SFT initialization. Route Teacher selects the listed specialist for each benchmark. Bold marks the best consolidated student.

Method	BFCL v4	TAU2	Claw-Eval	SkillsBench	Total	Recovery
Student (mixed SFT)	69.79	81.46	72.65	35.90	64.95	—
Teacher 1	72.04	—	—	—	—	—
Teacher 2	—	84.96	—	—	—	—
Teacher 3	—	—	74.03	49.99	—	—
Route Teacher	72.04	84.96	74.03	49.99	70.26	100%
Open-MOPD	71.31	83.46	73.42	49.80	69.50	85.7%
SA-MOPD (ours)	**72.17**	**84.36**	**73.72**	**52.83**	**70.77**	**109.7%**

specialists. The agent experiments below further examine this integration dynamic in settings where the student ultimately surpasses the routed reference.

5.3 Extension to Agentic Tasks

We further evaluate SA-MOPD on Qwen-3.6-27B across four agentic benchmarks. In this setting, selective attenuation is gated by task-level success; for multi-turn interactions, the gate uses the outcome of the full interaction (Appendix B.2). As shown in Table 2, SA-MOPD outperforms both mixed SFT and Open-MOPD on all four benchmarks, reaching a Total of 70.77, compared with 64.95 and 69.50, respectively. The largest gain over Open-MOPD occurs on SkillsBench ($+3.03$ pp). Relative to the routed specialist reference, SA-MOPD increases recovery from Open-MOPD's 85.7% to 109.7%, exceeding the reference Total by 0.51 pp. The student surpasses the designated specialists on BFCL v4 and SkillsBench, while remaining below them on TAU2 and Claw-Eval. These results extend the capability-integration gains to a larger model and agentic tasks, with a single consolidated student exceeding the routed specialists in aggregate.

5.4 More Analysis

Negative feedback density across task outcomes. We examine how teacher disagreement relates to task success across the 1,024 source responses used in the controlled study below (Figure 3). Negative feedback occurs in every verified success, while its per-token density overlaps substantially across outcome groups. The success group has a lower median in each domain. Thus the presence or magnitude of negative feedback alone does not separate successful responses from the remaining responses, motivating the use of verification as an additional selection signal. Appendix F provides the length-stratified analysis.

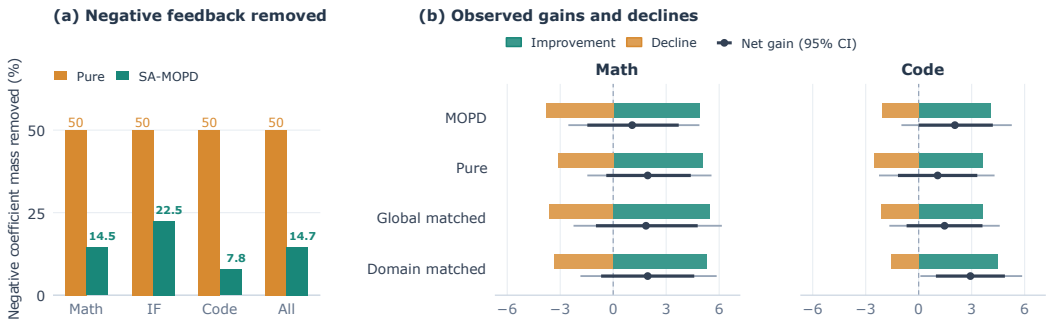

Figure 4: **Feedback removal and task gains.** (a) Negative-coefficient mass removed at $\lambda_u = 0.5$. (b) SA-MOPD minus control after one update (pp): bars show mean positive/negative contributions; dots show net gains. Thick/thin whiskers indicate pointwise/ Bonferroni-adjusted 95% paired prompt-bootstrap CIs.

Retaining supervision within verified responses. A verifier identifies a successful response, but the response can still contain both positive and negative teacher feedback. We therefore examine response-level gating and selective attenuation of negative coefficients. For comparison with concurrent work, we adapt RG-OPD (Akhondzadeh et al., 2026) to the multi-teacher On-MOPD setting. RG-OPD† retains the original trajectory-gating rule and applies it using each response's routed teacher. A retained response receives its full distillation supervision; a rejected response contributes no distillation loss (Appendix B.1). Table 3 compares this baseline with SA-MOPD, which reaches a higher Total (32.11 vs. 31.57), with gains of 0.27, 0.38, and 0.98 pp in Math, Code, and IF, respectively. The gains across all three domains favor SA-MOPD's selective-retention rule in this comparison.

Table 3: **Feedback-retention ablation.** †: multi-teacher adaptation of RG-OPD with its original trajectory-gating rule applied using the routed teacher. Domain averages and Total; bold marks higher scores.

Method	Math	Code	IF	Total
RG-OPD†	23.90	22.10	48.71	31.57
SA-MOPD	**24.17**	**22.48**	**49.69**	**32.11**

Controlled feedback-selection ablation. To distinguish the value of response selection from the amount of feedback removed, we next compare five updates from the same student state (Table 4). MOPD retains the original feedback; *Pure* halves negative feedback on every response; *SA-MOPD* halves it only on verified successes. *Global matched* removes the same total negative-coefficient mass as SA-MOPD by uniform attenuation; *Domain matched* matches the removed mass within each source domain. All branches preserve positive feedback and use the same 1,024 source responses, cached teacher scores, initial optimizer state, and learning rate of 1.5×10^{-6}. The unchanged starting model is also reported. We evaluate each model on a held-out diagnostic set of 128 prompts per domain, distinct from the main benchmarks and the update batch, with eight fresh responses per prompt. Appendix D details the protocol.

Table 4: **Single-update ablation of feedback selection.** Verifier success (%); Avg. is the unweighted mean across the three domains. Bold marks the highest score in each column.

Method	**Math**	**IF**	**Code**	**Avg.**
Before update	41.89	43.46	28.71	38.02
MOPD	42.68	45.02	29.00	38.90
Pure	41.80	**45.31**	29.98	39.03
Global matched	41.89	43.65	29.59	38.38
Domain matched	41.80	44.92	28.12	38.28
SA-MOPD	**43.75**	45.21	**31.05**	**40.01**

Selection beyond attenuation amount. The controlled Code comparison shows that choosing *where* to relax feedback improves task performance beyond controlling *how much* is removed. At the same per-teacher coefficient budget, SA-MOPD exceeds Domain matched by 2.93 pp (Bonferroni-adjusted 95% paired prompt-bootstrap CI: $[0.10, 5.86]$). The prompt-level analysis (Figure 4b) reveals a broader set of improvements than regressions, connecting this advantage to how the update

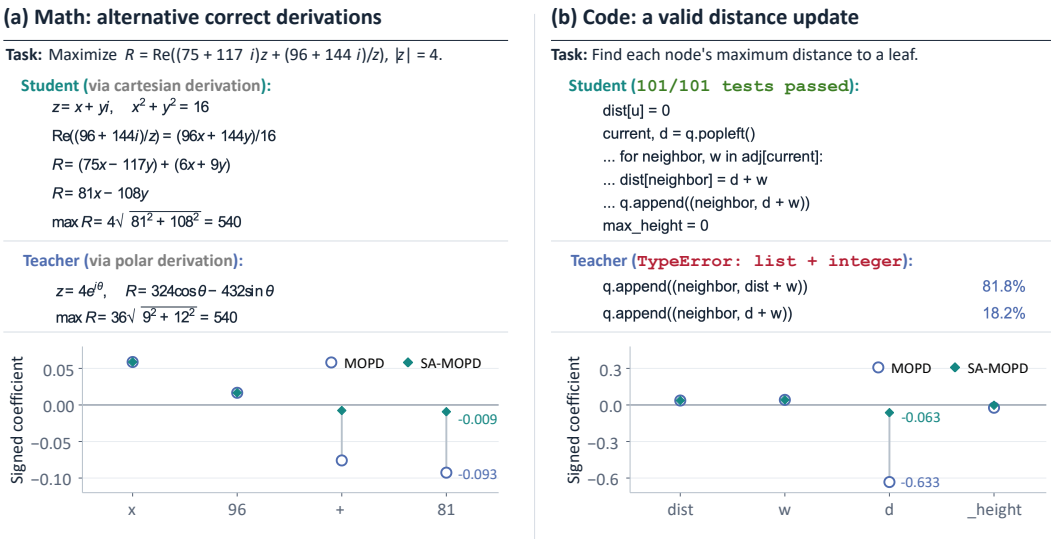

Figure 5: **Selective attenuation on verified solutions.** (a) Student and teacher reach the same correct Math answer through different derivations. (b) At the student's Code prefix, the teacher favors `dist` over the valid scalar `d`; probabilities score these tokens with the surrounding code held fixed. SA-MOPD ($\lambda_u = 0.1$) preserves positive feedback and reduces negative coefficients by 90%.

affects individual tasks. Across domains, SA-MOPD achieves the highest average, while Pure remains slightly better on IF. Together, these findings support response selection as a useful control over the task effects of distillation, with benefits that vary across domains.

5.5 Qualitative Examples

Figure 5 connects concrete solution choices to their teacher feedback. For the Math problem, the teacher's polar derivation and the student's Cartesian derivation both yield 540. Scoring the student's actual prefixes reveals different signs within its correct reasoning: the parameterization token `x` receives positive feedback ($+0.0588$), while `81` in the valid reduction $R = 81x - 108y$ receives negative feedback (-0.0926). SA-MOPD retains the former and reduces the latter to -0.00926. In Code, the teacher favors a different variable at a prefix where the student's scalar-distance update belongs to a program passing all 101 tests. Its negative coefficient decreases from -0.633 to -0.063, while positive coefficients elsewhere in the same program remain intact. These audits show how selective attenuation reduces penalties on verified student reasoning while retaining teacher support. Further complete-response comparisons appear in Appendix E.

6 Discussion and Conclusion

SA-MOPD studies how to relax teacher matching on verified student solutions while retaining supervision for multi-teacher capability integration. It softly attenuates only negative top-K distillation coefficients on verified-successful responses, preserving positive coefficients and the full supervision on failed or unverifiable responses. Our analysis identifies a conflict between teacher matching and task success and characterizes the local value of feedback selection at equal attenuation mass.

Across base-capability and agentic evaluations, SA-MOPD improves over Open-MOPD, narrowing the aggregate gap to routed specialists in the base setting and exceeding the routed reference in aggregate in the agent setting. The controlled single-update study further finds a Code advantage over uniform attenuation matched within each source domain, providing evidence that selecting where to attenuate can matter beyond the amount of feedback removed. Together, these findings support selective feedback retention as a way to balance protecting verified student solutions with continued learning from specialist teachers.

AI USE STATEMENT

Generative AI assisted with literature organization, manuscript drafting, mathematical analysis, implementation debugging, and numerical checks. The reported tables were checked against the supplied result tables, including score aggregation and numerical comparisons.

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*, 2024. URL https://arxiv.org/abs/2306.13649.

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12248–12267, 2024. URL https://aclanthology.org/2024.acl-long.662/.

Mohammad Sadegh Akhondzadeh, Vijay Lingam, Atula Tejaswi, Chanakya Ekbote, Sujay Sanghavi, and Aleksandar Bojchevski. Reward-gated on-policy distillation, 2026. URL https://arxiv.org/abs/2607.04037.

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2-Bench: Evaluating conversational agents in a dual-control environment, 2025. URL https://arxiv.org/abs/2506.07982.

BytedTsinghua-SIA. DAPO-Math-17k. Hugging Face dataset, 2025. URL https://huggingface.co/datasets/BytedTsinghua-SIA/DAPO-Math-17k.

BytedTsinghua-SIA. Open-MOPD-Data. Hugging Face dataset, 2026. URL https://huggingface.co/datasets/BytedTsinghua-SIA/Open-MOPD-Data.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Huan-ang Gao, Haohan Chi, Yong Yan, Shiyuan Feng, Hanlin Wu, Zheng Jiang, Bingxiang He, Wei-Ying Ma, Ya-Qin Zhang, and Hao Zhou. Open-MOPD: Diagnosing and fixing capability imbalance in multi-teacher on-policy distillation, 2026. URL https://arxiv.org/abs/2608.19098.

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://arxiv.org/abs/2306.08543v4.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2212.04089.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code, 2024. URL https://arxiv.org/abs/2403.07974.

Nan Jia, Haojin Yang, Xing Ma, Jiesong Lian, Shuailiang Zhang, Weipeng Zhang, Ke Zeng, Xunliang Cai, and Zequn Sun. Asymmetric on-policy distillation: Bridging exploitation and imitation at the token level, 2026. URL https://arxiv.org/abs/2605.06387.

Rongao Li, Jie Fu, Bo-Wen Zhang, Tao Huang, Zhihong Sun, Chen Lyu, Guang Liu, Zhi Jin, and Ge Li. TACO: Topics in algorithmic COde generation dataset, 2023. URL https://arxiv.org/abs/2312.14852.

Xiangyi Li, Yimin Liu, Wenbo Chen, Bingran You, Zonglin Di, Yifeng He, Shenghan Zheng, et al. SkillsBench: Benchmarking how well agent skills work across diverse tasks, 2026. URL https://arxiv.org/abs/2602.12670.

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

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

Mathematical Association of America. American Invitational Mathematics Examination (AIME). MAA Invitational Competitions, n.d. URL https://maa.org/maa-invitational-competitions/. Competition source for AIME 2024 and 2025; accessed September 24, 2026.

NVIDIA. OpenCodeReasoning. Hugging Face dataset, 2025. URL https://huggingface.co/datasets/nvidia/OpenCodeReasoning.

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

Shishir G. Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. The Berkeley Function Calling Leaderboard (BFCL): From tool use to agentic evaluation of large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 48371–48392, 2025. URL https://proceedings.mlr.press/v267/patil25a.html.

Valentina Pyatkin, Saumya Malik, Victoria Graf, Hamish Ivison, Shengyi Huang, Pradeep Dasigi, Nathan Lambert, and Hannaneh Hajishirzi. Generalizing verifiable instruction following, 2025. URL https://arxiv.org/abs/2507.02833.

Qwen Team. Qwen3.6-27B: Flagship-level coding in a 27B dense model, April 2026. URL https://qwen.ai/blog?id=qwen3.6-27b.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992. doi: 10.1007/BF00992696. URL https://doi.org/10.1007/BF00992696.

Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: Averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 23965–23998, 2022. URL https://proceedings.mlr.press/v162/wortsman22a.html.

Haoran Xu, Hongyu Wang, Yifei Gao, Jiaze Li, Xiaofeng Zhang, and Xiaosong Yuan. SG-OPD: Sign-gated on-policy distillation via sign-consistency gating and phased teacher sampling, 2026. URL https://arxiv.org/abs/2606.09304.

Bowen Ye, Rang Li, Qibin Yang, Yuanxin Liu, Linli Yao, Hanglong Lv, Zhihui Xie, Chenxin An, Lei Li, Lingpeng Kong, Qi Liu, Zhifang Sui, and Tong Yang. Claw-Eval: Towards trustworthy evaluation of autonomous agents, 2026. URL https://arxiv.org/abs/2604.06132.

Deheng Ye, Zhao Liu, Mingfei Sun, Bei Shi, Peilin Zhao, Hao Wu, Hongsheng Yu, Shaojie Yang, Xipeng Wu, Qingwei Guo, Qiaobo Chen, Yinyuting Yin, Hao Zhang, Tengfei Shi, Liang Wang, Qiang Fu, Wei Yang, and Lanxiao Huang. Mastering complex control in MOBA games with deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020. URL https://arxiv.org/abs/1912.09729.

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, 2026. URL https://arxiv.org/abs/2609.02998.

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

A PROOFS OF THE LOCAL ANALYSIS

A.1 GRADIENT DIFFERENCE UNDER CLIPPING

Use the baseline notation in Eq. 1 with detached advantages. At refreshed feedback, write $p_{itk} = \pi_\theta(k \mid h_{it})$, $\tau_{itk} = \tau_{d_i}(k \mid h_{it})$, $\zeta_{itk} = [\log(p_{itk}/\tau_{itk})]_+$, and $\chi^-_{itk} = \mathbf{1}\{1 - \epsilon_- < r_{itk} < c_{\text{dual}}\}$. Away from clipping boundaries and with the numerical log-ratio clamp inactive, the release coefficients are

$$\omega_{itk} = \frac{m_{it} w_i}{N} r_{itk} \chi^-_{itk} q_{itk} \zeta_{itk}, \qquad U_i = \sum_{t,k \in S_{it}} \omega_{itk} \nabla_\theta \log p_{itk}, \qquad M_i = \sum_{t,k \in S_{it}} \omega_{itk}. \tag{9}$$

For a negative advantage $a < 0$, the dual-clipped surrogate in Eq. 19 has derivative ra with respect to $\log r$ when $1 - \epsilon_- < r < c_{\text{dual}}$, and zero outside this interval. Multiplying a by any positive attenuation coefficient preserves both its sign and the active interval. At coefficient zero the term vanishes, so the same difference formula follows by continuity. Because $-A^-_{itk} = q_{itk}\zeta_{itk}$, subtraction of the baseline gradient gives

$$G_g - G_0 = \frac{\beta}{N} \sum_{i,t,k \in S_{it}} m_{it} w_i g_i r_{itk} \chi^-_{itk} q_{itk} \zeta_{itk} \nabla_\theta \log p_{itk} = \beta \sum_i g_i U_i. \tag{10}$$

The response-level gate controls negative-feedback attenuation across all supervised candidates within the response.

A.2 PROOF OF PROPOSITION 2

For a fixed target domain, write $J = J_d$ and $e = e_d$. The deterministic uniform control uses $\bar{g}$ on every response. Its removed mass is $\beta \bar{g} M = \beta \sum_i g_i M_i$, exactly matching selection. Taylor expansion around the common initial parameters gives

$$J(\theta'_g) - J(\theta'_{\text{uniform}}) = \eta\beta \sum_i (g_i - \bar{g}) \nabla J^\top B U_i + O(\eta^2) = \eta\beta M \operatorname{Cov}_\mu(g, e) + O(\eta^2).$$

For a binary gate with $0 < \bar{g} < 1$,

$$\operatorname{Cov}_\mu(g, e_d) = \bar{g}(1 - \bar{g})\big(\mathbb{E}_\mu[e_d \mid g = 1] - \mathbb{E}_\mu[e_d \mid g = 0]\big).$$

Thus selection has a positive first-order advantage when selected responses have greater release benefit per unit mass. If no active negative mass is present, both interventions coincide.

For a domain-matched control, set $\bar{g}_e = \mathbb{E}_{\mu_e} g$ within source domain e, where $\mu_{e,i} = M_i / M_e$ and $M_e = \sum_{i:d_i=e} M_i$. Subtracting that control gives

$$J_d(\theta'_g) - J_d(\theta'_{\text{domain}}) = \eta\beta \sum_e M_e \operatorname{Cov}_{\mu_e}(g, e_d) + O(\eta^2).$$

Domains with zero mass contribute zero.

A.3 REVERSAL RELATIVE TO THE MOPD UPDATE

For comparison with the original MOPD update, write $\bar{e}_d = \mathbb{E}_\mu e_d$.

Corollary 1 (Reversal of the local task effect) *Under Proposition 2, with $\eta\beta M > 0$, let $\Delta^a_d = J_d(\theta'_a) - J_d(\theta'_0)$ compare attenuation a with the same MOPD step $\theta'_0 = \theta + \eta B G_0$. Then*

$$\begin{aligned} \Delta^{\text{all}}_d &= \eta\beta M \bar{e}_d + O(\eta^2), \\ \Delta^g_d &= \eta\beta M [\bar{g}\bar{e}_d + \operatorname{Cov}_\mu(g, e_d)] + O(\eta^2). \end{aligned} \tag{11}$$

Unconditional attenuation has a negative first-order effect and selective attenuation a positive one exactly when

$$\bar{e}_d < 0, \qquad \operatorname{Cov}_\mu(g, e_d) > -\bar{g}\bar{e}_d. \tag{12}$$

Proof. Taylor expansion against $\theta'_0 = \theta + \eta BG_0$ gives, for any fixed gate a_i,

$$J_d(\theta'_a) - J_d(\theta'_0) = \eta\beta \sum_i a_i \nabla J_d^\top BU_i + O(\eta^2) = \eta\beta M \mathbb{E}_\mu[ae_d] + O(\eta^2). \tag{13}$$

Taking $a_i = 1$ and $a_i = g_i$, and expanding $\mathbb{E}_\mu[ge_d] = \bar{g}\bar{e}_d + \mathrm{Cov}_\mu(g, e_d)$, proves Eq. 11. The strict inequalities in Eq. 12 give opposite first-order signs, which persist for sufficiently small positive steps.

Example: preserving correction while relaxing imitation. Consider two equally likely reasoning routes whose success probabilities share a scalar parameter: $s_A(\theta) = \sigma(\theta + \log 4)$ and $s_B(\theta) = \sigma(\theta - \log 4)$, where σ is logistic. At $\theta = 0$, these are 0.8 and 0.2; let the teacher's probabilities be 0.6 and t. Both policies use the same fixed routing probabilities. The task score $J(\theta) = (s_A + s_B)/2$ is strictly increasing in θ. With full binary candidate support, detached feedback, and $B = I$, removing negative feedback releases directions $u_A = 0.16a$ and $u_B = -0.16b$, where $a = \log(4/3)$ and $b = \log[0.8/(1-t)]$ for $t > 0.2$. Thus release favors the correct candidate on route A but weakens correction of the incorrect candidate on route B.

Enumerating all four trajectories, the success gate activates with probabilities 0.8 and 0.2 on the respective routes. Equal routing and two-token normalization give

$$U_{\mathrm{all}} = 0.04(a - b), \qquad U_g = 0.04(0.8a - 0.2b).$$

Whenever $a < b < 4a$, unconditional release is negative and gated release positive. Adding these releases with the same positive attenuation strength to the common MOPD direction therefore yields $J_{\mathrm{Pure}} < J_{\mathrm{MOPD}} < J_{\mathrm{Gated}}$ after a single finite gradient step. The condition holds throughout $0.4 < t < 239/320$, where the teacher also has higher overall success than the initial student. This instantiates Corollary 1 with a gate determined solely by response correctness: it more often relaxes penalties on the stronger student route while retaining correction on the weaker route.

B Implementation and Training Details

B.1 Distillation Coefficients and Learner Objective

Let τ_d denote the specialist for domain d, π_θ the shared student, and π_{old} the rollout policy. For a generated response, $h_{it} = (x_i, y_{i,<t})$ is its prefix and $m_{it} \in \{0, 1\}$ masks the response positions used in the loss.

The signed distillation signal is computed on the student's top-K candidates. At each prefix, the cached support is $S_{it} = \mathrm{TopK}(\pi_{\mathrm{old}}(\cdot \mid h_{it}))$. At the learner parameters $\bar{\theta}$ used to construct advantages, define

$$q_{itk} = \mathrm{sg}\left[\frac{\pi_{\bar{\theta}}(k \mid h_{it})}{\sum_{j \in S_{it}} \pi_{\bar{\theta}}(j \mid h_{it})}\right], \qquad \Delta_{itk} = \mathrm{sg}\left[\log \frac{\tau_{d_i}(k \mid h_{it})}{\pi_{\bar{\theta}}(k \mid h_{it})}\right], \tag{14}$$

$$A_{itk} = q_{itk}\Delta_{itk}, \qquad A^+ = \max(A, 0), \qquad A^- = \min(A, 0). \tag{15}$$

The stop-gradient operator sg treats these quantities as fixed feedback. Only q is normalized on S_{it}; the probabilities in Δ retain their full-vocabulary normalization. A negative A_{itk} therefore means that the student assigns candidate k more probability than the teacher does, regardless of its task correctness.

Let $w_i \geq 0$ be the domain weight, held fixed during the update, and let $a_{itk} = w_i \widetilde{A}_{itk}$. We retain the baseline proximal policy optimization (PPO) surrogate with dual clipping (Schulman et al., 2017; Ye et al., 2020),

$$\begin{aligned} \mathcal{L}_{\mathrm{att}}(\theta) &= -\frac{1}{N} \sum_{i,t} m_{it} \sum_{k \in S_{it}} \psi\left(r_{itk}(\theta), a_{itk}\right), \\ r_{itk}(\theta) &= \frac{\pi_\theta(k \mid h_{it})}{\pi_{\mathrm{old}}(k \mid h_{it})}, \qquad N = \sum_{i,t} m_{it}. \end{aligned} \tag{16}$$

The clipping function ψ is specified below. Domain weights multiply the advantages, while N counts effective response tokens. Any baseline regularizers retain their original coefficients. Verified outcomes enter only through the transformation in Eq. 2.

Each update constructs or refreshes the baseline advantages, applies Eq. 2 once, multiplies by domain weights, and evaluates the loss. Refresh updates q and Δ using the current student and cached teacher scores; prefixes, candidate IDs, and rollout-policy probabilities remain fixed. Thus refresh updates the feedback, while new rollouts update the visited states.

Multi-teacher adaptation of RG-OPD. RG-OPD† uses the original trajectory-level gating rule of Akhondzadeh et al. (2026). For a prompt from domain d_i, the routed specialist τ_{d_i} takes the place of the single teacher in that rule. The gate retains or excludes the response's entire distillation loss: retained responses keep both positive and negative feedback, while rejected responses contribute no distillation loss. The dagger denotes the multi-teacher adaptation; the trajectory-gating rule is unchanged.

Attenuation schedule. With recorded global step u, start u_0, duration $H > 0$, and final retention $\lambda_{\min}$, we use

$$\lambda_u = 1 - (1 - \lambda_{\min}) \operatorname{clip}\left(\frac{u - u_0}{H}, 0, 1\right). \tag{17}$$

The baseline corresponds to $\lambda_u = 1$. A scheduled retention factor controls intervention strength independently of the outcome gate.

Table 5 distinguishes the scheduled base-capability training rule from the fixed settings used in diagnostics. The base-capability schedule decreases linearly from $\lambda_u = 1$ to $\lambda_u = 0$, and remains zero thereafter.

Table 5: Attenuation settings. λ is the retained fraction of negative feedback on verified-successful responses; $\beta = 1 - \lambda$ is the removed fraction.

Setting	Retention setting
Base-capability training	Linear schedule from $\lambda_u = 1$ to $\lambda_u = 0$
Controlled single update	Fixed $\lambda_u = 0.5$ ($\beta = 0.5$)
Figure 5 coefficient audit	Apply $\lambda_u = 0.1$ to the same response's feedback

Clipped surrogate. For lower and upper PPO tolerances ϵ_-, ϵ_+ and $c_{\text{dual}} > 1$, define

$$b(r, a) = \min\{ra, \operatorname{clip}(r, 1 - \epsilon_-, 1 + \epsilon_+)a\}, \tag{18}$$

$$\psi(r, a) = \begin{cases} \max\{b(r, a), c_{\text{dual}} a\}, & a < 0, \\ b(r, a), & a \geq 0. \end{cases} \tag{19}$$

The baseline and selective objectives share clipping, domain weights, and regularizer settings. At extreme ratios, the implementation's numerical log-ratio clamp must also be included in update diagnostics.

Refresh and reduction. Teacher scores, candidate IDs, old-policy probabilities, and response gates are cached for each rollout. Advantage refresh recomputes the student-dependent quantities before the single attenuation step. Gates remain aligned with their responses through shuffling and dynamic packing. For the token-mean objective, N in Eq. 16 counts effective policy tokens across the entire optimizer minibatch and all ranks. With W ranks and gradient averaging, a local microbatch mean over n tokens receives multiplier Wn/N. Domain weights remain in the numerator. Policy format masks apply to both the numerator and denominator, while entropy and reference-KL terms retain their own response masks.

The local analysis uses the response and token weights induced by the chosen loss reduction. Changing the reduction changes these weights and the corresponding release directions.

Verifier semantics. A valid result requires a supported task-specific check. For mathematics, we extract the final answer from a boxed expression, an explicit answer statement, or a standalone number, and compare it with the reference using normalized equality or exact numerical equivalence. For code, we extract the generated program and execute it in SandboxFusion against the supplied

test cases; success requires all tests to pass within the execution limits. For instruction following, we use the `verifiable_instructions` rule-based checkers on the response after removing its thinking block. Each training example supplies constraint identifiers and parameters, such as keyword, length, case, and formatting requirements. The gate opens only when at least one constraint is present, every constraint passes, and no checker raises an error. It uses the all-constraints-passed indicator rather than a partial-credit or format reward; no LLM judge is used for this IF gate. An unavailable or failed verification process leaves the gate closed. For a batch of n responses, coverage $n^{-1}\sum_i v_i$ and activation $n^{-1}\sum_i g_i$ distinguish available judgments from verified success.

B.2 Agent teachers and task-outcome verification

Teacher construction and reference routing. The student and all three teachers start from the same mixed-domain SFT checkpoint of Qwen-3.6-27B. Three teacher branches are then specialized through RL. The mixed-SFT student initializes the Open-MOPD and SA-MOPD distillation runs; the mixed-SFT baseline is evaluated before distillation. Table 2 uses Teacher 1 as the BFCL-v4 reference, Teacher 2 as the τ^2-Bench reference, and Teacher 3 as the reference for both Claw-Eval and SkillsBench. Route Teacher combines these designated specialist scores; it is a multi-model reference. Mixed SFT, Open-MOPD, and SA-MOPD are evaluated as individual students.

Task-level gate. For agent tasks, the gate uses task completion across the full interaction. The availability and success indicators retain the definition $g_i = v_i c_i$ from Section 3. A valid success judgment sets $v_i = c_i = 1$; partial progress that does not satisfy the task's success criterion leaves $c_i = 0$. An unavailable judgment, including an environment or verifier failure that prevents task assessment, sets $v_i = 0$. Thus an unsuccessful assessed task and an unassessable interaction both leave the attenuation gate closed, while remaining distinct verification states. Recovery from an intermediate tool error is assessed by the final task outcome. This gate determines feedback attenuation; benchmark scores retain their task-specific evaluation metrics.

C Score aggregation and training variability

Domain scores, Total, and Recovery follow Section 5. Scores are displayed to two decimal places and Recovery to one; reported main-text differences use the displayed scores. Base-capability Recovery uses the reported totals, whereas agent Recovery uses the four-benchmark means before total-score rounding. These conventions reproduce 96.4% and 109.7% for SA-MOPD, respectively.

C.1 Training-seed variability of SA-MOPD

To quantify variability across independent training runs, we train SA-MOPD with seeds 42, 43, and 44 and evaluate each resulting model. Table 6 reports the sample variance and standard deviation of each benchmark score. For scores x_s expressed in percentage points, we use $s^2 = \frac{1}{3-1}\sum_{s\in\{42,43,44\}}(x_s - \bar{x})^2$ and report $s = \sqrt{s^2}$. Instruction-following scores use strict prompt-level evaluation. For LiveCodeBench, each run averages correctness over ten sampled responses per problem and then over problems; this estimates pass@1, not pass@10.

Table 6: Training-seed variability of SA-MOPD across independent runs with seeds 42, 43, and 44. Sample variances use the $n-1$ denominator ($n = 3$). IF scores are strict prompt-level; LCB scores estimate pass@1 using ten sampled responses per problem.

Benchmark	Sample variance (pp^2)	Std. dev. (pp)
AIME24	0.303819	0.5512
AIME25	0.398763	0.6315
IFEval	0.649171	0.8057
IFBench	1.925926	1.3878
LCB v5	0.172111	0.4149
LCB v6	0.088163	0.2969

Across the three training runs, the standard deviations are 0.5512 and 0.6315 pp on AIME24 and AIME25, and 0.4149 and 0.2969 pp on LCB v5 and v6. IFEval and IFBench have standard deviations of 0.8057 and 1.3878 pp, respectively. Together with the three-run means in Table 1, these statistics characterize both the average performance and the training-run variability of SA-MOPD.

D ISOLATING THE VALUE OF FEEDBACK SELECTION

This study asks whether verified-success selection contributes beyond reducing the amount of negative feedback. We compare alternative feedback rules through a single controlled update from a common MOPD-trained student. Holding the model, optimizer state, source responses, and teacher scores fixed isolates the intervention before training trajectories diverge.

Separating attenuation amount, domain allocation, and selection. MOPD retains the original feedback. Pure halves negative feedback on every response, testing unconditional attenuation. Global matched distributes SA-MOPD's total attenuation budget uniformly across responses. Domain matched additionally preserves that budget within each source domain, controlling for its allocation across teachers. SA-MOPD instead halves negative feedback only on verified-successful responses. Every branch preserves positive feedback; SA-MOPD also preserves the original signal on failed or unverifiable responses.

To specify the matched budgets, let M_i be response i's active negative coefficient mass. The mass-weighted gate activations are

$$\rho_e = \frac{\sum_{i:d_i=e} g_i M_i}{\sum_{i:d_i=e} M_i}, \qquad \rho = \frac{\sum_i g_i M_i}{\sum_i M_i}. \tag{20}$$

With attenuation strength $\beta = 1 - \lambda_u = 0.5$, Global matched retains $1 - \beta\rho$ of each negative coefficient, and Domain matched retains $1 - \beta\rho_e$ within domain e. Thus comparison with Global matched isolates selection at a fixed total budget; comparison with Domain matched further isolates selection within domains, following Proposition 2.

Intervention and evaluation. Each branch receives the same frozen set of 1,024 responses spanning Math, IF, and Code. SA-MOPD and both matched controls remove 14.66% of the total negative coefficient mass, versus 50% for Pure. The domain-specific mass activations are 0.2899, 0.4495, and 0.1570, respectively. We measure the resulting optimizer displacements separately in Table 8.

All five updated models and the original student are evaluated on the same 384 prompts, with 128 per domain and eight fresh responses per prompt (18,432 responses in total). Scores average binary verifier outcomes to estimate single-response success. Evaluation prompts are disjoint from the update data and come from the existing mechanism-diagnostic set constructed from Open-MOPD-Data (BytedTsinghua-SIA, 2026). Its Math and Code records identify DAPO-Math-17k (BytedTsinghua-SIA, 2025) and TACO (Li et al., 2023) as their respective sources; IF records use the Nemotron instruction-following subset distributed in the same data release. Verification infrastructure errors are kept separate from task failures.

Score provenance. Table 4 reports the scores from the updated evaluation summary. Math and Code aggregates are reproduced by the archived prompt-level records. Revised IF scores are available only as aggregates; the prompt-level gain decomposition therefore covers Math and Code.

Reproduction settings. The shared initialization is an intermediate ordinary-MOPD training checkpoint designated C80 in the experiment records. We restore FP32 parameters and Adam moments at step 320 and apply one AdamW update with learning rate 1.5×10^{-6} and gradient-norm clipping threshold 0.5; all branch norms remain below this threshold. The frozen data contain 408 Math, 208 IF, and 408 Code responses across four batches, with teacher scores on the cached student top-16 support. Gradients use BF16 differentiation with FP32 accumulation, equal domain token-mean weighting, and averaging across source batches. Evaluation uses temperature 1, top-$p = 1$, unrestricted token sampling, and a 16,384-token generation limit.

E QUALITATIVE EXAMPLES

Sources and selection. These retrospectively selected cases examine feedback on verified code and constraint handling in generated responses. Example 1 scores teacher probabilities at a student prefix and tests a single-token substitution with the continuation fixed. Example 2 and Figure 6 compare complete responses sampled from the controlled update branches, with all-generation counts reported. Probabilities in Example 1 use full-vocabulary normalization and are shown as *student / Code teacher*; pass counts refer to the original program. The substitution tests compatibility with that program's continuation.

Sources and scoring for Figure 5. The Math panel uses AIME 2024 sample 72, with archived RL-teacher completion 55 and MOPD-student completion 54. Both final derivations yield 540: the teacher uses a polar representation and the student uses Cartesian coordinates and the Cauchy–Schwarz inequality. The latter's bound is attained at $(x, y) = (12/5, -16/5)$. We recompute full-vocabulary log probabilities in FP32 using the RL Math teacher and the student checkpoint that generated the response, retaining its complete preceding reasoning. The four audited token positions correspond to the Cartesian parameterization, the reciprocal's real part, addition of the two real parts, and the simplified real part. The Code panel retains the intermediate-checkpoint tree-distance example and its cached teacher scores; its four tokens occur in distance initialization, edge traversal, the scalar update, and result initialization. For both panels, q is the student probability normalized on its top-16 support, and the SA-MOPD coefficients apply Eq. 2 with $\lambda_u = 0.1$ to the same verified response. The plotted values precede common loss normalization. The Math rollout comparison shows distinct correct derivations; the prefix audit measures local feedback on the student's own derivation.

Example 1: Reading a light's initial state

Task (summary): Find the maximum number of simultaneously lit lamps. Lamp i toggles every a_i seconds starting at b_i; string `s` records the initial on/off states.

Student context (excerpt):

```
s = input().strip()
...
num_toggles = numerator // a_list[i] + 1
```

Observed student continuation:

```
state = 1 if (s[i] == '1') ^ (num_toggles % 2) else 0
```

Original verification: 132/132 tests passed.

Teacher scoring: The prefix ends in `state = 1 if (`.

`s`: $72.6\%/37.6\%$; `int`: $26.7\%/62.0\%$. $\Delta(\texttt{s}) \simeq -0.658$. The teacher's top-1 token is `int`.

Analyst edit (continuation fixed): Replace only `s[i]` with `int[i]` in the displayed condition.

Edited-program verifier output (first test):

```
TypeError:  type 'int' is not subscriptable
```

With verified success, Eq. 2 attenuates the negative feedback on the observed `s` token while preserving positive coefficients. The teacher-preferred `int` token is incompatible with the fixed continuation, causing the edited program to fail its first test.

Example 2: Double-counting an exceptional case (Math)

Task: Count ordered positive integer pairs (a, b) with $1 \leq a, b \leq 100$, prime $a + b$, and integer $(ab + 1)/(a + b)$.

Reference reasoning: Let $p = a + b$. Divisibility implies $a^2 \equiv 1 \pmod{p}$, hence the pairs are $(1, p-1)$ and $(p-1, 1)$ for primes $p \leq 101$. There are 26 such primes; the two pairs coincide at $p = 2$. Thus the answer is $2 \cdot 26 - 1 = 51$, also confirmed by exhaustive enumeration of the 10,000 allowed pairs.

Before the update (final-response excerpt):

"The prime $s = 2$ contributes one pair $(1, 1)$."

"25 primes $\times$ 2 pairs per prime + 1 pair for $s = 2 = 50 + 1 = 51$."

Verifier: Correct.

After the MOPD update (final-response excerpt):

"Each prime p contributes exactly two ordered pairs: $(1, p-1)$ and $(p-1, 1)$."

"26 primes $\times$ $2 = 52$."

Verifier: Incorrect; $(1, 1)$ is counted twice.

After the SA-MOPD update (final-response excerpt):

"However, for $S = 2$, the pair $(1, 1)$ is the only valid pair, while for each prime S from 3 to 101, there are two distinct pairs."

"$1 + 25 \times 2 = 1 + 50 = 51$."

Verifier: Correct.

All eight generations for this prompt: Before the update, 3/8 correct; MOPD, 1/8; SA-MOPD, 4/8. Respectively 5/8, 6/8, and 4/8 responses reach the generation limit. The displayed responses all terminate normally and are excerpts of unmodified generations.

Errors in complete Math and Code responses. Figure 6 presents two further cases from the controlled single-update evaluation. The Math response adds a case that violates its own parity condition; the Code response uses an input layout absent from the task despite stating the correct waiting-time formula. Original and SA-MOPD responses shown for comparison retain the relevant constraint. All eight responses from each of these three models terminate normally on both prompts.

Final-answer verification and intermediate arithmetic. A separate archived teacher response to the Math problem in Figure 5 obtains the correct expression $324 \cos\theta - 432 \sin\theta$, but then writes

$$\sqrt{324^2 + 432^2} = \sqrt{95076 + 186496} = \sqrt{281072} = 540.$$

The final answer is correct, while the intermediate arithmetic is not: $324^2 + 432^2 = 104976 + 186624 = 291600 = 540^2$. This example distinguishes the final-answer property checked by the verifier from the validity of every intermediate calculation.

F Feedback allocation and optimizer response

Negative feedback across response lengths. We audit the frozen source responses used in the controlled ablation. Figure 7 separates cumulative negative-feedback coefficient mass by verifier outcome. Each curve is normalized by the total negative mass in its domain, so its endpoint gives the share associated with that outcome. All 377 verified-success responses carry nonzero negative mass, showing that local teacher disagreement also occurs on successful student trajectories.

Feedback density within length groups. After excluding length-truncated responses, we divide each domain into four pooled response-length quartiles. Figure 8 shows negative-feedback density for verified-successful and other responses within these groups. Negative feedback remains present on successful responses at different lengths; the median and interquartile range characterize its magnitude conditional on length and termination.

Attenuation after excluding truncated responses. Table 7 recomputes relative attenuation reduction after excluding responses terminated at the token limit. The reduction remains 42.27%, 53.41%, and 79.27% in Math, IF, and Code. The larger change in Math indicates greater dependence on the truncation distribution. Code retains a substantial reduction without truncated responses.

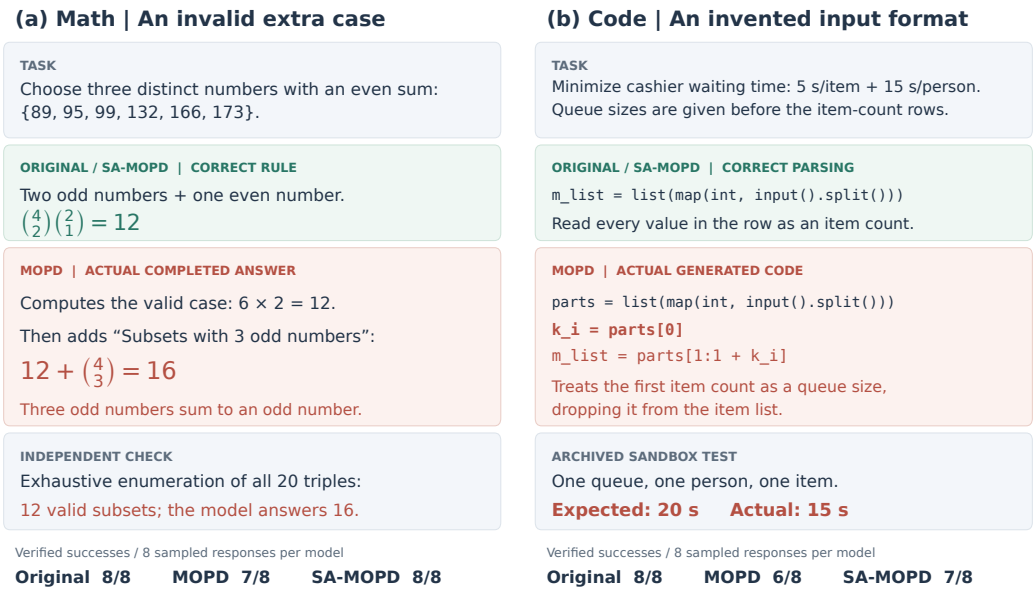

Figure 6: **Concrete errors after a controlled MOPD update.** (a) The Math response counts three-odd-number subsets even though their sum is odd. Enumeration confirms 12 valid subsets, rather than 16. (b) The Code response mistakes an item count for a queue size; the archived sandbox test returns 15 instead of 20. Code excerpts are verbatim; the Math reasoning is condensed. Displayed responses use sample indices 0/6/0 (Math) and 0/2/0 (Code) for Original/MOPD/SA-MOPD, respectively. Counts cover all eight generations per model and prompt.

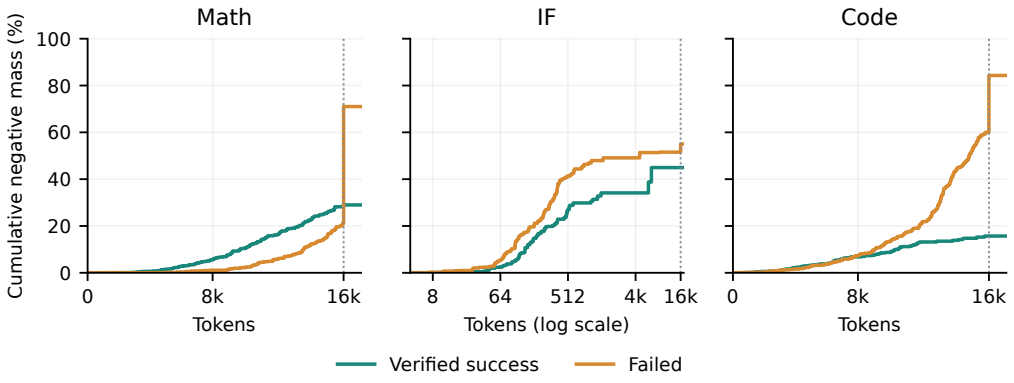

Figure 7: **Length-resolved feedback allocation.** Empirical cumulative negative coefficient mass on successful and failed source responses. The dotted line marks the generation limit. IF uses a logarithmic length axis; Math and Code use linear axes. The curves summarize the frozen source-batch distribution.

Response length and verifier selection. Within the frozen source batch, we rank responses by increasing length, breaking ties by response ID, and measure success among the shortest selected fraction (Figure 9). Selecting as many responses as the verifier gate yields gate overlaps of 80.0%, 42.0%, and 53.5% in Math, IF, and Code. Shortness predicts success strongly in Math (AUC 0.911), moderately in Code (0.797), and almost not at all in IF (0.511). These statistics describe the domain-dependent association between response length and verifier selection in this batch.

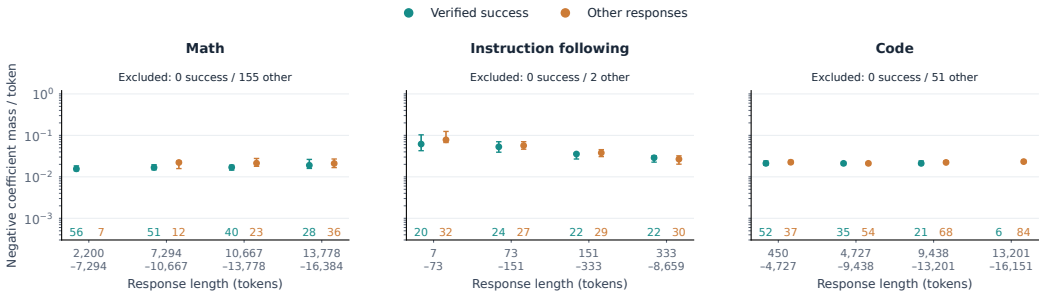

Figure 8: **Negative feedback within response-length groups.** Nontruncated responses from the frozen source batch. Points and whiskers show median negative-coefficient mass per token and interquartile range. Colored counts give group sizes; cells with fewer than eight responses are omitted. The other group includes responses without verified success.

Table 7: Source-allocation diagnostics. Reduction is the relative decrease in removed negative-feedback coefficient mass under SA-MOPD versus Pure. Nontruncated recomputes this ratio after excluding responses terminated at the token limit. Length AUC predicts verified success from shorter responses, with half credit for ties.

Domain	Truncated	Reduction (%)	Nontruncated (%)	Length AUC
Math	155	71.01	42.27	0.911
IF	2	55.05	53.41	0.511
Code	51	84.30	79.27	0.797

Optimizer response at matched coefficient mass. Table 8 reports actual optimizer responses. Global matched, Domain matched, and SA-MOPD remove the same aggregate mass, but yield different gradient norms and parameter displacements relative to MOPD. Clipping is inactive in every branch.

Table 8: All updates use identical initial weights and Adam state. Removed mass is a percentage of original negative coefficient mass. Parameter distance is the FP32 Euclidean distance to the MOPD-updated parameters; clipping was inactive for every branch.

Update	Removed mass (%)	$\|G\|_2$	$\|\theta_b - \theta_{\mathrm{MOPD}}\|_2$
MOPD	0.0000	0.345200	0.00000000
Pure	50.0000	0.300456	0.00121780
Global matched	14.6551	0.329162	0.00035498
Domain matched	14.6551	0.319944	0.00051522
SA-MOPD	14.6551	0.326337	0.00064618

Performance by prior student capability. We group evaluation prompts using eight generations from the shared pre-update student: zero successes, one to seven successes, or eight successes. Group membership is fixed before inspecting the updated models' outcomes. Each method is then evaluated using eight fresh generations on the same prompts. Math group sizes are 43, 64, and 21; Code group sizes are 69, 41, and 18. We bootstrap prompts within each group 100,000 times to obtain pointwise 95% intervals for its mean score.

On Code prompts with eight pre-update successes, SA-MOPD reaches 98.61%, compared with 94.44% for MOPD and 91.67% for Domain matched. For one to seven successes, the corresponding scores are 51.52%, 46.95%, and 45.43%; for zero successes, all three score 1.27%. Figure 10 shows both domains under the same protocol. On Math prompts with one to seven pre-update successes, SA-MOPD reaches 53.91%, versus 51.56% for MOPD and 50.39% for Domain matched. On the

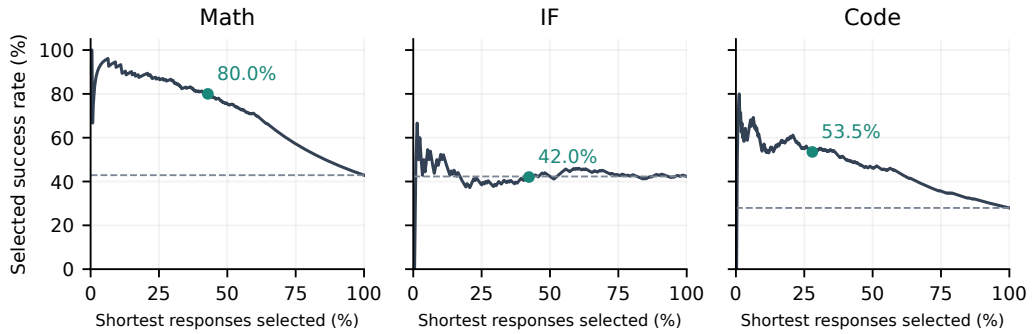

Figure 9: **Response length and verifier selection.** Success rate among the shortest selected responses in the frozen source batch. Green markers select the same response count as the verifier gate; dashed lines show each domain's source success rate. At the markers, precision also equals overlap with the verified-success set because the two selections have equal size. This comparison measures selection overlap; no length-based training update is applied.

eight-of-eight group, these scores are 97.02%, 95.83%, and 96.43%, respectively. On the zero-of-eight group, SA-MOPD scores 2.62%, between MOPD's 3.49% and Domain matched's 2.33%. In this exploratory stratification, SA-MOPD has higher observed scores in groups with some pre-update success in both domains, with a larger separation in Code.



Figure 10: **Task performance by prior student capability.** (a) Math and (b) Code post-update success, using all 128 prompts per domain, grouped by success counts in eight pre-update generations. Group sizes are shown below the axes. Bars average eight fresh generations per prompt and method. Whiskers show pointwise 95% prompt-bootstrap intervals for each group mean, conditional on the pre-update groups.

G Task geometry and verifier-based diagnostics

This section proves the distribution-level result, develops counterexamples, and specifies verifier-based measurements of local task influence.

G.1 Outcome decomposition and proof of Proposition 1

For a fixed prompt, partition a finite response space into $\mathcal{Y}_+ = \{y : r(y) = 1\}$ and $\mathcal{Y}_- = \{y : r(y) = 0\}$. Assume strictly positive p, τ and $0 < v, u < 1$. Define $\ell = \log(\tau/p)$, $m_+ = \mathbb{E}_p[\ell \mid r = 1]$, and $m_- = \mathbb{E}_p[\ell \mid r = 0]$. On $\mathcal{Y}_+$, $p(y) = vp_+(y)$ and $\tau(y) = u\tau_+(y)$, with the analogous factorization on $\mathcal{Y}_-$. Substitution into KL gives Eq. 5 and

$$m_+ = \log(u/v) - K_+, \qquad m_- = \log((1-u)/(1-v)) - K_-. \tag{21}$$

The Fisher inner product on probability tangents is $\langle z, z' \rangle_p = \sum_y z(y)z'(y)/p(y)$, with $\sum_y z(y) = 0$. Projecting the negative KL derivative onto this tangent space gives

$$\dot{p}_0(y) = p(y)[\ell(y) - \mathbb{E}_p \ell], \qquad \ell(y) = \log(\tau(y)/p(y)). \tag{22}$$

Thus $\dot{v} = \sum_y r(y)\dot{p}_0(y) = \mathrm{Cov}_p(r, \ell)$. Conditioning on r yields $v(1-v)(m_+ - m_-)$; inserting Eq. 21 proves Proposition 1.

A stronger-teacher counterexample. Take

$$p_0 = (0.60, 0.10, 0.30), \quad \tau = (0.01, 0.89, 0.10), \quad r = (1, 1, 0). \tag{23}$$

The student succeeds with probability $v = 0.7$ and the teacher with $u = 0.9$. Nevertheless, $K_+ = 3.448460$, $K_- = 0$, and $\dot{v} = 0.21[\log(27/7) - 3.448460] = -0.440692$. The exact solution of Eq. 22 is

$$p_t(y) = \frac{p_0(y)^{\exp(-t)} \tau(y)^{1-\exp(-t)}}{\sum_z p_0(z)^{\exp(-t)} \tau(z)^{1-\exp(-t)}}.$$

Differentiating its log probabilities recovers the centered log-ratio flow. Figure 2a plots this solution: success initially falls to approximately 0.630, before converging to the teacher's 0.9. This analytical example instantiates the failure condition in Proposition 1.

An orthogonal decomposition. Put $a = m_+ - m_-$ and $m_r(y) = \mathbb{E}_p[\ell \mid r = r(y)]$. Then

$$\dot{p}_0 = \underbrace{ap(r - v)}_{z_{\mathrm{out}}} + \underbrace{p(\ell - m_r)}_{z_{\mathrm{within}}}, \qquad \langle z_{\mathrm{out}}, z_{\mathrm{within}} \rangle_p = 0. \tag{24}$$

The conditional mean of $\ell - m_r$ vanishes within each outcome class. Hence $\sum_y r(y) z_{\mathrm{within}}(y) = 0$, and orthogonality follows. The first component changes success mass; the second changes relative probabilities within outcome classes. The latter is neutral only to the current success rate at first order, not to subsequent learning.

G.2 Imitation-gradient alignment and task benefit

Let F_e denote imitation-loss gradients and h_d task gradients. An update $\theta' = \theta - \eta q$ has task effect $-\eta h_d^\top q + O(\eta^2)$. Products $F_e^\top F_f$ do not determine this sign. For example, the scalar losses $L_e(\theta) = (\theta + 1)^2/2$ all have gradient 1 at zero, but their common descent direction reduces $J(\theta) = \mathrm{sigmoid}(\theta)$.

This example motivates measuring alignment with the task gradient h_d when evaluating a change to distillation feedback.

G.3 Cross-domain task influence and verifier-based estimation

Because parameters are shared, releases from source domain e can affect any target domain d. With $H_e = \sum_{i:d_i=e} g_i U_i$,

$$\Delta_d^g = \eta\beta \sum_e T_{d \leftarrow e} + O(\eta^2), \qquad T_{d \leftarrow e} = h_d^\top B H_e. \tag{25}$$

The score-function identity gives $h_d = \mathbb{E}[r_d \nabla_\theta \log \pi_\theta(y \mid x)]$ (Williams, 1992). It needs outcomes and student log probabilities, not a differentiable verifier or learned critic. For M_d independent calibration prompts $x_a \sim \mathcal{D}_d$ and $n \geq 2$ independent responses per prompt, define $s_{aj} = \nabla_\theta \log \pi_\theta(y_{aj} \mid x_a)$ and $b_{a,-j} = (n-1)^{-1} \sum_{k \neq j} r_{ak}$.

Proposition 3 (Verifier-based influence estimation) *Assume responses are sampled from the same policy whose score is s_{aj}, labels define J_d, and differentiation and expectation commute. Then the leave-one-out estimator*

$$\widehat{h}_d = \frac{1}{M_d n} \sum_{a=1}^{M_d} \sum_{j=1}^{n} (r_{aj} - b_{a,-j}) s_{aj}, \qquad \widehat{T}_{d \leftarrow e} = \widehat{h}_d^\top B H_e \tag{26}$$

obeys $\mathbb{E}\widehat{h}_d = h_d$ and $\mathbb{E}\widehat{T}_{d \leftarrow e} = T_{d \leftarrow e}$ when BH_e is fixed independently of the calibration responses.

The estimator uses standard score-function and leave-one-out techniques (Williams, 1992; Ahmadian et al., 2024) to diagnose task influence. Appendix G.4 specifies sampling and uncertainty estimation.

Proof and interpretation. Assume a parameter-independent prompt distribution and response support, a normalized policy, and no explicit parameter dependence in the task score. With differentiation under the expectation justified,

$$h_d = \mathbb{E}_{x\sim\mathcal{D}_d} \sum_y r_d(x,y)\nabla_\theta \pi_\theta(y \mid x) = \mathbb{E}[r_d(x,y)s(x,y)], \qquad \mathbb{E}[s(x,y) \mid x] = 0. \tag{27}$$

The identity also holds for bounded-length trajectories when stopping and scoring rules are included in the objective. The verifier need not be differentiated (Williams, 1992). Given a prompt, $b_{a,-j}$ uses responses independent of y_{aj}, so $\mathbb{E}[b_{a,-j}s_{aj}] = \mathbb{E}[b_{a,-j}]\mathbb{E}[s_{aj}] = 0$. Linearity proves $\mathbb{E}\widehat{h}_d = h_d$. Conditioning also on an independently chosen BH_e proves Proposition 3's influence claim. Using the same responses to construct both the release direction and its task-gradient estimate removes this independence; splitting or cross-fitting can restore it.

The leave-one-out baseline avoids a value network. Same-outcome groups contribute zero and remain in the sample average. In the optimizer audit below, task projections are computed by contracting the estimated gradient with the corresponding parameter displacement.

G.4 Diagnostic sampling and uncertainty

The estimator requires calibration prompts independent of the release directions and responses sampled from the policy defining J_d. The optimizer audit below uses fresh full-support samples at temperature 1, with log probabilities evaluated at the same parameters and an explicit stopping horizon.

The verifiable prompt subset is fixed before observing outcomes. Verification errors and unknown judgments are recorded separately from task failures. For influence uncertainty, prompt-level bootstrap resampling retains all responses within each prompt, with simultaneous intervals for jointly tested domain influences. The resulting intervals condition on the checkpoint and source update.

G.5 Fixed-preconditioner approximation of an actual Adam update

We test how closely the common-B model in Proposition 2 approximates an implemented Adam step. At an intermediate checkpoint, we freeze the source rollouts and cached top-16 gradients, and recompute each branch's Adam moments from the same initial optimizer state, using $\beta = 0.5$ and learning rate 1.5×10^{-6}. The common diagonal B uses the MOPD-updated second moment, frozen across branches, together with the first-moment response factor and bias corrections. The five gradient norms range from 0.3005 to 0.3452, below the clipping threshold 0.5 (Table 8).

For each of 768 independent calibration prompts (256 per domain), we draw four full-support FP32 responses at temperature one and estimate the prompt-level task gradient $\widehat{h}_x$ using Eq. 26. Using this same gradient estimate for branches b, c, we compare

$$z_x^B = \eta \widehat{h}_x^\top B(G_b - G_c), \qquad z_x^{\mathrm{Adam}} = \widehat{h}_x^\top (\Delta\theta_b^{\mathrm{Adam}} - \Delta\theta_c^{\mathrm{Adam}}).$$

The latter uses measured parameter displacements with both Adam moments recomputed separately for each branch. All prompts are retained; 532 have identical labels across their four responses and hence zero leave-one-out estimates. Correlation and sign agreement are computed on the remaining 236 mixed-label prompts.

Table 9: **Local Adam approximation audit.** Fixed-B and actual-Adam task projections for SA-MOPD minus Domain matched at a fixed checkpoint and source update. Each domain includes 256 calibration prompts; the table reports the mixed-label subset. Relative error is $\|z^B - z^{\mathrm{Adam}}\|_2 / \|z^{\mathrm{Adam}}\|_2$ and is unchanged when the zero-estimate prompts are included.

Domain	Mixed prompts	Pearson r	Relative L_2 error	Sign agreement
Math	101	0.99356	15.15%	96.04%
IF	71	0.99999	0.39%	100.00%
Code	64	0.99731	13.86%	100.00%

For SA-MOPD versus Domain matched, correlations range from 0.9936 to 0.99999, with sign agreement of 96.04%–100%. Across all four contrasts—Pure versus MOPD and SA-MOPD versus MOPD, Global matched, and Domain matched—the 12 domain-wise correlations exceed 0.992. The audit shows high correlation and sign agreement in the local task projections at this checkpoint. Both projections use the same $\widehat{h}_x$, so the comparison isolates the optimizer approximation.