BEYOND DOMAIN-ROUTED SUPERVISION: CROSS-TEACHER GUARDING FOR MULTI-TEACHER ON-POLICY DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-teacher on-policy distillation (MOPD) extends on-policy distillation (OPD) by routing domain-specific data to specialized teachers, integrating their capabilities through dense token-level supervision on student-generated trajectories. However, domain routing only distinguishes the supervision sources for different domains; updates from all domains still act on shared parameters. When teachers differ substantially in response style and domain expertise, supervision that is effective in isolation may still lead to persistent gradient conflicts and capability degradation during joint optimization. To address this challenge, we propose C**ro**ss-Teacher Gu**ard**ing for MOPD (ROAD MOPD), which introduces non-routed teachers as auxiliary cross-domain critics on the same student trajectories and coordinates multi-teacher supervision through token-level credit assignment and gradient updates. Specifically, cross-teacher-guided credit assignment uses teacher agreement to strengthen token-level corrections jointly supported by the teachers and attenuate conflicting signals; preventive multi-teacher gradient projection uses auxiliary gradients from the same trajectories to constrain potentially conflicting updates and adaptively controls projection strength to retain target-domain learning signals. In experiments, ROAD MOPD mitigates degradation in the disadvantaged domain, gains over 5 percentage points under high conflict, and improves overall performance in low-conflict and three-domain settings.

1 INTRODUCTION

Reinforcement learning (RL) has emerged as an important paradigm for post-training large language models (OpenAI, 2023; DeepSeek-AI, 2025). However, training with outcome-level rewards lacks direct feedback on intermediate reasoning steps, making it difficult to attribute a response's success or failure to specific generation decisions. Optimization therefore remains largely dependent on policy exploration, posing challenges for credit assignment and stable optimization. On-policy distillation (OPD) (Gu et al., 2024; Agarwal et al., 2024; Ko et al., 2025) offers a promising alternative for addressing this limitation: it leverages a strong teacher to provide dense token-level supervision on trajectories generated by the student's current policy, combining on-policy sampling aligned with the student's current generation distribution with fine-grained teacher feedback.

Prior work has improved OPD capability transfer through better distillation objectives and optimization strategies (Li et al., 2026a; Lin et al., 2026; Yang et al., 2026a; Zhang et al., 2026). A natural extension is to consolidate specialized teachers' complementary capabilities in a single student across domains. Multi-teacher on-policy distillation (MOPD) (Ma et al., 2026; DeepSeek-AI et al., 2026) pursues this goal, typically through direct domain routing: a sample from domain d receives supervision only from its routed teacher T_d, while updates from all domains act on the same student model. This design exploits each teacher's domain expertise but does not explicitly coordinate updates from different domains to shared parameters. Consequently, supervision that is effective in single-domain training may induce interfering update directions during joint optimization, limiting multi-domain capability integration.

Prior MOPD studies have demonstrated the potential of joint learning on domain combinations such as math and code (Yang et al., 2026b; Hou et al., 2026), but these gains may not generalize to

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.

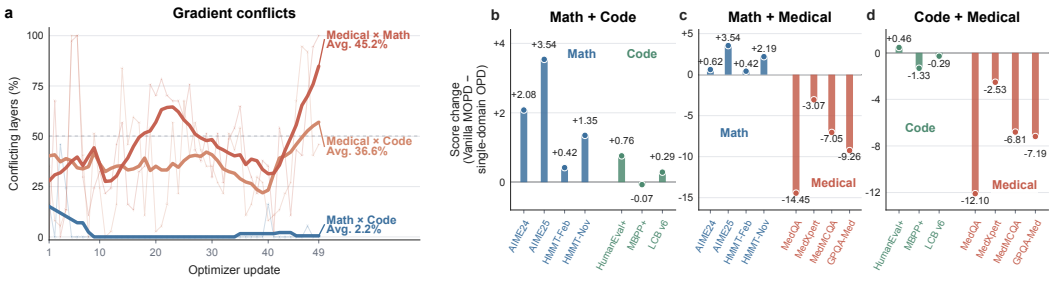

Figure 1: **Update conflicts reveal a failure mode of domain-routed MOPD.** (a) Fraction of conflicting layers during training; labels show training averages. (b–d) Per-benchmark score changes of Vanilla MOPD relative to single-domain OPD. Math–Code remains compatible, whereas both Medical pairings consistently degrade Medical performance.

other combinations. As shown in Figure 1, joint training on math and code generally preserves or improves upon the corresponding single-domain OPD performance, with relatively few gradient conflicts. In contrast, jointly training medicine with either math or code consistently yields a high proportion of layers with conflicting gradients, and this proportion increases further in later stages of training. In both pairings, medical performance falls below the single-domain OPD baseline, showing that the joint-learning benefits observed for math and code do not extend to these combinations. This contrast highlights compatibility between domain updates as an important factor in MOPD and suggests that allocating supervision through domain routing alone is insufficient to coordinate capability transfer through shared parameters.

To address this limitation, we reconsider teachers' roles in MOPD: can non-routed teachers help anticipate and mitigate update interference while preserving the routed teacher's domain-specific supervision? We retain this supervision and introduce non-routed teachers as cross-domain critics on the same student-generated trajectories. Their feedback on the same behavior provides local cues about potentially incompatible learning signals. We aim to reduce joint-optimization interference through this feedback while retaining target-domain learning signals.

Specifically, we propose C**ro**ss-Teacher Gu**ard**ing for MOPD (ROAD MOPD), to exploit cross-teacher feedback at both the token and gradient levels. At the token level, ROAD MOPD compares evidence from the routed and auxiliary teachers relative to a shared frozen reference model, strengthening supervision when their implied correction directions agree and downweighting the corresponding tokens when they conflict. At the gradient level, auxiliary supervision on the same trajectory defines a cross-domain reference direction, and preventive gradient projection suppresses components of the target-domain update that conflict with this direction. In high-conflict settings, ROAD MOPD improves Medical accuracy by 5.6–8.8 percentage points over Vanilla MOPD. It also improves overall performance in low-conflict and three-domain settings.

Our contributions are as follows:

- We identify a failure mode of domain-routed MOPD: when domain updates are poorly compatible, separately assigned supervision can induce persistent gradient conflicts in shared parameters. These conflicts accompany asymmetric performance degradation, highlighting the importance of update compatibility for joint learning.
- We propose ROAD MOPD, using non-routed teachers as cross-domain critics on the same student trajectories. Cross-teacher agreement guides token weighting, while preventive gradient projection adapts its strength under a norm-retention constraint to protect target-domain learning signals.
- Across domain combinations and teacher–student configurations, ROAD MOPD improves adversely affected domains under high conflict, preserves low-conflict gains, and generalizes to three domains, outperforming standard MOPD and gradient-conflict correction baselines in overall performance.

2 Related Work

2.1 On-Policy Distillation

Classical knowledge distillation trains a student to match a teacher's predictive distribution (Hinton et al., 2015). Sequence-level distillation applies this idea to autoregressive models using teacher-generated responses, but creates a mismatch between training and inference prefixes (Kim & Rush, 2016). On-policy distillation (OPD) instead queries the teacher on student-generated trajectories, using reverse-KL, general on-policy, or contrastive objectives (Gu et al., 2024; Agarwal et al., 2024; Ko et al., 2025). Subsequent work analyzes sampled-token estimators (Shao et al., 2026); makes supervision more selective through token filtering, reweighting, verifiable rewards, or reasoning progress (Li et al., 2026a; Lin et al., 2026; Yang et al., 2026a; Zhang et al., 2026); and extends OPD across tokenizers or intermediate generation steps (He et al., 2026; Sun et al., 2026a). Reference-based extrapolation further scales teacher corrections against a frozen student initialization (Yang et al., 2026b). Despite these advances, OPD can generalize unevenly across capabilities (Li et al., 2026b). These methods refine how evidence from one assigned teacher is estimated, selected, or scaled; they do not test its compatibility with capabilities represented by other teachers. We instead study how domain-specialized teachers can jointly assess the same student-generated prefixes.

2.2 Multi-Teacher On-Policy Distillation

Multi-teacher on-policy distillation (MOPD) transfers capabilities from domain specialists to a shared student, typically routing each rollout to its corresponding teacher for dense supervision (Ma et al., 2026). Domain routing is also used in large-scale and staged multi-domain post-training systems (Xiaomi LLM-Core Team et al., 2026; Yang et al., 2026c). Related methods adjust domain proportions during pretraining or multi-domain reinforcement learning (Xie et al., 2023; Liang et al., 2025), but do not coordinate teacher feedback on the same trajectory. Among recent MOPD methods, Open-MOPD balances token shares and teacher–student gaps while refreshing stale rewards (Gao et al., 2026), and D^3-MOPD schedules domains by their remaining distillation gaps and learning progress (Sun et al., 2026b). Other methods separate capability-recovery and domain-preservation updates (Chen et al., 2026), or calibrate confidence across heterogeneous teachers (Yin et al., 2026). Gradient-conflict methods such as PCGrad and conflict-averse gradient descent instead modify already-computed task gradients (Yu et al., 2020; Liu et al., 2021). None uses non-routed teachers to evaluate the current target-domain trajectory. ROAD MOPD lets routed and auxiliary teachers assess the same prefixes: their agreement guides token-level credit assignment, while auxiliary supervision defines preventive gradient constraints before domain updates are aggregated.

3 Method

3.1 Problem Setup and On-Policy Distillation

Let π_θ be the student policy, π_T a teacher, and $\mathcal{D}$ a prompt distribution. For $x \sim \mathcal{D}$, the student samples $y = (y_1, \ldots, y_T)$ with prefixes $h_t = (x, y_{<t})$. OPD minimizes the sequence-level reverse-KL objective:

$$\mathcal{L}_{\mathrm{OPD}}(\theta; \pi_T, \mathcal{D}) = \mathbb{E}_{x\sim\mathcal{D}} D_{\mathrm{KL}}(\pi_\theta(\cdot \mid x) \,\|\, \pi_T(\cdot \mid x)) \,. \tag{1}$$

Following prior work (Yang et al., 2026b), we express OPD as relative-reward maximization with a KL penalty to a frozen reference π_{ref}. With reward scale $\lambda > 0$, we maximize

$$\mathcal{J}_\lambda(\theta) = \mathbb{E}_\theta \left[\lambda \log \frac{\pi_T(y \mid x)}{\pi_{\mathrm{ref}}(y \mid x)} - D_{\mathrm{KL}}(\pi_\theta(\cdot \mid x) \,\|\, \pi_{\mathrm{ref}}(\cdot \mid x)) \right] . \tag{2}$$

where $\mathbb{E}_\theta$ averages over $x \sim \mathcal{D}$ and $y \sim \pi_\theta(\cdot \mid x)$. Maximizing this objective recovers standard OPD at $\lambda = 1$; $0 < \lambda < 1$ attenuates the teacher's relative reward, while $\lambda > 1$ amplifies it (ExOPD). The reference provides a shared baseline, not an additional teacher.

For K domain–teacher pairs $\{(\mathcal{D}_d, \pi_{T_d})\}_{d=1}^K$, MOPD routes each prompt to its domain teacher and minimizes the equal-weight objective:

$$\mathcal{L}_{\mathrm{MOPD}}(\theta) = \frac{1}{K} \sum_{d=1}^{K} \mathcal{L}_{\mathrm{OPD}}(\theta; \pi_{T_d}, \mathcal{D}_d). \tag{3}$$

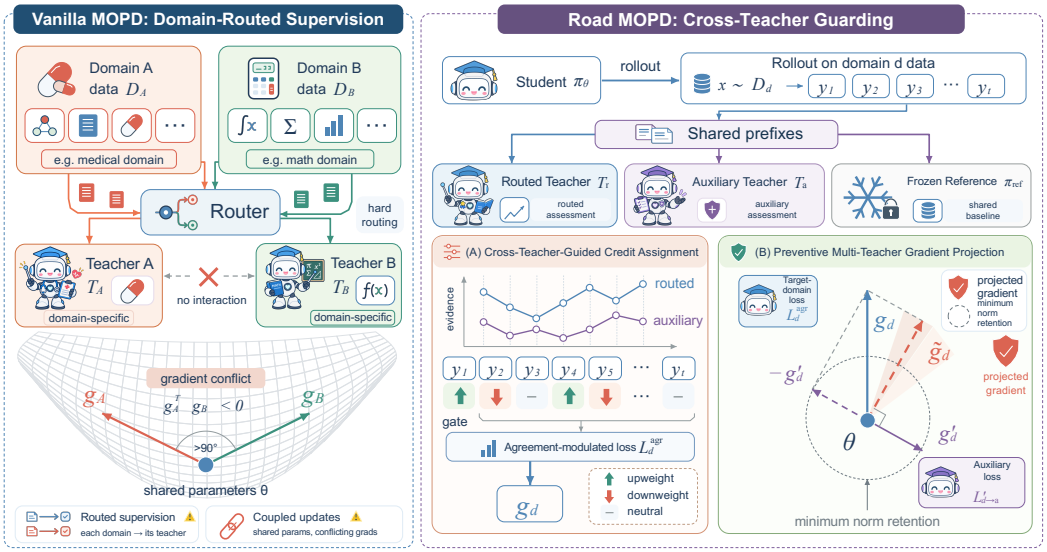

Figure 2: **Vanilla MOPD and ROAD MOPD.** Left: Vanilla MOPD's domain-routed supervision can produce conflicting updates to shared parameters. Right: ROAD MOPD uses same-trajectory auxiliary-teacher feedback for (A) agreement-guided token credit assignment and (B) preventive gradient projection under a target-gradient norm-retention constraint.

For a fixed target domain $d \in \{1, \ldots, K\}$, denote its routed teacher by $T_{\mathrm{r}} = T_d$ and a teacher from an auxiliary domain by T_{a}. The upright labels r and a denote teacher roles, while d indexes domains. Both teachers and the reference score the same student-generated tokens under teacher forcing, without additional rollouts. We write ℓ_t^{r}, ℓ_t^{a}, and ℓ_t^{ref} for their respective log-probabilities of y_t given h_t.

3.2 CROSS-TEACHER-GUIDED CREDIT ASSIGNMENT

A fixed λ in Eq. (2) applies the same reward scale to every token, although a trajectory may contain both transferable reasoning patterns and domain-specific corrections. The routed teacher alone provides no cross-domain check on these corrections. We therefore query an auxiliary teacher on the same student-generated prefixes and use its feedback to adjust the routed teacher's token-level reward. This strengthens supervision supported by both teachers and attenuates conflicting signals. Importantly, the auxiliary teacher is not treated as a supervisor for solving the target-domain task. Its relative preference is used only as a compatibility signal for the capability represented by that teacher.

To compare teacher preferences, we measure each teacher's log-probability relative to the shared frozen reference. Their token-wise agreement is

$$s_t = \underbrace{\tanh\left(\frac{\ell_t^{\mathrm{r}} - \ell_t^{\mathrm{ref}}}{\tau}\right)}_{\text{routed-teacher evidence}} \times \underbrace{\tanh\left(\frac{\ell_t^{\mathrm{a}} - \ell_t^{\mathrm{ref}}}{\tau}\right)}_{\text{auxiliary-teacher evidence}}, \qquad s_t \in [-1, 1]. \tag{4}$$

Positive s_t means that both teachers favor the same direction of correction relative to the reference. Negative s_t indicates opposing preferences, with values near -1 corresponding to strong cross-teacher disagreement. The score approaches zero when either teacher provides little evidence beyond the reference.

We use this agreement to replace the global reward scale in Eq. (2) with a token-wise coefficient:

$$\lambda_t = \mathrm{clip}(1 + \rho_{\mathrm{g}} s_t, \lambda_{\min}, \lambda_{\max}). \tag{5}$$

Here, $\rho_{\mathrm{g}} > 0$ controls the modulation strength; the subscript g denotes the agreement gate. When $s_t > 0$, $\lambda_t > 1$ strengthens the routed teacher's relative reward on tokens supported by both teachers. When $s_t < 0$, $\lambda_t < 1$ attenuates the reward, with stronger disagreement producing greater

attenuation. When $s_t = 0$, $\lambda_t = 1$ recovers standard OPD. Clipping limits both amplification and attenuation.

For a sampled token y_t and a comparison policy q, the k_1 log-ratio estimator is $k_{1,t}(\pi_\theta \| q) = \log \pi_\theta(y_t \mid h_t) - \log q(y_t \mid h_t)$. The modulated signal $\lambda_t(\ell_t^{\mathrm{r}} - \ell_t^{\mathrm{ref}}) - k_{1,t}(\pi_\theta \| \pi_{\mathrm{ref}})$ is clipped and detached as the advantage in the PPO-style surrogate $\ell_{d,i,t}^{\mathrm{agr}}(\theta)$. Gradients flow through the policy importance ratio, not the advantage (Appendix A.1). With multiple auxiliary teachers, we average their token-wise agreement with the routed teacher before applying the same reward modulation (Appendix A.1).

3.3 Preventive Multi-Teacher Gradient Projection

Token-level agreement does not guarantee compatible parameter updates, since token supervision signals are aggregated through shared parameters. We therefore formulate gradient correction as a constrained optimization problem: minimize conflict with an auxiliary teacher on the same trajectories while preserving a minimum fraction of the target-gradient norm.

We first compute the target-domain gradient $g_d = \nabla_\theta \mathcal{L}_d^{\mathrm{agr}}$ from the agreement-modulated loss. To construct the auxiliary gradient, we replace the routed teacher with the auxiliary teacher and disable agreement modulation. Let $\ell_{d\to \mathrm{a},i,t}^{\mathrm{OPD}}$ denote the resulting plain token-level OPD loss. Applying the same response-level reduction gives the auxiliary loss and gradient, which we distinguish from their target-domain counterparts using primes:

$$\mathcal{L}'_{d\to \mathrm{a}} = \frac{1}{|\mathcal{B}_d|} \sum_{i \in \mathcal{B}_d} \frac{1}{T_i} \sum_{t=1}^{T_i} \ell_{d\to \mathrm{a},i,t}^{\mathrm{OPD}}, \qquad g'_d = \nabla_\theta \mathcal{L}'_{d\to \mathrm{a}}. \tag{6}$$

Both losses use the same stored domain-d trajectories. Thus, g'_d reflects the auxiliary teacher's preferences on the current behavior, rather than its training gradient on auxiliary-domain data. Differences in knowledge, response style, or confidence can shape this signal; its usefulness as a constraint is assessed through joint-domain performance.

We optimize the corrected gradient separately for each trainable parameter tensor. Let g and g' be nonzero slices of g_d and g'_d, and let v denote a candidate corrected gradient. For an intended tensor-level step $-\eta v$ with $\eta > 0$, the first-order change in the same-trajectory auxiliary loss is $-\eta\langle v, g'\rangle$. Thus, $\langle g, g'\rangle < 0$ indicates local conflict; otherwise, we retain g unchanged. For a conflicting pair, we restrict v to the line segment between g and its full projection onto the hyperplane orthogonal to g' (Yu et al., 2020):

$$g_{\mathrm{full}} = g - \frac{\langle g, g'\rangle}{\|g'\|_2^2} g', \qquad \mathcal{S}(g, g') = \mathrm{conv}\{g, g_{\mathrm{full}}\}. \tag{7}$$

Here, conv denotes the convex hull. This restriction preserves the component orthogonal to g' and only removes the conflicting component, up to full projection. Full projection eliminates local conflict but can nearly erase g when the gradients oppose each other. We therefore choose the gradient with the least residual conflict subject to a norm-retention floor $m \in (0, 1]$:

$$\widetilde{g} = \operatorname*{arg\,min}_{v \in \mathcal{S}(g,g')} \; -\langle v, g'\rangle \quad \text{subject to} \quad \|v\|_2^2 \geq m^2 \|g\|_2^2. \tag{8}$$

On this segment, $-\langle v, g'\rangle \geq 0$ measures the predicted auxiliary-loss increase per unit step size, while the constraint preserves the target-gradient norm. The solution can be written as a partial projection:

$$\widetilde{g} = g - \alpha^\star \frac{\langle g, g'\rangle}{\|g'\|_2^2} g', \qquad \alpha^\star \in [0, 1]. \tag{9}$$

Here, $\alpha^\star$ is the fraction of the conflicting component removed. We use full projection ($\alpha^\star = 1$) whenever it satisfies the norm floor; otherwise, we remove as much conflict as the floor permits, with $\|\widetilde{g}\|_2 = m\|g\|_2$. For non-conflicting pairs, $\alpha^\star = 0$. The closed-form coefficient and its derivation are given in Appendix A.2. These are local statements about the same-trajectory auxiliary loss, not guarantees of auxiliary-domain performance after an optimizer step. Tensors with negligible gradient norm are left unchanged; auxiliary gradients are never applied as separate optimizer updates. The projected tensors form $\widetilde{g}_d$.

With multiple auxiliary teachers, we apply the projections sequentially under a shared norm-retention floor relative to the original target gradient (Appendix A.2). Method-specific hyperparameters are summarized in Appendix B.3.

A related approach in multi-task learning is PCGrad (Yu et al., 2020), which performs **post-hoc** gradient surgery by subtracting conflicting components from independently computed task gradients. Our projection is **preventive**: the auxiliary teacher assesses the current target-domain trajectory to construct a gradient constraint specifically for that update, using its feedback to constrain potentially incompatible corrections while retaining target-domain learning signals. We empirically compare these two projection strategies in Section 4.3.

3.4 Overall Objective and Training

We use sequence-mean aggregation, averaging token losses within each response and then across responses and domains. Auxiliary losses use the same reduction but guide only the gradient correction, so the overall distillation objective includes only target-domain losses:

$$\mathcal{L}_{\text{overall}}(\theta) = \frac{1}{K}\sum_{d=1}^{K}\frac{1}{|\mathcal{B}_d|}\sum_{i\in\mathcal{B}_d}\frac{1}{T_i}\sum_{t=1}^{T_i}\ell_{d,i,t}^{\text{agr}}(\theta). \tag{10}$$

After applying the preventive projection in Section 3.3, we average only the projected target-domain gradients, using the same domain weights as Eq. (10). The resulting g_{update} is globally clipped for one AdamW step:

$$g_{\text{update}} = \frac{1}{K}\sum_{d=1}^{K}\widetilde{g}_d. \tag{11}$$

4 Experiments

4.1 Experimental Setup

Evaluation datasets. We evaluate 11 benchmarks across Code (HumanEval+, MBPP+, and LiveCodeBench) (Liu et al., 2023; Jain et al., 2024), Math (AIME 2024/2025 and HMMT February/November 2025) (MAA, 2024; 2025; Harvard–MIT Mathematics Tournament, 2025a;b), and Medical (MedQA, MedXpertQA, MedMCQA, and SuperGPQA-Med) (Jin et al., 2020; Zuo et al., 2025; Pal et al., 2022; M-A-P Team et al., 2025). We report Avg@4 accuracy for Code, Avg@32 for Math, and Avg@1 for Medical. Avg@n averages correctness over n completions per problem and then over problems, equivalent to the same-sample pass@1 estimate for Math and Code.

Baselines. We compare with four representative approaches: (1) Vanilla MOPD routes each on-policy rollout only to its corresponding domain teacher and applies token-level reverse-KL supervision (Ma et al., 2026); (2) ExOPD constructs an extrapolated teacher target using the frozen student initialization as a reference (Yang et al., 2026b); (3) CaMOPD separates capability-recovery and domain-preservation updates and prioritizes samples with larger teacher–student gaps (Chen et al., 2026); and (4) Open-MOPD mitigates cross-domain imbalance through token-share balancing, dynamic domain loss weighting, and reward refresh (Gao et al., 2026).

Implementation details. We use Qwen3-4B and Qwen3-1.7B students with thinking disabled (Yang et al., 2025). Math and Code teachers use specialized Qwen3-4B checkpoints (Yang et al., 2026b). The Medical teacher is II-Medical-8B, a Qwen3-8B RL checkpoint (Intelligent Internet, 2025). Each paired setting uses 10,000 sampled prompts per domain from DeepMath, Eurus-RL-Code, and Medical sources including the MedMCQA training split (Table 5). Full configurations are in Appendix B.

4.2 Main Results

Best overall performance across domain pairs and student sizes. Across Math $\times$ Medical and Code $\times$ Medical with both student sizes, ROAD MOPD achieves the highest two-domain average among the compared multi-domain methods in all four settings (Table 1), exceeding the strongest overall baseline in each by 0.72–2.89 points.

Table 1: **Main results across paired-domain settings.** Domain Avg. averages benchmarks within each domain; Avg. averages the two domains. Bold/underlined scores mark the best/second-best multi-domain methods per student model.

(a) Math × Medical

Student Model: Qwen3-4B

Method	Math					Medical					All
	AIME24	AIME25	HMMT25 Feb.	HMMT25 Nov.	Domain Avg.	MedQA	MedXpert QA	MedMCQA	SuperGPQA -Med	Domain Avg.	*Avg.*
Student	21.46	21.77	10.00	8.23	15.37	64.73	13.80	55.63	29.87	41.01	28.19
Single OPD	58.13	51.67	31.56	35.73	44.27	77.06	15.31	62.44	36.81	47.90	46.09
Vanilla MOPD	58.75	55.21	31.98	37.92	45.97	62.61	12.24	55.39	27.55	39.45	42.71
ExOPD	61.67	**55.62**	33.54	37.92	47.19	69.99	13.76	59.50	32.23	43.87	45.53
Open-MOPD	56.56	55.10	32.60	37.08	45.34	70.31	13.06	60.94	33.76	44.52	44.93
CaMOPD	61.15	54.27	33.12	38.02	46.64	**77.45**	**16.61**	61.89	35.03	47.75	47.20
Ours	**63.85**	53.96	**33.65**	**38.96**	**47.60**	77.06	15.67	**62.83**	**37.35**	**48.23**	**47.92**

Student Model: Qwen3-1.7B

Method	Math					Medical					All
	AIME24	AIME25	HMMT25 Feb.	HMMT25 Nov.	Domain Avg.	MedQA	MedXpert QA	MedMCQA	SuperGPQA -Med	Domain Avg.	*Avg.*
Student	11.98	10.21	6.04	3.96	8.05	46.03	10.33	42.55	22.58	30.37	19.21
Single OPD	39.90	34.48	18.75	17.29	27.60	53.81	13.39	50.99	24.50	35.67	31.64
Vanilla MOPD	41.46	35.10	18.85	18.02	28.36	47.37	**11.39**	45.25	20.69	31.18	29.77
ExOPD	**41.56**	**35.42**	18.13	18.65	**28.44**	49.88	9.80	49.49	22.87	33.01	30.72
Open-MOPD	37.40	31.98	18.96	18.33	26.67	46.58	**11.39**	44.20	20.04	30.55	28.61
CaMOPD	36.35	33.13	**19.06**	18.23	26.69	47.84	10.16	44.63	21.49	31.03	28.86
Ours	40.10	31.88	17.81	**19.06**	27.21	**58.13**	11.06	**52.35**	**26.03**	**36.89**	**32.05**

(b) Code × Medical

Student Model: Qwen3-4B

Method	Code				Medical					All
	HumanEval+	MBPP+	LCB	Domain Avg.	MedQA	MedXpert QA	MedMCQA	SuperGPQA -Med	Domain Avg.	*Avg.*
Student	76.68	65.08	17.14	52.97	64.73	13.80	55.63	29.87	41.01	46.99
Single OPD	84.30	70.44	28.00	60.91	77.06	15.31	62.44	36.81	47.90	54.41
Vanilla MOPD	84.76	69.11	27.71	60.53	64.96	12.78	55.63	29.62	40.75	50.64
ExOPD	85.98	68.98	**30.00**	61.65	72.82	13.63	60.46	32.81	44.93	53.29
Open-MOPD	84.45	70.77	27.29	60.83	**76.83**	15.59	61.68	35.25	47.34	54.09
CaMOPD	84.91	69.58	27.43	60.64	76.12	15.39	62.49	35.10	47.28	53.96
Ours	**87.65**	**72.69**	28.71	**63.02**	75.73	**15.96**	**62.97**	**36.33**	**47.75**	**55.38**

Student Model: Qwen3-1.7B

Method	Code				Medical					All
	HumanEval+	MBPP+	LCB v6	Domain Avg.	MedQA	MedXpert QA	MedMCQA	SuperGPQA -Med	Domain Avg.	*Avg.*
Student	60.37	53.44	12.57	42.13	46.03	10.33	42.55	22.58	30.37	36.25
Single OPD	73.48	59.13	22.43	51.68	53.81	13.39	50.99	24.50	35.67	43.67
Vanilla MOPD	71.95	58.27	23.00	51.07	46.11	11.02	45.68	20.91	30.93	41.00
ExOPD	**75.76**	**62.30**	**24.43**	**54.16**	44.62	10.37	44.85	19.24	29.77	41.97
Open-MOPD	71.95	58.60	20.43	50.33	45.80	10.78	43.46	19.64	29.92	40.12
CaMOPD	71.65	55.29	23.00	49.98	47.53	10.33	45.80	21.71	31.34	40.66
Ours	73.78	61.44	**24.43**	53.22	**56.56**	**12.24**	**51.35**	**25.88**	**36.51**	**44.86**

4B students: Medical recovery accompanies gains in the other domain. In both Math × Medical and Code × Medical, Vanilla MOPD lowers Medical accuracy below the initial student's 41.01, while ExOPD, Open-MOPD, and CaMOPD mitigate this degradation to varying degrees. ROAD MOPD reaches Medical averages of 48.23 and 47.75, respectively, recovering to at or near the Single OPD reference of 47.90 and exceeding the strongest multi-domain Medical baseline in each setting by 0.48 and 0.41 points. It also achieves the highest Math and Code averages in the respective settings, indicating that Medical recovery accompanies gains in the other domain's average performance.

1.7B students: Larger Medical gains improve the cross-domain trade-off. With the smaller student, all multi-domain baselines fall below the Single OPD Medical average of 35.67 in both pairs. ROAD MOPD instead reaches 36.89 and 36.51, respectively, exceeding this reference in both

Table 2: **Low-conflict generalization: Math × Code.** Qwen3-4B domain averages (%) and their mean (All).

Method	Math	Code	All
Student	15.37	52.97	34.17
Single OPD	44.27	60.91	52.59
Vanilla MOPD	46.12	61.24	53.68
ExOPD	**48.28**	61.82	55.05
Open-MOPD	44.32	59.82	52.07
CaMOPD	46.30	60.37	53.34
Ours	47.73	**63.07**	**55.40**

Table 3: **Three-domain generalization: Math × Code × Medical.** Qwen3-4B domain averages (%) and their mean (All).

Method	Math	Code	Medical	All
Student	15.37	52.97	41.01	36.45
Single OPD	44.27	60.91	47.90	51.03
Vanilla MOPD	46.98	61.26	41.03	49.76
ExOPD	47.09	61.52	45.62	51.41
Open-MOPD	44.06	60.56	38.83	47.82
CaMOPD	45.96	60.97	48.20	51.71
Ours	**47.29**	**62.94**	**48.45**	**52.89**

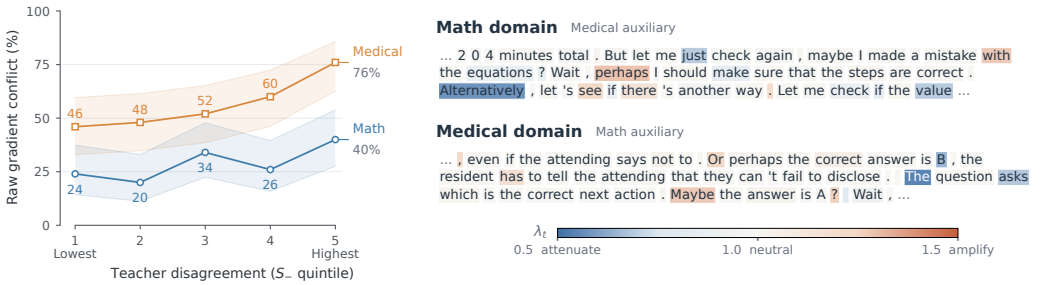

Figure 3: **Teacher disagreement and local supervision.** Left: raw gradient conflict by disagreement quintile (Math: blue; Medical: orange). Right: Math (top) and Medical (bottom) response windows; blue attenuates ($\lambda_t < 1$) and orange amplifies ($\lambda_t > 1$) token supervision.

settings. Its margins over the strongest multi-domain Medical baseline in each setting increase to 3.88 and 5.17 points. Although its Math and Code averages trail ExOPD by 1.23 and 0.94 points, respectively, the Medical gains yield two-domain averages that exceed ExOPD by 1.33 and 2.89 points, demonstrating a better overall result when domains are weighted equally.

4.3 Further Analysis

Generalizing to lower-conflict settings and more domains. Tables 2 and 3 show that ROAD MOPD also improves joint performance in low-conflict and three-domain settings. For Math × Code, it achieves the best Code and second-best Math averages among the compared multi-domain methods. Although its Math average trails ExOPD by 0.55 points, its 1.25-point Code gain yields the highest All Avg. of 55.40, exceeding ExOPD by 0.35 points. With all three domains, ROAD MOPD leads in every domain and achieves the best All Avg. of 52.89, surpassing CaMOPD by 1.18 points. This setting also tests multiple auxiliary teachers, using aggregated agreement for credit assignment and sequential constraints for gradient protection. Full per-benchmark results are in Appendix C.2 (Tables 9 and 10).

Cross-teacher disagreement is associated with gradient conflict, supporting token-level credit assignment. To examine whether cross-teacher agreement reflects update compatibility, we sample response batches at a fixed student checkpoint, averaging $\max(-s_t, 0)$ (Section 3.2) over all valid tokens per batch to obtain disagreement S_-. Within each domain, we sort batches by S_- from low to high into five equal-sized groups. Figure 3 (left) plots each group's fraction of batches with negative cosine between raw full-parameter OPD gradients under the two teachers on identical responses. Conflict rates trend upward from 24% to 40% in Math and 46% to 76% in Medical. This suggests that teacher disagreement can signal potential update conflicts, providing empirical support for agreement-guided credit assignment. Figure 3 (right) visualizes token-level credit assignment: relative to the shared reference, opposing teacher corrections on *Alternatively* in Math and *The* in Medical attenuate the routed teacher's relative reward signal, whereas aligned corrections on *perhaps*

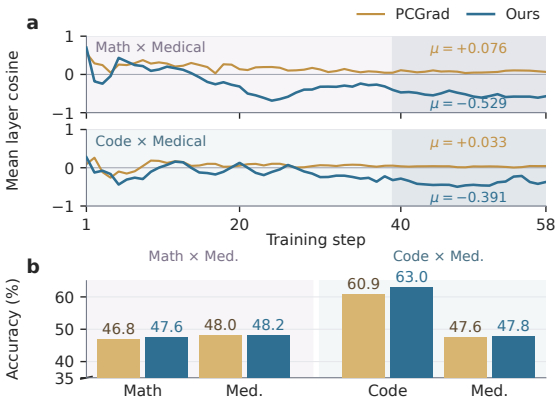

Figure 4: **Preventive projection vs. PCGrad.** (a) Mean layerwise cosine; labels average shaded steps 39–58. (b) Domain accuracy (%).

Table 4: **Ablations of ROAD MOPD.** Domain-average accuracy (%). Avg.: mean across the two domains.

Math × Med.	**Math**	**Med.**	**Avg.**
w/o Credit Assign.	45.78	47.74	46.76
w/o Grad. Protect.	47.19	48.11	47.65
w/o Adaptive Ret.	45.78	**48.40**	47.09
ROAD MOPD	**47.60**	48.23	**47.92**
Code × Med.	**Code**	**Med.**	**Avg.**
w/o Credit Assign.	59.54	47.46	53.50
w/o Grad. Protect.	63.01	46.87	54.94
w/o Adaptive Ret.	62.85	47.03	54.94
ROAD MOPD	**63.02**	**47.75**	**55.38**

and *Maybe* in the respective domains amplify it. Probe details are provided in Appendix B.5; full prompts and visualizations of extended responses appear in Appendix C.3 (Figure 5).

Preventive projection improves joint learning over post-hoc correction. Figure 4 compares preventive projection with PCGrad under matched training settings; both retain cross-teacher credit assignment and the same norm-retention mechanism. PCGrad corrects conflicts between gradients computed independently on each domain's own trajectories. Our method instead constrains target-domain updates using auxiliary-teacher gradients on the same rollout. This directly tests compatibility between the target gradient and auxiliary-teacher feedback on the same behavior. Unlike PCGrad's comparison of independent domain gradients, this constraint can address local conflicts that may persist even when those gradients align. Figure 4a shows negative mean layerwise cosines for same-trajectory pairs during much of training, whereas independently formed domain-gradient pairs are mostly non-negative. These patterns reflect different local compatibility criteria and do not directly measure capability damage. Across Math × Medical and Code × Medical, ROAD MOPD improves all four domain-average accuracies over PCGrad by 0.18–2.12 percentage points (Figure 4b). These gains support the utility of same-trajectory constraints for joint learning.

Each component improves joint performance. Table 4 shows that removing either credit assignment or gradient protection lowers all four domain averages across the two domain pairs. Without credit assignment, Math and Code accuracy drop by 1.82 and 3.48 points, respectively; removing gradient protection reduces Medical accuracy by 0.88 points in Code × Medical. Replacing adaptive retention with full projection lowers the two-domain mean by 0.83 and 0.44 points in Math × Medical and Code × Medical, respectively, despite a 0.17-point Medical gain in the former. The full method achieves the highest equally weighted two-domain mean in both settings. These results support adaptive retention as a way to balance conflict reduction with target-domain learning, alongside credit assignment and gradient protection.

5 CONCLUSION

We identify update incompatibility as a central challenge in multi-teacher on-policy distillation: domain-routed supervision can produce conflicting updates to shared student parameters. ROAD MOPD addresses this challenge using auxiliary teachers to assess the same student trajectories. Agreement-aware token weighting and preventive gradient projection suppress conflicting supervision and updates while retaining target-domain learning signals. Across mathematics, medicine, and code, ROAD MOPD mitigates high-conflict degradation, retains gains in compatible settings, and achieves the best distilled-model average in each domain under joint three-domain training. Our evaluation is limited to three text-based reasoning domains, and auxiliary scoring and gradient computation add training cost. Extending to broader domains and larger teacher pools calls for efficient teacher selection and compatibility estimation.

AI USE STATEMENT

We used generative AI tools to assist with English translation, language polishing, and the presentation of figures and tables. This assistance included refining wording, improving clarity, and adjusting captions and layouts. The authors retain full responsibility for the paper's technical content, results, citations, and conclusions, as well as the accuracy and integrity of the final manuscript.

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 *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/5be69a584901a26c521c2b51e40a4c20-Abstract-Conference.html.

Junying Chen, Zhenyang Cai, Ke Ji, Xidong Wang, Wanlong Liu, Rongsheng Wang, Jianye Hou, and Benyou Wang. HuatuoGPT-o1, towards medical complex reasoning with LLMs. *arXiv preprint arXiv:2412.18925*, 2024.

Tianlei Chen, Jiao Ou, Ziyuan Liu, Ruiming Tang, Jian Liang, and Han Li. Counteraction-aware multi-teacher on-policy distillation for general capability recovery with domain preservation. *arXiv preprint arXiv:2605.27115*, 2026.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, et al. Process Reinforcement through Implicit Rewards. *arXiv preprint arXiv:2502.01456*, 2025. URL https://arxiv.org/abs/2502.01456.

DeepSeek-AI. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv preprint arXiv:2501.12948*, 2025. URL https://arxiv.org/abs/2501.12948.

DeepSeek-AI, Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, et al. DeepSeek-V4: Towards Highly Efficient Million-Token Context Intelligence. *arXiv preprint arXiv:2606.19348*, 2026. URL https://arxiv.org/abs/2606.19348.

FreedomIntelligence. Medical-R1-Distill-Data. Hugging Face dataset, 2025. URL https://huggingface.co/datasets/FreedomIntelligence/Medical-R1-Distill-Data.

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. *arXiv preprint arXiv:2608.19098*, 2026.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *Proceedings of ICLR*, 2024.

Harvard–MIT Mathematics Tournament. Archive of february 2025. Official tournament archive, 2025a. URL https://www.hmmt.org/www/archive/282.

Harvard–MIT Mathematics Tournament. Archive of november 2025. Official tournament archive, 2025b. URL https://www.hmmt.org/www/archive/291.

Haonan He, Haodi Lei, Yun Luo, Haoran Zhang, Shunkai Zhang, Yizhuo Li, Shengji Tang, Zhilin Wang, Runzhe Zhan, Lei Bai, Ganqu Cui, Fangchen Yu, Yafu Li, Peng Ye, Ning Ding, and Yu Cheng. Simpleopd: Simple tokenizer-agnostic on-policy distillation for long-context reasoning. *arXiv preprint arXiv:2608.14277*, 2026.

Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. DeepMath-103K: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. *arXiv preprint arXiv:2504.11456*, 2025.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531*, 2015. URL https://arxiv.org/abs/1503.02531.

Wenjin Hou, Shangpin Peng, Weinong Wang, Zheng Ruan, Yue Zhang, Zhenglin Zhou, Mingqi Gao, Yifei Chen, Kaiqi Wang, Hongming Yang, Chengquan Zhang, Zhuotao Tian, Han Hu, Yi Yang, Fei Wu, and Hehe Fan. Uni-OPD: Unifying On-Policy Distillation with a Dual-Perspective Recipe. *arXiv preprint arXiv:2605.03677*, 2026. URL https://arxiv.org/abs/2605.03677.

Xiaoke Huang, Juncheng Wu, Hui Liu, Xianfeng Tang, and Yuyin Zhou. m1: Unleash the Potential of Test-Time Scaling for Medical Reasoning with Large Language Models. *arXiv preprint arXiv:2504.00869*, 2025. URL https://arxiv.org/abs/2504.00869.

Intelligent Internet. II-Medical-8B: Medical reasoning model. Hugging Face model, 2025. URL https://huggingface.co/Intelligent-Internet/II-Medical-8B.

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

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *arXiv preprint arXiv:2009.13081*, 2020.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In Jian Su, Kevin Duh, and Xavier Carreras (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1317–1327, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1139. URL https://aclanthology.org/D16-1139/.

Jongwoo Ko, Tianyi Chen, Sungnyun Kim, Tianyu Ding, Luming Liang, Ilya Zharkov, and Se-Young Yun. DistiLLM-2: A contrastive approach boosts the distillation of LLMs. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 31044–31062. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/ko25a.html.

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*, 2026a.

Zhaoyi Li, Deyang Kong, Yuan Wei, Evan Yang, Ranran Shen, Mahardika Krisna Ihsani, Ming Yang, Wei Zhang, Chuan Hao, Jian Yang, Ran Tao, Bryan Dai, Shikun Zhang, Wei Ye, Ying Wei, and Defu Lian. Every coin has two sides: On the dual nature of generalization in on-policy distillation of large language models. *arXiv preprint arXiv:2608.16647*, 2026b.

Yiqing Liang, Jielin Qiu, Wenhao Ding, Zuxin Liu, James Tompkin, Mengdi Xu, Mengzhou Xia, Zhengzhong Tu, Laixi Shi, and Jiacheng Zhu. Modomodo: Multi-domain data mixtures for multimodal llm reinforcement learning. *arXiv preprint arXiv:2505.24871*, 2025.

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.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. In *Conference on Neural Information Processing Systems (NeurIPS), 2021*, December 2021.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is Your Code Generated by ChatGPT Really Correct? Rigorous Evaluation of Large Language Models for Code Generation. In *Thirty-Seventh Conference on Neural Information Processing Systems*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/43e9d647ccd3e4b7b5baab53f0368686-Abstract.html.

M-A-P Team, Xinrun Du, Yifan Yao, Kaijing Ma, Bingli Wang, Tianyu Zheng, et al. SuperGPQA: Scaling LLM Evaluation across 285 Graduate Disciplines. *arXiv preprint arXiv:2502.14739*, 2025. URL https://arxiv.org/abs/2502.14739.

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

Mathematical Association of America. 2024 AIME I Problems. Problem sets archived by Art of Problem Solving, 2024. URL https://artofproblemsolving.com/wiki/index.php?title=2024_AIME_I_Problems.

Mathematical Association of America. 2025 AIME I Problems. Problem sets archived by Art of Problem Solving, 2025. URL https://artofproblemsolving.com/wiki/index.php?title=2025_AIME_I_Problems.

OpenAI. GPT-4 Technical Report. Technical Report arXiv:2303.08774, OpenAI, 2023. URL https://arxiv.org/abs/2303.08774.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. MedMCQA: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pp. 248–260. PMLR, 2022. URL https://proceedings.mlr.press/v174/pal22a.html.

Bing Shao, Jiazheng Zhang, Long Ma, Yujiong Shen, Senjie Jin, Xin Guo, Yuming Yang, Mingxu Chai, Zhiheng Xi, Boyang Liu, Junlin Shang, Tao Gui, Qi Zhang, and Xuanjing Huang. A Token-Level Analysis of Sampled-Token Reverse-KL On-Policy Distillation. *arXiv preprint arXiv:2608.25643*, 2026. URL https://arxiv.org/abs/2608.25643.

Changhui Sun, Lanbo Liu, Hang Lei, Tong Ling, Jiahang Xie, Zhiyong Zheng, Yujia Wang, Hao Liu, Feng Xiao, Lu Liu, Yanlong Du, Zifeng Cheng, Ziwei Jiang, and Qing Gu. Step-level on-policy distillation: Interpolating between on-policy distillation and supervised fine-tuning. *arXiv preprint arXiv:2608.16333*, 2026a.

Zechen Sun, Zhiwei Zhang, Fei Zhao, Juntao Li, Mu Chuan, Huayu Deng, Guojian Zhan, Wenliang Chen, Yao Hu, and Min Zhang. D^3-MOPD: Dynamic Domain ScheDuling for Efficient Multi-Teacher Distillation. *arXiv preprint arXiv:2608.24987*, 2026b. URL https://arxiv.org/abs/2608.24987v2.

Juncheng Wu, Wenlong Deng, Xingxuan Li, Sheng Liu, Taomian Mi, Yifan Peng, Ziyang Xu, Yi Liu, Hyunjin Cho, Chang-In Choi, Yihan Cao, Hui Ren, Xiang Li, Xiaoxiao Li, and Yuyin Zhou. MedReason: Eliciting factual medical reasoning steps in LLMs via knowledge graphs. *arXiv preprint arXiv:2504.00993*, 2025.

Xiaomi LLM-Core Team, Bangjun Xiao, Bingquan Xia, Bo Yang, Bofei Gao, Bowen Shen, et al. MiMo-V2-Flash Technical Report. *arXiv preprint arXiv:2601.02780*, 2026. URL https://arxiv.org/abs/2601.02780.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V. Le, Tengyu Ma, and Adams Wei Yu. Doremi: Optimizing data mixtures speeds up language model pretraining. *Advances in Neural Information Processing Systems*, 36:69798–69818, 2023.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, et al. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

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

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*, 2026b. URL https://arxiv.org/abs/2602.12125.

Zhuolin Yang, Zihan Liu, Yang Chen, Wenliang Dai, Boxin Wang, Sheng-Chieh Lin, Chankyu Lee, Yangyi Chen, Dongfu Jiang, Jiafan He, Renjie Pi, Grace Lam, Nayeon Lee, Alexander Bukharin, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. Nemotron-Cascade 2: Post-Training LLMs with Cascade RL and Multi-Domain On-Policy Distillation. *arXiv preprint arXiv:2603.19220*, 2026c. URL https://arxiv.org/abs/2603.19220.

Qixiang Yin, Huanjin Yao, Yuchen Cai, Jianghao Chen, Ziyi Wang, Min Yang, Fei Su, and Zhicheng Zhao. H-opd: Confidence aware heterogeneous multi-teacher multimodal on-policy distillation. *arXiv preprint arXiv:2607.02592*, 2026.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33:5824–5836, 2020.

Zhu Zhang, Jixun Wang, Xiaoang Xu, Xiaorong Wang, Zihan Zhou, Zhiyuan Wang, Shuo Wang, Chaojun Xiao, and Yuezhi Zhou. Beyond teacher likelihood: Group-calibrated on-policy distillation for long-context reasoning. *arXiv preprint arXiv:2608.19181*, 2026.

Yuxin Zuo, Shang Qu, Yifei Li, Zhang-Ren Chen, Xuekai Zhu, Ermo Hua, Kaiyan Zhang, Ning Ding, and Bowen Zhou. MedXpertQA: Benchmarking expert-level medical reasoning and understanding. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 80961–80990. PMLR, 2025. URL https://proceedings.mlr.press/v267/zuo25a.html.

APPENDIX CONTENTS

Page

A Additional Method Details 15
A.1 OPD Derivation and Training Loss 15
A.2 Derivation of Norm-Constrained Projection 16
A.3 Complete Training Procedure 17

B Experimental Setup and Reproducibility 18
B.1 Models and Teacher–Student Configurations 18
B.2 Training Data and Prompt Construction 18
B.3 Training Hyperparameters and Infrastructure 19
B.4 Baseline Implementations 20
B.5 Evaluation Protocols 21

C Additional Experimental Results 22
C.1 Detailed Benchmark Results 22
C.2 Full Results for Low-Conflict and Three-Domain Settings 23
C.3 Extended Token-Level Visualizations 24
C.4 Training Dynamics 28

D Limitations 30

A Additional Method Details

A.1 OPD Derivation and Training Loss

From sequence-level reverse KL to token-level supervision. We expand the OPD reformulation in Section 3.1 before specifying its sampled-token implementation. For a prompt $x \sim \mathcal{D}$, let $y = (y_1, \ldots, y_T) \sim \pi_\theta(\cdot \mid x)$ and $h_t = (x, y_{<t})$. Write $p_t = \pi_\theta(\cdot \mid h_t)$ and $q_t = \pi_T(\cdot \mid h_t)$ for the student and teacher distributions on the same prefix. With $\mathbb{E}_\theta$ denoting expectation over these prompts and student-generated responses, autoregressive factorization and the KL chain rule give

$$\begin{aligned}\mathcal{L}_{\mathrm{OPD}}(\theta) &= \mathbb{E}_{x\sim\mathcal{D}} D_{\mathrm{KL}}(\pi_\theta(\cdot \mid x) \,\|\, \pi_T(\cdot \mid x)) \\ &= \mathbb{E}_\theta\left[\log \frac{\pi_\theta(y \mid x)}{\pi_T(y \mid x)}\right] \\ &= \mathbb{E}_\theta\left[\sum_{t=1}^{T} \log \frac{p_t(y_t)}{q_t(y_t)}\right] = \mathbb{E}_\theta\left[\sum_{t=1}^{T} D_{\mathrm{KL}}(p_t \,\|\, q_t)\right].\end{aligned} \tag{12}$$

The expectations assume finite divergences and common policy support. Thus, the token-level teacher distributions are evaluated on prefixes visited by the student.

Equivalent relative-reward objective. Introducing a frozen reference π_{ref} and adding and subtracting its log-probability exposes the teacher's relative reward (Yang et al., 2026b):

$$\begin{aligned}-\mathcal{L}_{\mathrm{OPD}}(\theta) &= \mathbb{E}_\theta[\log \pi_T(y \mid x) - \log \pi_\theta(y \mid x)] \\ &= \mathbb{E}_\theta\left[\big(\log \pi_T(y \mid x) - \log \pi_{\mathrm{ref}}(y \mid x)\big) - \big(\log \pi_\theta(y \mid x) - \log \pi_{\mathrm{ref}}(y \mid x)\big)\right] \\ &= \mathbb{E}_\theta\left[\log \frac{\pi_T(y \mid x)}{\pi_{\mathrm{ref}}(y \mid x)}\right] - \mathbb{E}_{x\sim\mathcal{D}} D_{\mathrm{KL}}(\pi_\theta(\cdot \mid x) \,\|\, \pi_{\mathrm{ref}}(\cdot \mid x)) \equiv \mathcal{J}_1(\theta).\end{aligned} \tag{13}$$

Consequently, $\arg\min_\theta \mathcal{L}_{\mathrm{OPD}}(\theta) = \arg\max_\theta \mathcal{J}_1(\theta)$: the reward-maximization objective is the negative of the reverse-KL loss. Scaling only the relative-reward term by $\lambda > 0$ gives the objective in Eq. (2), whose sampled-response form is

$$\begin{aligned}\mathcal{J}_\lambda(\theta) &= \mathbb{E}_\theta\left[\lambda \log \frac{\pi_T(y \mid x)}{\pi_{\mathrm{ref}}(y \mid x)} - \log \frac{\pi_\theta(y \mid x)}{\pi_{\mathrm{ref}}(y \mid x)}\right] \\ &= \mathbb{E}_\theta\left[\sum_{t=1}^{T} \left\{\lambda\big(\ell_t^T - \ell_t^{\mathrm{ref}}\big) - \big(\ell_t^\theta - \ell_t^{\mathrm{ref}}\big)\right\}\right],\end{aligned} \tag{14}$$

where ℓ_t^T, ℓ_t^θ, and ℓ_t^{ref} are the teacher, student, and reference log-probabilities of the sampled token. At $\lambda = 1$, the reference terms cancel and recover OPD; $0 < \lambda < 1$ attenuates the relative reward, whereas $\lambda > 1$ amplifies it (ExOPD). Domain routing applies this construction to each teacher–domain pair in Eq. (3).

Credit assignment with multiple teachers. For the $K > 2$ extension of Section 3.2, let $\mathcal{P}_d = \{j \mid j \neq d\}$ be the peer teachers for target domain d. Here we use explicit teacher indices: for each peer j, the main-text roles are $T_{\mathrm{r}} = T_d$ and $T_{\mathrm{a}} = T_j$, with $\ell_t^{\mathrm{r}} = \ell_t^d$ and $\ell_t^{\mathrm{a}} = \ell_t^j$. All peers score the same trajectory as the routed teacher, and their token agreements are averaged:

$$\bar{s}_{d,t} = \frac{1}{K-1} \sum_{j\in\mathcal{P}_d} \tanh\left(\frac{\ell_t^d - \ell_t^{\mathrm{ref}}}{\tau}\right) \tanh\left(\frac{\ell_t^j - \ell_t^{\mathrm{ref}}}{\tau}\right). \tag{15}$$

We obtain $\lambda_{d,t}$ by applying Eq. (5) with $s_t = \bar{s}_{d,t}$. For $K = 2$, this recovers Eqs. (4)–(5). For $K > 2$, the $K - 1$ peers reinforce or attenuate the routed signal according to their mean evidence; opposing assessments can cancel, recovering standard OPD when $\bar{s}_{d,t} = 0$.

Sampled-token learning signal. The implementation replaces the fixed reward scale in Eq. (14) with the agreement coefficient from Section 3.2, using the sampled-token k_1 log-ratio estimator and a PPO-style policy surrogate. Let π_{old} denote the student policy that generates a stored response,

and let $\ell^{\theta}_{i,t} = \log \pi_\theta(y_{i,t} \mid h_{i,t})$ and $\ell^{\text{old}}_{i,t} = \log \pi_{\text{old}}(y_{i,t} \mid h_{i,t})$. For a response from domain d, the routed teacher and shared frozen reference define $\Delta^{d}_{i,t} = \ell^{d}_{i,t} - \ell^{\text{ref}}_{i,t}$. The advantage supplied to the policy loss is

$$A^{\text{agr}}_{d,i,t} = \text{sg}\big[\text{clip}\big(\lambda_{d,i,t}\Delta^{d}_{i,t} - \big(\ell^{\theta}_{i,t} - \ell^{\text{ref}}_{i,t}\big), -10, 10\big)\big]\,, \tag{16}$$

where sg denotes stop-gradient. The entire recomputed advantage is clipped to $[-10, 10]$ after agreement modulation and detached before entering the policy surrogate, including its student log-probability term. Teacher and reference parameters are frozen throughout training. When $\lambda_{d,i,t} = 1$, Eq. (16) reduces to the plain OPD learning signal $\text{sg}[\text{clip}(\ell^{d}_{i,t} - \ell^{\theta}_{i,t}, -10, 10)]$. Thus, the coefficient rescales the teacher's reward relative to the reference; it does not multiply the entire OPD loss.

Clipped policy surrogate and masking. The importance ratio and per-token loss are

$$r_{i,t}(\theta) = \exp\big(\ell^{\theta}_{i,t} - \ell^{\text{old}}_{i,t}\big)\,, \tag{17}$$

$$\ell^{\text{agr}}_{d,i,t}(\theta) = -\min\Big\{r_{i,t}(\theta)A^{\text{agr}}_{d,i,t},\ \text{clip}(r_{i,t}(\theta), 1-\epsilon_{\text{PPO}}, 1+\epsilon_{\text{PPO}})\,A^{\text{agr}}_{d,i,t}\Big\}\,. \tag{18}$$

Let $M_{i,t}$ indicate a valid response token and $T_i = \sum_t M_{i,t}$. The target-domain loss is

$$\mathcal{L}^{\text{agr}}_{d} = \frac{1}{|\mathcal{B}_d|}\sum_{i\in\mathcal{B}_d}\frac{\sum_t M_{i,t}\ell^{\text{agr}}_{d,i,t}}{T_i}. \tag{19}$$

Prompt and padding positions do not contribute to the loss. The recorded training configuration uses $\epsilon_{\text{PPO}} = 0.2$, without an outcome-based task reward or a separate KL reward added by the trainer. The single-turn rollout mask retains the returned generated tokens, including EOS when returned, and excludes padding. A response reaching the length cap retains supervision on its available prefix. Fully masked responses contribute zero. The agreement coefficient $\lambda_{d,i,t}$ is bounded by $[0.5, 1.5]$.

Auxiliary supervision. For each peer teacher $j \neq d$, we reuse the same stored domain-d responses and replace the advantage by $A^{\text{OPD}}_{d\to j,i,t} = \text{sg}[\text{clip}(\ell^{j}_{i,t} - \ell^{\theta}_{i,t}, -10, 10)]$. The auxiliary loss uses the same clipped surrogate, with agreement modulation disabled and the same advantage clipping range. For the selected auxiliary teacher $T_{\text{a}} = T_j$, we write the main-text loss $\mathcal{L}'_{d\to\text{a}}$ as $\mathcal{L}'_{d\to j}$. The full method and all ablations retain this range, including when credit assignment is disabled. Advantage clipping is separate from the importance-ratio clipping in Eq. (18).

A.2 Derivation of Norm-Constrained Projection

For nonzero g and g', let $c = \langle g, g'\rangle/(\|g\|_2\|g'\|_2)$. When $c < 0$, every candidate in $\mathcal{S}(g, g')$ can be written as

$$v(\alpha) = g - \alpha\frac{\langle g, g'\rangle}{\|g'\|_2^2}g', \qquad 0 \le \alpha \le 1. \tag{20}$$

The objective in Eq. (8) is $-(1-\alpha)\langle g, g'\rangle$, which strictly decreases with α. The optimum therefore uses the largest feasible coefficient. Expanding the squared norm gives

$$\frac{\|v(\alpha)\|_2^2}{\|g\|_2^2} = 1 - (2\alpha - \alpha^2)c^2. \tag{21}$$

This decreases monotonically on $[0, 1]$. Full projection is feasible if $\sqrt{1-c^2} \ge m$; otherwise, the optimum satisfies $1 - (2\alpha - \alpha^2)c^2 = m^2$. Including the no-conflict case gives

$$\alpha^\star = \begin{cases} 0, & c \ge 0, \\ 1, & c < 0 \text{ and } \sqrt{1-c^2} \ge m, \\ 1 - \sqrt{1 - \dfrac{1-m^2}{c^2}}, & c < 0 \text{ and } \sqrt{1-c^2} < m. \end{cases} \tag{22}$$

Substitution into Eq. (9) recovers the unique solution of Eq. (8) for a conflicting pair.

The norm constraint trades complete conflict removal for retention of the target-domain signal. In particular,

$$\langle \widetilde{g}, g'\rangle = (1-\alpha^\star)\langle g, g'\rangle, \tag{23}$$

which remains negative when $c < 0$ and $\alpha^\star < 1$. These statements concern the projected tensor before domain aggregation, global clipping, and AdamW preconditioning. They do not guarantee a non-increasing auxiliary loss after the final optimizer step, nor a lower bound on the norm of the merged update.

Multi-teacher projection and shared norm retention. For $K > 2$, each peer $j \in \mathcal{P}_d$ induces a same-trajectory auxiliary gradient $g'_{d\to j}$, computed as in Eq. (6). For each parameter tensor, we select the remaining valid peer with the most negative cosine, project against it, and recompute the cosines; processing stops when no valid negative cosine remains. These projections use the norm-constrained form of Eq. (9) with a shared floor relative to the original target tensor:

$$\|g_d^{(q)}\|_2 \geq m\|g_d^{(0)}\|_2, \qquad q = 1, \ldots, K-1, \tag{24}$$

where $g_d^{(0)}$ is the original target tensor and $g_d^{(q)}$ is the tensor after q projections. Once processing is complete, the resulting tensors form $\widetilde{g}_d$. The floor is measured against the original tensor throughout the sequence. Applying a fresh local retention ratio m independently at every projection would provide only a product bound and would not implement this shared constraint. In addition, a later projection can change alignment with a previously processed peer; sequential projection alone does not establish simultaneous non-conflict with every peer.

For each conflicting peer, the implementation computes the coefficient from the remaining squared-norm budget. Write $a = \|g_d^{(q-1)}\|_2^2$, $b = m^2\|g_d^{(0)}\|_2^2$, and $u = \langle g_d^{(q-1)}, g'\rangle^2/\|g'\|_2^2$. It uses $\alpha = 1$ if $a - u \geq b$; otherwise, $\alpha = 1 - \sqrt{1 - \mathrm{clip}((a-b)/u, 0, 1)}$. Thus, a tensor that has exhausted its shared budget receives no further projection. These comparisons use numerical tolerance 10^{-12}.

A.3 Complete Training Procedure

Algorithm 1 combines teacher scoring, token modulation, and all-peer gradient projection within one optimizer step, using the domain-mean gradient convention of the main text. All target and auxiliary gradients are evaluated at the same student parameters on stored trajectories; only projected target gradients enter the update. Batch normalization is detailed below, and archived projection configurations are specified in Appendix B.3.

Token operations are elementwise over the valid responses in $\mathcal{B}_d$; stored scores and masks are restricted to that domain when used. MaskedLoss computes the loss in Eqs. (18) and (19) using valid response tokens. TensorwiseProject applies Eq. (9) separately to each parameter tensor. ProjectPeersSharedFloor denotes the sequential multi-peer procedure detailed in Appendix A.2, subject to the shared norm floor in Eq. (24). Auxiliary gradients are discarded after projection and never merged into g_{update}.

Domain means and global-batch normalization. Equation (19) and the algorithm define each gradient from a domain mean, and the algorithm averages the projected gradients across domains as in Eq. (11). In the implementation path supplied with full-batch normalization metadata, masking or slicing to domain d retains the denominator $|\mathcal{B}|$. After microbatch accumulation and data-parallel averaging, its loss contribution is therefore $\mathcal{L}_d^{\text{batch}} = w_d\mathcal{L}_d^{\text{agr}}$, where $w_d = |\mathcal{B}_d|/|\mathcal{B}|$; auxiliary losses on the same responses receive the same factor. Scaling both gradients by $w_d > 0$ preserves the projection cosines and retention coefficients in exact arithmetic. To express this implementation using the algorithm's domain-mean gradients, replace its accumulation by $g_{\text{update}} \leftarrow g_{\text{update}} + w_d\widetilde{g}_d$. Balanced batches give $w_d = 1/K$, recovering the equal-weight average in Eq. (11); shuffled batches use their observed domain fractions. For imbalanced batches, these weights differ from $1/K$ before global clipping and AdamW, so the two weighting conventions are not assumed to produce identical optimizer steps.

Projection implementation. The paired-domain implementation projects each trainable FSDP parameter tensor separately. Gradient snapshots are stored in BF16, while dot products and squared norms are computed in FP64 and reduced across the data-parallel group. Shards of the same tensor therefore use a common cosine and projection coefficient. Tensors with negligible gradient norms are left unchanged; the numerical stability constant is 10^{-12}. With multiple peers, each tensor selects the most negative cosine among its unprocessed peers and recomputes these cosines after

Algorithm 1 One optimizer step of ROAD MOPD

Require: Student parameters θ, optimizer state $\mathcal{S}$, rollout policy π_{old}
Require: Frozen teachers $\{\pi_{T_d}\}_{d=1}^{K}$, frozen reference π_{ref}, labelled prompts $\mathcal{X}$
Require: Modulation parameters $(\tau, \rho_{\text{g}}, \lambda_{\min}, \lambda_{\max})$, retention m, clip threshold γ
Ensure: Updated student parameters θ and optimizer state $\mathcal{S}$

$\mathcal{B}, \ell^{\text{old}}, M \leftarrow \textsc{Rollout}(\pi_{\text{old}}, \mathcal{X})$
$\ell^{1:K}, \ell^{\text{ref}} \leftarrow \textsc{ScoreSameTrajectories}(\{\pi_{T_d}\}_{d=1}^{K}, \pi_{\text{ref}}, \mathcal{B})$
$g_{\text{update}} \leftarrow 0$
for $d = 1, \dots, K$ **do**
  $\mathcal{B}_d \leftarrow \{i \in \mathcal{B} : \text{domain}(i) = d\}$; $\mathcal{P}_d \leftarrow \{1, \dots, K\} \setminus \{d\}$
  $\ell^{\theta} \leftarrow \textsc{StudentLogProbs}(\pi_\theta, \mathcal{B}_d)$; $r \leftarrow \exp(\ell^{\theta} - \ell^{\text{old}})$
  $\bar{s}_{d,i,t} \leftarrow \frac{1}{K-1} \sum_{j \in \mathcal{P}_d} \tanh\big((\ell^{d}_{i,t} - \ell^{\text{ref}}_{i,t})/\tau\big) \tanh\big((\ell^{j}_{i,t} - \ell^{\text{ref}}_{i,t})/\tau\big)$
  $\lambda_{d,i,t} \leftarrow \text{clip}(1 + \rho_{\text{g}} \bar{s}_{d,i,t}, \lambda_{\min}, \lambda_{\max})$
  $A^{\text{agr}}_{d,i,t} \leftarrow \text{sg}\big[\text{clip}\big(\lambda_{d,i,t}(\ell^{d}_{i,t} - \ell^{\text{ref}}_{i,t}) - (\ell^{\theta}_{i,t} - \ell^{\text{ref}}_{i,t}), -10, 10\big)\big]$
  $\mathcal{L}^{\text{agr}}_{d} \leftarrow \textsc{MaskedLoss}(r, A^{\text{agr}}_{d}, M_d)$
  $g_d \leftarrow \nabla_\theta \mathcal{L}^{\text{agr}}_{d}$
  for $j \in \mathcal{P}_d$ **do**
    $A^{\text{OPD}}_{d \to j,i,t} \leftarrow \text{sg}[\text{clip}(\ell^{j}_{i,t} - \ell^{\theta}_{i,t}, -10, 10)]$
    $\mathcal{L}'_{d \to j} \leftarrow \textsc{MaskedLoss}(r, A^{\text{OPD}}_{d \to j}, M_d)$
    $g'_{d \to j} \leftarrow \nabla_\theta \mathcal{L}'_{d \to j}$
  end for
  if $K = 2$ **then**
    $\widetilde{g}_d \leftarrow \textsc{TensorwiseProject}(g_d, g'_{d \to j}, m)$, where $\mathcal{P}_d = \{j\}$
  else
    $\widetilde{g}_d \leftarrow \textsc{ProjectPeersSharedFloor}(g_d, \{g'_{d \to j}\}_{j \in \mathcal{P}_d}, m)$
  end if
  $g_{\text{update}} \leftarrow g_{\text{update}} + \frac{1}{K} \widetilde{g}_d$
end for
$g_{\text{update}} \leftarrow \textsc{GlobalNormClip}(g_{\text{update}}, \gamma)$
$(\theta, \mathcal{S}) \leftarrow \textsc{AdamWStep}(\theta, \mathcal{S}, g_{\text{update}})$
return $\theta, \mathcal{S}$

each projection. Ties are broken by the configured peer order. Peers with negligible norms are skipped, and processing stops when no valid negative cosine remains. A selected peer is removed from the candidate set, even if the norm budget forces its coefficient to zero; previously selected peers are not revisited.

B Experimental Setup and Reproducibility

B.1 Models and Teacher–Student Configurations

Section 4.1 introduces the student and domain teachers. The specific Math and Code teacher checkpoints are Qwen3-4B-Non-Thinking-RL-Math-Step500 and Qwen3-4B-Non-Thinking-RL-Code-Step300, respectively (Yang et al., 2026b). The medical teacher is II-Medical-8B, a reinforcement-learning (RL) checkpoint based on Qwen3-8B (Intelligent Internet, 2025). The student undergoes full-parameter training; all teachers and the shared reference remain frozen and are used only for scoring.

The models use compatible Qwen3 tokenizers and vocabularies, allowing the scorers to evaluate identical response token IDs at identical prefixes. Scoring checks require matching token IDs and log-probability tensor shapes.

B.2 Training Data and Prompt Construction

Each paired training set contains 10,000 prompts per domain, for 20,000 prompts in total, and is randomly shuffled before training. For Math, we filter DeepMath (He et al., 2025) to samples with difficulty level ≥ 6, yielding a pool of 57K candidates, and sample 10,000 training prompts from this pool. For Code, we sample 10,000 prompts from Eurus-RL-Code (Cui et al., 2025). The Medical pool contains 10,000 prompts from four sources, as detailed in Table 5, including 4,000 sampled

from the MedMCQA training split (Pal et al., 2022). The MedMCQA validation split is used only for evaluation, not for constructing the training pool.

Table 5: **Composition of the training prompt pools.** Counts denote sampled training prompts. Each paired setting uses the two corresponding domain pools.

Domain	Source	Sampled prompts
Math	DeepMath (He et al., 2025), difficulty level ≥ 6 (57K candidates)	10,000
Code	Eurus-RL-Code (Cui et al., 2025)	10,000
Medical	MedMCQA training split (Pal et al., 2022)	4,000
	MedReason (Wu et al., 2025)	3,000
	Medical-R1-Distill-Data (FreedomIntelligence, 2025; Chen et al., 2024)	2,000
	m23k (Huang et al., 2025)	1,000

Domain labels select the routed teacher; only the prompt is supplied as the model input. Ground-truth answers are not used as outcome rewards in the distillation loss. Medical prompts request step-by-step reasoning with the final answer enclosed in `\boxed{}`. Thinking mode is disabled through the chat template, although generated responses can still contain intermediate reasoning.

B.3 Training Hyperparameters and Infrastructure

Table 6 records the shared optimization and rollout settings of the archived ROAD MOPD runs. The implementation uses verl with an FSDP student and vLLM for rollout generation and teacher scoring. Baseline-specific update schedules are described in Appendix B.4.

Table 6: **Shared optimization and rollout settings of ROAD MOPD.** Each rollout batch produces one optimizer update.

Hyperparameter	Value
Training / PPO minibatch size	1,008 / 1,008 prompts
Rollout batches (paired / three-domain)	58 / 116
PPO epochs per rollout batch	1
Responses per prompt	1
Rollout temperature / top-p	1.0 / 1.0
Maximum prompt / response length	2,048 / 16,384 tokens
Optimizer	AdamW
Learning rate / schedule	10^{-5} / constant, no warm-up
Adam betas / weight decay	$(0.9, 0.999)$ / 0.01
Global gradient norm clipping	1.0, after merging domain gradients
Advantage clipping	$[-10, 10]$, full method and all ablations
PPO ratio-clipping parameter	0.2
Distillation estimator	Sampled-token k_1, policy-gradient surrogate
Task reward / separate trainer KL reward	Disabled / disabled
Checkpoint interval	29 optimizer steps
In-training evaluation	Disabled; offline checkpoint evaluation

The shared modulation parameters are $\tau = 1$, $\rho_{\mathrm{g}} = 0.5$, and $[\lambda_{\min}, \lambda_{\max}] = [0.5, 1.5]$, with 10^{-12} for numerical stability. Paired-domain training uses bidirectional projection with $m = 0.6$. Because agreement lies in $[-1, 1]$, the unbounded mapping $1 + 0.5 s_t$ already lies in $[0.5, 1.5]$; clipping does not further restrict this mapping in exact arithmetic.

The paired- and three-domain configurations use one node with eight NVIDIA B200 GPUs. Six GPUs host the FSDP student and three rollout replicas with tensor-parallel size two. The remaining two GPUs host teacher scoring. In paired settings, one 4B teacher shares a GPU with the frozen reference; in three-domain training, the Math teacher, Code teacher, and reference share one GPU, while the Medical teacher occupies the other. Gradient checkpointing, padding removal, and dynamic batching are enabled. The three-domain ROAD MOPD sampler forms balanced batches of 336 prompts per domain.

The audited ExOPD, Open-MOPD, and CaMOPD runs use 1,008 prompts per rollout batch, with 58 batches in paired settings and 116 in three-domain settings. ExOPD and CaMOPD perform one optimizer update per batch. Open-MOPD divides that batch into four sequential minibatches of 252 prompts, giving 232 and 464 optimizer updates, respectively, while retaining one response per prompt. Thus, matching rollout budgets does not imply matching optimizer-update counts; CaMOPD also changes the allocation of prompts across domains through its alternating schedule.

Student sizes, reference models, and Single OPD. For both ROAD MOPD and ExOPD, the reference is a frozen copy of the student's initialization: Qwen3-4B in 4B experiments and Qwen3-1.7B in 1.7B experiments. For Qwen3-1.7B, Table 1(a)–(b) reports multi-domain checkpoints at step 58; Student denotes the original, undistilled model. Open-MOPD uses the sampled-token k_1 estimator. Within each student size, Math, Code, and Medical Single OPD share the same training configuration and differ only in their domain-specific datasets. All three Qwen3-1.7B Single OPD runs start from the original model and train for 50 steps, evaluated with Avg@32, Avg@4, and Avg@1, respectively. Qwen3-4B Single OPD uses the matching in-domain checkpoints in Table 8.

B.4 BASELINE IMPLEMENTATIONS

We implement the baselines within the same MyOPD/verl training stack, using the corresponding Qwen3-4B or Qwen3-1.7B initialization and domain teachers introduced in Section 4.1. Teacher scoring uses the sampled tokens of student-generated responses. The baseline paths use the sampled-token k_1 reverse-KL estimator and a clipped policy-gradient surrogate with clipping parameter 0.2; task rewards, entropy regularization, and a separate trainer KL reward are disabled. The method-specific implementations are as follows.

Vanilla MOPD. Each prompt is routed to its domain teacher (Ma et al., 2026). The token advantage $\log \pi_{T_d}(y_t \mid h_t) - \log \pi_\theta(y_t \mid h_t)$ is clipped to $[-10, 10]$ and detached, without cross-teacher modulation or gradient projection. The archived Math–Code and Code–Medical runs each use 58 optimizer updates, while the three-domain run uses 116.

ExOPD. We use a fixed extrapolation coefficient of 1.25. The reference is the frozen student initialization, Qwen3-4B or Qwen3-1.7B according to student size (Yang et al., 2026b). For each sampled token, the detached advantage is $1.25(\log \pi_{T_d} - \log \pi_{\text{ref}}) - (\log \pi_\theta - \log \pi_{\text{ref}})$, with all log probabilities computed at the same response prefix. The cross-teacher gate, gradient projection, and domain-balancing modules are disabled for this baseline.

CaMOPD. The paired-domain implementation alternates homogeneous domain batches and selects examples using the rollout-time gap $\Delta_t = \log \pi_{T_d}(y_t \mid h_t) - \log \pi_{\text{old}}(y_t \mid h_t)$ (Chen et al., 2026). Recovery examples are scored by the mean of $|\Delta_t|$, while preservation examples use the mean of $\max(\Delta_t, 0)$. Within each branch, examples are ranked by this score and the smallest prefix covering 80% of the total score mass is retained. For Medical–Math and Medical–Code, Medical is the recovery branch: the update ratio is 3:1, and selected-example advantage scales are 2.0 for Medical and 1.0 for the other domain. Math–Code uses Math as recovery and Code as preservation, a 1:1 update ratio, and unit scales for both branches. The three-domain adaptation treats domains as peer branches, using the positive-gap score, the same 0.8 score-mass threshold, and unit selected-example weights. We sample Medical, Math, and Code data in a 3:1:1 ratio, prioritizing the weaker Medical domain in line with CaMOPD's capability-recovery objective.

Open-MOPD. We adapt its balancing and reward-refresh mechanisms to the sampled-token OPD path (Gao et al., 2026). At rollout time, a domain's weight is proportional to its uniform target share divided by its observed response-token share, multiplied by its mean absolute teacher–student gap relative to the mean gap across domains. This gap factor uses exponent 1.0 and is clipped to $[0.05, 20]$. The weights are normalized to have token-weighted mean one and remain fixed within the rollout batch. Dynamic allocation is therefore implemented through domain loss weights. Each batch is split into four sequential optimizer minibatches. On each minibatch, the OPD advantage is recomputed using the current student's log probabilities and cached teacher scores, providing reward refresh.

PCGrad. For the two-domain comparison (Yu et al., 2020), separate backward passes mask the shared minibatch to each domain's own responses. For each trainable parameter tensor, the implementation computes the two gradients' dot product and squared norms across FSDP shards. If the dot product is negative, both gradients are projected against the original opposing gradient; the two corrected gradients are then summed. This symmetric two-task calculation requires no random task ordering. Gradient snapshots use BF16 and dot products and squared norms use FP64. The comparison applies global gradient clipping after merging, followed by one optimizer step. These gradients are formed on their respective domains' responses, following the independent-domain comparison in Section 4.3; same-trajectory auxiliary gradients are not used for this projection. The archived PCGrad runs whose benchmark scores appear in Table 7 also enable reward-agreement modulation and are marked with a dagger in that table.

B.5 Evaluation Protocols

We evaluate 11 benchmarks across Code (HumanEval+, MBPP+, and LiveCodeBench) (Liu et al., 2023; Jain et al., 2024), Math (AIME 2024/2025 and HMMT February/November 2025) (MAA, 2024; 2025; Harvard–MIT Mathematics Tournament, 2025a;b), and Medical (MedQA, MedXpertQA, MedMCQA, and SuperGPQA-Med) (Jin et al., 2020; Zuo et al., 2025; Pal et al., 2022; M-A-P Team et al., 2025).

Generation setup. Offline evaluation uses vLLM with temperature 1.0, top-p 1.0, and a maximum of 16,384 generated tokens per completion. Chat prompts disable thinking mode via `enable_thinking=False`.

Mathematics. Each of the four Math test sets contains 30 questions, with 32 sampled completions per question. We append the instruction "Please reason step by step, and put your final answer within `\boxed{}`." The evaluator extracts the last boxed answer and uses `math-verify` to compare it with the reference answer; a missing or malformed boxed answer is incorrect. The reported Avg@32 is the mean correctness over completions, averaged over questions:

$$\text{Avg@32} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{32}\sum_{r=1}^{32}\mathbf{1}[\hat{a}_{i,r} = a_i]. \tag{25}$$

Here equality denotes equivalence under the answer verifier. Avg@32 equals the pass@1 estimate computed from the same 32 samples. This differs from pass@32, which counts a question as solved when at least one of its 32 completions is correct.

Code. All three Code benchmarks use four completions per problem. We report Avg@4, the fraction of passing completions averaged over problems, which equals the pass@1 estimate from the same four samples. HumanEval+ contains 164 problems and MBPP+ contains 378. A completion must pass both the original and additional EvalPlus tests to count as correct. Prompts request a Python solution in a fenced code block. Our local evaluation protocol (v3) prioritizes Python or untagged Markdown code blocks, selects the candidate with the most nonempty lines, and uses tree-sitter to retain the target entry point and its dependencies. Generation continues until EOS or the token limit, without stopping at the first closing code fence. EvalPlus batches up to 256 generation requests across problems and sets each test's execution limit to $\max(5\,\text{s}, 6t_{\text{ref}})$, where t_{ref} is the reference solution's runtime for that test.

LiveCodeBench. We use the v6 code-generation split: 175 problems with contest dates from January 4 to April 6, 2025, giving 700 completions in total. The local loader selects `test6.jsonl`, with no additional date filtering. The runner uses the Qwen3-4B non-thinking prompt template and extracts the final fenced code block. Each completion is checked against the problem's public and private tests with a 60-second per-test timeout. Compilation failures, runtime errors, timeouts, and failed tests count as incorrect completions.

Medical. The evaluation uses one completion per question and reports Avg@1 multiple-choice accuracy. It covers the four-option MedQA-USMLE test set (1,273 questions), MedXpertQA Text test set (2,450; ten options), MedMCQA validation set (4,183; four options), and the SuperGPQA subset selected by `discipline=Medicine` (2,755; ten options). Each prompt includes the question

and labeled options, and requests step-by-step reasoning with the final answer letter in `\boxed{}`. The parser first checks the last boxed letter, then falls back to an explicit answer phrase such as "Answer: B" or a terminal "(B)". Letters are normalized to uppercase and must belong to the question's available options. Exact match to the reference letter determines correctness; unparseable answers are incorrect.

Aggregation. All scores are reported as percentages. Each benchmark weights its questions equally, and domain averages are arithmetic means over the benchmarks within that domain, irrespective of their different numbers of questions. The 11-task average in Table 8 weights all 11 benchmarks equally; it is distinct from averaging the three domain means, because Code contains three benchmarks and the other domains contain four.

Teacher-disagreement and gradient-conflict probe. For the fixed-checkpoint probe in Figure 3 (left), we compute $S_- = \text{mean}_t[\max(-s_t, 0)]$ over all valid response tokens pooled across each four-trajectory batch, using Eq. (4) with $\tau = 1$. We partition batches into five within-domain quintiles, with 50 batches per quintile. A raw conflict is a negative cosine between the full-parameter routed- and auxiliary-teacher OPD gradients on the same batch, without token modulation, projection, or optimizer updates. This five-bin analysis is exploratory; shading connects pointwise 95% Wilson interval limits. The probe is separate from the token-visualization samples in Appendix C.3.

C Additional Experimental Results

C.1 Detailed Benchmark Results

Table 7 provides available benchmark breakdowns for the ablations and PCGrad comparison.

Table 7: **Per-benchmark breakdown of the ablations and PCGrad comparison.** Avg. is the benchmark mean; rounding may cause small differences from the mean of displayed scores. Scores are percentages (higher is better).

(a) Math × Medical

	Math					Medical				
Method	AIME24	AIME25	HMMT 25 Feb.	HMMT 25 Nov.	Avg.	MedQA	Med Xpert QA	Med-MCQA	GPQA Med	Avg.
w/o Credit Assign.	59.69	53.54	32.71	37.19	45.78	76.98	15.88	62.08	36.01	47.74
w/o Grad. Protection	61.46	54.48	33.75	39.06	47.19	76.04	15.39	63.28	37.71	48.11
w/o Adaptive Retention	58.13	55.42	33.02	36.56	45.78	77.53	15.59	63.61	36.84	48.40
PCGrad†	62.60	53.54	33.75	37.50	46.85	77.93	14.65	62.54	36.99	48.03
ROAD MOPD	63.85	53.96	33.65	38.96	47.60	77.06	15.67	62.83	37.35	48.23

(b) Code × Medical

	Code				Medical				
Method	Human Eval+	MBPP+	LCB	Avg.	MedQA	Med Xpert QA	Med-MCQA	GPQA Med	Avg.
w/o Credit Assign.	84.15	69.05	25.43	59.54	76.20	16.08	61.77	35.79	47.46
w/o Grad. Protection	87.65	71.96	29.43	63.01	75.18	14.16	61.53	36.62	46.87
w/o Adaptive Retention	86.74	73.68	28.14	62.85	76.12	14.08	62.25	35.68	47.03
PCGrad†	84.45	70.11	28.14	60.90	76.90	15.71	62.49	35.17	47.57
ROAD MOPD	87.65	72.69	28.71	63.02	75.73	15.96	62.97	36.33	47.75

† The archived PCGrad runs also enable reward-agreement modulation.

Single-domain reference results. Table 8 lists the available single-domain checkpoints across all 11 benchmarks, complementing Figure 1.

Table 8: **Cross-domain evaluation of single-domain training checkpoints.** Each Qwen3-4B single-domain model is trained with OPD on 10,000 in-domain prompts for 3 epochs and evaluated across all 11 benchmarks. The best and second-best results among the three checkpoints are shown in bold and underlined, respectively.

Benchmark	Teacher	Student	Math-only	Medical-only	Code-only
Code					
HumanEval+	85.98	76.68	81.40	81.10	**84.30**
MBPP+	79.63	65.08	69.71	**72.95**	70.44
LCB v6	27.00	17.14	22.00	22.14	**28.00**
Avg.	64.20	52.97	57.70	58.73	**60.91**
Math					
AIME24	57.71	21.46	58.13	**59.38**	43.75
AIME25	55.00	21.77	**51.67**	46.15	37.29
HMMT25 Feb.	32.29	10.00	**31.56**	30.31	20.00
HMMT25 Nov.	39.27	8.23	35.73	**36.56**	24.79
Avg.	46.07	15.37	**44.27**	43.10	31.46
Medical					
MedQA	87.90	64.73	70.86	**77.06**	67.71
MedXpertQA	25.71	13.80	14.90	**15.31**	14.00
MedMCQA	71.38	55.63	60.17	**62.44**	57.33
SuperGPQA-Med	47.19	29.87	34.70	**36.81**	33.32
Avg.	58.05	41.01	45.16	**47.90**	43.09
11-Task Avg.	55.37	34.94	48.26	**49.11**	43.72

C.2 Full Results for Low-Conflict and Three-Domain Settings

Tables 9 and 10 provide the full per-benchmark Qwen3-4B results underlying the summaries in Tables 2 and 3, respectively. Domain averages weight benchmarks equally. Single OPD uses the matching in-domain checkpoints from Table 8.

Table 9: **Full results for low-conflict Math × Code.** Per-benchmark Qwen3-4B scores underlying Table 2. Scores are percentages (higher is better); Avg. weights benchmarks equally within each domain. Bold and underlining mark the best and second-best multi-domain methods, respectively.

Benchmark	Student	Single OPD	Vanilla MOPD	ExOPD	Open-MOPD	CaMOPD	Ours
Math							
AIME24	21.46	58.13	60.21	**63.33**	56.77	61.04	60.73
AIME25	21.77	51.67	55.21	**57.71**	53.33	54.17	57.40
HMMT25 Feb.	10.00	31.56	31.98	**33.85**	30.52	31.87	33.33
HMMT25 Nov.	8.23	35.73	37.08	38.23	36.67	38.12	**39.48**
Avg.	15.37	44.27	46.12	**48.28**	44.32	46.30	47.73
Code							
HumanEval+	76.68	84.30	85.06	86.59	83.08	84.91	**86.89**
MBPP+	65.08	70.44	70.37	71.16	70.37	70.50	**72.75**
LCB v6	17.14	28.00	28.29	27.71	26.00	25.71	**29.57**
Avg.	52.97	60.91	61.24	61.82	59.82	60.37	**63.07**

Table 10: **Full results for generalization to three domains.** Per-benchmark Qwen3-4B scores underlying Table 3. Scores are percentages (higher is better); Avg. weights benchmarks equally within each domain. Single OPD uses the matching in-domain checkpoint from Table 8. The best and second-best results among the compared multi-domain methods are shown in bold and underlined, respectively.

Benchmark	Student	Single OPD	Vanilla MOPD	ExOPD	Open-MOPD	CaMOPD	Ours
Code							
HumanEval+	76.68	84.30	<u>86.28</u>	84.76	85.52	84.45	**86.59**
MBPP+	65.08	70.44	70.63	<u>71.23</u>	70.44	70.90	**72.95**
LCB	17.14	28.00	26.86	<u>28.57</u>	25.71	27.57	**29.29**
Avg.	52.97	60.91	61.26	<u>61.52</u>	60.56	60.97	**62.94**
Math							
AIME24	21.46	58.13	<u>61.04</u>	59.90	55.73	59.48	**61.88**
AIME25	21.77	51.67	**57.50**	<u>55.31</u>	54.17	54.90	54.48
HMMT25 Feb.	10.00	31.56	31.67	**34.90**	31.25	31.98	<u>33.54</u>
HMMT25 Nov.	8.23	35.73	37.71	<u>38.23</u>	35.10	37.50	**39.27**
Avg.	15.37	44.27	46.98	<u>47.09</u>	44.06	45.96	**47.29**
Medical							
MedQA	64.73	77.06	64.49	73.29	60.17	<u>76.75</u>	**77.61**
MedXpertQA	13.80	15.31	12.78	14.57	12.78	**16.69**	<u>15.96</u>
MedMCQA	55.63	62.44	57.14	60.94	55.06	<u>62.54</u>	**63.11**
SuperGPQA-Med	29.87	36.81	29.69	33.68	27.30	<u>36.81</u>	**37.10**
Avg.	41.01	47.90	41.03	45.62	38.83	<u>48.20</u>	**48.45**

C.3 Extended Token-Level Visualizations

Figure 5 expands the exact trajectories shown in the right panels of Figure 3 to 256-token contexts. It includes the original prompts and marks the 52-token main-text windows, preserving every displayed token's original coefficient. These illustrative windows were selected from 20 sampled responses per prompt to show both attenuation and amplification near an answer mention; they do not estimate the overall coefficient distribution. The visualization samples use the token-modulation-only checkpoint at step 58, temperature/top-p $1/1$, and a 2,048-token generation cap. Both selected trajectories reached that cap.

(a) Mathematics

AIME2024_0 | rollout 7 | response tokens [1792, 2048)

Every morning Aya goes for a 9-kilometer-long walk and stops at a coffee shop afterwards. When she walks at a constant speed of s kilometers per hour, the walk takes her 4 hours, including t minutes spent in the coffee shop. When she walks s+2 kilometers per hour, the walk takes her 2 hours and 24 minutes, including t minutes spent in the coffee shop. Suppose Aya walks at s+1/2 kilometers per hour. Find the number of minutes the walk takes her, including the t minutes spent in the coffee shop.
Please reason step by step, and put your final answer within \boxed{}.

Scored response context | outlined tokens: Figure 3 [1988, 2040)

says : " Find the number of minutes the walk takes her , including the t minutes spent in the coffee shop ." So , when she walks at s + · 1 / 2 km /h , the time spent walking is · 9 /(s + · 1 / 2) hours . Then add t minutes . But to convert the entire thing to minutes , we need to convert the walking time to minutes and add t . So : · Walking time : · 9 / (s + · 1 / 2) hours . Convert to minutes : · 9 /(s + · 1 / 2) * · 6 0 minutes . Then add t minutes , which is · 2 4 . But s + · 1 / 2 is · 2 . 5 + · 0 . 5 = · 3 , as above . So · 9 / · 3 = · 3 hours . · 3 hours is · 1 8 0 minutes . · 1 8 0 + · 2 4 = · 2 0 4 minutes . So that 's · 2 0 4 minutes total . But let me just check again , maybe I made a mistake with the equations ? Wait , perhaps I should make sure that the steps are correct . Alternatively , let 's see if there 's another way . Let me check if the value of s is correct . From the initial

Earlier context omitted. Generation limit reached at token 2048.

(b) Medical

MedQA_test_0 | rollout 2 | response tokens [869, 1125)

A junior orthopaedic surgery resident is completing a carpal tunnel repair with the department chairman as the attending physician. During the case, the resident inadvertently cuts a flexor tendon. The tendon is repaired without complication. The attending tells the resident that the patient will do fine, and there is no need to report this minor complication that will not harm the patient, as he does not want to make the patient worry unnecessarily. He tells the resident to leave this complication out of the operative report. Which of the following is the correct next action for the resident to take?

Options:
A. Disclose the error to the patient and put it in the operative report
B. Tell the attending that he cannot fail to disclose this mistake
C. Report the physician to the ethics committee
D. Refuse to dictate the operative report

Reason briefly, then state the final answer as a single option letter.

Scored response context | outlined tokens: Figure 3 [971, 1023)

the patient . · Alternatively , maybe the resident should follow the attending 's orders if the attending has authority . However , this can get tricky . · Also , if the resident is not disclosing errors , they might be violating their own ethical training . · Wait , the key is that even minor errors should be reported . The attending 's reasoning is perhaps that it 's not harmful , but the mistake itself needs to be documented . So , perhaps the correct answer is A , because the resident should disclose the error to the patient , even if the attending says not to . Or perhaps the correct answer is B , the resident has to tell the attending that they can 't fail to disclose . · The question asks which is the correct next action . Maybe the answer is A ? · Wait , the operative report is part of the medical record . So if the resident is dict ating the operative report , perhaps they can 't omit the error . So if the attending says to not report it , but the resident is in a position to tell the truth , then perhaps the resident has to go against the attending . · Alternatively , maybe the attending is the one who has the authority to decide what is reported . But generally , medical records must be accurate . · Hmm . Maybe the answer is A , because the resident

Earlier context omitted. Response continues beyond this excerpt.

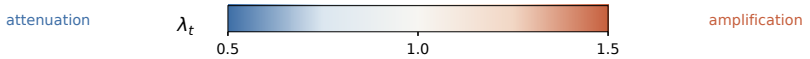

Figure 5: **Extended Math and Medical examples from the right panels of Figure 3.** Outlines identify the main-text windows within the longer contexts. Blue and orange denote attenuation and amplification on the unchanged scale $\lambda_t \in [0.5, 1.5]$. Token indices are zero-based, with half-open intervals. Omitted context and the generation limit are marked.

Additional fixed-window examples. Figures 6 and 7 show three further responses per prompt. We take the first three rollout IDs after excluding the response already shown in Figure 5: Math IDs 0, 1, and 2; Medical IDs 0, 1, and 3. In that order, we display the fixed 192-token windows $[0, 192)$, $[928, 1120)$, and $[1856, 2048)$. This selection does not inspect coefficients or answer correctness. Prompt summaries accompany the plots; the full prompts appear in Figure 5.



Figure 6: **Three additional Math response excerpts.** Panels show the beginning, middle, and end of three distinct sampled responses to `AIME2024_0`. Original token coefficients and the $[0.5, 1.5]$ color scale are preserved. A dot represents a whitespace-only token; omitted context and the generation cap are marked.



Figure 7: **Three additional Medical response excerpts.** Panels show the beginning, middle, and end of three distinct sampled responses to `MedQA_test_0`, using the same fixed-window rule and unchanged color scale as Figure 6. These are generated reasoning excerpts, not verified answer explanations.

C.4 Training Dynamics

We complement the endpoint evaluations with complete training traces from three archived ROAD MOPD runs: Math × Medical, Code × Medical, and Math × Code. Each run has 58 updates with 1,008 prompts per update, one PPO epoch, and a 16,384-token response cap. Both domains use reward-agreement modulation and tensor-wise adaptive projection with norm-retention floor $m = 0.6$. All plots retain every logged step without smoothing.

Loss and generation behavior. Figure 8 shows the logged signed policy surrogate, per-domain response lengths, and the fraction of responses reaching the cap. The logger captures only the task-0 primary loss contribution: Math in Math × Medical, and Code in the other two runs. This quantity can be negative and is neither the full two-domain objective nor a nonnegative KL estimate. It decreases sharply early in training, while generation lengths and cap rates continue to vary. These single-run traces describe training behavior; they do not establish convergence of both domain objectives or comparative test accuracy.

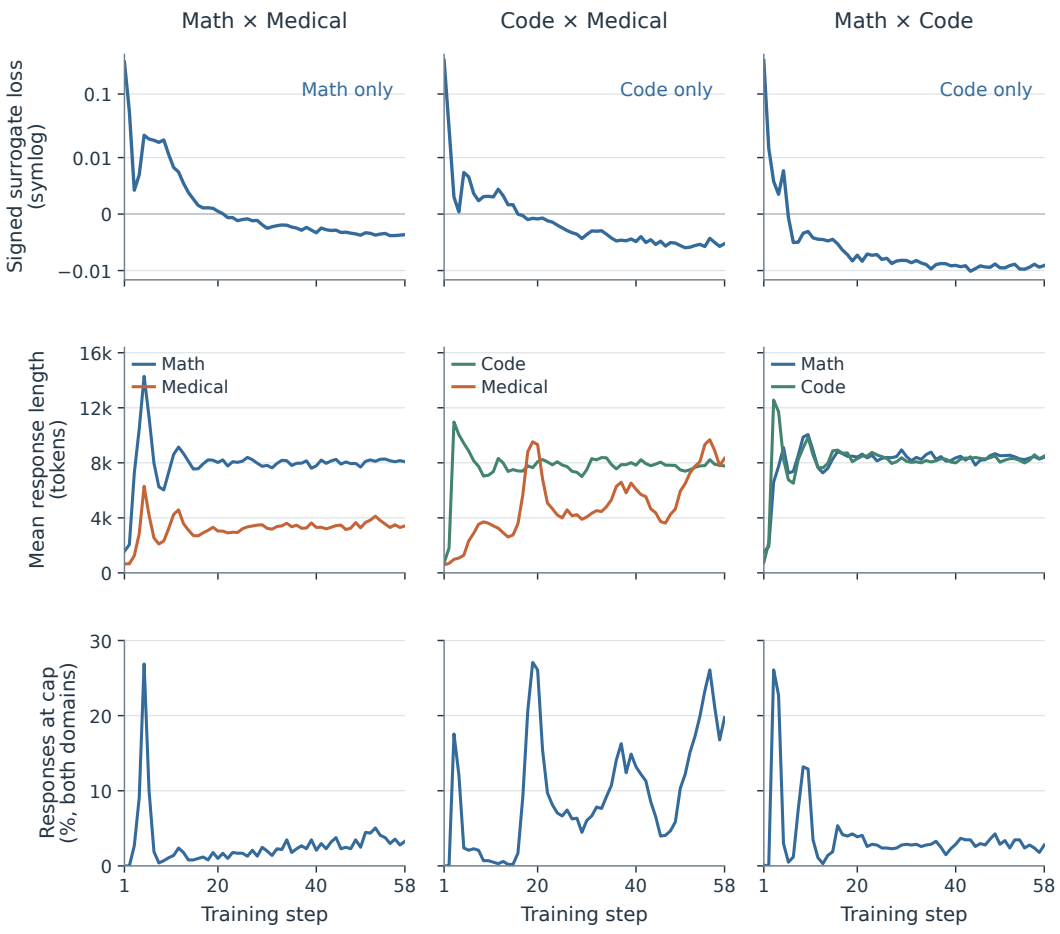

Figure 8: **Training loss and generation behavior across three domain pairings.** Columns show individual ROAD MOPD runs. Top: the logged task-0 surrogate contribution on a symmetric-log scale (linear within ± 0.01); each panel identifies the recorded domain. Middle: mean response length for each domain. Bottom: percentage of responses across both domains that reach 16,384 tokens. Curves connect all 58 logged updates; negative loss values and fluctuations are preserved.

Gradient-norm retention during projection. Figure 9 examines when the norm constraint changes the projection strength. For each primary domain, the logger computes two norm ratios from the same pre-projection gradients: the ratio under full projection and the ratio under the chosen adaptive strength. Each curve reports the minimum over conflicting parameter-tensor units at that update (or one if none conflict). These are analytical projection diagnostics before merging and global gradient clipping, rather than measurements of the final optimizer displacement.

The adaptive ratios stay at or above 0.6, within numerical precision, in all six domain traces. For example, in the Math branch of Math $\times$ Medical, the full-projection estimate falls below 0.6 in 40 of 58 updates and reaches a minimum of 0.253, whereas the adaptive minimum is 0.600. The Medical branch of that same run never crosses the floor, so the two curves coincide. This illustrates how the constraint limits norm loss when needed without changing full projection when its retained norm already satisfies the floor. The full-projection curve uses the gradients encountered by ROAD MOPD; it is not a separately trained baseline.

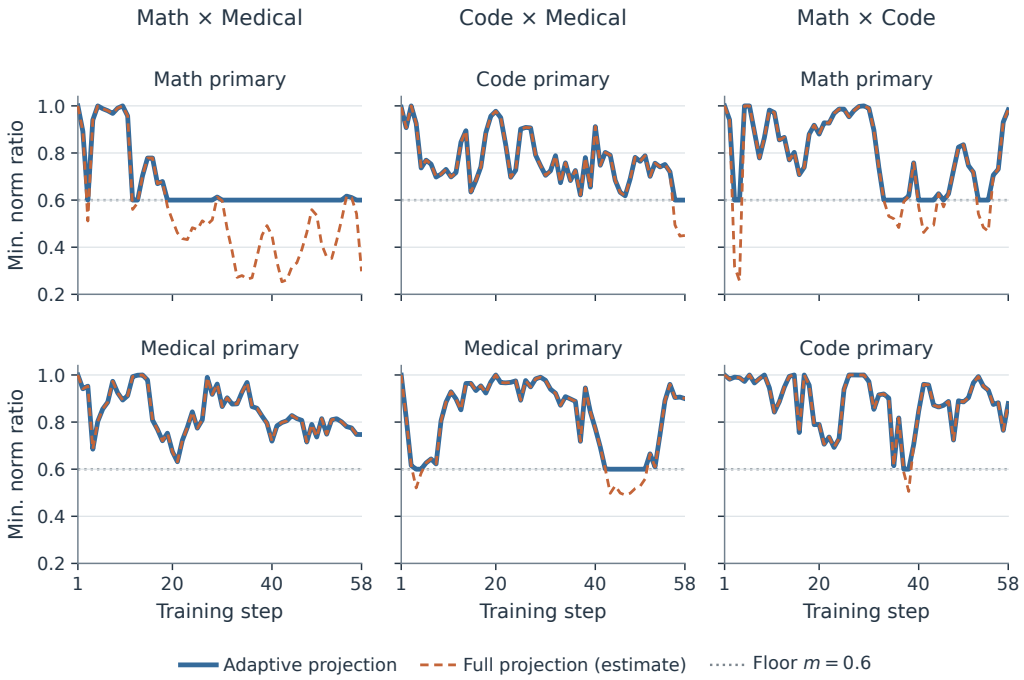

Figure 9: **Minimum norm retention over training.** Each panel identifies the primary domain whose gradient is projected; columns use the same runs as Figure 8. Blue shows the logged ratio under adaptive projection; dashed orange shows the analytical full-projection estimate on the same gradients. The dotted line marks $m = 0.6$. Ratios are minima over conflicting parameter-tensor units, before merged-gradient clipping. Overlapping curves indicate identical minimum retention. All logged steps are shown without smoothing.

D LIMITATIONS

Computational overhead. ROAD MOPD incurs additional training overhead compared with Vanilla MOPD because of auxiliary-teacher scoring and gradient processing. Reusing student trajectories avoids additional rollout generation. Future work will explore more efficient teacher evaluation and gradient correction to reduce this overhead.

Limited evaluation of domain scalability. Our experiments cover mathematics, code, and medicine, with joint training evaluated on at most three domains (Table 3). Future work will assess whether the observed gains and training stability extend to broader domain collections and larger teacher pools.