DOUBLEFREE: EXPLOITING EXPLORATION HISTORY FOR ROLLOUT-FREE MULTI-TASK LLMS TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-task post-training can develop domain specialists through independent reinforcement learning (RL) and consolidate their capabilities through multi-teacher on-policy distillation (MOPD). However, standard MOPD generates on-policy student responses and keeps multiple teachers online throughout consolidation, introducing substantial system costs. We view this pipeline as domain-wise exploration followed by cross-domain exploitation. Domain RL produces not only the final specialists, but also the trajectories collected throughout their training. Existing methods primarily use the former while leaving the latter underutilized. We introduce DoubleFree, which reuses these historical trajectories for rollout-free consolidation and precomputes teacher supervision for online-teacher-free student training. To understand why this works, we analyze when updates from historical trajectories can approximate on-policy distillation updates and how combining trajectories across exploration stages can better support an evolving student. DoubleFree requires only minimal changes to existing post-training frameworks. Across math, code, and instruction following on Qwen3-4B, Qwen3-8B, and SmolLM3-3B, DoubleFree achieves 2.99-5.75$\times$ consolidation speedups and reduces GPU-hours by 80.9-90.1%, while maintaining comparable multi-domain performance.

1 INTRODUCTION

Multi-task LLM post-training seeks to combine capabilities such as mathematical reasoning, code generation, and instruction following in a single model (Lambert et al., 2025; Qwen Team, 2025). An increasingly studied approach first develops domain specialists through independent reinforcement learning (RL), then consolidates their capabilities through multi-teacher on-policy distillation (MOPD) (Ma et al., 2026; Bai et al., 2026).

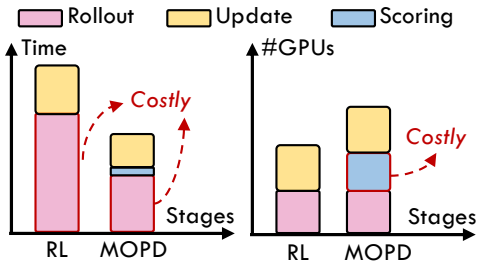


Figure 1: Two system costs in RL-then-MOPD: repeated response generation and GPU resources reserved for online multi-teacher serving.

This RL-then-MOPD pipeline incurs two major system costs (Figure 1). First, rollout is a major cost in both stages Gao et al. (2026b), and accounts for about 70% of per-batch time during consolidation. Yet consolidation generates new student responses despite the domain-specific experience already captured in RL trajectories. Second, student updates depend on online teacher scoring, typically requiring dedicated teacher GPUs throughout training. Specifically, teachers reserve 42.9% of GPUs, while remaining idle for over 90% of their allocated time (Section 2).

We revisit this pipeline through an *exploration-exploitation* view (Section 3.1): domain-wise exploration develops specialist capabilities, while shared exploitation transfers these capabilities into a single policy. This view reveals two outputs of domain-wise exploration available for shared exploitation: **final specialist policies** and **historical response trajectories**. Existing distillation methods exploit the former for supervision but largely discard the latter, instead generating additional responses from the current student, initial student, or final specialists (Ma et al., 2026; Wu

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

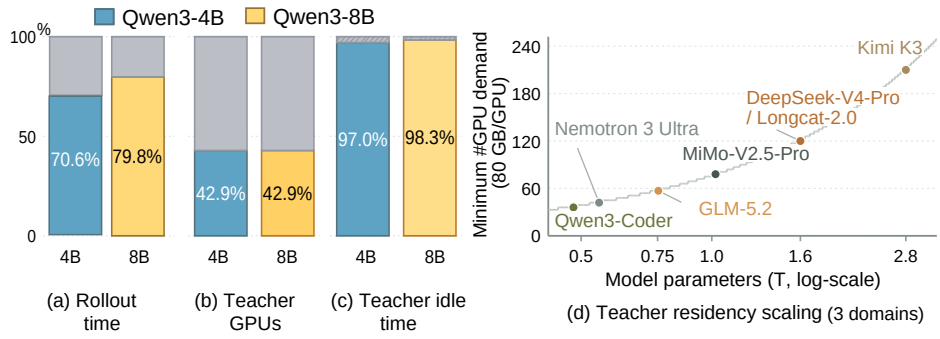


Figure 2: MOPD system costs. (a-c) Rollout time, teacher GPU allocation, and teacher idle time under two settings. (d) Minimum GPU demand for three resident domain teachers (BF16 weights).

et al., 2026; Kim & Rush, 2016). This presents the opportunity to exploit both outputs to improve the consolidation efficiency.

We introduce DoubleFree, which exploits both outputs of domain-wise exploration: it reuses historical trajectories for **rollout-free** consolidation and pre-scores them with final specialists for **online-teacher-free** student training. For policy distillation, the student learns from cached responses and teacher scores while recomputing its own log-probabilities. Together, trajectory reuse and teacher pre-scoring eliminate additional response generation and release teacher GPUs before student training. A key question is whether exploration trajectories can support an evolving shared student. Collected across different RL substages, these trajectories may jointly provide broader coverage of student behavior than responses from either the initial policy or final specialists alone. In Section 4, we bound the discrepancy between on-policy and reuse distillation updates in terms of distribution mismatch and characterize potential coverage benefits of historical policy mixtures.

Our experiments are conducted on NVIDIA H100-80GB GPUs, with DoubleFree implemented in verl (Sheng et al., 2024). Across math, code, and instruction following on Qwen3-4B, Qwen3-8B, and SmolLM3-3B, DoubleFree reduces consolidation GPU-hours by 80.9-90.1% and achieves 2.99-5.75× speedups over Standard MOPD across DoubleFree's SFT and MPD variants, while retaining comparable multi-domain performance. Our contributions are as follows:

- We formulate RL-then-MOPD as an explore-exploit framework and identify redundant response generation and persistent teacher GPU allocation as two system bottlenecks.
- DoubleFree combines exploration trajectory reuse with teacher pre-scoring for rollout-free consolidation and online-teacher-free student training. Our analysis further characterizes when such reuse can approximate on-policy updates and how trajectories across exploration stages can complement each other.
- Experiments across three models of different scales and families cover nine benchmarks in math, code, and instruction following. DoubleFree substantially reduces consolidation time and GPU-hours while maintaining comparable multi-domain accuracy.

2 MOPD System Bottlenecks

We profile Standard MOPD on Qwen3-series across three domains, with separate GPU pools for student training and teacher serving. Student/teacher GPU allocations are 4/3 for 4B and 8/6 for 8B model. Each mixed-domain batch is globally synchronized in our setting. Figure 2 summarizes rollout time, teacher allocation and idle time, and teacher GPU demand as model size increases.

Costly rollout across stages. Standard MOPD generates fresh responses from the current student for every update. As shown in Figure 2(a), rollout accounts for 70.6% of per-batch time on Qwen3-4B (55.02 of 77.95 seconds) and 79.8% on Qwen3-8B (76.06 of 95.27 seconds). This generation cost is paid again during consolidation even though domain RL has already produced substantial trajectories through domain-specific exploration. Offline alternatives move response generation by the initial student or final specialists ahead of training, but do not eliminate this additional cost. Such

redundant generation is particularly costly for long-response reasoning workloads. These observations raise a key research question: *Can existing exploration trajectories replace on-policy rollouts during consolidation while preserving distillation quality?*

Intermittent multi-teacher serving. Online teachers provide supervision for trajectories from their respective domains, and each student update proceeds only after scoring is complete across all domains. In a conventional resident-teacher deployment, domain specialists occupy dedicated GPUs throughout training, accounting for 42.9% of the total GPU allocation in both profiled settings. Yet request-level timings indicate that over 97% of allocated teacher GPU-time falls outside scoring calls in both settings (Figure 2(b-c)). Moreover, the resource requirement of teacher residency scales with model size. As shown in Figure 2(d), three domain teachers require at least 120 80 GB GPUs at 1.6T parameters per teacher and 210 at 2.8T for their BF16 weights alone. This raises a complementary question: *Can teacher supervision be delivered without continuously reserving substantial GPU resources for intermittent scoring?*

3 Method: DoubleFree

DoubleFree addresses the two bottlenecks by exploiting both outputs of domain-wise exploration: historical trajectories are reused for consolidation, while final specialists provide precomputed supervision (Figure 3). Trajectory reuse eliminates additional response generation, while fixed responses allow specialist supervision to be computed offline. Student training is therefore *rollout-free* and *online-teacher-free*.

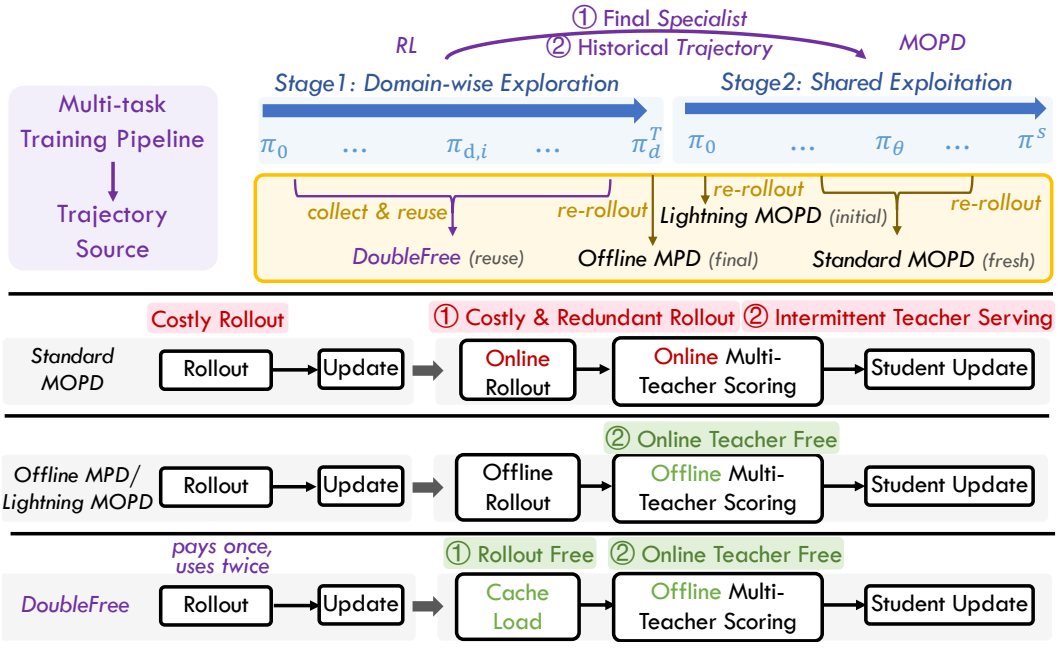


Figure 3: Overview of DoubleFree. Exploration stage produces both specialist policies and historical trajectories. DoubleFree reuses the trajectories and precomputes specialist scores, removing additional generation and online teacher residency from student training.

3.1 Explore-Exploit Formulation

Let π_0 be an initial shared policy and $\{\mathcal{D}_d, r_d\}_{d=1}^{M}$ a collection of domains, with prompt distribution $\mathcal{D}_d$ and verifiable reward r_d. The goal is a shared policy with strong aggregate performance,

$$\max_{\pi} J(\pi) = \sum_{d=1}^{M} w_d J_d(\pi), \qquad J_d(\pi) = \mathbb{E}_{x\sim\mathcal{D}_d,\, y\sim\pi(\cdot|x)}[r_d(x,y)], \tag{1}$$

where $w_d \geq 0$ and $\sum_d w_d = 1$. We organize RL-then-MOPD into domain-wise exploration and shared exploitation toward the aggregate objective in Eq. 1.

Domain-wise exploration. Each explorer is initialized from π_0 and independently optimized on J_d, producing two outputs: a final specialist π_d^T and a historical trajectory buffer $\mathcal{B}_d$ of prompt-response pairs $\tau = (x, y)$ collected throughout RL. We denote the policies that generated these responses by $\{\pi_{d,i}\}_{i \in \mathcal{I}_d}$, where i indexes RL substages within exploration.

Shared exploitation. A student π_θ, initialized from π_0, integrates the specialists through multi-domain policy distillation (MPD). For a response prefix $h_t = (x, y_{<t})$, the update signal is

$$\log \pi_d^T(y_t \mid h_t) - \log \pi_\theta(y_t \mid h_t). \tag{2}$$

The teacher defines the supervision, while the trajectory source determines the prefixes and tokens on which it acts.

3.2 Trajectory Reuse for Rollout-Free Consolidation

DoubleFree loads complete responses from $\mathcal{B}_d$ rather than generating new student rollouts. The exploration buffer contains responses generated throughout domain-specific capability development, spanning policies from different RL substages. With prompts drawn from the same domain distribution $\mathcal{D}_d$, the underlying response distribution (*reuse*) is

$$p_{\text{reuse},d}(y \mid x) = \sum_{i \in \mathcal{I}_d} \alpha_{d,i} \pi_{d,i}(y \mid x), \tag{3}$$

where $\alpha_{d,i} \geq 0$ is the mixture weight of historical policy $\pi_{d,i}$ and $\sum_i \alpha_{d,i} = 1$.

Within the framework of Section 3.1, we express alternative trajectory sources as

$$p_{\text{fresh},d}(y \mid x; \theta) = \pi_\theta(y \mid x), \tag{4}$$

$$p_{\text{initial},d}(y \mid x) = \pi_0(y \mid x), \tag{5}$$

$$p_{\text{final},d}(y \mid x) = \pi_d^T(y \mid x). \tag{6}$$

We extend these trajectory sources to multi-domain consolidation, yielding Standard MOPD (*fresh*), Lightning MOPD (*initial*), Offline MPD (*final*), and DoubleFree (*reuse*), all under the same final-specialist supervision. Here, reused responses need not be positive demonstrations: the teacher-student signal can increase or decrease the probability of their observed tokens. Exploration history supplies contexts visited during capability development, while the final specialist determines how the student should learn from them. This changes the sampling distribution rather than reproducing on-policy distillation exactly.

3.3 Pre-scoring: Online-Teacher-Free Training

For fixed responses, scores from frozen teachers are independent of the evolving student. We therefore precompute and cache observed-token log-probabilities under teacher forcing,

$$\mathbf{q}_d^T(\tau) = \left(\log \pi_d^T(y_t \mid h_t)\right)_{t=1}^{|y|}. \tag{7}$$

During student training, cached teacher scores remain fixed while student log-probabilities are recomputed on the reused responses. Pre-scoring separates teacher and student GPU allocations in time, removing online teacher queries and scoring barriers. Teachers can also be loaded and released sequentially during preparation, further reducing peak GPU demand.

Exploration typically generates more responses per prompt than distillation uses, enabling optional trajectory selection. We construct the scored training buffer as

$$\widetilde{\mathcal{B}}_d = \text{Select}_\phi \left\{(\tau, \mathbf{q}_d^T(\tau)) : \tau \in \mathcal{B}_d\right\}. \tag{8}$$

Here, ϕ may retain all responses or select a subset, for example by random sampling.

3.4 SHARED-STUDENT OPTIMIZATION

Our primary instantiation, DoubleFree-MPD, optimizes the shared student using final-specialist supervision on reused trajectories. We follow a PPO-style clipped objective: cached teacher log-probabilities and recomputed student log-probabilities define clipped, detached token-level distillation advantages. Updates are anchored to the pre-update student policy, and the loss is averaged over valid response tokens. The full objective is given in Appendix A.1. Trajectory reuse is also compatible with SFT. We include DoubleFree-SFT as a variant that directly minimizes response-token cross-entropy on reused trajectories, without teacher scoring.

4 THEORETICAL ANALYSIS OF TRAJECTORY REUSE

4.1 BOUNDED DISCREPANCY FROM ON-POLICY UPDATES

The key question is how replacing fresh student rollouts with historical exploration trajectories changes the distillation update. To characterize this discrepancy, we fix the student parameters θ and final specialists π_d^T. In each domain d, *fresh* and *reuse* draw prompts from the same $\mathcal{D}_d$ but generally induce different response distributions. Following the local-surrogate perspective of Wu et al. (2026), we compare unnormalized population update directions at the PPO student anchor.

Let $f_d(x, y; \theta)$ be a response's update contribution in domain d and $g_b(\theta)$ the expected update under source $b \in \{\text{fresh}, \text{reuse}\}$:

$$\begin{aligned} f_d(x, y; \theta) &= \sum_{t=1}^{|y|} A_{d,t}(x, y; \theta) \nabla_\theta \log \pi_\theta(y_t \mid h_t), \\ g_b(\theta) &= \sum_d w_d \mathbb{E}_{x \sim \mathcal{D}_d,\, y \sim p_{b,d}(\cdot|x)}[f_d(x, y; \theta)]. \end{aligned} \tag{9}$$

Here, $|y|$ is the response length, y_t its t-th token, and $h_t = (x, y_{<t})$ its prefix. The advantage $A_{d,t}$ is the teacher-student log-probability difference in Eq. 2, clipped to $[-c, c]$ for $c > 0$ and held constant during differentiation. The domain weights satisfy $w_d \geq 0$ and $\sum_d w_d = 1$.

Theorem 4.1 (Bounded update discrepancy)**.** *At a fixed student state θ, assume for each domain d that $p_{\text{fresh},d}(\cdot \mid x; \theta) \ll p_{\text{reuse},d}(\cdot \mid x)$ for every prompt x with $\mathcal{D}_d(x) > 0$. (Here, $\ll$ denotes absolute continuity: every response with positive fresh probability also has positive reuse probability.) Let $B_d \geq 0$ be a finite bound on the root-mean-square update contribution,*

$$\mathbb{E}_{x \sim \mathcal{D}_d,\, y \sim p_{\text{reuse},d}(\cdot|x)} \| f_d(x, y; \theta) \|_2^2 \leq B_d^2,$$

where $\| \cdot \|_2$ is the Euclidean norm. Assume the nonnegative distribution mismatch $\delta_d(\theta)$, defined by

$$\delta_d^2(\theta) = \mathbb{E}_{x \sim \mathcal{D}_d} \chi^2(p_{\text{fresh},d}(\cdot \mid x; \theta) \,\|\, p_{\text{reuse},d}(\cdot \mid x)), \tag{10}$$

is finite. For response distributions $p \ll q$, $\chi^2(p \| q) = \sum_{y: q(y) > 0} (p(y) - q(y))^2 / q(y)$. Then

$$\| g_{\text{reuse}}(\theta) - g_{\text{fresh}}(\theta) \|_2 \leq \sum_d w_d B_d \delta_d(\theta) \equiv \Delta(\theta), \tag{11}$$

where $\Delta(\theta)$ is the aggregate update-discrepancy bound.

Theorem 4.1 shows that reuse remains close to the fresh update when distribution mismatch and update magnitudes are controlled (proof in Appendix B.1). Importantly, the direction of the divergence penalizes responses that are likely under the current student but unlikely under *reuse*. The discrepancy therefore depends on *how well the historical source covers the responses the current student would generate*.

4.2 COVERAGE BENEFITS OF HISTORICAL POLICY MIXTURES

Historical reuse combines trajectories generated by policies $\{\pi_{d,i}\}$ across exploration substages. This mixture can inherit coverage from any retained historical policy, as formalized below.

Table 1: Comparison of multi-task training methods.

Method	Trajectory Source	Additional Rollout	Consolidation objective	Online Teacher	Supervision Source
Standard MOPD	Current student	Online	PD	Yes	Final specialist
Mixed SFT	Final specialist	Offline	SFT	No	Generated response tokens
Offline MPD	Final specialist	Offline	PD	No	Final specialist
Lightning MOPD	Initial student	Offline	PD	No	Final specialist
DoubleFree-SFT	RL training policies	None	SFT	No	RL response tokens
DoubleFree-MPD	RL training policies	None	PD	No	Final specialist

Proposition 4.2 (Coverage inherited from history)**.** *For each domain d, let $\pi_{d,i}$ be a retained historical policy with mixture weight $\alpha_{d,i} > 0$ in Eq. 3. The distribution mismatch defined in Eq. 10 is bounded as follows:*

$$1 + \delta_d^2(\theta) \leq \frac{1 + \mathbb{E}_{x \sim \mathcal{D}_d} \chi^2(\pi_\theta(\cdot \mid x) \,\|\, \pi_{d,i}(\cdot \mid x))}{\alpha_{d,i}}. \tag{12}$$

Because *reuse* assigns each response at least an $\alpha_{d,i}$ fraction of its probability under $\pi_{d,i}$, good coverage by any sufficiently weighted historical policy transfers to the reuse mixture (Appendix B.2). More importantly, historical policies may provide complementary coverage. Responses rarely generated by either endpoint may be more common at intermediate substages, for example due to differences in reward, teacher-student gap, or response length. Their mixture can therefore cover student behavior that no single historical policy covers well (Appendix B.3).

Takeaway. Reusing exploration trajectories stays close to on-policy distillation as long as the history covers what the student would generate. Historical mixtures are naturally suited to provide this coverage, as they combine complementary responses from every exploration stage.

5 Experiments

5.1 Experimental Setup

Models and datasets. We evaluate Qwen3-4B and Qwen3-8B (Qwen Team, 2025), as well as SmolLM3-3B (Bakouch et al., 2025), in non-thinking mode across math, code, and instruction following. For domain RL, we sample 17K examples each from DAPO-Math-17k (Yu et al., 2025), DeepCoder-Preview-Dataset (Luo et al., 2025), and Nemotron-RL-instruction_following (NVIDIA, 2025), respectively. Evaluation spans nine benchmarks across the three domains, with benchmark details and protocols in Appendix A.3.

Baselines. We compare DoubleFree-MPD with Standard MOPD, Offline MPD, and Lightning MOPD, which use different trajectory sources under the same final-specialist supervision. Mixed SFT and DoubleFree-SFT provide response-token supervision baselines. We focus comprehensive baseline comparisons and ablations on Qwen3-4B. On Qwen3-8B and SmolLM3-3B, we evaluate both DoubleFree variants against Standard MOPD to assess cross-model generalization. For the main accuracy comparisons, we evaluate DoubleFree-MPD over three random seeds on each model, while using the same fixed seed for all other methods. Table 1 summarizes the method configurations.

Training protocol. Within each model, methods share the student initialization and use equal numbers of training prompts. We implement all methods in verl (Sheng et al., 2024) with vLLM (Kwon et al., 2023) on a testbed with 16 NVIDIA H100-80GB GPUs. For RL, actor training and rollout generation are colocated on the same GPU group. Standard MOPD uses separate student and teacher GPU pools. For example, four GPUs support student training and rollouts, while each of the three domain teachers occupies one dedicated GPU. More details in Appendix A.

Metrics. *GPU-hours* measure allocated resource cost, summing each phase's duration multiplied by its allocated GPU count. *Time* reports consolidation duration in hours, combining measured training time and offline preparation time. *Speedup* is Standard MOPD's Time divided by the method's *Time* for the same model. For accuracy preservation, we report math accuracy, code pass@1, and strict prompt-level accuracy for instruction following. AVG equally weights within-domain benchmarks, and All AVG across the three domains.

5.2 Training Efficiency

Table 2: Training efficiency across three models. Teacher costs include offline pre-scoring or online teacher residency, including idle time. Cache load for offline methods includes cached trajectory access and associated batch-processing overhead.

Method	GPUs	GPU-hours ↓						Time (h) ↓	Speedup ↑
		Offline/Online Rollout	Offline/Online Teacher	Cache load	Update	Others	**Total**		
Qwen3-4B									
Math RL	8	51.37	–	–	75.70	2.05	129.13	16.14	–
Code RL	8	34.62	–	–	35.05	2.21	71.88	8.98	–
IF RL	8	17.35	–	–	6.31	1.95	25.61	3.20	–
Mixed SFT	4	24.10	–	0.63	10.56	2.31	37.60	9.44	1.40×
Offline MPD	4	24.10	1.73	0.60	13.36	2.34	42.13	10.53	1.25×
Lightning MOPD	4	9.22	1.34	0.68	6.14	2.47	19.85	4.96	2.66×
Standard MOPD	4+3	36.68	39.56	–	13.49	2.58	92.31	13.19	1.00×
DoubleFree-SFT	4	–	–	0.62	10.20	2.17	12.99	3.25	4.06×
DoubleFree-MPD	4	–	1.34	1.01	12.64	2.63	17.61	4.40	2.99×
Qwen3-8B									
Standard MOPD	8+6	101.41	96.71	–	19.63	7.90	225.65	16.12	1.00×
DoubleFree-SFT	8	–	–	1.54	14.24	6.65	22.43	2.80	5.75×
DoubleFree-MPD	8	–	1.57	2.40	16.88	7.65	28.50	3.56	4.52×
SmolLM3-3B									
Standard MOPD	4+3	28.16	26.39	–	5.57	1.45	61.58	8.80	1.00×
DoubleFree-SFT	4	–	–	0.53	4.25	1.78	6.56	1.67	5.26×
DoubleFree-MPD	4	–	1.19	0.77	5.40	2.33	9.70	2.43	3.63×

Table 2 shows that DoubleFree reduces consolidation GPU-hours by 80.9-90.1% and achieves 2.99-5.75× speedups across three models and both variants. For example, on Qwen3-4B, DoubleFree-MPD reduces GPU-hours from 92.31 to 17.61 and consolidation time from 13.19 to 4.40 hours. Resource savings exceed time savings because DoubleFree both shortens consolidation and eliminates dedicated teacher GPU reservations, including idle periods. The benefits extend to Qwen3-8B and SmolLM3-3B, where the MPD variant achieves 4.52× and 3.63× speedups, respectively.

The cost breakdown for Qwen3-4B illustrates where these savings arise. Domain RL already incurs 103.34 rollout GPU-hours, yet Standard MOPD adds 36.68 student rollout GPU-hours and 39.56 teacher-residency GPU-hours, including idle time. Offline MPD and Lightning MOPD avoid online teacher residency but still require 24.10 and 9.22 generation GPU-hours, respectively. Here, Lightning benefits from shorter initial-student responses. DoubleFree eliminates this additional overhead, requiring only offline scoring for MPD and no teacher scoring for SFT. Its higher cache-loading overhead reflects loading and validating full candidate shards before optional selection, but remains small relative to the generation and residency costs avoided.

5.3 Multi-Task Accuracy

Table 3 demonstrates that exploration trajectories support effective consolidation without additional response generation. On Qwen3-4B, DoubleFree-MPD achieves 58.18 overall accuracy, slightly above Offline MPD (57.78) and Lightning MOPD (58.13). The exploration history spans multiple stages of capability development, potentially providing broader coverage than either initial-

Table 3: Multi-task accuracy (%). Bold marks the best consolidated student per model. DoubleFree-MPD reports mean ± sample standard deviation over three seeds.

Method	Math					Code				Instruction Following			All AVG
	AIME24 mean@32	AIME25 mean@32	AMC23 mean@32	MATH500	AVG	LCB v6 pass@1	BCB pass@1	HumanEval+ pass@1	AVG	IFBench	IFEval	AVG	
Qwen3-4B													
Base	21.46	20.31	65.55	80.20	46.88	26.07	8.11	76.22	36.80	26.33	79.85	53.09	45.59
Final Specialists	54.69	54.48	92.27	90.40	72.96	31.79	11.82	81.71	41.77	40.00	83.92	61.96	58.90
Mixed SFT	54.69	52.71	**92.19**	89.60	72.30	**32.57**	10.14	78.05	40.25	38.67	83.55	61.11	57.89
Offline MPD	55.52	**53.13**	91.80	90.40	72.71	32.00	9.54	77.44	39.66	39.33	82.62	60.98	57.78
Lightning MOPD	55.21	51.46	92.11	**90.60**	72.34	31.93	**10.98**	74.39	39.10	40.67	**85.21**	62.94	58.13
Standard MOPD	**55.94**	52.92	91.87	90.20	**72.73**	32.14	9.97	79.27	40.46	39.67	81.89	60.78	57.99
DoubleFree-SFT	50.63	53.02	91.64	89.40	71.17	32.21	10.73	**81.71**	**41.55**	**43.00**	84.10	**63.55**	**58.76**
DoubleFree-MPD	54.76 ±1.49	52.26 ±0.94	91.88 ±0.16	90.40 ±0.20	72.32 ±0.37	32.12 ±0.64	10.33 ±0.27	79.07 ±1.53	40.51 ±0.64	39.44 ±2.36	83.98 ±0.47	61.71 ±1.07	58.18 ±0.50
Qwen3-8B													
Base	27.29	20.31	66.72	79.60	48.48	25.29	17.23	81.10	41.20	27.00	82.81	54.90	48.20
Final Specialists	62.40	54.27	92.27	88.80	74.44	34.93	17.06	82.93	44.97	45.67	88.35	67.01	62.14
Standard MOPD	59.17	51.04	**91.33**	87.40	72.23	**36.07**	**17.99**	82.93	**45.66**	**46.33**	**87.80**	**67.07**	**61.65**
DoubleFree-SFT	58.44	51.88	90.08	87.20	71.90	34.50	**17.99**	80.49	44.33	44.00	87.43	65.72	60.65
DoubleFree-MPD	**59.93** ±0.69	**52.15** ±0.22	91.12 ±0.66	**88.27** ±0.61	**72.87** ±0.41	34.90 ±0.48	16.53 ±0.18	**83.54** ±2.79	44.99 ±0.95	44.00 ±0.58	87.49 ±0.75	65.75 ±0.08	61.20 ±0.20
SmolLM3-3B													
Base	8.02	9.48	42.34	63.60	30.86	18.79	3.80	59.15	27.25	15.67	69.13	42.40	33.50
Final Specialists	36.67	27.29	76.48	83.80	56.06	25.36	5.57	67.07	32.67	46.33	82.81	64.57	51.10
Standard MOPD	**37.29**	26.88	73.44	**83.20**	**55.20**	**27.00**	**7.01**	61.59	31.87	**44.00**	**81.33**	**62.67**	**49.91**
DoubleFree-SFT	36.25	**26.98**	74.45	83.00	55.17	25.50	6.00	58.54	30.01	42.00	**81.33**	61.67	48.95
DoubleFree-MPD	36.39 ±1.36	26.39 ±0.77	**75.00** ±0.28	81.47 ±0.64	54.81 ±0.44	26.38 ±0.37	6.17 ±0.59	**63.21** ±2.14	**31.92** ±0.91	41.78 ±0.38	81.08 ±0.28	61.43 ±0.22	49.39 ±0.19

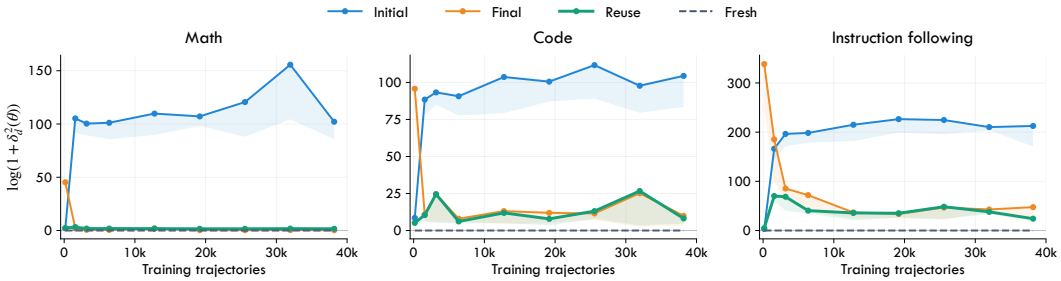


Figure 4: Trajectory-source coverage during Qwen3-4B consolidation. Lower values indicate less distribution mismatch or better coverage.

student or final-specialist responses alone. Its comparable performance to Standard MOPD (57.99) also shows that *reuse* approximates *fresh*-based updates. The additional models exhibit a similar accuracy-efficiency trade-off. DoubleFree-MPD reaches 61.20 ± 0.20 on Qwen3-8B and 49.39 ± 0.19 on SmolLM3-3B, with modest deficits of 0.45 and 0.52 percentage points relative to Standard MOPD. Together with the cost reductions in Section 5.2, these results support reuse as a practical alternative to on-policy consolidation. Within DoubleFree variants, cached teacher supervision generally yields more balanced results than response-token training alone. SFT attains the highest overall score on Qwen3-4B (58.76), as its stronger code and instruction-following results offset lower math accuracy, notably on AIME24 (50.63 vs. MPD's 54.76). MPD further improves overall accuracy over SFT by 0.55 and 0.44 points on Qwen3-8B and SmolLM3-3B, respectively, highlighting the value of teacher supervision beyond the reused responses themselves.

5.4 Trajectory Analysis

Trajectory coverage comparison. To examine how trajectory sources cover the evolving student, we evaluate their likelihoods on common on-policy responses sampled throughout consolidation on Qwen3-4B. We approximate exploration history with a uniform mixture of policy checkpoints

spaced across domain RL, including the shared initialization and final specialists. Following the mismatch definition in Eq. 10, we estimate its response-prefix counterpart and report $\log(1+\delta_d^2(\theta))$. Figure 4 shows that the initial policy quickly becomes mismatched as the student evolves, whereas final specialists exhibit substantial early mismatch. The historical mixture (*reuse*) moderates these stage-dependent mismatches, consistent with complementary coverage across exploration stages, though it does not uniformly outperform the final specialists. These diagnostics illustrate coverage dynamics rather than directly validate the full-trajectory update bound.

Sensitivity to trajectory selection. To assess how strongly reuse depends on trajectory selection, we compare three strategies on Qwen3-4B under MPD settings. Each retains one of eight responses per prompt: *Random* samples uniformly, *Reward* selects the highest-reward response, and *Balance* samples uniformly from the top four by reward. Table 4 shows relatively modest differences in overall accuracy, with Random already approaching Standard MOPD (57.63 vs. 57.99). Reward improves only slightly over Random, while Balance achieves the highest overall score in this comparison. These results suggest that effective reuse does not hinge on careful trajectory selection, while leaving room for improved selectors that better support capability transfer (Appendix C.2).

Table 4: Trajectory-selection ablation.

Selector	Math	Code	IF	AVG
Random	71.21	40.40	61.28	57.63
Reward	71.30	41.30	60.61	57.74
Balance	72.65	41.14	62.77	58.85

6 Related Work

Multi-task post-training. Common pipelines include sequential or mixed-domain training with SFT or RL objectives (Luo et al., 2023; Lambert et al., 2025; Wang et al., 2025) and training-free model merging (Yadav et al., 2023). To mitigate conflicts among task objectives, PCGrad (Yu et al., 2020) and MGS (Cai et al., 2026) apply gradient surgery during joint training. Recent studies further refine training: NPO (Qin et al., 2026) uses near-future policy guidance to improve exploration, CoPD (Gu et al., 2026) interleaves RL and mutual distillation, D^3-MOPD (Sun et al., 2026) adapts domain sampling to learning progress during consolidation, and Open-MOPD (Gao et al., 2026a) rebalances domain supervision to address capability imbalance. These refinements are complementary to DoubleFree: they modify how consolidation learns, whereas DoubleFree changes the source of trajectories and teacher supervision.

Efficient policy distillation and trajectory reuse. Offline sequence-level distillation learns from teacher-generated responses (Kim & Rush, 2016), while Lightning OPD (Wu et al., 2026) uses fixed responses from the SFT-initialized student. These sources correspond to the two endpoints of domain exploration: the final specialist policy and the initial policy. AsyncOPD (Kang et al., 2026) enables asynchronous distillation using stale student rollouts. At finer granularity, ReOPD (Liao et al., 2026) reuses teacher-generated prefixes while generating new student actions at selected steps. With fixed responses and frozen teachers, both endpoint sources and historical reuse permit offline pre-scoring for online-teacher-free student training. DoubleFree additionally eliminates response generation for consolidation by reusing trajectories already collected during domain RL. By combining complementary behaviors across exploration substages, its historical policy mixture may cover the evolving student better than either endpoint alone.

7 Conclusion

Conventional RL-then-MOPD incurs substantial system costs from redundant response generation and persistent teacher GPU residency. We formulate this pipeline as domain-wise exploration followed by shared exploitation and introduce DoubleFree to reuse both outputs of exploration: final specialist policies and historical trajectories. By combining trajectory reuse with teacher pre-scoring, DoubleFree enables rollout-free consolidation and online-teacher-free student training. Our analysis relates the discrepancy from on-policy distillation updates to distribution mismatch and characterizes how historical policy mixtures can improve coverage. Experiments across three models show substantial reductions in GPU-hours and training time, while lowering peak GPU demand and maintaining comparable multi-domain performance.

AI USE STATEMENT

Generative AI tools were used to assist with language polishing and improving the clarity and presentation of the manuscript. All AI-assisted content was reviewed and verified by the authors, who take full responsibility for the content of this paper.

ETHICS STATEMENT

Our work uses publicly available models and benchmarks and involves no human subjects or personally identifiable information. The training and evaluation prompts come from established public datasets cited in Section 5.1; we do not introduce new data collection. Our method reduces the GPU-hours required for multi-task consolidation, which lowers the energy footprint of this training stage; we do not foresee specific negative societal impacts beyond those generally associated with improved language-model training efficiency.

REPRODUCIBILITY STATEMENT

We describe the explore-exploit formulation and the full DoubleFree training procedure in Section 3, with the student objective detailed in Appendix A.1. Experimental setup, including models, datasets, baselines, training protocol, and metrics, is given in Section 5.1 and Appendix A. Proofs of Theorem 4.1 and Proposition 4.2 appear in Appendices B.1 and B.2. All evaluated models, datasets, and benchmarks are publicly available, and all methods are implemented in the open-source verl framework (Sheng et al., 2024). We will release our implementation codes and scripts upon publication.

REFERENCES

Lei Bai, Jiaqi Cao, Chiyu Chen, Guanzhou Chen, Kai Chen, Guangran Cheng, Erfei Cui, Xuanlang Dai, Shengyuan Ding, Shangheng Du, Yanhui Duan, Yue Fan, Youqing Fang, Quan Gan, Yuanyuan Gao, Jiaye Ge, Lixin Gu, Yuzhe Gu, Qipeng Guo, Junjun He, Xin Hong, Ming Hu, Zhouqi Hua, Haian Huang, Junhao Huang, Zixian Huang, Minxi Jin, Lingkai Kong, Alexander Lam, Zehao Li, Zonglin Li, Tianhao Liang, Dahua Lin, Junyao Lin, Tianyang Lin, Zhouhan Lin, Jiangning Liu, Jin Liu, Kuikun Liu, Wenran Liu, Yifei Liu, Yuhong Liu, Yuhong Liu, Zhoumianze Liu, Ziyan Liu, Ziyu Liu, Haijun Lv, Han Lv, Chengqi Lyu, Le Ma, Ningsheng Ma, Zerun Ma, Haoyang Peng, Runyu Peng, Jifei Shan, Zixin Shang, Kou Shi, Xiang Shi, Qisheng Su, Xuerui Su, Hao Sun, Xiao Sun, Yanan Sun, Yu Sun, Huanze Tang, Yinghao Tang, Wenhui Tian, Zhongbo Tian, Bingli Wang, Haomin Wang, Jiarui Wang, Jingzhi Wang, Rui Wang, Xiquan Wang, Yi Wang, Zhecan Wang, Ziyi Wang, Zun Wang, Rubin Wei, Lianyi Wu, Wen Wu, Yue Wu, Yuhan Wu, Zhenyu Wu, Zijian Wu, Shuhao Xing, Jun Xu, Xingle Xu, Xuenan Xu, Xiangchao Yan, Ziang Yan, Bowen Yang, Danni Yang, Lin Yang, Zhiqi Yang, Qian Yao, Haochen Ye, Peng Ye, Jinhui Yin, Jiashuo Yu, Dingbo Yuan, Fei Yuan, Yuhang Zang, Bo Zhang, Chao Zhang, Chen Zhang, Hongjie Zhang, Junming Zhang, Wenlong Zhang, Wenwei Zhang, Yiming Zhang, Zhuo Zhang, Ziyang Zhang, Haiteng Zhao, Penghao Zhao, Yibo Zhao, Zhonghan Zhao, Zhihang Zhong, Bowen Zhou, Peiheng Zhou, Xin Zhou, Xinyu Zhou, Yunhua Zhou, Dongsheng Zhu, and Yicheng Zou. Intern-s2-preview: Scientific agentic foundation model, 2026. URL https://arxiv.org/abs/2608.13505.

Elie Bakouch, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, Lewis Tunstall, Carlos Miguel Patiño, Edward Beeching, Aymeric Roucher, Aksel Joonas Reedi, Quentin Gallouédec, Kashif Rasul, Nathan Habib, Clémentine Fourrier, Hynek Kydlicek, Guilherme Penedo, Hugo Larcher, Mathieu Morlon, Vaibhav Srivastav, Joshua Lochner, Xuan-Son Nguyen, Colin Raffel, Leandro von Werra, and Thomas Wolf. SmolLM3: smol, multilingual, long-context reasoner. Hugging Face Blog, 2025. URL https://huggingface.co/blog/smollm3.

Min Cai, Yu Liang, Longzheng Wang, Yan Wang, Yueyang Zhang, Long Xia, Zhiyuan Sun, Xi Ye, and Daiting Shi. Advancing general-purpose reasoning models with modular gradient surgery, 2026. URL https://arxiv.org/abs/2602.02301.

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, 2026a. URL https://arxiv.org/abs/2608.19098.

Wei Gao, Yuheng Zhao, Dakai An, Tianyuan Wu, Lunxi Cao, Shaopan Xiong, Ju Huang, Weixun Wang, Siran Yang, Wenbo Su, Jiamang Wang, Lin Qu, Bo Zheng, and Wei Wang. RollPacker: Taming Long-Tail rollouts for RL Post-Training with tail batching. In *23rd USENIX Symposium on Networked Systems Design and Implementation (NSDI 26)*, pp. 849–866, Renton, WA, May 2026b. USENIX Association. ISBN 978-1-939133-54-0. URL https://www.usenix.org/conference/nsdi26/presentation/gao-wei.

Naibin Gu, Chenxu Yang, Qingyi Si, Chuanyu Qin, Dingyu Yao, Peng Fu, Zheng Lin, Weiping Wang, Nan Duan, and Jiaqi Wang. Co-evolving policy distillation. arXiv preprint, 2026. URL https://arxiv.org/abs/2604.27083.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring Mathematical Problem Solving With the MATH Dataset. arXiv preprint, 2021. URL https://arxiv.org/abs/2103.03874.

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, 2024. URL https://arxiv.org/abs/2403.07974.

Wonjun Kang, Kevin Galim, Seunghyuk Oh, Minjun Kang, Sanghyun Park, Donghoon Kim, Minjae Lee, Minseo Kim, Rishabh Tiwari, Yuchen Zeng, Hyung Il Koo, and Kangwook Lee. AsyncOPD: How stale can on-policy distillation be? arXiv preprint, 2026. URL https://arxiv.org/abs/2606.24143.

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/.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient Memory Management for Large Language Model Serving with PagedAttention. arXiv preprint, 2023. URL https://arxiv.org/abs/2309.06180.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tulu 3: Pushing frontiers in open language model post-training, 2025. URL https://arxiv.org/abs/2411.15124.

Baohao Liao, Hanze Dong, Christof Monz, Xinxing Xu, Li Dong, and Furu Wei. Multi-turn on-policy distillation with prefix replay. arXiv preprint, 2026. URL https://arxiv.org/abs/2607.04763.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's Verify Step by Step. arXiv preprint, 2023. URL https://arxiv.org/abs/2305.20050.

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. arXiv preprint, 2023. URL https://arxiv.org/abs/2305.01210.

Michael Luo, Sijun Tan, Roy Huang, Ameen Patel, Alpay Ariyak, Qingyang Wu, Xiaoxiang Shi, Rachel Xin, Colin Cai, Maurice Weber, Ce Zhang, Li Erran Li, Raluca Ada Popa, and Ion Stoica. DeepCoder: A Fully Open-Source 14B Coder at O3-mini Level. Together AI Blog, 2025. URL https://www.together.ai/blog/deepcoder.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. arXiv preprint, 2023. URL https://arxiv.org/abs/2308.08747.

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

NVIDIA. Nemotron-RL-instruction_following. Hugging Face dataset, 2025. URL https://huggingface.co/datasets/nvidia/Nemotron-RL-instruction_following.

Valentina Pyatkin, Saumya Malik, Victoria Graf, Hamish Ivison, Shengyi Huang, Pradeep Dasigi, Nathan Lambert, and Hannaneh Hajishirzi. Generalizing Verifiable Instruction Following. arXiv preprint, 2025. URL https://arxiv.org/abs/2507.02833.

Chuanyu Qin, Chenxu Yang, Qingyi Si, Naibin Gu, Dingyu Yao, Zheng Lin, Peng Fu, Nan Duan, and Jiaqi Wang. Near-future policy optimization. arXiv preprint, 2026. URL https://arxiv.org/abs/2604.20733.

Qwen Team. Qwen3 technical report. arXiv preprint, 2025. URL https://arxiv.org/abs/2505.09388.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. HybridFlow: A flexible and efficient RLHF framework. *arXiv preprint arXiv:2409.19256*, 2024. URL https://arxiv.org/abs/2409.19256.

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, 2026. URL https://arxiv.org/abs/2608.24987.

Boxin Wang, Chankyu Lee, Nayeon Lee, Sheng-Chieh Lin, Wenliang Dai, Yang Chen, Yangyi Chen, Zhuolin Yang, Zihan Liu, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. Nemotron-Cascade: Scaling cascaded reinforcement learning for general-purpose reasoning models. arXiv preprint, 2025. URL https://arxiv.org/abs/2512.13607.

Yecheng Wu, Song Han, and Hai Cai. Lightning OPD: Efficient post-training for large reasoning models with offline on-policy distillation. arXiv preprint, 2026. URL https://arxiv.org/abs/2604.13010.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. TIES-Merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2306.01708.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. DAPO: An Open-Source LLM Reinforcement Learning System at Scale. arXiv preprint, 2025. URL https://arxiv.org/abs/2503.14476.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://arxiv.org/abs/2001.06782.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-Following Evaluation for Large Language Models. arXiv preprint, 2023. URL https://arxiv.org/abs/2311.07911.

Terry Yue Zhuo, Minh Chien Vu, Jenny Chim, Han Hu, Wenhao Yu, Ratnadira Widyasari, Imam Nur Bani Yusuf, Haolan Zhan, Junda He, Indraneil Paul, Simon Brunner, Chen Gong, Thong Hoang, Armel Randy Zebaze, Xiaoheng Hong, Wen-Ding Li, Jean Kaddour, Ming Xu, Zhihan Zhang, Prateek Yadav, Naman Jain, Alex Gu, Zhoujun Cheng, Jiawei Liu, Qian Liu, Zijian Wang, Binyuan Hui, Niklas Muennighoff, David Lo, Daniel Fried, Xiaoning Du, Harm de Vries, and Leandro Von Werra. BigCodeBench: Benchmarking Code Generation with Diverse Function Calls and Complex Instructions. arXiv preprint, 2024. URL https://arxiv.org/abs/2406.15877.

A Implementation Details

A.1 Student Objective

For each mixed-domain minibatch, let $\mathcal{C}_d$ be the subset sampled from $\widetilde{\mathcal{B}}_d$. For each observed token, define

$$A_t(\tau) = \mathrm{sg}\left[\mathrm{clip}\big(q^T_{d,t}(\tau) - \log \pi_\theta(y_t \mid h_t), -c, c\big)\right], \tag{13}$$

$$\rho_t(\tau) = \frac{\pi_\theta(y_t \mid h_t)}{\bar{\pi}(y_t \mid h_t)}, \tag{14}$$

where sg denotes stop-gradient, $c > 0$ bounds the distillation signal, and $\bar{\pi}$ is the student checkpoint anchoring the minibatch updates. We optimize

$$\mathcal{L}_{\mathrm{PD}}(\theta) = -\frac{1}{N_{\mathcal{C}}} \sum_d \sum_{(\tau,\mathbf{q}^T)\in\mathcal{C}_d} \sum_t \min\{\rho_t A_t, \mathrm{clip}(\rho_t, 1-\epsilon, 1+\epsilon)A_t\}, \tag{15}$$

where $N_{\mathcal{C}}$ is the total number of valid response tokens in the combined minibatch and $\epsilon > 0$ controls ratio clipping. Student and anchor log-probabilities are evaluated on the reused responses without generating new ones.

A.2 Training Configurations

Models and initialization. We use the released post-trained checkpoints of Qwen3-4B[1], Qwen3-8B[2], and SmolLM3-3B[3], all in non-thinking mode. Qwen3-4B hosts the full baseline comparison; the other models test generalization across sizes and families. Domain specialists and the shared student are independently initialized from the same checkpoint, denoted *Base* in the result tables. PD methods use the fixed final specialists as teachers.

Implementation and evaluation. All methods are implemented in verl with vLLM and run on two nodes with eight NVIDIA H100 GPUs each; the efficiency tables report per-method resource allocations. We evaluate final student checkpoints.

Random seeds. The default training seed is 42. The DoubleFree-MPD accuracy rows in Table 3 report the mean and sample standard deviation over seeds 42, 43, and 44, with RL trajectories and final-teacher scores held fixed across seeds. Other rows report single-seed results unless noted otherwise. Each seed's domain and aggregate scores are computed before averaging across seeds.

Baseline construction. Lightning MOPD adapts the fixed initial-student trajectory source of Lightning OPD to multi-domain PD; it does not reproduce the full SFT-OPD pipeline. Offline MPD, Lightning MOPD, and DoubleFree-MPD cache final-specialist scores before student training. Mixed SFT and DoubleFree-SFT use final-specialist response tokens and historical RL response tokens, respectively, without teacher scoring during training.

Trajectory selection. *Random* samples one of the eight RL responses per prompt occurrence; *Reward* selects the response with the highest recorded RL reward; *Balance* samples uniformly among the four highest-reward responses. Table 3 uses *Balance* on Qwen3-4B and *Random* on Qwen3-8B and SmolLM3-3B for both DoubleFree variants.

Training configurations. Tables 5 and 6 summarize the key RL and distillation settings. Both stages use 64 prompts per batch, a learning rate of 1×10^{-6}, a PPO mini-batch of 16, sampling at temperature 1.0 with top-$p = 1$, a maximum response length of 10,240 tokens, token-mean loss aggregation, and seed 42. Each domain uses 17K training examples from DAPO-Math-17k (Yu et al., 2025), DeepCoder-Preview-Dataset (Luo et al., 2025), or Nemotron-RL-instruction_following (NVIDIA, 2025). Math uses a final-answer verifier; code uses an all-tests-pass reward with at most 15 tests, a 10-second timeout, and a 1,024 MB memory limit; instruction following uses the fraction of verified constraints satisfied.

[1] https://huggingface.co/Qwen/Qwen3-4B

[2] https://huggingface.co/Qwen/Qwen3-8B

[3] https://huggingface.co/HuggingFaceTB/SmolLM3-3B

Table 5: Domain-RL settings.

Setting	Value
Objective	GRPO
Training batches	200
Responses per prompt	8
PPO clipping (low / high)	0.2 / 0.28
KL reward	Disabled
Overlong buffer	4,096 tokens

Table 6: Policy-distillation settings.

Setting	Value
Objective	PPO-style clipped PD
Responses per prompt	1
Prompt proportions	1:1:1
Teacher scoring	Temperature $= 1.0$
Distillation signal	Teacher$-$student log-prob. difference

A.3 Evaluation Configurations

For math, we report mean@32 (averaging correctness over 32 responses per problem) on AIME 2024/2025 and AMC 2023, and single-response accuracy on MATH500 (Hendrycks et al., 2021; Lightman et al., 2023). For code, we report pass@1 with eight samples per problem on the 175-problem LiveCodeBench v6 (Jain et al., 2024) `test6` generation subset and the 148-problem BigCodeBench (Zhuo et al., 2024) Complete-Hard subset, and pass@1 with a single sample on HumanEval+ (Liu et al., 2023). IFBench (Pyatkin et al., 2025) and IFEval (Zhou et al., 2023) use strict prompt-level accuracy. Math evaluation uses temperature 0.6 and top-$p = 0.95$; code and instruction-following evaluation use temperature 0.7, top-$p = 0.8$, top-$k = 20$, and seed 1234, with an additional presence penalty of 1.5 for code.

B Theoretical Details

We fix the student state θ, final teachers, and historical policies. Write $P_d(x, y) = \mathcal{D}_d(x)p_{\mathrm{fresh},d}(y \mid x; \theta)$ and $Q_d(x, y) = \mathcal{D}_d(x)p_{\mathrm{reuse},d}(y \mid x)$. The common prompt marginal gives $\chi^2(P_d\|Q_d) = \delta_d^2(\theta)$, where

$$\chi^2(P\|Q) = \mathbb{E}_Q\left[\left(\frac{\mathrm{d}P}{\mathrm{d}Q} - 1\right)^2\right], \qquad P \ll Q. \tag{16}$$

These are population distributions, rather than realized finite buffers.

B.1 Proof of Theorem 4.1

Proof. Let $W_d = \mathrm{d}P_d/\mathrm{d}Q_d$ and $g_{b,d} = \mathbb{E}_{\mathcal{D}_d, p_{b,d}} f_d$. Changing measure and applying Cauchy-Schwarz yields

$$g_{\mathrm{fresh},d} - g_{\mathrm{reuse},d} = \mathbb{E}_{Q_d}[(W_d - 1)f_d], \tag{17}$$

$$\|g_{\mathrm{fresh},d} - g_{\mathrm{reuse},d}\|_2 \le \sqrt{\mathbb{E}_{Q_d}(W_d - 1)^2}\sqrt{\mathbb{E}_{Q_d}\|f_d\|_2^2} \le \delta_d B_d. \tag{18}$$

The triangle inequality over the domain-weighted sum gives Eq. 11. □

Absolute continuity holds if a retained policy with positive mixture weight assigns positive probability to every response allowed by the fresh source. Hard sampling filters can violate this condition. An admissible moment bound is $B_d = cL_{\max}G$ when response lengths are bounded by $L_{\max}$ and $\|\nabla_\theta \log \pi_\theta(y_t \mid h_t)\|_2 \le G$. Clipping the advantage alone does not bound the score function or distribution mismatch. The ratio W_d is a proof device, not an importance weight used by the algorithm.

B.2 Proof of Proposition 4.2

Proof. Fix d, x and a retained component with $\alpha_{d,i} > 0$. Abbreviate $p = \pi_\theta(\cdot \mid x)$, $q = p_{\mathrm{reuse},d}(\cdot \mid x)$, and $q_i = \pi_{d,i}(\cdot \mid x)$. Since $q \ge \alpha_{d,i}q_i$ pointwise,

$$1 + \chi^2(p\|q) = \sum_y \frac{p(y)^2}{q(y)} \le \frac{1}{\alpha_{d,i}} \sum_y \frac{p(y)^2}{q_i(y)} = \frac{1 + \chi^2(p\|q_i)}{\alpha_{d,i}}. \tag{19}$$

Zero-over-zero terms are zero; if the right-hand divergence is infinite, the bound is vacuous. Averaging over $x \sim \mathcal{D}_d$ proves the proposition. □

If initial-policy responses are retained with weight $\alpha_{d,0} > 0$, the bound at $\pi_{\theta_0} = \pi_0$ becomes $\delta_d^2(\theta_0) \le \alpha_{d,0}^{-1} - 1$. This gives an initial coverage component, not control over later drift.

B.3 An Example of Complementary Historical Coverage

For one prompt and $0 < \varepsilon < 1/3$, consider three responses:

Distribution	A	B	C
Initial	$1-2\varepsilon$	ε	ε
Final	ε	$1-2\varepsilon$	ε
Intermediate	ε	ε	$1-2\varepsilon$
Current student	$1/3$	$1/3$	$1/3$

The uniform historical mixture equals the current student p, so $\chi^2(p\|q_{\text{reuse}}) = 0$. Each endpoint instead has mismatch

$$\chi^2(p\|q_{\text{initial}}) = \chi^2(p\|q_{\text{final}}) = \frac{1}{9}\left(\frac{1}{1-2\varepsilon} + \frac{2}{\varepsilon}\right) - 1 > 0.$$

Even the best endpoint-only mixture retains positive mismatch:

$$\min_{\lambda\in[0,1]} \chi^2(p\|\lambda q_{\text{initial}} + (1-\lambda)q_{\text{final}}) = \frac{(1-3\varepsilon)^2}{9\varepsilon(1-\varepsilon)} > 0.$$

Both endpoints underrepresent C; the intermediate policy fills this gap. Conversely, an endpoint has zero mismatch when the student equals that endpoint. The example establishes a possible benefit, not a universal ranking or an assumption that students follow historical mixtures.

B.4 Connection to the Implemented Update

PPO anchor and inner updates. The implemented objective is Eq. 15. At $\pi_\theta = \bar{\pi}$, all token ratios equal one, ratio clipping is inactive, and the negative gradient is the sum of the f_d contributions divided by the valid token count. With the sampling distribution and detached advantages held fixed, g_b is also the gradient at $\vartheta = \theta$ of

$$\mathcal{S}_b(\vartheta;\theta) = \sum_d w_d \mathbb{E}_{\mathcal{D}_d, p_{b,d}}\left[\sum_t A_{d,t}(\theta)\log \pi_\vartheta(y_t \mid h_t)\right].$$

Thus the comparison concerns raw local surrogate gradients, not task-reward gradients or optimizer-preconditioned steps. Away from the anchor, token contributions acquire the PPO ratio and clipping indicator. The same change-of-measure argument applies to that common contribution function if its second moment is finite. Note that a fresh minibatch is still sampled from the anchor $\bar{\pi}$ rather than the updated π_θ; the anchor ratio therefore corrects for student updates within the minibatch window, not for the mismatch between the student and the historical explorer that generated a reused response.

Global token normalization. Let $P_b(d,x,y) = w_d \mathcal{D}_d(x) p_{b,d}(y \mid x)$, $\ell = |y|$, and $Z_b = \mathbb{E}_{P_b}\ell \in (0,\infty)$. Population token-mean updates are $G_b = g_b/Z_b$. For $W = \mathrm{d}P_{\text{fresh}}/\mathrm{d}P_{\text{reuse}}$ and $f(d,x,y) = f_d(x,y)$,

$$G_{\text{fresh}} - G_{\text{reuse}} = \frac{\mathbb{E}_{P_{\text{reuse}}}[(W-1)(f - \ell G_{\text{reuse}})]}{Z_{\text{fresh}}}. \tag{20}$$

This follows from $\mathbb{E}_{P_{\text{reuse}}}(f - \ell G_{\text{reuse}}) = 0$. Cauchy-Schwarz then gives

$$\|G_{\text{fresh}} - G_{\text{reuse}}\|_2 \le \frac{\sqrt{\sum_d w_d \delta_d^2}\,\sqrt{\mathbb{E}_{P_{\text{reuse}}}\|f - \ell G_{\text{reuse}}\|_2^2}}{Z_{\text{fresh}}},$$

when these moments are finite. Different response lengths therefore affect normalized update magnitudes. A finite minibatch has a random token-count denominator; its expected update need not equal the population ratio above.

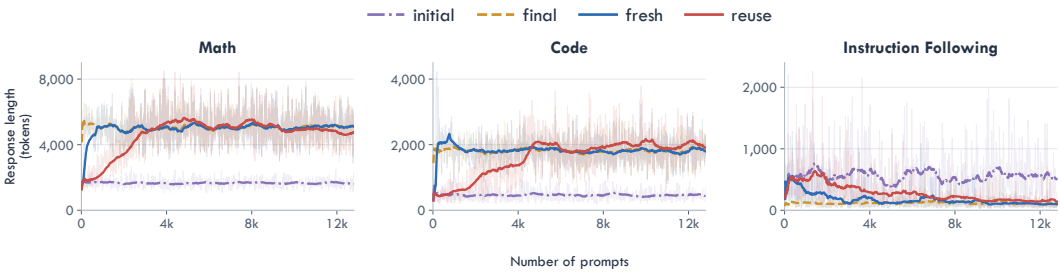
Finite buffers and selection. If $\widehat{g}_{\text{reuse}}$ is the expected unnormalized update under sampling from a realized buffer, then

$$\|\widehat{g}_{\text{reuse}} - g_{\text{fresh}}\|_2 \leq \Delta(\theta) + \|\widehat{g}_{\text{reuse}} - g_{\text{reuse}}\|_2.$$

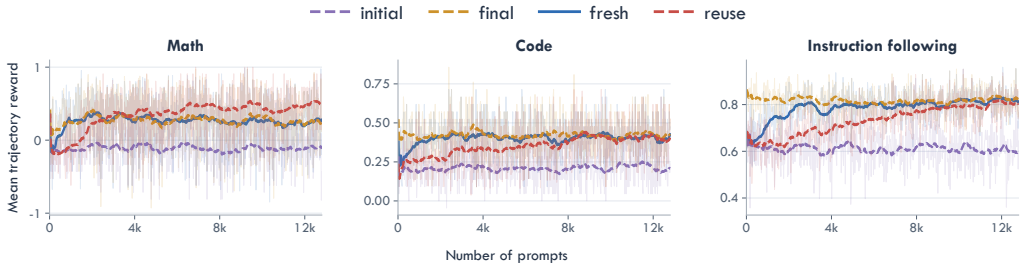
The additional finite-buffer error, minibatch variation, and adaptive student evolution are not bounded by the population theorem. Content-dependent selection requires using the resulting replay distribution; differing prompt marginals require joint prompt-response distributions. The results describe local update discrepancy, not convergence or final task accuracy.

C Training Dynamics and Additional Results

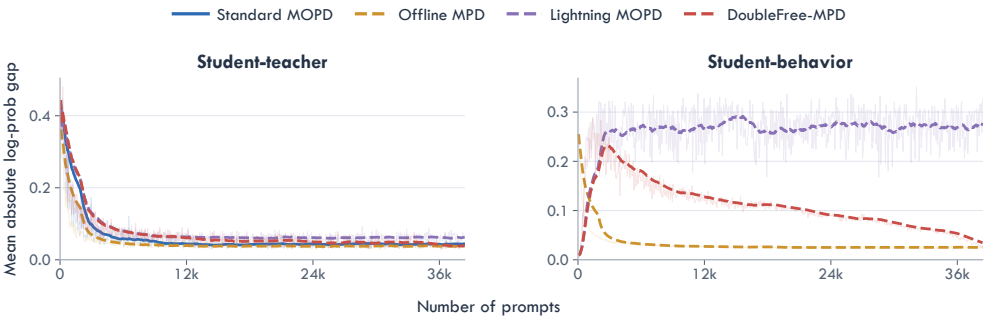
C.1 Training Dynamics



(a) Response length by trajectory source. Mixed SFT and Offline MPD share the same *final* responses.



(b) Reward by trajectory source.



(c) Student-teacher and student-behavior absolute sampled-token log-probability gaps. Standard MOPD is omitted from the behavior gap, as its responses are generated by the pre-update student itself.

Figure 5: Training dynamics by trajectory source on Qwen3-4B.

Figure 5 compares response sources during Qwen3-4B consolidation. Figures 5a and 5b analyze the trajectories themselves through their lengths and rewards, while Figure 5c tracks training dynamics through the student's log-probability gaps. Throughout, *fresh* denotes current-student responses from Standard MOPD, *reuse* the historical RL responses consumed by DoubleFree-MPD, *initial* the initial-student responses from Lightning MOPD, and *final* the final-specialist responses shared by Mixed SFT and Offline MPD. Cached-source curves follow the consumption order of fixed responses; *fresh* instead follows the evolving student's online responses. The *reuse* trace uses *Random* selection to exclude reward-based selection effects.

The *final* responses average 5,061, 1,820, and 120 tokens in math, code, and instruction following, respectively, versus *initial*'s 1,660, 464, and 556. The longer math and code responses dominate the total token volume, so generating from final specialists is correspondingly more expensive. Lightning MOPD instead generates from the weaker initial student, whose responses are substantially shorter, so its additional generation is correspondingly faster. DoubleFree reuses responses already produced during domain RL and incurs no additional preparation generation. Weighting the distillation loss to balance the per-domain token-count differences observed above is a natural direction for future work.

Figure 5b shows that *reuse* responses attain higher mean reward than *initial* in all three domains, while remaining below *fresh* and *final* in code and instruction following. Since *reuse* mixes responses from policies across RL substages, it also naturally spans a wider reward range than any single-policy source, providing more diverse responses for the student to learn from.

In Figure 5c, DoubleFree-MPD's student-teacher gap steadily decreases and approaches Offline MPD's late-stage level, indicating that reused trajectories move the student toward the teachers about as effectively as teacher-generated responses do. The teacher gap is logged during actor updates, and the behavior gap compares the pre-update student with the policy that generated each cached response. Each method is logged on its own responses. Meanwhile, its student-behavior gap also *decreases* as training consumes later RL trajectories, rather than growing as a fixed offline source would suggest. The consumption order explains this: consolidation replays the exploration progression from early, initial-like policies toward the final specialist, so the behavior distribution advances in the same direction as the student itself.

C.2 Sensitivity to Trajectory Selection

We ablate the trajectory selector ϕ in Eq. 8 for both DoubleFree variants on Qwen3-4B; each selector retains one of eight RL responses per prompt occurrence. Table 7 shows that overall accuracy varies modestly across selectors: *Random* already approaches the reference method in each block, *Reward* brings small gains, and *Balance* performs best for both variants. Effective reuse therefore does not hinge on careful response selection, while better selectors still leave room for further gains.

Table 7: Trajectory-selection ablations for DoubleFree-SFT and DoubleFree-MPD on Qwen3-4B (%). Bold indicates the best score in each column within each block.

Method	Math					Code				Instruction Following			All AVG
	AIME24 mean@32	AIME25 mean@32	AMC23 mean@32	MATH500	**AVG**	LCB v6 pass@1	BCB pass@1	HumanEval+ pass@1	**AVG**	IFBench	IFEval	**AVG**	
DoubleFree-SFT													
Mixed SFT	**54.69**	52.71	**92.19**	89.60	**72.30**	**32.57**	10.14	78.05	40.25	38.67	83.55	61.11	57.89
Random	52.71	51.04	90.39	**90.40**	71.14	31.93	**10.90**	80.49	41.10	39.67	83.92	61.79	58.01
Reward	53.44	**53.33**	91.88	88.60	71.81	31.50	10.56	**81.71**	41.25	41.33	83.92	62.63	58.56
Balance	50.63	53.02	91.64	89.40	71.17	32.21	10.73	**81.71**	**41.55**	**43.00**	**84.10**	**63.55**	**58.76**
DoubleFree-MPD													
Standard MOPD	55.94	**52.92**	91.87	**90.20**	**72.73**	32.14	9.97	79.27	40.46	39.67	81.89	60.78	57.99
Random	53.85	50.31	**91.88**	88.80	71.21	32.07	10.47	78.66	40.40	39.00	**83.55**	61.28	57.63
Reward	54.38	50.21	91.41	89.20	71.30	32.00	**10.81**	**81.10**	**41.30**	37.67	**83.55**	60.61	57.74
Balance	**56.04**	52.50	**91.88**	**90.20**	72.65	**32.79**	10.14	80.49	41.14	**42.00**	**83.55**	**62.77**	**58.85**