REGEN: REPLAY-RECYCLING FOR EXPERT-TO-GENERALIST DISTILLATION WITH OFFLINE REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large-scale online reinforcement learning (RL) is the predominant means of eliciting advanced abilities including long-term reasoning and agentic tool use in large language models (LLMs). However, continuing to scale it across vast task domains of interest remains challenging in both computational infrastructure and cost, especially when considering RL as merely a one-off learning stage. Recently, a widely used technique for distilling knowledge across various domains and training stages, multi-teacher on-policy distillation (MOPD), helps to decouple the RL stage, saving costs, while maintaining generality across vast domains. Nonetheless, similar to online RL, MOPD requires coupled inference and backward passes, which continue to limit its scalability and computational efficiency. To address these challenges, we propose **REGEN: Replay-recycling for Expert-to-Generalist distillation with Offline RL**. Instead of distilling from multiple teacher models, REGEN trains a generalist by simply recycling the replay memory—the free by-product of the teachers' specialized RL training—and employing offline RL algorithms. REGEN completely decouples the rollout sampling from the backward training process and thus greatly reduces the training cost. Across mathematical reasoning, code generation, and instruction following, REGEN matches the accuracy of MOPD at substantially lower cost. It potentially turns online RL into a data synthesis process instead of a one-off learning stage, and can be extended to large-scale post-training without requiring heavy computational load.

1 INTRODUCTION

Reinforcement learning (RL) has emerged as the dominant means of endowing large language models (LLMs) with strong reasoning and decision-making ability, underpinning recent advances in mathematical problem solving, program synthesis, scientific question answering, and agentic tool use (DeepSeek-AI, 2025; OpenAI, 2024). However, as the field shifts from optimizing models for individual tasks toward training generalist reasoning agents, large-scale online RL is approaching a structural bottleneck due to the following reasons. First, jointly optimizing a single policy across many task domains can be relatively challenging and expensive owing to the phenomenon of negative transfer (Yu et al., 2020; Ahn et al., 2025; Wu et al., 2026), especially when it comes to RL, which requires complex infrastructure and substantial exploration costs, or even different reward models and hyper-parameters across different domains. Second, online RL is inherently a one-off optimization process that does not lend itself to straightforward aggregation across domains or stages. Historical rollouts are seldom reused in subsequent training runs—even when the training tasks remain unchanged—leading to a significant waste of computational resources whenever base models or task distributions are switched.

To mitigate the cost of large-scale RL across diverse domains and training stages, multi-teacher distillation (Hinton et al., 2015; You et al., 2017) has become a practical alternative, in which multiple domain-specialized RL experts are first trained and then serve as teachers for a separate distillation stage. A central obstacle in distilling RL teachers is the well-known problem of *distribution shift* (Ross & Bagnell, 2010), whereby the trajectory distribution induced by the student diverges from that of the teacher. To address this issue, data aggregation (Ross & Bagnell, 2010) and on-policy distillation (OPD) (Agarwal et al., 2024; Gu et al., 2024; Kim & Rush, 2016) supervise the student on its own

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.

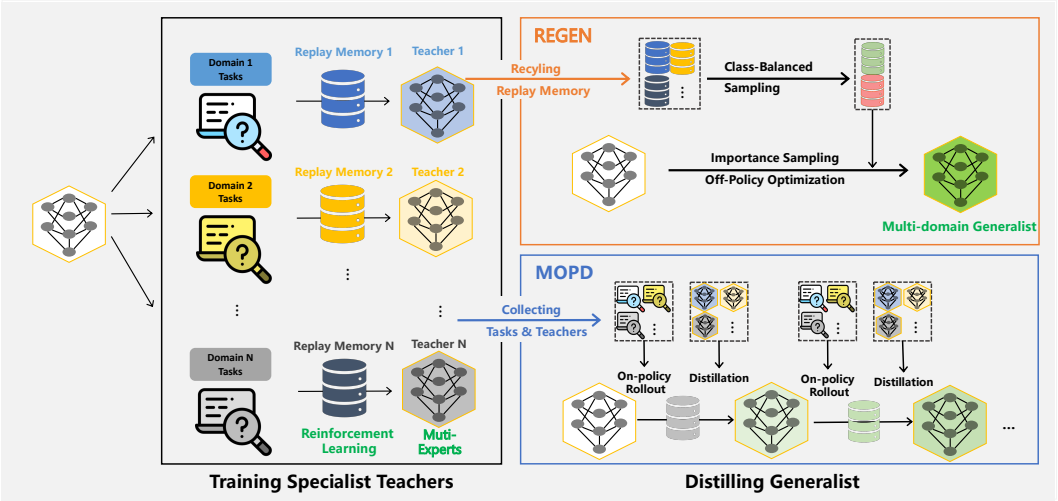


Figure 1: Sketch of the Replay-recycling for Expert-to-Generalist Distillation with Offline RL (REGEN) method in comparison with Multi-Teacher On-Policy Distillation (MOPD)

generated trajectories using teacher-provided signals. In the multi-domain setting, the Multi-teacher OPD (MOPD) (Xiaomi, 2026; Ma et al., 2026) routes each prompt to the corresponding domain-expert teacher, and thus serves as the natural distillation baseline. Although these methods effectively alleviate distribution shift, they require online inference, supervision, and backpropagation to proceed simultaneously, as in RL, and consequently remain computationally expensive.

Motivated by the observation that the rollout trajectories generated as a by-product of an online RL run constitute a rich source of supervision typically discarded by existing pipelines, we propose **REGEN**: **R**eplay-recycling for **E**xpert-to-**GEN**eralist Distillation with Offline RL as an alternative to MOPD. As shown in Figure 1, instead of distilling from trained teachers, REGEN aggregates the replay memory collected during specialist training into a single offline dataset, on which a generalist student is optimized. This method fully decouples data generation from student training, eliminating the need for teachers or rollout engines in the training loop. Through an appropriate combination of a data sampling and an off-policy offline RL objective, REGEN achieves performance and robustness comparable to MOPD at substantially lower computational cost. More broadly, it reframes any RL training run as a data-production pipeline, offering a scalable path toward large-scale RL on large models, including LLMs and beyond.

To summarize, our contributions are threefold. First, we propose **REGEN**, an offline post-training framework that reuses the replay memories generated during RL teacher training. By recycling these trajectories and optimizing the generalist with offline RL, REGEN achieves performance close to MOPD at substantially lower computational cost, as demonstrated in Section 4.2. Second, Section 4.3 shows that REGEN provides significantly stronger diversity exploration than MOPD, yielding superior performance in terms of pass@k. Third, REGEN demonstrates the value of replay memory as a reusable asset for LLM post-training, suggesting a new direction in which existing RL runs can serve not only as optimization procedures but also as data-generation processes for subsequent training stages.

2 Related Work

2.1 Reinforcement Learning for LLM Post-Training

Building on the vast abilities and knowledge from pre-training and supervised fine-tuning (SFT) (Ouyang et al., 2022a), RL adapts LLMs to human preferences (Ouyang et al., 2022a; Christiano et al., 2017; Ouyang et al., 2022b) and other more specific objectives and advanced abilities, including mathematical and programming tasks (DeepSeek-AI, 2025), chain-of-thought

reasoning (Guo et al., 2025), and agentic tool use (Zhang et al., 2026). Commonly used LLM RL algorithms range from PPO (Schulman et al., 2017) to GRPO (Shao et al., 2024), mostly relying on on-policy optimization, where the optimized model continually generates rollouts for its next training stage. A key constraint in on-policy RL is that the distribution gap between training trajectories and the current policy cannot be too large, requiring consecutive inference, evaluation, and training. This coupling raises considerable infrastructure and learning challenges: (i) RL can be much less sample-efficient, spending most cost exploring relatively low-valued trajectories; (ii) considerable effort is required to synchronize inference, evaluation, and training to maximize compute utility (Sheng et al., 2025; Hu et al., 2024); and (iii) modern cross-domain LLM RL pipelines can differ greatly in environment settings, including evaluation methods, raising additional challenges for unified training infrastructure.

2.2 Policy Distillation from Multiple Teachers

To address the bottlenecks of building a unified RL framework across domains, divide-and-conquer RL training plus multi-teacher policy distillation offers a lower-complexity, more flexible alternative: LLMs are first trained separately per domain, then these specialists are distilled into a cross-domain generalist. Policy distillation is long-standing, with Behavior Cloning (BC) (Pomerleau, 1988) as its most direct realization, formulating policy learning as supervised maximum-likelihood estimation over expert trajectories (Pomerleau, 1988). A well-documented limitation of BC is *distribution shift*: since the policy is trained under the expert-induced state distribution but deployed under its own, per-step prediction errors accumulate over the horizon and degrade long-sequence performance (Ross & Bagnell, 2010). This challenge has been addressed by shifting the student's state distribution, leading to approaches such as Data Aggregation (DAgger) (Ross et al., 2011) and inverse RL (Ng & Russell, 2000). For LLMs, BC and DAgger take the form of knowledge distillation (Hinton et al., 2015) and on-policy distillation (OPD) (Agarwal et al., 2024), respectively. DAgger and OPD achieve significantly better performance and much lower regret than BC and naive knowledge distillation by largely bridging the state-distribution gap, but at the cost of coupling inference and optimization, much like RL. Nonetheless, OPD inspired Multi-teacher OPD (MOPD) (Xiaomi, 2026; Ma et al., 2026), which has become the state of the art for training LLMs across vast domains.

2.3 Offline Reinforcement Learning

Addressing the distribution mismatch between training trajectories and the policy being optimized is a long-standing topic in RL. Offline RL, in particular, must address the most significant distribution gap, where static trajectories generated by any behavior policy may be used to optimize the performance of a completely unrelated policy (Levine et al., 2020). Distribution mismatch can cause convergence issues (Baird, 1995) and value-estimation biases (Fujimoto et al., 2019), and deteriorate or completely ruin the performance of existing RL algorithms. Typical remedies include explicit policy regularization to restrict the deviation of the policy distribution (Fujimoto et al., 2019; Kumar et al., 2019), conservative value estimation to correct value overestimation (Kumar et al., 2020), and importance-sampling-based off-policy correction to compensate for trajectory-distribution deviation. Replay memory, or the replay buffer, has been widely applied in off-policy RL algorithms to enhance data reuse and improve sample efficiency (Lange & Riedmiller, 2010; Mnih et al., 2015; Ritter et al., 2026; Arnal et al., 2026). However, most methods retain only a time window of historical trajectories to avoid a large distribution gap. The application of replay memories from entirely different models remains underexplored, especially in the context of training LLMs.

3 Methodology

3.1 Preliminaries

Consider x to be the context, which includes the user prompt and dialogue history, and y to be the answer output of the LLM. We use the policy $\pi(\cdot|x)$ to represent the distribution of the LLM's generation. We define each task domain as $\tau_d = (X_d = \{x_d^{(i)}\}, r_d(\cdot))$, where X_d is a collection of tasks or user queries, $r_d(\cdot)$ is the evaluation function for domain τ_d, which can potentially be a reward model or rules. Given D domains $\tau_1, \ldots, \tau_D$, a LLM π is first trained independently in each domain

as follows:

$$\pi \xrightarrow{\text{RL}(X_d, r_d)} \pi_d^*, \tag{1}$$

with π_d^* representing the teacher policy for domain d. However, π_d^* is a domain specialist, as it typically fails in different domains. The most straightforward approach to distilling the teachers is behavior cloning (BC) (Pomerleau, 1988), which forces each teacher to perform inference at its own mastered domain $y_d^* \sim \pi_d^*(\cdot|x_d)$, and minimizes the following imitation loss for the policy π_θ to be optimized:

$$\mathcal{L}_{\text{BC}}(\theta) \propto -\sum_{d \le D} \mathbb{E}_{x \sim X_d, y \sim \pi_d^*(\cdot|x)} \left[\frac{1}{|y|} \sum_t \log \pi_\theta(y_t \mid x, y_{<t}) \right]. \tag{2}$$

The risk of BC is that when the self-generated trajectory $y_{<t}$ deviates from the teacher generated trajectory $y_{d,<t}^*$, the remaining trajectories might enter a "unfamiliar zone" and thus lead to the explosion of prediction error. To avoid the impact of this distribution shift, OPD (Agarwal et al., 2024) utilizes supervision on self-generated trajectory instead of teacher generated trajectory. The loss function takes the following form:

$$\mathcal{L}_{\text{OPD}}(\theta) \propto -\sum_{d \le D} \mathbb{E}_{x \sim X_d,\, y \sim \pi_\theta(\cdot|x)} \left[\frac{1}{|y|} \sum_t A_t^{\text{OPD}} \log \pi_\theta(y_t \mid x, y_{<t}) \right], \tag{3}$$

$$A_t^{\text{OPD}} = \text{stop-gradient} \left[\log \frac{\pi_d^*(y_t|x, y_{<t})}{\pi_\theta(y_t|x, y_{<t})} \right].$$

where the KL divergence between the teacher and the student is minimized on the student-generated trajectories instead of the teacher's. The advantage A_t^{OPD} penalizes tokens that are less likely to be generated by the teacher and more likely to be generated by the student, while encouraging the reverse. MOPD (Xiaomi, 2026; Ma et al., 2026) further notices the lag between the student policy that generates the rollouts and the current policy under optimization, and introduces an importance weight w_t to re-weight the advantage. It also introduces a reward-based advantage (such as the GRPO advantage) in addition to the KL divergence, forming a hybrid method integrating RL and policy distillation, which gives the advantage value of the following final form, with $[\cdot]_a^b$ representing clipping the importance ratio to $[a, b]$.

$$A_t^{\text{MOPD}} = w_t(A_t^{\text{OPD}} + \alpha A_t^{\text{GRPO}}), \tag{4}$$

$$y \sim \pi_{\theta_{\text{lag}}}(\cdot|x), w_t = \text{stop-gradient} \left[\left[\frac{\pi_\theta(y_t \mid x, y_{<t})}{\pi_{\theta_{\text{lag}}}(y_t \mid x, y_{<t})} \right]_a^b \right]$$

3.2 Replay-recycling for Expert-to-Generalist Distillation with Offline RL

Instead of utilizing the teacher's policy π_d^* to supervise the trajectory, we choose to recycle the experience within the RL training process of each teacher RL(X_d, r_d), which we denote as $\mathcal{E}_d = \{(x^{(i)}, y^{(i)}, r^{(i)})\}$. The motivation for recycling the teacher's experience is straightforward: since domain experience drives the model to become domain specialists, accumulating experiences across those domains should at least be capable of driving the model to become competent in those domains. The only problem is the distribution shift; therefore, it is desirable to apply off-policy correction and offline learning algorithms. We name this method Replay-recycling for expert-to-generalist distillation with offline RL, or REGEN for short. The target loss of REGEN is motivated by two key considerations:

- Existing Offline RL methods without importance sampling such as DPO (Rafailov et al., 2023) render most offline samples ineffective, while off-policy GRPO (Mroueh et al., 2025) introduces importance sampling but employs symmetric clipping designed for near on-policy updates, zeroing out the gradient of any sequence whose importance ratio falls outside the given clipping interval and thereby discarding most learning signals under the large distribution shift of accumulated experiences. In contrast, TOPR (Roux et al., 2025) adopts Truncated Importance Sampling (TIS), which clips only the importance weight to $[0, 1]$ that modulates the gradient while leaving the underlying policy gradient term intact.

- Concerning unified cross-domain optimization, the original reward signal employed by TOPR can be adversely affected by discrepancies in reward settings across different domains. A more principled alternative is to replace the raw reward with a normalized advantage following GRPO. However, GRPO normalizes the advantage exclusively within rollouts generated consecutively by a unique policy, here we need to compute the advantage over the history, with different rollouts possibly coming from different policy snapshots.

Consequently, REGEN adopts the following loss function for offline trajectories:

$$\mathcal{L}_{\text{REGEN}}(\theta) \;=\; -\sum_{d\le D}\mathbb{E}_{(x,y,r)\sim\mathcal{E}_d}\left[A^{\text{REGEN}}\cdot\frac{1}{|y|}\sum_t \log\pi_\theta(y_t|x,y_{<t})\right], \tag{5}$$

$$A^{\text{REGEN}} = \begin{cases}\hat{A}, & \text{if } r \ge r_d^H\\ \text{stop-gradient}\left[\left[\dfrac{\pi_\theta(y|x)}{\mu(y|x)}\right]_0^1\right]\hat{A}, & \text{else}\end{cases},$$

$$\hat{A} = \frac{r-\hat{\mathbb{E}}(r|x,\mathcal{E}_d)}{\sqrt{\hat{\mathbb{V}}(r|x,\mathcal{E}_d)+\varepsilon}}. \tag{6}$$

r_d^H is a reward threshold defined per domain, serving as the criterion for classifying a sample as positive or negative. $\hat{\mathbb{E}}$ and $\hat{\mathbb{V}}$ are the empirical mean and variance, respectively. Notice that μ here is the policy that was used to sample the rollout y at that moment. Therefore, it does not refer to a single uniform policy, but rather a collection of policies accumulated across the training process and across different domains. Another issue is that for each $x \in X_d$, the corresponding output trajectories recorded in $\mathcal{E}_d$ may be generated by models of various training stages and can become excessively large in number. To address these challenges, REGEN introduces two additional techniques: augmented experience recording and class-balanced sampling.

Augmented experience recording. This phase harvests all rollouts from each teacher's online RL training without forgetting. To ensure $\mu(y|x)$ in Equation 5 can be effectively retrieved, we record the Negative Log Likelihood (NLL) $z = -\sum_t \log\mu(y_t|x,y_{<t})$ as the policy generating the rollout, and therefore the replay buffer is augmented by the NLL and represented as:

$$\Xi_d \;=\; \{(\,x^{(i)},y^{(i)},z^{(i)},r^{(i)})\}, \tag{7}$$

this replay buffer then ensures the estimation of μ can be efficiently recovered by $\mu(y^{(i)}|x^{(i)}) = \exp(-z^{(i)})$.

Class-balanced sampling. Similar to GRPO, we group the responses in the replay buffer by the same query. Since an overwhelming number of negative samples has been found to be harmful to RL training (Hong et al., 2023), while uniform sampling—in accumulated rollouts where positives dominate—repeatedly draws highly similar positive samples, reducing sample diversity, collapsing the advantage signal and driving the model to overfit the concentrated positive mode, and positives-only filtering discards negatives entirely, losing the contrastive signal needed to localize decision boundaries (Roux et al., 2025), we additionally apply Class-Balanced Sampling (CBS) (Cui et al., 2019) to adjust the distribution between positive and negative samples. Specifically, we retain only the *valid* queries, i.e., those that have at least one positive response and one negative response. For each query x, define the positive and negative groups:

$$\Xi_d^{\pm}(x) = \{(x,y,z,r)\in\Xi_d : r \gtrless r_d^H\}, \tag{8}$$

where $\gtrless$ is $\ge$ for $+$ and $<$ for $-$. To balance them, we downsample each group with probability

$$p^{\pm}(x) = \frac{\min(|\Xi_d^+(x)|,|\Xi_d^-(x)|)}{|\Xi_d^{\pm}(x)|}, \tag{9}$$

and form $\mathcal{E}_d(x) = \text{Down-Sample}(\Xi_d^+(x),p^+(x)) \cup \text{Down-Sample}(\Xi_d^-(x),p^-(x))$. Offline training is based on the down-sampled trajectories over all queries $\mathcal{E}_d = \{\mathcal{E}_d(x) : x \in X_d\}$. Notice that in

Algorithm 1 Replay-recycling for Expert-to-Generalist Distillation with Offline RL (REGEN)

Require: Tasks from D domains: $\{(X_d, r_d)\}$; Initial model: π
1: **Experience Recording**: Independently train teachers and collect experiences following Equation 7.
2: **Class-balanced Sampling**: Filter and down-sample the replay buffer according to Equations 8 and 9.
3: **Off-Policy Optimization**: Update the student model using the REGEN objective functions (Equation 5).
4: **return** The generalist policy π_θ.

Equation 5, the advantage $\hat{A}$ is calculated over the down-sampled group $\mathcal{E}_d(x)$ with respect to each query x. A review of the REGEN method is shown in Algorithm 1.

Note that REGEN can be readily extended to multi-turn tasks, where x represents both the query and the dialogue context at each turn. Moreover, it does not require the rollout-generating policy and the policy under optimization to share the same backbone or the same vocabulary as the teacher. These features endow the method with exceptional flexibility in data accumulation and offline training.

4 Experiments

4.1 Experiment Settings

We evaluate REGEN for generalist model training through joint training on three domains, using Qwen2.5-1.5B-Instruct (Yang et al., 2024) as the base model: mathematical problem solving (**Math**), code generation (**Code**), and instruction following (**Alignment**). The training corpora comprise GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021) for Math, KodCode-Light-RL-10K (Xu et al., 2025) for Code, and an IFEval-style instruction-following dataset (Hugging Face H4, 2024) for Alignment. Evaluation reports inference-time decoding accuracies on GSM8K, MATH, HumanEval (Chen et al., 2021), and MBPP (Austin et al., 2021); for IFEval (Zhou et al., 2023), both loose- and strict-match accuracies are reported.

Domain-expert teachers are trained via online RL, specifically GRPO (Shao et al., 2024), using the same base model under a unified cross-domain configuration; domain-specific reward allocation is detailed in Appendix A. Teacher training uses maximum generation length 1024, $n = 4$ responses per prompt, global batch size 128, learning rate 1×10^{-6}, and 4 epochs, yielding one expert per domain and approximately $800K$ logged trajectories. Using these teachers and trajectories, we compare:

- *Behavior Cloning* (BC) (Pomerleau, 1988): the best checkpoint from each online RL run serves as the domain-expert teacher and re-labels answers to training queries across all three domains; the student base model is trained directly on this re-labeled corpus via Eq. 2.
- *Multi-Teacher On-Policy Distillation* (MOPD) (Xiaomi, 2026; Ma et al., 2026): for each training query across all three domains, a new rollout is generated by the policy under optimization, evaluated with a reward, and relabeled token-wise by the corresponding domain expert; the student is trained via Eq. 4.
- *REGEN*: CBS (Cui et al., 2019) (The reward threshold r_d^H for each domain is $+1$) is applied directly to the logged trajectories, yielding $140K$ trajectories with equal positive and negative examples; the student base model is trained on this sampled dataset by minimizing Eq. 5.

BC, MOPD, and REGEN follow settings similar to teacher training: maximum sequence length 1024, batch size 128, learning rate 5.0×10^{-7}, and 4 epochs. For fair comparison, although REGEN can exploit a much larger corpus at minimal additional cost, all three methods use an identical total corpus size of $140K$, distributed roughly uniformly across domains; replay-memory distribution statistics are in Appendix B. All experiments are conducted on a single node with 4 NVIDIA L40S GPUs (48 GB each), using maximum local batch size 4 and 8-step gradient accumulation to achieve an effective global batch size of 128.

4.2 Main Results

Table 1 summarizes the performance of different methods across five benchmarks, reported under multiple evaluation metrics. Overall, REGEN consistently matches or surpasses MOPD on most

metrics, while substantially outperforming BC, particularly on Code and Alignment. For the 1.5B student, REGEN achieves the best results on MATH (pass@1 and maj@4), HumanEval (pass@10), and MBPP (pass@10) compared to BC and MOPD. For the 0.5B student, REGEN leads on all benchmarks (except GSM8K and IFEval) under multiple metrics, demonstrating its robustness across model scales. Notably, REGEN achieves this competitive accuracy with considerably higher training efficiency than MOPD. Since the student trainer maintains only π_θ, without hosting either the teacher model or a rollout engine during training, REGEN eliminates the online teacher inference that dominates MOPD's computational cost, thereby achieving substantially higher training throughput and lower per-token latency, as shown in Table 2.

Despite these gains, on IFEval, REGEN slightly lags MOPD under loose/strict metrics (e.g., -2.8 on loose metric for 0.5B, -3.5 on strict metric for 1.5B) due to coarser trajectory-level supervision, whereas MOPD benefits from dense token-level KL guidance from online teacher rollouts. In addition, both REGEN and MOPD remain noticeably behind the code teacher on HumanEval under pass@1. We attribute this gap to a fundamental limitation of policy distillation under distribution shift (Ross & Bagnell, 2010). Specifically, the training corpus KodCode-Light-RL-10K consists primarily of code generation tasks similar to MBPP, whereas HumanEval evaluates code completion. Since both methods distill the teacher's behavior only on the training distribution, the mismatch between generation and completion prevents full inheritance of the teacher's capability on the target benchmark.

Table 1: Accuracy comparison across benchmarks and multiple evaluation metrics. Teacher is the same online-RL expert (using the Qwen2.5-1.5B-instruct as the base model). For each student model and each metric column, the best among BC, MOPD, and REGEN is bolded. Superscripts in the REGEN rows indicate the difference relative to the MOPD value in the same column.

Method	GSM8K		MATH		HumanEval		MBPP		IFEval	
	pass@1	maj@4	pass@1	maj@4	pass@1	pass@10	pass@1	pass@10	loose	strict
Teacher	81.2	83.1	57.0	59.6	66.5	70.1	70.4	76.7	73.0	65.2
Student: Qwen2.5-1.5B-Instruct										
Base	76.6	77.3	53.6	57.8	55.5	73.8	65.3	84.9	52.7	49.7
BC	79.4	81.3	57.0	60.8	57.3	68.9	66.9	81.7	65.4	60.1
MOPD	**79.9**	81.9	57.3	59.0	62.2	65.9	**71.4**	77.2	**73.2**	**67.8**
REGEN	$79.5^{-0.4}$	$\mathbf{82.6}^{+0.7}$	$\mathbf{58.4}^{+1.1}$	$\mathbf{61.4}^{+2.4}$	$\mathbf{62.2}^{+0.0}$	$\mathbf{70.1}^{+4.2}$	$69.3^{-2.1}$	$\mathbf{77.8}^{+0.6}$	$71.5^{-1.7}$	$64.3^{-3.5}$
Student: Qwen2.5-0.5B-Instruct										
Base	48.7	52.1	32.6	35.5	39.0	48.8	48.4	60.3	38.1	36.2
BC	51.9	55.5	35.2	37.5	40.6	45.7	47.1	57.7	46.2	44.0
MOPD	**60.6**	**63.8**	36.5	38.8	40.2	45.7	51.9	54.8	**59.9**	**55.8**
REGEN	$57.8^{-2.8}$	$59.0^{-4.8}$	$\mathbf{38.1}^{+1.6}$	$\mathbf{39.8}^{+1.0}$	$\mathbf{41.5}^{+1.3}$	$\mathbf{48.8}^{+3.1}$	$\mathbf{52.6}^{+0.7}$	$\mathbf{63.5}^{+8.7}$	$57.1^{-2.8}$	$52.7^{-3.1}$

4.3 Ablations and Analysis

RQ1: How does REGEN compare to OPD in diversity exploration under pass@k when Teacher fails? The main experiments 4.2 reveal an interesting empirical phenomenon: on the two code generation tasks, HumanEval and MBPP, the Teacher fails to preserve its advantage under pass@k and can even fall below the original base model. We mainly attribute this to entropy collapse in the late stage of RL training, which limits the Teacher's diverse exploration. This limitation is intrinsic to online RL-style optimization and therefore directly affects OPD, since OPD continues to train the student through online rollouts and thus tends to inherit the same entropy-collapse behavior. For REGEN, however, the effect is only indirect: entropy collapse in the Code RL Teacher may produce many duplicated responses in the replay memory, and such duplicates can induce overfitting and degrade offline training. To investigate this, we use DeepSeek-V4 to remove duplicated responses, obtaining a $10K$ code training set, and retrain both REGEN and OPD on it (using Qwen2.5-1.5B-Instruct as the student model). Figure 2 shows the entropy curves of the Code RL Teacher and the newly trained OPD actor; OPD still exhibits entropy collapse. We regard this as an inherent limitation of RL training: since OPD is essentially RL, it naturally inherits this limitation, whereas REGEN is an offline method and is therefore not directly subject to entropy collapse. Existing work (Chen et al., 2026; Yu et al., 2026) strongly supports this argument. As shown in Table 3, after removing

Table 2: Training efficiency of REGEN versus MOPD on four 48 GB NVIDIA L40S GPUs. MOPD places each of the three teachers on one GPU and the student on the remaining one, while REGEN trains the student on all four GPUs without any teacher serving.

	Training Throughput		Per-1k-Token Latency			Overall
Method	**Tokens/s**	**Speedup**	**Forward**	**Backward**	**Total**	**Speedup**
MOPD	1,528.0	1.00×	463.3	218.7	682.0	1.00×
REGEN (Ours)	**7,714.5**	**5.05×**	**43.2**	**86.4**	**129.6**	**5.26×**

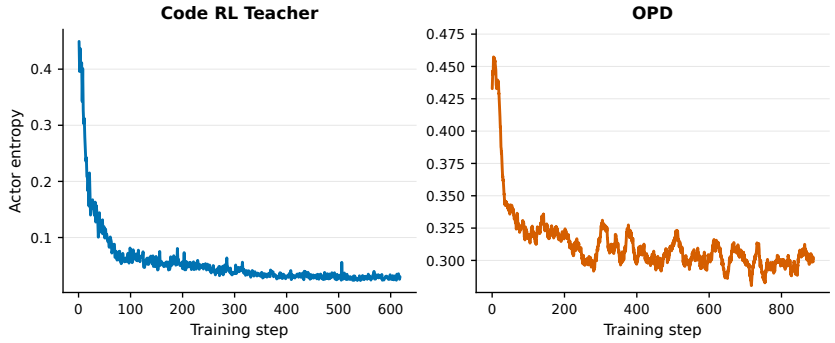


Figure 2: Training trajectories of the actor entropy for Code RL Teacher and OPD. The left and right panels show the corresponding entropy evolution over training steps, respectively.

duplicated data, REGEN significantly outperforms OPD under pass@k. Together with the main results in Section 4.2, where the 0.5B model achieves pass@k no lower than the base model and higher than MOPD on HumanEval and MBPP, we tentatively conclude that REGEN is superior to OPD in diversity exploration.

Table 3: Pass@1 and pass@10 comparison of REGEN and OPD on HumanEval and MBPP under a failed Teacher. Results are obtained on a $10K$ code training set with duplicated responses removed, using Qwen2.5-1.5B-Instruct as the student model. Teacher selects the same code teacher as the main experiment. For each metric column, the best among Base, OPD, and REGEN is bolded.

Method	HumanEval		MBPP	
	pass@1	pass@10	pass@1	pass@10
Teacher	66.5	70.1	70.4	76.7
Student: Qwen2.5-1.5B-Instruct				
Base	55.5	73.8	65.3	84.9
OPD	**60.4**	67.6	**70.1**	79.2
REGEN	$59.1^{-1.3}$	$\mathbf{75.6}^{+8.0}$	$69.3^{-0.8}$	$\mathbf{88.3}^{+9.1}$

RQ2: How do the sampling strategy and training objective affect model performance? We compare all combinations of sampling strategy and training objective in Table 4. Overall, the combination of CBS and REGEN achieves the best performance on almost all benchmarks. This confirms the effectiveness of CBS in correcting the imbalance inherent in accumulated rollouts and demonstrates that the asymmetric TIS objective in REGEN handles distribution shift more effectively than symmetric clipping and preference-based alternatives. The only exception is MBPP, where positives-only SFT achieves the best performance. This difference may stem from the nature of code generation: a program forms an integrated computation through its control flow, so an error typically propagates globally rather than remaining localized. Consequently, a fully correct program already serves as a clean imitation target, whereas an incorrect one provides little transferable supervision. By contrast, in mathematical reasoning, positive and negative solutions usually share most intermediate reasoning steps and diverge only at the erroneous step, while in IFEval the soft compositional

Table 4: Ablation of the sampling strategy and training objective on Qwen2.5-1.5B-Instruct, using the accumulated rollout trajectories logged along the online teachers' RL runs. The best result in each column is highlighted in bold. GRPO denotes off-policy GRPO.

Sampling	Objective	GSM8K	MATH	HumanEval	MBPP	IFEval	Avg.
Positives Only	**SFT**	77.7	57.3	61.0	**72.0**	68.2	67.2
Uniform	**TOPR**	77.7	56.7	59.1	69.6	70.1	66.6
	GRPO	74.9	54.1	59.1	66.9	67.8	64.6
	REGEN	77.2	56.7	60.4	68.5	71.0	66.8
CBS	**TOPR**	78.5	57.6	61.6	70.1	66.7	66.9
	GRPO	76.5	52.6	60.4	68.8	60.0	63.7
	DPO	75.3	54.7	60.4	65.9	56.9	62.6
	REGEN	**79.5**	**58.4**	**62.2**	69.3	**71.5**	**68.2**

Table 5: Multi-domain versus single-domain training with REGEN on Qwen2.5-1.5B-Instruct. *Multi* trains one student jointly over all three domains, whereas *Single* trains a separate student per domain. The two settings share identical training hyperparameters.The gap is computed as Single-domain minus Multi-domain.

Training Setting	GSM8K	MATH	HumanEval	MBPP	IFEval
Multi-domain	79.5	58.4	62.2	**69.3**	71.5
Single-domain	**80.2**	**58.3**	**63.4**	69.0	**73.2**
Gap	+0.7	-0.1	+1.2	-0.3	+1.7

constraints preclude a simple binary correctness criterion. In both settings, contrasting a correct solution with a near-miss negative provides critical supervision for localizing the decision boundary.

RQ 3: How does the performance of REGEN differ between multi-domain and single-domain training? We investigate whether training a single student jointly across all domains incurs any performance degradation compared with training a dedicated specialist for each domain. Table 5 compares our multi-domain offline student with single-domain students trained using the same objective on each domain independently. Across all five benchmarks, the performance gap never exceeds 2 accuracy points, and neither setting consistently outperforms the other. These results indicate that aggregating data from multiple domains introduces little to no cross-domain interference, demonstrating that REGEN remains stable and effective when a single student is shared across diverse domains.

5 Conclusion and Discussion

We propose REGEN, a method that recycles the replay memory accumulated during RL training of domain-expert teachers to subsequently train a generalist student. Through class-balanced sampling and an advantage-weighted asymmetric offline objective, REGEN achieves accuracy comparable to MOPD at substantially lower computational cost, surpassing naive behavior cloning by a significant margin. More broadly, REGEN reframes any RL run as a recyclable data-production process rather than a one-off learning stage, offering a scalable path toward large-scale post-training.

Our work effectively demonstrates the potential of reusing replay memory; however, the optimal strategy for leveraging this resource remains an open question worthy of deeper investigation. For example, are the negative samples discarded by CBS truly devoid of value, or could more sophisticated reuse methods unlock additional utility? Furthermore, while REGEN currently treats all replay data uniformly regardless of their generation stage, incorporating stage-specific information may enable more effective utilization. We leave these directions to future work.

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*. OpenReview.net, 2024. URL https://openreview.net/forum?id=3zKtaqxLhW.

Hongjoon Ahn, Jinu Hyeon, Youngmin Oh, Bosun Hwang, and Taesup Moon. Prevalence of negative transfer in continual reinforcement learning: Analyses and a simple baseline. In *The Thirteenth International Conference on Learning Representations*, 2025.

Charles Arnal, Vivien Cabannes, Taco Cohen, Julia Kempe, and Remi Munos. Efficient rl training for llms with experience replay. *arXiv preprint arXiv:2604.08706*, 2026.

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

Leemon Baird. Residual algorithms: Reinforcement learning with function approximation. In *Machine learning proceedings 1995*, pp. 30–37. Elsevier, 1995.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. *CoRR*, abs/2107.03374, 2021. URL https://arxiv.org/abs/2107.03374.

Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *Advances in Neural Information Processing Systems*, 38:57654–57689, 2026.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 4299–4307, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/d5e2c0adad503c91f91df240d0cd4e49-Abstract.html.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168, 2021. URL https://arxiv.org/abs/2110.14168.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9268–9277, 2019.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948, 2025. doi:10.48550/ARXIV.2501.12948. URL https://doi.org/10.48550/arXiv.2501.12948.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2052–2062. PMLR, 2019. URL http://proceedings.mlr.press/v97/fujimoto19a.html.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=5h0qf7IBZZ.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

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. In Joaquin Vanschoren and Sai-Kit Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015. URL http://arxiv.org/abs/1503.02531.

Zhang-Wei Hong, Aviral Kumar, Sathwik Karnik, Abhishek Bhandwaldar, Akash Srivastava, Joni Pajarinen, Romain Laroche, Abhishek Gupta, and Pulkit Agrawal. Beyond uniform sampling: Offline reinforcement learning with imbalanced datasets. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/0ff3502bb29570b219967278db150a50-Abstract-Conference.html.

Jian Hu, Xibin Wu, Zilin Zhu, Weixun Wang, Dehao Zhang, Yu Cao, et al. Openrlhf: An easy-to-use, scalable and high-performance rlhf framework. *arXiv preprint arXiv:2405.11143*, 6, 2024.

Hugging Face H4. Ifeval-like data. https://huggingface.co/datasets/HuggingFaceH4/ifeval-like-data, 2024. Hugging Face Datasets.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In Jian Su, Xavier Carreras, and Kevin Duh (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pp. 1317–1327. The Association for Computational Linguistics, 2016. doi:10.18653/V1/D16-1139. URL https://doi.org/10.18653/v1/d16-1139.

Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 11761–11771, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/c2073ffa77b5357a498057413bb09d3a-Abstract.html.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/0d2b2061826a5df3221116a5085a6052-Abstract.html.

Sascha Lange and Martin Riedmiller. Deep auto-encoder neural networks in reinforcement learning. In *The 2010 international joint conference on neural networks (IJCNN)*, pp. 1–8. IEEE, 2010.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *CoRR*, abs/2005.01643, 2020. URL https://arxiv.org/abs/2005.01643.

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

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Youssef Mroueh, Nicolas Dupuis, Brian Belgodere, Apoorva Nitsure, Mattia Rigotti, Kristjan Greenewald, Jiri Navratil, Jerret Ross, and Jesus Rios. Revisiting group relative policy optimization: Insights into on-policy and off-policy training. *arXiv preprint arXiv:2505.22257*, 2025.

Andrew Y Ng and Stuart J Russell. Algorithms for inverse reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pp. 663–670, 2000.

OpenAI. Openai o1 system card. *CoRR*, abs/2412.16720, 2024. doi:10.48550/ARXIV.2412.16720. URL https://doi.org/10.48550/arXiv.2412.16720.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022a.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022b. URL http://papers.nips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html.

Dean Pomerleau. ALVINN: an autonomous land vehicle in a neural network. In David S. Touretzky (ed.), *Advances in Neural Information Processing Systems 1, [NIPS Conference, Denver, Colorado, USA, 1988]*, pp. 305–313. Morgan Kaufmann, 1988. URL http://papers.nips.cc/paper/95-alvinn-an-autonomous-land-vehicle-in-a-neural-network.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/a85b405ed65c6477a4fe8302b5e06ce7-Abstract-Conference.html.

Daniel Ritter, Owen Oertell, Bradley Guo, Jonathan Chang, Kianté Brantley, and Wen Sun. Llms can learn to reason via off-policy rl. *arXiv preprint arXiv:2602.19362*, 2026.

Stéphane Ross and Drew Bagnell. Efficient reductions for imitation learning. In Yee Whye Teh and D. Mike Titterington (eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010*, volume 9 of *JMLR Proceedings*, pp. 661–668. JMLR.org, 2010. URL http://proceedings.mlr.press/v9/ross10a.html.

Stéphane Ross, Geoffrey J. Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In Geoffrey J. Gordon, David B. Dunson, and Miroslav Dudík (eds.), *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2011, Fort Lauderdale, USA, April 11-13, 2011*, volume 15 of *JMLR Proceedings*, pp. 627–635. JMLR.org, 2011. URL http://proceedings.mlr.press/v15/ross11a/ross11a.pdf.

Nicolas Le Roux, Marc G. Bellemare, Jonathan Lebensold, Arnaud Bergeron, Joshua Greaves, Alexandre Fréchette, Carolyne Pelletier, Eric Thibodeau-Laufer, Sándor Tóth, and Sam Work. Tapered off-policy REINFORCE: stable and efficient reinforcement learning for llms. *CoRR*, abs/2503.14286, 2025. doi:10.48550/ARXIV.2503.14286. URL https://doi.org/10.48550/arXiv.2503.14286.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300, 2024. doi:10.48550/ARXIV.2402.03300. URL https://doi.org/10.48550/arXiv.2402.03300.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025.

Runzhe Wu, Ankur Samanta, Ayush Jain, Scott Fujimoto, Jeongyeol Kwon, Ben Kretzu, Youliang Yu, Kaveh Hassani, Boris Vidolov, and Yonathan Efroni. Imbalanced gradients in rl post-training of multi-task llms. In *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 3137–3150, 2026.

LLM-Core Xiaomi. Mimo-v2-flash technical report. *CoRR*, abs/2601.02780, 2026. doi:10.48550/ARXIV.2601.02780. URL https://doi.org/10.48550/arXiv.2601.02780.

Zhangchen Xu, Yang Liu, Yueqin Yin, Mingyuan Zhou, and Radha Poovendran. Kodcode: A diverse, challenging, and verifiable synthetic dataset for coding. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, volume ACL 2025 of *Findings of ACL*, pp. 6980–7008. Association for Computational Linguistics, 2025. doi:10.18653/V1/2025.FINDINGS-ACL.365. URL https://doi.org/10.18653/v1/2025.findings-acl.365.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *CoRR*, abs/2412.15115, 2024. doi:10.48550/ARXIV.2412.15115. URL https://doi.org/10.48550/arXiv.2412.15115.

Shan You, Chang Xu, Chao Xu, and Dacheng Tao. Learning from multiple teacher networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*, pp. 1285–1294. ACM, 2017. doi:10.1145/3097983.3098135. URL https://doi.org/10.1145/3097983.3098135.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 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.

Guibin Zhang, Hejia Geng, Xiaohang Yu, Zhenfei Yin, Zaibin Zhang, Zelin Tan, Heng Zhou, Zhong-Zhi Li, Xiangyuan Xue, Yijiang Li, Yifan Zhou, Yang Chen, Chen Zhang, Yutao Fan, Zihu Wang, Songtao Huang, Francisco Piedrahita Velez, Yue Liao, Hongru WANG, Mengyue Yang, Heng Ji, Jun Wang, Shuicheng YAN, Philip Torr, and LEI BAI. The landscape of agentic reinforcement learning for LLMs: A survey. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=RY19y2RI1O. Survey Certification.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *CoRR*, abs/2311.07911, 2023. doi:10.48550/ARXIV.2311.07911. URL https://doi.org/10.48550/arXiv.2311.07911.

A Reward Design Across Domains

The three task domains adopt distinct reward schemes tailored to their evaluation characteristics.

For the **Math** domain, the reward is binary. Both GSM8K and MATH employ prompts that constrain the final answer within a `box{}` format, and the reward is determined by matching the extracted answer against the ground truth. During the online RL training of each Math teacher, we assign a reward of 1 for correct answers and 0 for incorrect ones, as negative rewards would dilute the learning signal and hinder teacher training. However, in the TOPR baseline, where trajectories are weighted by their rewards instead of advantages, a reward of 0 for negative samples would cause them to be entirely discarded. To address this, we redefine the trajectory-level reward of negative samples as -1 during TOPR training, which aligns with the original TOPR design and ensures that negative samples contribute a meaningful contrastive signal.

For the **Code** domain, the reward is continuous in $[0, 1]$, reflecting the fraction of passed test cases. A program that passes all tests receives a reward of 1, while a completely failing program receives 0, with intermediate values proportional to the test pass rate. For the **Alignment** domain, the reward is also continuous in $[0, 1]$, representing the proportion of verifiable constraints satisfied by the generated response. This fine-grained reward provides nuanced supervision compared to the binary Math domain.

B Distribution Statistics of Replay Memory

Figure 3 illustrates the reward and advantage distributions of the accumulated replay memory across the three domains. As expected from the reward design described in Appendix A, the Math domain exhibits a bimodal reward distribution concentrated at -1.0 and 1.0, while Code and Alignment display continuous distributions over $[0, 1]$. After group-relative normalization, the advantage distributions of Code and Alignment are centered around 0.0 with a continuous range, whereas the Math advantage distribution mirrors its binary reward structure. Despite the heterogeneity in reward scales, the advantage normalization in REGEN provides a unified, scale-invariant learning signal across all domains.

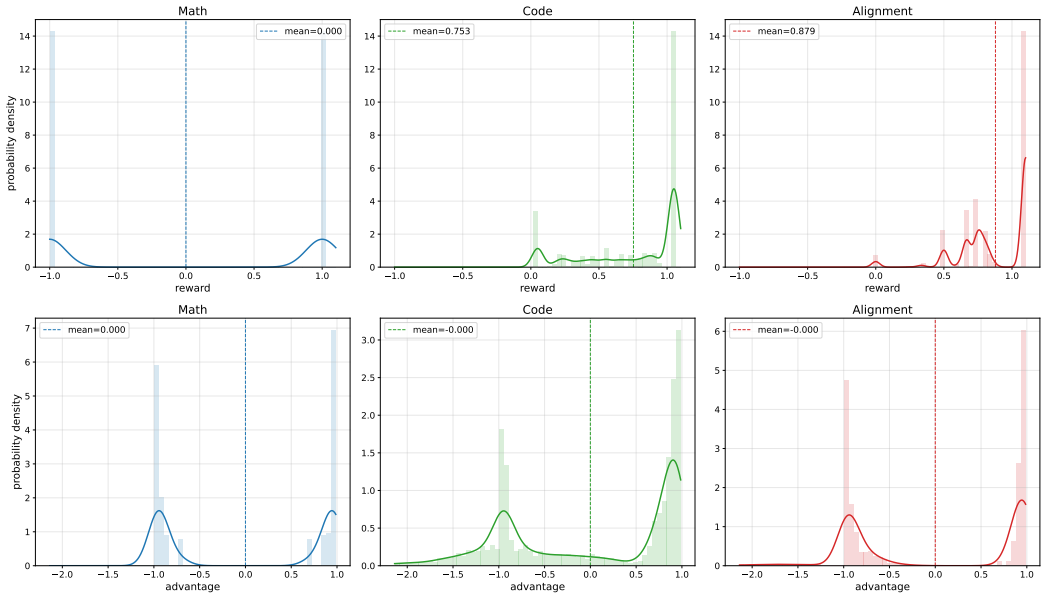


Figure 3: Reward (Top) vs Advantage (Bottom) Distribution by Domain. The dashed lines indicate the mean values.