POPULATION-BASED DIVERSITY FOR REINFORCEMENT LEARNING IN LLMS VIA SEQUENTIAL MULTI-ADAPTER OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning (RL) fine-tuning of large language models (LLMs) tends to collapse the policy onto a few strategies. This induces a severe quality-diversity (QD) trade-off: while single-policy performance improves, solution coverage (e.g., pass@k) shrinks, leaving the model brittle when its preferred strategy fails. While other methods may opt to maintain diverse modes of behavior within a single policy, we instead use a framework that sequentially builds a population of LoRA adapters over a shared frozen backbone, optimizing each policy for task performance while rewarding distribution-level diversity against previously learned policies conditioned on sufficient task competence. Unlike response-level diversity objectives, our method compares the cross-policy probability distributions, avoiding surface-level similarity metrics or additional learned diversity models. On long-horizon BabyAI tasks, our method improves held-out pass@k and is more robust to prompt-based distribution shift than independently trained policies. We also demonstrate that our approach seamlessly transfers to large-scale jailbreak red-teaming

1 INTRODUCTION

Reinforcement learning (RL) with algorithms such as PPO (Schulman et al., 2017) and GRPO (Shao et al., 2024) has become the standard way to improve large language models (LLMs) on reasoning and agentic tasks. However, RL fine-tuning tends to collapse the policy onto a few strategies, even when many valid ones exist (Cui et al., 2025; Kirk et al., 2024): the policy's entropy falls, and repeated samples increasingly follow the same trajectory. This strategy collapse directly limits pass@k, since additional attempts add little once they all make the same choices. It also makes the policy brittle: when a new environment breaks its preferred strategy, it has no alternative to fall back on (Kumar et al., 2020).

To address this, prior research aims to learn a set of policies that all solve the task well but solve it in different ways, a goal known as quality diversity (Mouret & Clune, 2015; Zhou et al., 2022). Existing approaches to this goal face two core limitations (Figure 1a). First, rule-based diversity objectives like lexical instance-level differences may ignore the rich semantic and structural knowledge within the pre-trained prior of LLMs, forcing models to exploit shortcuts such as lexical drift and length inflation (Lee et al., 2026). Second, single-policy methods rely on passive sampling: standard RL, and even objectives that optimize pass@k directly (Tajwar et al., 2026), contain no term that encourages new strategies, so any diversity comes from strategies that sampling happens to find.

To overcome these challenges, a natural direction is to train multiple policies simultaneously, rewarding each for behaving differently from the others (Kumar et al., 2020; Zhou et al., 2022; Fu et al., 2023). In classical RL, however, these methods must learn how to measure the difference between policies, through state distances or a trained discriminator (Chen et al., 2024), which requires large amounts of interaction data that are costly to collect for LLM agents. A language model does not need such a learned measure: its parameters already store a rich prior over text, and the output distribution of each policy can be compared directly.

We build on this observation (Figure 1b). We train LoRA adapters on a shared backbone one at a time, each against the frozen adapters trained before it. The diversity score is distribution-level:

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.

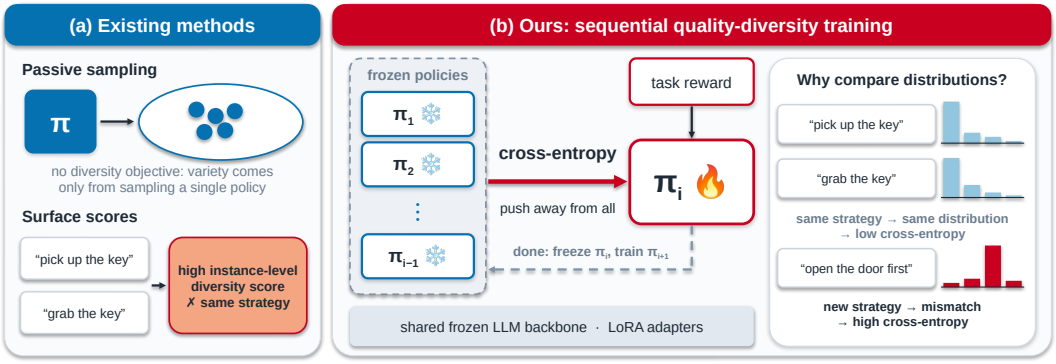


Figure 1: **(a)** Existing diversity methods either rely on sampling a single policy, reward instance-level scores that rewording can satisfy, or introduce additional overhead through learned semantic classifiers. **(b)** Our method trains policies one at a time: the current policy π_i maximizes task reward plus the cross-entropy against its frozen predecessors, and is then frozen and added to the pool. Comparing full output distributions makes superficial rewording less directly rewarded and encourages the new policy to make semantic strategy changes.

a new adapter is rewarded by the cross-entropy of its actions under the frozen adapters, that is, by how unlikely its behavior is under their output distributions. Because this score is computed from the language model's own distributions rather than from sampled text, it draws on the model's learned prior and needs neither a hand-designed similarity score nor a learned discriminator. To safeguard task performance, we introduce a performance gate. This gate activates the diversity reward only after the new adapter successfully clears a task performance threshold, ensuring that diversity does not compromise quality. Finally, maintaining multiple adapters fits naturally into existing population-based or multi-agent workflows.

We evaluate the method on BabyAI, a long-horizon, multi-turn agentic benchmark with a 3B model, and on jailbreak red-teaming with an 8B attacker. On BabyAI, we train on two task types and test on two task types never seen during training. Our adapters outperform adapters trained independently with PPO on these unseen tasks, solving up to 21 percentage points more of them when the test prompts also differ from the training format. In red-teaming, two attacks from our adapters succeed on 93.2% of 1,000 tasks, versus 80.1% for GRPO. Ablations show that neither a surface-level diversity reward nor optimizing pass@k within a single policy brings these gains.

In summary, our contributions are threefold:

1. We formulate LLM diversity as a quality-diversity problem over a population of policies: maximizing distribution-level diversity between policies subject to a task-performance constraint. We decompose it into a sequence of single-policy problems and relax the constraint through its Lagrangian into a simple gate on the diversity reward.
2. We implement this algorithm for long-horizon LLM agents with LoRA adapters on a shared frozen backbone, computing the diversity reward per token from the frozen adapters' output distributions and training with a separate value channel for diversity; the implementation works with both PPO and GRPO. Code can be found here.
3. We show that this explicit cross-policy diversity improves population-level coverage on long-horizon BabyAI tasks, achieves better performance across a variety of OOD settings, and transfers to jailbreak red-teaming. We further show that directly increasing surface-level textual diversity does not provide the same improvement in held-out task coverage.

2 Two Traps in Existing Diversity Methods

Existing approaches to diversity in RL fine-tuning fall into one of two traps (Figure 1a). The first is passive sampling: standard RL (PPO, GRPO) has no term that pushes strategies apart, while pass@k objectives such as MaxRL (Tajwar et al., 2026) retain the successful strategies that sampling happens to find but never encourage the policy to discover new ones. Keeping several strategies inside one

policy is also hard: RL steadily lowers the policy's entropy (Cui et al., 2025), and a single policy trained toward several objectives suffers from interference between their learning signals and from representational mode collapse (Ambadkar et al., 2026). On BabyAI, MaxRL matches GRPO at every k (§4.6). The second is surface scores: methods that do reward diversity compare individual responses by edit distance, n-gram overlap or a learned similarity (Li et al., 2025b; Lanchantin et al., 2025; Hu et al., 2026), and such scores can be raised by rewording without changing strategy. Rewarding POS-tag edit distance raises surface diversity over PPO and inflates response length by 57%, yet lowers out-of-distribution pass@2 from 0.609 to 0.555 (Table 1).

Our quality-diversity algorithm (Figure 1b) avoids both traps by training a population of policies sequentially, each against its frozen predecessors. A single policy defines only one output distribution, so its diversity can only be measured between samples; a population defines several, so we can add an explicit diversity term that pushes policies apart and measure it as the cross-entropy between their full output distributions. Because this signal is computed from the probabilities the language model assigns to every token, it leverages the pre-trained prior rather than the sampled strings alone. Consider two responses with the same strategy, "pick up the key" and "grab the key": an instance-level metric sees different strings and pays a large bonus, whereas the cross-entropy stays small because the reference policy already assigns high probability to both phrasings. Only a change in behavior moves the new policy away from its predecessors. Keeping a pool of policy weights adds little parameter-storage and serving overhead in multi-agent RL, which already maintain several models (Castanyer et al., 2026), so our method is naturally compatible with these settings.

3 METHODS

3.1 PRELIMINARIES

A Markov Decision Process (MDP) can be described as a tuple $(\mathcal{S}, \mathcal{A}, P, r, \gamma, \rho_0)$. Here, $\mathcal{S}$ and $\mathcal{A}$ represent the state and action space, respectively; $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition kernel; $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a reward function; $\gamma \in [0, 1)$ is the discount factor; $\rho_0(\cdot)$ is the distribution of the initial state. A trajectory is written $\tau = \{(s_t, a_t, r_t)\}_{t\geq 0}$. The goal of an RL agent is to learn a policy $\pi : \mathcal{S}\times\mathcal{A} \to [0, 1]$ that maximizes the expected return: $J_{\text{task}}(\pi) = \mathbb{E}_{s_0,a_0,s_1,\cdots}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]$, where $s_0 \sim \rho_0(\cdot), a_t \sim \pi(\cdot|s_t), s_{t+1} \sim P(\cdot \mid s_t, a_t)$. We write J_{task} for this return under the environment's task reward throughout.

3.2 PROBLEM STATEMENT: DIVERSITY AS CONSTRAINED OPTIMIZATION

Rather than learning a single policy, we train a set of K policies $\{\pi_1, \ldots, \pi_K\}$ to optimize both quality and diversity. Here, quality serves as a baseline constraint: each policy must independently achieve sufficient performance to solve the task. Diversity, quantified by a dissimilarity measure D between policy distributions (Section 3.3), encourages the policies to adopt fundamentally different strategies rather than collapsing into identical solutions. Together, these two criteria define a trade-off space, and our goal is to find K distinct policies along the Pareto frontier of quality and diversity.

Each policy is realized as a lightweight LoRA adapter over a shared frozen base model (Appendix A.1). Since the adapters are the only per-policy parameters, adding a policy to the set costs a small fraction of a full model rather than a copy of one, and K can be raised without a proportional growth in memory.

Alongside the task return J_{task} of Section 3.1, we score a set by how distinct its members are from one another. Writing D for a dissimilarity measure between two policies, instantiated in Section 3.3, the goal is to make the members of the set as mutually distinct as possible while every one of them remains competent at the task:

$$\max_{\pi_1,\ldots,\pi_K} \frac{2}{K(K-1)} \sum_{1\leq i<j\leq K} D(\pi_i, \pi_j) \qquad \text{s.t.} \qquad J_{\text{task}}(\pi_i) \geq \kappa, \quad \forall\, 1 \leq i \leq K, \tag{1}$$

where κ denotes the minimum level of task performance that each member of the set is required to reach. Eq. 1 is the objective this work aims at, and it states the quality–diversity trade-off directly: quality enters as a constraint that every member must satisfy, and diversity is the quantity maximized subject to it. Neither half is meaningful alone, since dropping the constraint is satisfied by policies that no longer solve the task and dropping the objective is satisfied by K copies of a single policy.

Eq. 1 cannot be optimized as written. It couples all K policies through $\binom{K}{2}$ pairwise terms, so every policy's diversity signal moves as the others update; it fixes K in advance, so a set cannot be extended without re-solving the whole problem; and its constraint is hard, offering no gradient once satisfied and no diversity signal while violated. Section 3.4 derives the algorithm we actually run as a sequence of relaxations of Eq. 1.

3.3 Cross-Adapter Diversity Reward

We take D to be the accumulative cross-entropy of the current policy against a reference,

$$D(\pi_i, \pi_j) \;=\; \mathbb{E}_{\tau\sim\pi_i}\Big[-\sum\nolimits_t \log \pi_j(a_t \mid s_t)\Big], \tag{2}$$

which scores the current policy's actions under the reference policy's own distribution. The comparison is therefore made between distributions rather than between generated texts, and it is evaluated using the reference model itself rather than a hand-designed surface statistic.

Eq. 2 is an expectation over whole trajectories, but the summand is already per-token, so it can be collected as a dense reward during training. Averaging over the k previously trained frozen reference policies, the diversity reward at token t within a turn is

$$r_t^{\text{div}} = -\frac{1}{k}\sum_{j=1}^{k} \log \pi_j(a_t \mid s_t), \qquad a_t \sim \pi_{\text{curr}},$$

We use cross entropy rather than a KL-based diversity objective. The two differ by the actor's own entropy, $H(\pi_i\|\pi_j) = H(\pi_i) + D_{\mathrm{KL}}(\pi_i\|\pi_j)$, so maximizing forward KL implicitly rewards lowering $H(\pi_i)$, while maximizing cross entropy does not. Given that entropy collapse is one of the failure modes this work sets out to avoid, an objective that pays for it is the wrong starting point. Appendix A.6 gives the fuller comparison. Because the reward is dense and per-token, diversity credit is confined to the turn in which it is earned and does not accumulate across turns of a trajectory.

3.4 From the Set Objective to a Practical Algorithm

We now relax Eq. 1 into a form that can be optimized with standard single-policy reinforcement learning, addressing its three obstacles in turn: the coupling across policies, the fixed set size, and the hard constraint.

Step 1: sequential decomposition. We build the set one policy at a time. When training the i-th policy, the earlier policies $\pi_1, \ldots, \pi_{i-1}$ are held frozen and enter only as references, so the pairwise terms among them are constant and the terms involving policies not yet trained are unavailable. Maximizing the terms that couple π_i to its predecessors leaves

$$\max_{\pi_i} \;\; J_{\text{div}}(\pi_i) = \frac{1}{i-1}\sum_{j=1}^{i-1} D(\pi_i, \pi_j) \qquad \text{s.t.} \qquad J_{\text{task}}(\pi_i) \geq \kappa_i, \tag{3}$$

where π_i (synonymous with π_{curr}) is the current policy adapter being optimized and π_j for $j < i$ a previously trained and frozen reference policy within the set. Each stage is now an ordinary single-policy problem against a stationary reference, and the set can be extended by appending a policy without retraining its existing members. The relaxation is greedy: π_i cannot influence the policies that preceded it, so the resulting set need not be the joint optimum of Eq. 1. The per-stage threshold κ_i also replaces the single κ, allowing the requirement on each policy to be set from what the set has already achieved. Finally, because policies are added one at a time, the size of the set need not be fixed in advance: training can continue until an additional policy no longer improves the set, whereas methods that optimize the set jointly must choose K before training begins.

Step 2: Lagrangian relaxation of the constraint. The constraint in Eq. 3 is still hard. Applying the Lagrangian multiplier method to relax it turns the problem into a minimax optimization of the generalized Lagrangian function,

$$\min_{\lambda\geq 0}\; \max_{\pi_i} \;\; \mathcal{L}(\pi_i, \lambda) = J_{\text{div}}(\pi_i) + \lambda\big(J_{\text{task}}(\pi_i) - \kappa_i\big). \tag{4}$$

For any $\lambda > 0$, dividing through by λ and discarding the constant κ_i leaves an inner problem with the same maximizer,

$$\max_{\pi_i} \; J_{\text{task}}(\pi_i) + \tfrac{1}{\lambda} J_{\text{div}}(\pi_i), \tag{5}$$

so the multiplier sets nothing more than the relative weight of the two returns.

Step 3: a switching surrogate for the multiplier. Solving Eq. 4 requires maintaining and updating λ alongside π_i, introducing a second quantity that must itself be tuned and stabilized. We instead fix the weight and switch it on and off. Writing $w_{\text{div}} \geq 0$ for the fixed counterpart of $1/\lambda$ in Eq. 5, and defining $\phi(\pi_i) = \mathbb{1}[J_{\text{task}}(\pi_i) \geq \kappa_i]$, produces the surrogate objective

$$J_{\text{switch}}(\pi_i) = J_{\text{task}}(\pi_i) + w_{\text{div}} \phi(\pi_i) J_{\text{div}}(\pi_i). \tag{6}$$

Eq. 6 is a two-valued approximation of the dual variable. While the constraint is violated the gate is closed, $w_{\text{div}}\phi = 0$, which corresponds to $\lambda \to \infty$ in Eq. 5 and places all weight on restoring feasibility; once the constraint is met the gate opens and the trade-off is held at $\lambda = 1/w_{\text{div}}$.

This places our formulation in the switching-objective family introduced by Reward Switching Policy Optimization (Zhou et al., 2022), of which the present method can be read as a large language model counterpart: each policy is trained against frozen predecessors, and the diversity incentive is switched on and off over the course of training rather than blended in at a fixed weight throughout. Relative to soft objectives, which weigh task and diversity returns against each other at every step and are correspondingly sensitive to the relative scale of the two reward landscapes, a switching objective keeps the two signals separated in time. The two formulations differ in where the switch is placed. RSPO constrains diversity and optimizes the environment objective; we place the switch on the other side, activating a diversity incentive only once the policy's task performance clears a threshold. This choice is motivated by the setting: over the long generations typical of multi-turn LLM tasks, a trajectory-level switch leaves room for non-task tokens to act as a cheap source of diversity, and gating on task performance closes that route by construction.

Step 4: estimating the gate during training. Gating the diversity incentive on task performance preserves foundational structure such as format and relevance in token outputs, as opposed to optimizing for diversity from the start. For training stability we evaluate ϕ against a mean of expected returns over a rolling window of training steps rather than a single batch, and hold the gate closed for a short warmup at the start of each policy's phase. In practice we also cap the diversity incentive from above. Without such a cap, even with the performance gate active, the diversity term eventually pushes the policy to regress on learned behaviors optimal for the environment, most evident in reduced abilities to follow format instructions. We therefore scale w_{div} by a factor that ramps smoothly to zero as the current adapter's average divergence from the previously trained adapters approaches a target level D_{target}, rather than cutting the incentive off abruptly. We treat this as a measure for training stability rather than a component of the objective itself, and defer its definition to Appendix A.7. The threshold κ_i is a hyperparameter of the i-th training phase, set from the performance already attained by the preceding policy so that the requirement placed on each new adapter is calibrated to what the set has demonstrated to be achievable; Appendix A.5 gives the exact rule and the values used.

As a note, we apply diversity incentives independently of the task rewards, creating two GAE channels for the critic, allowing for weighted blend of advantages, so as to not modify the original reward function (§A.4)

4 Experiment Results

4.1 Experimental Setup

Benchmark and training. We evaluate on BabyAI, a long-horizon agentic benchmark. Policies are trained on the *pickup* and *goto* task families of `BabyAI-MixedTrainLocal-v0` and evaluated on the held-out families *open* and *pick_up_seq_go_to*. Each policy is a LoRA adapter with scaling $\alpha = 32$ over a shared frozen Qwen2.5-3B-Instruct backbone, trained with the PPO implementation of Verlog (Chen et al., 2025), a multi-turn RL framework for long-horizon LLM agents; the remaining hyperparameters are given in Appendix A.2. The method itself is compatible with both PPO and GRPO, and we use PPO here.

Evaluation. To test out-of-distribution robustness, policies are trained with a single-step history ($h = 1$) and evaluated with histories of $h \in \{1, 2, 4\}$ steps, corresponding to Low, Medium, and High shift. At turn t, we format the observation by including h past turns, $\text{obs}_t = \{s_{t-h}, a_{t-h}, \ldots, s_{t-1}, a_{t-1}, s_t\}$. Increasing h alters the prompt structure without changing the environment, instructions, or action space. The Low shift matches the training format, isolating task novelty as the main challenge, while Medium and High shifts test how well the policy generalizes to longer historical contexts unseen during training. We report pass@k, but define it over a set of k distinct policies (one sample per policy) rather than k samples from a single policy. This maintains the exact same generation budget (k generations per task) while shifting the source of diversity from temperature sampling to policy variance.

Baselines. We compare against: (1) **PPO** (Schulman et al., 2017) and **GRPO** (Shao et al., 2024), standard single-policy RL algorithms without diversity incentives; (2) **MaxRL** (Tajwar et al., 2026), which optimizes pass@k within a single policy rather than a population; and (3) **SMERL** (Kumar et al., 2020), an information-theoretic method maximizing mutual information via a latent skill (adapted to LMs via prompt prefixes). Implementation details for all baselines are given in Appendix A.3.

4.2 RQ1: Does Explicit Policy Diversity Translate into Higher Pass@k?

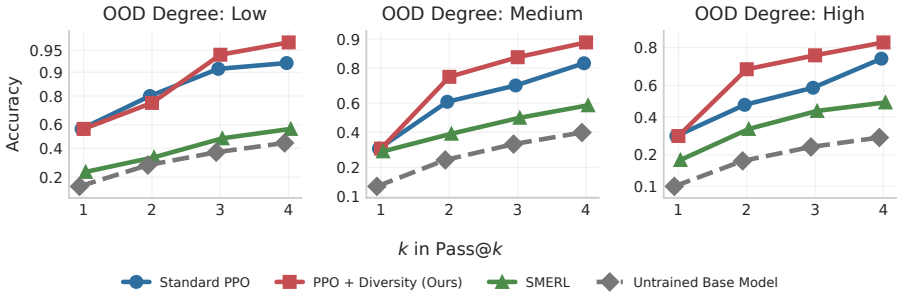


Figure 2: Pass@k averaged over the held-out task families (*open* and *pick_up_seq_go_to*), evaluated at three degrees of distribution shift (corresponding to prompt history lengths 1, 2, 4 respectively). Higher is better.

Training a set of adapters with explicit diversity can be shown to improve pass@k over training them independently. Figure 2 reports pass@k on held-out tasks at three degrees of distribution shift. At pass@1 our method matches Standard PPO, which is expected since each adapter is trained with the same RL recipe. The advantage appears at $k \geq 2$: because the diversity objective encourages each adapter to cover a different region of the solution space, the set compensates for individual blind spots. At $k = 2$ under Medium shift our method reaches 0.75 versus 0.61 for PPO; under High shift the gap is wider, 0.68 versus 0.47. By $k = 4$ the figures are 0.89 versus 0.82 (Medium) and 0.83 versus 0.74 (High). The margin grows with the severity of the distribution shift: at $k = 2$ the gap over PPO is -0.06 (Low), $+0.14$ (Medium) and $+0.21$ (High). We analyze this trend in §4.3. For in-distribution tasks all methods exceed 0.94 (Appendix A).

SMERL (Kumar et al., 2020) underperforms all other methods, including standard PPO. SMERL requires training a state discriminator $q_\phi(z \mid s)$ alongside the policy to provide the diversity reward; until the discriminator learns to separate the latents, the reward is uninformative. In our setting this warm-up takes an order of magnitude higher than the training budget used by the other methods, making it impractical for LLM-scale RL(A.12). Our cross-entropy signal requires no learned component: the reference adapters are frozen, so the diversity reward is exact from the first update.

4.3 RQ2: Does Diversity Training Improve Robustness to Distribution Shift?

Our method is more robust to distribution shift than Standard PPO. Figure 3 fixes k and varies the degree of shift from Low to High. Under Low shift the two methods perform similarly, but as the shift increases Standard PPO degrades sharply: at $k = 2$ it drops from 0.80 (Low) to 0.48 (High), losing 0.32 absolute, while our method drops from 0.75 to 0.68, losing only 0.07. The pattern holds at larger k: at $k = 4$, PPO falls from 0.93 to 0.74 (-0.19) while ours falls from 0.97 to 0.83 (-0.14).

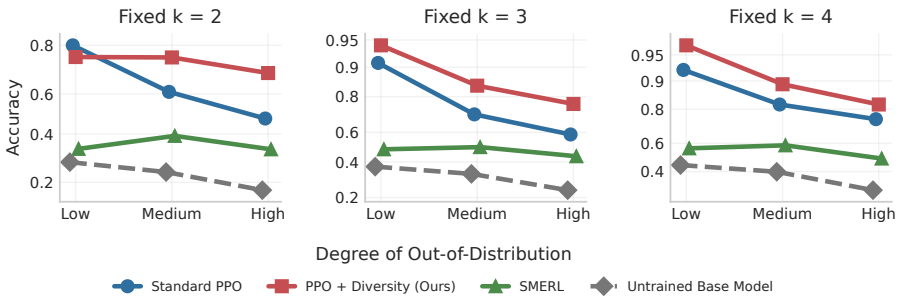


Figure 3: Robustness of population pass@k as the degree of distribution shift (prompt history length) increases (Low $\rightarrow$ Medium $\rightarrow$ High), at fixed $k \in \{2, 3, 4\}$. Our method degrades more slowly than Standard PPO.

Notably, our method under the hardest shift at $k = 4$ (0.83) still exceeds PPO's performance under Low shift at $k = 2$ (0.80).

No method has seen out-of-distribution trajectories during training. PPO adapters, trained independently on the same distribution, learn correlated strategies that break similarly under shift. The diversity objective decorrelates their failure modes: each adapter is pushed to cover strategies its predecessors miss, so the set degrades gracefully despite no single adapter trained on OOD data.

4.4 RQ3: Does Distribution-Level Diversity Better Leverage the Pre-Trained LLM Prior Than Surface-Level Metrics?

Table 1 highlights risks associated with using metrics like surface level textual diversity directly as diversity rewards, fostering reward hacking, as seen with mild length inflation, along with a failure to improve generalization given the performance in OOD prompt tasks.

In contrast, our distribution level objective still induces improvements in surface level diversity metrics, without ever directly optimizing BLEU or part-of-speech differences, but also yields gains in OOD performance. These results suggest distribution-level diversity produces substantial lexical variation as a by-product of policy separation rather than as an explicitly optimized target.

Table 1: Surface-level vs. distribution-level diversity objective on BabyAI. Overall Pass@2 averages all tasks with history length 1; OOD averages held-out tasks with history length 2 (Medium shift). BLEU-div $=$ $1 -$ BLEU and POS-div (length-normalized POS-tag edit distance) averaged over adapter pairs; response length in tokens.

	POS-div	**PPO**	**Ours**
Ovr. Pass@2↑	0.870	0.900	0.880
OOD Pass@2↑	0.555	0.609	0.758
BLEU-div↑	0.924	0.867	0.973
POS-div↑	0.598	0.525	0.605
Resp. length	61.7	39.2	42.6

Together, these results empirically address RQ3, suggesting that building on the pre-trained prior of the LLM can help avoid rewarding semantically similar phrasings that the model already considers likely, while simultaneously mitigating the length inflation or format collapse observed with surface-level objectives. These experiments further demonstrate that the learned behaviors transfer better to various OOD settings.

4.5 RQ4: Does Diversity Scale to Jailbreak Red-Teaming?

The diversity objective scales to a larger and harder domain: 8B-parameter jailbreak red-teaming, where the reward signal, action space, and model scale all differ substantially from BabyAI. The attacker is Meta-Llama-3.1-8B-Instruct-abliterated, fine-tuned with LoRA adapters. The defender is a frozen Mistral-7B-Instruct-v0.2, evaluated on 1,000 red-teaming tasks. All adapters are trained on the same 1,000 tasks regardless of iteration; diversity comes entirely from the objective, not from different data. We evaluate with *budget-matched coverage*: given a generation budget of B forward passes per task, coverage is the fraction of tasks with at least one successful attack. Our method and GRPO both maintain a pool of K adapters and query each adapter once, so budget $B{=}K$ reflects K distinct policies. MaxRL trains a single model and draws K samples from it, so the same budget measures sampling diversity within one policy rather than strategic diversity across policies. This

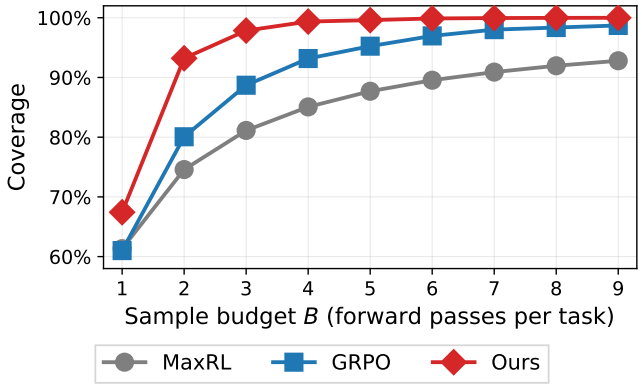


Figure 4: Budget-matched task coverage on 1,000 jailbreak red-teaming tasks. Each method is charged the same generation budget B per task. A pool distributes B forward passes across its members; a single model spends the whole budget on itself. Coverage is the fraction of tasks with at least one successful attack.

Table 2: MaxRL versus GRPO on BabyAI: pass@k with all k samples from one policy, on in-distribution (ID) and held-out (OOD) tasks.

	ID			OOD		
k	Base	GRPO	MaxRL	Base	GRPO	MaxRL
1	0.669	0.891	0.906	0.154	0.273	0.274
2	0.868	0.988	0.995	0.273	0.430	0.423
3	0.931	0.998	1.000	0.369	0.525	0.524
4	0.953	1.000	1.000	0.445	0.586	0.594

Table 3: Diversity reward on all versus passing-only trajectories (jailbreak, 40 steps). Pass@1: attack success rate of the adapter; pass@2: pooled with π_1; refusal: fraction of attacks the defender refuses.

Applied to	**Pass@1↑**	**Pass@2↑**	**Refusal↓**
Passing only	0.341	0.494	0.514
All (ours)	0.396	0.523	0.435

distinction matters for interpreting the comparison: part of the advantage of pool-based methods over MaxRL comes from maintaining multiple models, not just from the training objective.

Figure 4 reports the results. At $B = 1$ all methods are comparable ($\sim$61–67%). At $B = 2$ our method already reaches 93.2% coverage versus 80.1% for a GRPO pool and 74.6% for MaxRL, and it saturates near 100% by $B = 4$. Our method at $B = 2$ already exceeds what MaxRL achieves at $B = 9$ (92.8%), a saving of more than $4\times$ in generation cost. The cleanest comparison is between our method and GRPO: both use the same algorithm, pool size, and evaluation protocol; the only difference is the diversity term. The $+13.2$ gap at $B = 2$ therefore isolates the contribution of the diversity objective itself.

4.6 Ablation Studies

Is a pass@k objective within one policy enough? MaxRL (Tajwar et al., 2026) optimizes pass@k but samples from a single policy. Our method is agnostic to the policy-gradient algorithm and runs with either PPO or GRPO, whereas MaxRL's objective is defined over groups of rollouts and requires a GRPO-style estimator. Our main experiments use Verlog (Chen et al., 2025), a PPO-based codebase optimized for long-horizon agentic tasks, on which PPO reaches substantially higher performance than a naive GRPO implementation. We therefore compare MaxRL with GRPO separately. MaxRL stays within 0.02 of GRPO at every k (Table 2): 75% of GRPO's groups already have six or more successes out of eight (Appendix Figure 6), leaving few rare successes to reweight. We expect the same for PPO, so pass@k gains require explicit diversity across policies.

Diversity on all versus passing trajectories. Applying the diversity bonus to all trajectories works better than applying it only to successful ones (Hu et al., 2026; Yao et al., 2026). With the restriction, pass@2 drops from 0.523 to 0.494 and the fraction of attacks refused by the defender rises from 0.435 to 0.514 (Table 3, single runs). In jailbreak red-teaming most attacks fail, so new strategies must be found among failed attempts. The diversity bonus on these failures acts as an

exploration bonus, encouraging the policy to try attacks it has not tried before. If only successful attacks receive the bonus, the policy is instead rewarded for rewording attacks that already work, and these rewordings are refused more often.

Sensitivity to w_{div}. From $w_{\text{div}} = 0.05$ to 0.5, pass@2 varies by at most 0.013 (Appendix Table 6), so the weight needs little tuning. At this short 40-step budget no setting clearly exceeds $w_{\text{div}} = 0$; the gains in §4.5 come from adapters trained to convergence.

5 Related Work

Quality diversity in RL. Quality-diversity RL learns a set of policies that are both high-performing and distinct. Early methods maintain archives of behaviors, as in Novelty Search and MAP-Elites (Lehman & Stanley, 2011; Mouret & Clune, 2015); population methods such as MEP and DvD optimize several policies jointly with entropy- or kernel-based diversity terms (Zhao et al., 2023; Parker-Holder et al., 2020); and skill-discovery methods such as DIAYN and SMERL separate policies by a latent variable identified by a learned discriminator (Eysenbach et al., 2019; Kumar et al., 2020; Laskin et al., 2022). Constrained formulations replace fixed reward weights with adaptive constraints or Lagrangian objectives (Tessler et al., 2019; Zahavy et al., 2023), and sequential methods such as RSPO and SIPO build the set one policy at a time against frozen predecessors (Zhou et al., 2022; Fu et al., 2023). These methods measure diversity through behavior descriptors, visited states or a learned discriminator, and were developed for dense-reward control. Our method follows the sequential, switching line of RSPO, but measures diversity with the frozen language-model policies themselves, so no extra component is learned and the signal is available for sparse-reward, long-horizon LLM tasks.

Diversity and pass@k in LLM post-training. RL fine-tuning tends to sharpen an LLM onto a few successful modes (Kirk et al., 2024); with GRPO, already-likely correct trajectories are reinforced more strongly than rare ones (He et al., 2025). Proposed remedies stay within a single policy: MaxRL optimizes a pass@k objective (Tajwar et al., 2026), the choice of regularizing divergence affects mode coverage (Li et al., 2025a), and diversity rewards based on group frequency, surface metrics, semantic partitions or preference reweighting encourage varied responses (Anschel et al., 2025; Hu et al., 2026; Li et al., 2025b; Lanchantin et al., 2025). All of them must keep every mode inside one output distribution, and most score diversity per response, the two traps discussed in §2. We instead place distinct strategies in distinct policies and compare their probability output distributions.

Quality diversity for LLM agents. Rainbow Teaming (Samvelyan et al., 2024) applies MAP-Elites to red-teaming at the prompt level without training the attacker, and QDRT (Wang et al., 2025) trains multiple RL attackers, each assigned a region of a predefined behavior grid. Both define diversity through a fixed taxonomy of risk categories and attack styles; our objective requires no such grid. Populations of LoRA adapters are already standard in multi-agent LLM training (Castanyer et al., 2026), so the pool our method maintains adds little beyond what these settings already keep. An extended discussion is given in Appendix A.13.

6 Conclusion

We introduced a sequential population-based approach for quality-diversity reinforcement learning in LLM settings, where we sequentially train lightweight adapters to remain sufficiently task-competent while differing from frozen predecessors at the policy-distribution level. This explicit cross-policy diversity improves population-level coverage on long-horizon BabyAI tasks, remains effective under distribution shift, and transfers well to jailbreak red-teaming.

One limitation lies with how the training cost grows with the number of policies, as each additional adapter must be compared with all previously learned policies. In practice, this makes the method most attractive for relatively small populations, while scaling to much larger values of K may require more efficient reference aggregation or approximation.

AI USE DISCLOSURE

Generative AI tools were used to assist with drafting and editing the manuscript, including improving clarity, organization, and presentation, and with literature discovery. AI assistance was also used for feedback on experimental design and to support code design, such as debugging and implementation suggestions. The authors reviewed and verified the resulting code, experiments, analyses, and manuscript content.

REFERENCES

Tanmay Ambadkar, Sourav Panda, Shreyash Kale, Jonathan Dodge, and Abhinav Verma. Preference conditioned multi-objective reinforcement learning: Decomposed, diversity-driven policy optimization. *arXiv preprint arXiv:2602.07764*, 2026.

Oron Anschel, Alon Shoshan, Adam Botach, Shunit Haviv Hakimi, Asaf Gendler, Emanuel Ben Baruch, Nadav Bhonker, Igor Kviatkovsky, Manoj Aggarwal, and Gerard Medioni. Group-aware reinforcement learning for output diversity in large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 32394–32415, 2025.

Roger Creus Castanyer, Geoffrey Bradway, Lorenz Wolf, Maxwill Lin, Augustine N. Mavor-Parker, and Matthew James Sargent. PopuLoRA: Co-evolving llm populations for reasoning self-play. *arXiv preprint arXiv:2605.16727*, 2026.

Wentse Chen, Shiyu Huang, Yuan Chiang, Tim Pearce, Wei-Wei Tu, Ting Chen, and Jun Zhu. DGPO: Discovering multiple strategies with diversity-guided policy optimization. In *AAAI*, 2024.

Wentse Chen, Jiayu Chen, Hao Zhu, and Jeff Schneider. Verlog: Context-lite multi-turn reinforcement learning framework for long-horizon LLM agents. In *First Workshop on Multi-Turn Interactions in Large Language Models*, 2025. URL https://openreview.net/forum?id=GmodkWwMV3.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, Zhiyuan Liu, Hao Peng, Lei Bai, Wanli Ouyang, Yu Cheng, Bowen Zhou, and Ning Ding. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. In *ICLR*, 2019.

Wei Fu, Weihua Du, Jingwei Li, Sunli Chen, Jingzhao Zhang, and Yi Wu. Iteratively learn diverse strategies with state distance information. *arXiv preprint arXiv:2310.14509*, 2023.

Andre Wang He, Daniel Fried, and Sean Welleck. Rewarding the unlikely: Lifting grpo beyond distribution sharpening. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 25548–25560, 2025.

Zican Hu, Shilin Zhang, Yafu Li, Jianhao Yan, Xuyang Hu, Leyang Cui, Xiaoye Qu, Chunlin Chen, Yu Cheng, and Zhi Wang. Diversity-incentivized exploration for versatile reasoning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=9G7AbBrd27.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. Understanding the effects of RLHF on LLM generalisation and diversity. In *ICLR*, 2024.

Saurabh Kumar, Aviral Kumar, Sergey Levine, and Chelsea Finn. One solution is not all you need: Few-shot extrapolation via structured maxent rl. In *NeurIPS*, 2020.

Jack Lanchantin, Angelica Chen, Shehzaad Dhuliawala, Ping Yu, Jason Weston, Sainbayar Sukhbaatar, and Ilia Kulikov. Diverse preference optimization. *arXiv preprint arXiv:2501.18101*, 2025.

Michael Laskin, Hao Liu, Xue Bin Peng, Denis Yarats, Aravind Rajeswaran, and Pieter Abbeel. Cic: Contrastive intrinsic control for unsupervised skill discovery. *arXiv preprint arXiv:2202.00161*, 2022.

Sangmook Lee, Minbeom Kim, Jeonghye Kim, Dohyung Kim, Sojeong Rhee, and Kyomin Jung. Are we measuring strategy or phrasing? the gap between surface-and approach-level diversity in llm math reasoning. *arXiv preprint arXiv:2606.29985*, 2026.

Joel Lehman and Kenneth O. Stanley. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation*, 19(2):189–223, 2011.

Long Li, Zhijian Zhou, Jiaran Hao, Jason Klein Liu, Yanting Miao, Wei Pang, Xiaoyu Tan, Wei Chu, Zhe Wang, Shirui Pan, Chao Qu, and Yuan Qi. The choice of divergence: A neglected key to mitigating diversity collapse in reinforcement learning with verifiable reward. *arXiv preprint arXiv:2509.07430*, 2025a.

Tianjian Li, Yiming Zhang, Ping Yu, Swarnadeep Saha, Daniel Khashabi, Jason Weston, Jack Lanchantin, and Tianlu Wang. Jointly reinforcing diversity and quality in language model generations. *arXiv preprint arXiv:2509.02534*, 2025b.

Jean-Baptiste Mouret and Jeff Clune. Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909*, 2015.

Jack Parker-Holder, Aldo Pacchiano, Krzysztof Choromanski, and Stephen Roberts. Effective diversity in population based reinforcement learning. In *NeurIPS*, 2020.

Mikayel Samvelyan, Sharath C Raparthy, Andrei Lupu, Eric Hambro, Aram H Markosyan, Manish Bhatt, Yuning Mao, Minqi Jiang, Jack Parker-Holder, Jakob Foerster, et al. Rainbow teaming: Open-ended generation of diverse adversarial prompts. *Advances in Neural Information Processing Systems*, 37:69747–69786, 2024.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

DJ Strouse, Kate Baumli, David Warde-Farley, Volodymyr Mnih, and Steven Stenberg Hansen. Learning more skills through optimistic exploration. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=cU8rknuhxc.

Fahim Tajwar, Guanning Zeng, Yueer Zhou, Yuda Song, Daman Arora, Yiding Jiang, Jeff Schneider, Ruslan Salakhutdinov, Haiwen Feng, and Andrea Zanette. Maximum likelihood reinforcement learning. *arXiv preprint arXiv:2602.02710*, 2026.

Chen Tessler, Daniel J. Mankowitz, and Shie Mannor. Reward constrained policy optimization. In *ICLR*, 2019.

Ren-Jian Wang, Ke Xue, Zeyu Qin, Ziniu Li, Sheng Tang, Hao-Tian Li, Shengcai Liu, Zhi Yu, Yuanpeng Tan, and Chao Qian. Quality-diversity red-teaming: Automated generation of high-quality and diverse attackers for large language models. *arXiv preprint arXiv:2506.07121*, 2025.

Jian Yao, Ran Cheng, Xingyu Wu, Jibin Wu, and Kay Chen Tan. Diversity-aware policy optimization for large language model reasoning. *Advances in Neural Information Processing Systems*, 38: 94801–94826, 2026.

Tom Zahavy, Yannick Schroecker, Feryal Behbahani, Kate Baumli, Sebastian Flennerhag, Shaobo Hou, and Satinder Singh. Discovering policies with domino: Diversity optimization maintaining near optimality. In *ICLR*, 2023.

Rui Zhao, Jinming Song, Yufeng Yuan, Haifeng Hu, Yang Gao, Yi Wu, Zhongqian Sun, and Wei Yang. Maximum entropy population-based training for zero-shot human-ai coordination. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 6145–6153, 2023.

Zihan Zhou, Wei Fu, Bingliang Zhang, and Yi Wu. Continuously discovering novel strategies via reward-switching policy optimization. In *ICLR*, 2022.

A Appendix

A.1 Multi-Adapter LoRA Architecture

Full fine-tuning scales poorly with the number of policies k, with memory limits and managing independent LLMs likely creates a system that is overly complex. Instead, employing LoRA adapters offers a feasible solution that allows all policies to share a frozen backbone. Policies are trained sequentially, with prior adapters frozen for reference for the current policy.

Each policy slot consists of an actor LoRA adapter and a critic LoRA adapter, with trainable value head. During initialization of a new policy slot, we perform a critic warmup phase to fit the new critic adapter's estimates before training the actor adapter.

A.2 Training and Diversity Hyperparameters

Policies are LoRA adapters of rank 64 and scaling $\alpha = 32$ over a frozen Qwen2.5-3B-Instruct backbone. Training uses PPO with GAE and a low-variance KL penalty against the base model, a rollout batch of 256, a micro-batch of 16, two PPO epochs per update and 16 parallel environments. Adapters are trained sequentially, with a new adapter slot activated every 150 updates.

The diversity reward is the per-token cross-entropy of the current policy's actions under the frozen reference adapters, clipped at 5.0 nats per token, and enters the policy update through its own value head with blend weight $w_{\text{div}} = 0.25$. The performance gate is evaluated on a rolling window of recent updates, is held closed for the first 20 updates of each adapter's phase, and thereafter opens when the rolling task performance clears $\kappa_i = 0.4 \cdot \hat{V}_{i-1}$. All RQ1 experiments use this token-level cross-entropy diversity objective rather than a text-based diversity metric. We train a population of $K = 4$ sequential adapters, consisting of the initial task-trained policy followed by three diversity-trained policies.

A.3 Baseline Implementation Details

All baselines are trained in the same framework, on the same tasks, and with the same backbone, LoRA configuration and optimization settings as our method, so that the comparison isolates the training objective. Each adapter is trained for 150 iterations, with each iteration collecting 256 turn-level environment interactions per update (38,400 nominal interactions). GRPO and MaxRL instead perform 24 updates of eight groups with eight complete trajectories each (64 episodes per update; 1,536 episodes total, each trajectory capped at 50 steps). Their smaller number of outer updates reflects the greater sampling performed per grouped update and was chosen to approximately match the PPO environment-interaction budget with similar wall-clock training time under identical hardware. All methods reuse each collected batch for two optimization epochs.

For SMERL we use four discrete latents with a uniform prior, assigned to rollouts by a stratified round-robin sampler, and condition the policy on the latent through a short opaque prompt prefix. The diversity bonus is $\alpha\,[\log q_\phi(z \mid s) - \log p(z)]$ with $\alpha = 0.05$, where q_ϕ is a state discriminator trained alongside the policy, and following the original formulation it is applied only once task performance clears a threshold, evaluated on a rolling window of recent episodes using task success. The discriminator operates on a 72-dimensional compact BabyAI state encoding containing normalized agent position and direction, carried-object identity, and relative object features; it does not consume LM hidden states or generated text. We parameterize (q_ϕ) as a three-layer MLP, $(72 \to 256 \to 256 \to 4)$, with GELU activations, and optimized with AdamW with learning rate 3×10^{-4}. After each PPO update, we perform 40 full-batch discriminator updates, with no replay buffer or held-out discriminator split. SMERL is trained for 200 PPO updates with rollout batch size 256, corresponding to approximately 51,200 environment transitions. In practice, we also evaluated an auxiliary latent-classification head over frozen LM embeddings, but find its predictive accuracy lower than the state-space discriminator, which we use for the main SMERL baseline.

A.4 Two-Channel Critic

A single reward stream involving the task reward with a supplemented diversity reward creates the classic challenge of reward scaling and shifting of the reward landscape, especially when paired with

further threshold indicator functions, effectively causing the critic to learn to predict inconsistent returns. The two-channel distinction is key to separating the two rewards to help mitigate some of the challenges with balancing the task reward landscape with the injected intrinsic reward for diversity.

Specifically, this two-channel design allows us to operate on two independent scales for the diversity and task reward. With a shared backbone due to the implementation of critic adapters with trainable value heads, we scale down the diversity terms to limit the effect of diversity relevant updates to the shared backbone that would perturb accurate predictions to the return of the environment rewards.

A detail is that advantages are first whitened for each channel before a weighted sum, then re-whitened as the final set of advantages.

A.5 Threshold Selection

The performance threshold κ_i gating the diversity incentive for the i-th policy is set from the performance already attained by the preceding policy. Concretely, we take a fixed proportion of the highest mean critic value $\hat{V}_{i-1}$ recorded over the previous policy's training phase,

$$\kappa_i \;=\; c \cdot \hat{V}_{i-1},$$

with c a constant held fixed across a run. Anchoring the threshold to the previous policy calibrates the requirement placed on each new adapter to what the set has already demonstrated to be achievable on the task, rather than to a value fixed in advance that may be unreachable or trivially satisfied. The first policy has no predecessor and instead uses a constant threshold supplied in the configuration.

We evaluate the current policy against κ_i using the mean of the critic's value estimates over a rolling window of recent training steps rather than a single batch, and hold the gate closed for a short warmup period at the start of each phase while the value estimates settle.

A.6 Cross Entropy versus KL as the Diversity Objective

Writing the cross entropy of the current policy against a reference in terms of the forward KL divergence gives

$$H(\pi_i\|\pi_j) \;=\; H(\pi_i) + D_{\mathrm{KL}}(\pi_i\|\pi_j),$$

so the two candidate diversity objectives differ exactly by the actor's own entropy. Maximizing $D_{\mathrm{KL}}(\pi_i\|\pi_j) = H(\pi_i\|\pi_j) - H(\pi_i)$ rewards moving away from the reference while simultaneously rewarding a reduction in $H(\pi_i)$: a policy can increase the objective by sharpening onto a single alternative mode. Since entropy collapse under RL fine-tuning is one of the failure modes this work sets out to avoid, an objective carrying that incentive is a poor starting point, and we therefore maximize cross entropy directly.

The same decomposition shows what cross entropy buys and what it costs. It does not penalize the actor's entropy, but it does reward it: a policy can raise $H(\pi_i\|\pi_j)$ by flattening its own distribution without moving away from the reference at all. Cross entropy is, in this sense, a dissimilarity score rather than a divergence, and it does not vanish when $\pi_i = \pi_j$, where it takes the value $H(\pi_i)$. The performance gate of Section 3.4 bounds how far this can be exploited, since a policy that has flattened to the point of failing the task loses the diversity incentive altogether.

A.7 Optional Smoothing

Often, even with a performance gate, without a reasonable diversity cap, the added diversity incentives eventually push the policy to regress on learned behaviors optimal for the environment, most evident in reduced abilities to follow format instructions.
This yields the introduction of a diversity target threshold (set as a multiple of the diversity score of a freshly initialized adapter w.r.t. previous trained adapters) which we will denote D_{target}. Then we have

$$\bar{r}_n^{\mathrm{div}} := \frac{1}{T_n}\sum_{t=0}^{T_n} r_{n,t}^{\mathrm{div}}, \qquad\qquad D_t = \frac{1}{|\mathcal{B}|}\sum_{n\in\mathcal{B}} \bar{r}_n^{\mathrm{div}}$$

where $\mathcal{B}$ is the current batch and $|\mathcal{B}|$ the number of turns in it, and T_n the number of valid response tokens in turn n. This denotes the average cross-adapter negative log likelihood over the current batch. The effective diversity weight is then

$$w_{\text{div}}^{\text{eff}} = w_{\text{div}} \cdot \max\left(0,\ 1 - \frac{D_t}{D_{\text{target}}}\right).$$

so that as the current adapter's divergence from past adapters approaches D_{target}, the diversity weight is ramped smoothly to zero rather than cut off abruptly.

A.8 Additional Plots

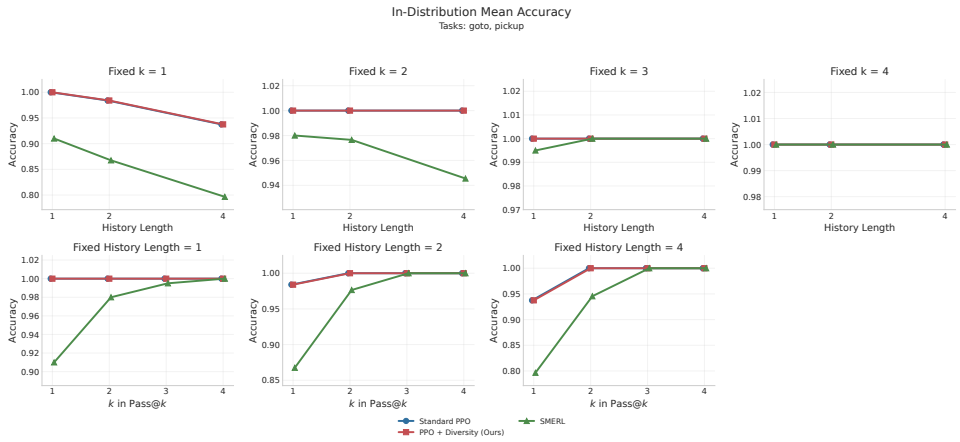


Figure 5: In Distribution Tasks

A.9 GRPO Group-Success Distribution

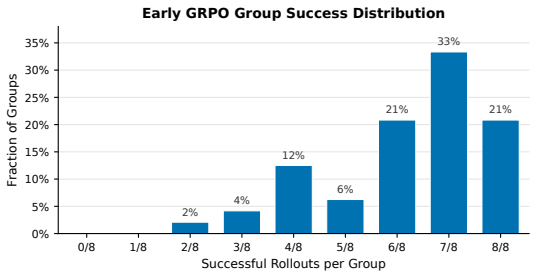


Figure 6: Number of successful rollouts per group of eight during the first half of GRPO training on BabyAI. Most groups are already largely successful, leaving MaxRL few rare successes to reweight (§4.6).

A.10 Surface-Level Diversity Detailed Results

Table 4: **Task coverage under Medium OOD shift.** Pass@2 evaluated over 64 fixed seeds per task family. ID Mean averages *goto* and *pickup*; OOD Mean averages *open* and *pick_up_seq_go_to*.

Method	ID Mean	open	pick_up_seq_go_to	OOD Mean
Levenshtein (Reduced Weight)	1.000	0.453	0.656	0.555
PPO Population	1.000	0.594	0.625	0.609
Ours (CE)	1.000	0.750	0.766	0.758

Table 5: **Diversity objective diagnostics.** Overall pass@2 and response-level diversity metrics. "Strong Weight" matches the CE diversity coefficient; "Reduced Weight" downweights the Levenshtein diversity term relative to task reward.

Setting	Overall Pass@2	Edit Dist.	Avg. Length
Levenshtein (Strong Weight)	0.721	0.7766	173.80
Levenshtein (Reduced Weight)	0.870	0.5983	61.72
Standard PPO Pair	0.900	0.5252	39.19
PPO + Diversity	0.880	0.6051	42.63

A.11 Sensitivity to Diversity Weight

Table 6: Sensitivity to the diversity weight w_{div} (jailbreak, same seed and data, 40 steps, pass@2 pooled with π_1). $w_{\text{div}} = 0$: mean and range over 5 runs.

w_{div}	**Pass@1**	**Pass@2**
0	0.367 (0.296–0.448)	0.503 (0.476–0.535)
0.05	0.394	0.514
0.2	0.399	0.510
0.5	0.343	0.501

A.12 SMERL Analysis

This training required to learn an accurate latent discriminator is particularly costly in our LLM setting. The matched SMERL run collects only $51{,}200$ environment transitions over 200 policy iterations and compensates with 40 discriminator updates per iteration, with each resampling the batch. Even then, post-update accuracy averages 71.1% over the final 50 iterations and drops to 62.1% on the next batch before refitting. By comparison, a two-latent classical-RL diagnostic reaches 91.3% held-out accuracy only after roughly 2.05 million transitions, about $40\times$ more interaction data. This suggests learning an accurate discriminator requires more varied state data than is practical in LLM-scale RL (Strouse et al., 2022).

A.13 Extended Related Work

Prior work on multi-LoRA and population-based methods demonstrates that maintaining a population of models, where each model fulfills a specialized role, is essential for achieving strong performance in MARL. Because our framework adopts this necessary population-based structure, the computational overhead introduced by running multiple specialized models is both expected and justified. (Castanyer et al., 2026)

Quality Diversity and Multi-Policy Learning in Classical Reinforcement Learning: QD-RL seeks to learn a collection of distinct high performing and distinct policies rather than overfitting an optimal solution, with earlier work exploring behavioral diversity such as Novelty Search and MAP-Elites (Lehman & Stanley, 2011; Mouret & Clune, 2015). Population-based methods such as MEP and DvD jointly optimize multiple policies while encouraging behavioral diversity across the population via population entropy and kernel-based diversity objectives. (Zhao et al., 2023; Parker-Holder et al., 2020). However, these methods must balance a reward-diversity tradeoff, and may be difficult to scale to LLM settings where model hosting, adapter switching, and rollout generation dominate training expense. Recently, Rainbow Teaming (Samvelyan et al., 2024) applies MAP-Elites to LLM red-teaming but operates at the prompt level without training attacker policies, while QDRT (Wang et al., 2025) extends this by RL-training multiple attackers partitioned by predefined behavioral descriptors. Both rely on discrete taxonomies (risk category, attack style) to define diversity, whereas our method uses token-level cross-entropy between output distributions and requires no predefined behavior grid.

Constrained RL offers a way to balance task performance and diversity, using adaptive constraints or Lagrangian objectives rather than fixed reward weights (Tessler et al., 2019; Zahavy et al., 2023).

Sequential methods such as RSPO and SIPO leverage this concept and construct diverse policy sets iteratively, training new policies against previously learned behaviors using trajectory or state-based diversity signals (Zhou et al., 2022; Fu et al., 2023). The benefits of this paradigm mitigate issues with joint optimization, but are primarily studied in dense reward RL settings, and its applicability to sparse reward settings especially in long-horizon LLM tasks remains comparatively unexplored.

Other popular QD-RL approaches separate policies through the use of latent priors. DIAYN (Eysenbach et al., 2019) establishes the concept of defining latent variables, inducing distinguishable behaviors by maximizing mutual information between visited states and latent variables. Building on this idea, Kumar et al. (2020) makes use of such a learned discriminator with a gated performance objective for quality-diversity learning. In LLM settings, however, the higher computational costs associated with gathering trajectories and different reward structures often introduce issues when scaling the method, given the large amount of interaction data required to reliably distinguish an increasingly diverse set of behaviors (Laskin et al., 2022).

Diversity and Pass@k in LLM Post-Training. Work in LLM post-training has focused on preserving response quality while encouraging diverse responses through resampling, countering tendencies of RL to simply sharpen successful modes of the distribution (Kirk et al., 2024). He et al. (2025) identify a similar effect with GRPO, showing that already-likely correct trajectories can receive disproportionately strong reinforcement while rarer successful trajectories are underweighted, while MaxRL(Tajwar et al., 2026) tackles this problem from a maximum likelihood perspective, formulating a pass@k performance objective over successful rollouts for a single policy. Alternate studies highlight the effects of the choice of divergence used for policy regularization on diversity and mode coverage (Li et al., 2025a).

A parallel line of work incorporates diversity directly into post-training objectives within a single policy using group-frequency signals, surface-level response metrics, learned semantic partitions, or preference reweighting (Anschel et al., 2025; Hu et al., 2026; Li et al., 2025b; Lanchantin et al., 2025). While these methods differ in how diversity is measured, they all require retaining multiple behavioral modes within the same policy distribution, while creating potential for reward hacking.

In contrast, our method distributes diversity across policies, which directly encourages separation between their output distributions. This eliminates the need to preserve all modes within a single policy, reduces chances for interference between two diverse behavioral modes, and does not require obtaining or maintaining an additional semantic classifier. Additionally, as opposed to most group-based methods, our method places no requirement on the underlying policy-gradient algorithm, whereas group-based methods often cannot be carried over to actor-critic settings.