MILO: MODE-WISE LIKELIHOOD OPTIMIZATION FOR POST-TRAINING DIVERSITY

Anonymous authors
Paper under double-blind review

ABSTRACT

People often ask language models for multiple candidate solutions, but sampling more responses does not necessarily produce more ways of solving a problem: many samples may repeat the same underlying strategy. We introduce **Mode-wise Likelihood Optimization (MILO)**, a post-training method for increasing the diversity of correct solution strategies while preserving accuracy. MILO groups verified-correct responses that use the same strategy into *correct modes*, gives each mode its own log-likelihood term, and combines the resulting mode-wise objective with pooled correctness using a weight λ. Its mode-wise update gives each observed mode equal total weight, rather than favoring modes simply because they appear more often.

We show that this update directly targets correct-mode coverage. For finite rollout groups, its expected mode-wise update is the gradient of a $1/k$-weighted sum of expected CorrectModes@k, where CorrectModes@k counts the distinct correct modes among k samples. At a local correctness optimum, a small MILO step with $\lambda > 0$ improves this coverage objective to first order when its gradient is nonzero, while correctness changes only at second order.

On Qwen3-4B-Base GSM8K, $\lambda = 0.01$ improves both single-sample accuracy and CorrectModes@64, raising coverage from 8.4 to 14.6. Increasing λ to 0.1 raises coverage to 22.6 while keeping single-sample accuracy comparable to the base model and pass@64 nearly unchanged. The gains extend across datasets, sampling budgets, and independent mode labelers, and are also reflected by labeler-free diversity metrics. A controlled verifier-blind-spot experiment further illustrates how broader correct-mode coverage can help candidate pools retain usable correct answers when a verifier fails on common strategies.

1 INTRODUCTION

A common way to make progress on a difficult problem is to find a new way to represent or reason about it. Galois theory, for example, used group theory to characterize which polynomial equations can be solved by radicals (Artin, 2011). Similarly, Erdős used probability to answer a question about graphs: instead of constructing a graph with no large clique or independent set, he showed that a random graph has this property with nonzero probability, so such a graph must exist (Erdős, 1947). These examples suggest a broader point: having multiple ways to approach the same problem can reveal structure that is hard to see from a single perspective.

When people use large language models (LLMs) to solve difficult problems (Cobbe et al., 2021; Shao et al., 2024; Guo et al., 2025), it is often useful to consider several candidate solutions rather than rely on a single answer. This is already common in practice: self-training and debate generate multiple candidates and use verifiers, selectors, or LLM judges to choose among them (Brown et al., 2024; Schaeffer et al., 2025; Snell et al., 2025; Feng et al., 2026; Liang et al., 2024). Repeated sampling can also improve the chance of finding a correct answer, while self-consistency selects the most frequent final answer across samples (Wang et al., 2023; Brown et al., 2024). But sampling more responses does not necessarily produce more ways of solving the problem: a large candidate pool may still contain many variations of the same underlying strategy.

We call a group of correct responses that use the same solution strategy a *correct mode*. For a pool of k sampled responses, CorrectModes@k measures mode coverage by counting the distinct correct

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.

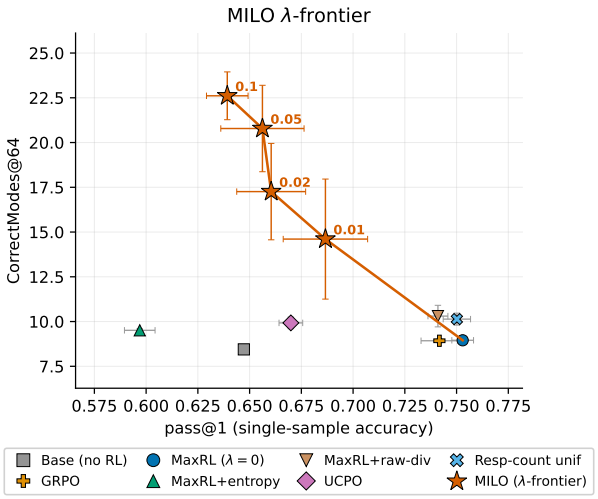


Figure 1: **Mode coverage and single-sample accuracy** on Qwen3-4B-Base GSM8K. MILO at $\lambda = 0.01$ improves both over the base model; $\lambda = 0.1$ gives higher coverage at comparable accuracy. All plotted points have pass@64 ≥ 0.996. Error bars show ± 1 standard error of the mean (SEM) over seeds.

modes that appear. For example, algebraic derivation, case analysis, and geometric reasoning may all reach the same answer but represent different modes. Having several such strategies can be useful downstream: if a verifier fails on one correct strategy, another may still be accepted, and a broader pool can also provide more varied data for self-training. Related evidence comes from multi-agent systems, where heterogeneous agents can provide complementary evidence (Yang et al., 2026), and a diverse starting set can help debate begin from a correct hypothesis (Zhu et al., 2026).

Current post-training methods, however, can reduce this variety. Prior work finds substantial output homogeneity within and across models (Jiang et al., 2025), and post-training can improve quality while concentrating the output distribution or reducing response diversity (Kirk et al., 2024; Huang et al., 2025; Li et al., 2025; Wang et al., 2024; Springer et al., 2026). This motivates our central question: can post-training increase CorrectModes@k while preserving pass@k, the probability of obtaining at least one correct response in k samples?

The difficulty is that standard correctness objectives do not distinguish how correct probability is distributed across modes. MaxRL, which we use as a pooled-correctness baseline, has this property (Tajwar et al., 2026). More broadly, prior work finds that reinforcement learning with verifiable rewards can improve accuracy while reducing the diversity of correct solutions available under repeated sampling (Yue et al., 2025). Exploration methods instead encourage broader sampling through repeated interaction (Xiong et al., 2023; Xie et al., 2024; Weng, 2020). Our focus is different: we use verified-correct samples to train the model to increase correct-mode coverage directly.

We introduce **Mode-wise Likelihood Optimization (MILO)**. Rather than treating all correct responses as one event, MILO gives each correct mode its own log-likelihood term and combines the mode-wise objective with pooled correctness using a weight λ. In the mode-wise update, gradients are averaged within each observed mode, so every observed mode receives the same total weight regardless of how often it appears. Common strategies therefore do not receive more weight in this part of the update simply because they are sampled more often (Section 3).

Our analysis connects this update directly to correct-mode coverage (Section 4). For a rollout group of size N, the expected MILO update is the gradient of a finite-sample objective that approaches the distributional objective as N grows. Its mode-wise term is, up to a constant, a $1/k$-weighted sum of expected CorrectModes@k over $k = 1, \ldots, N$. Moreover, at a local correctness optimum, a small MILO step with $\lambda > 0$ improves this coverage objective to first order when its gradient is nonzero, while correctness changes only at second order (Theorems 1 and 2).

On Qwen3-4B-Base GSM8K, MILO at $\lambda = 0.1$ increases CorrectModes@64 from 8.4 for the base model to 22.6, while pass@64 remains nearly unchanged (1.000 vs. 0.997) and pass@1 remains comparable. With a smaller weight, $\lambda = 0.01$, MILO improves both pass@1 and mode coverage, reaching 14.6 correct modes (Fig. 1 and Section 5.2). Across the full λ sweep, pass@64 remains at least 0.996 (Section 5.3). The coverage gains extend across datasets, sampling budgets, and independent mode labelers, and are also visible under labeler-free diversity metrics. Finally, a controlled verifier-blind-spot experiment illustrates one downstream use: when a verifier rejects common correct modes, MILO's broader candidate pools are more likely to retain a usable correct answer (Section 5.6).

2 Related work

Pooled correctness and pass@k. Repeated sampling is widely used to improve reasoning at inference time, for example through self-consistency (Wang et al., 2023), inference scaling (Brown et al., 2024), and particle-filter search (Golowich et al., 2026). These methods benefit from assigning more probability mass to high-quality responses, and recent work has studied how this probability mass arises and can be improved (Chen et al., 2026). RL with verifiable rewards can substantially improve reasoning accuracy (Guo et al., 2025), but it can also reduce the diversity of correct solutions available under repeated sampling (Yue et al., 2025; Wang et al., 2026). Methods such as MaxRL (Tajwar et al., 2026), Pass@k Policy Optimization (Walder & Karkhanis, 2025), and Pass@k Training (Chen et al., 2025) optimize pooled correctness: they increase the total probability of generating a correct response. For a fixed partition of correct responses into modes, however, these objectives do not distinguish how that probability is distributed across modes (Proposition 1). MILO is designed to optimize this distribution directly. We use "coverage" to mean coverage of distinct correct strategies; this differs from the use of coverage in online RL, where it typically refers to visiting states needed for learning (Xie et al., 2022).

Diversity and exploration. Post-training can reduce the diversity of model outputs (Kirk et al., 2024), motivating methods that explicitly encourage broader exploration. Existing approaches do this in several ways: by regularizing entropy (Williams & Peng, 1991; Mnih et al., 2016), rewarding semantic diversity or rare outcomes (Hu et al., 2026; Song et al., 2025), matching feature statistics (Jelassi et al., 2026), or optimizing diversity at the trajectory or set level (Hamid et al., 2026; Li et al., 2026). Other methods diversify generation through staged procedures (Cao et al., 2026), diversity-aware preference pairs (Lanchantin et al., 2025), preservation of alternative token-level candidates (Peng et al., 2025), or target reward distributions (Zhu et al., 2025). Related ideas also appear in determinantal point processes, population-based diversity, and quality-diversity methods (Kulesza & Taskar, 2012; Parker-Holder et al., 2020; Mouret & Clune, 2015). Multi-agent systems similarly use diversity to obtain different or complementary reasoning paths (Liang et al., 2024; Yang et al., 2026; Zhu et al., 2026). These methods encourage diversity in different forms. MILO instead focuses specifically on *mode coverage*: it partitions correct responses by solution strategy and directly optimizes probability mass across these modes.

Response- and strategy-level objectives. Some recent methods move beyond pooled correctness by explicitly shaping how probability is distributed among correct outputs. UCPO (Uniform-Correct Policy Optimization), for example, encourages uniformity over individual correct responses (Lochab et al., 2026). This differs from MILO when many responses express the same underlying strategy: UCPO treats them as separate outputs, whereas MILO groups them into one mode and gives that mode a single log-likelihood term (Section 3.2). Other work represents strategy diversity through latent variables learned with mutual information (Shah et al., 2025), or studies whether surface-level diversity metrics reflect differences in reasoning strategy (Lee et al., 2026). MILO instead starts from an explicit partition of correct responses into solution strategies and optimizes the probability assigned to each mode directly. Our analysis further shows that, at a local correctness optimum, the mode-wise objective can improve to first order while correctness changes only at second order (Theorem 2). This connects MILO to multi-objective optimization, where one seeks to improve a secondary objective while preserving a primary one (Désidéri, 2012; Boyd & Vandenberghe, 2004).

3 MODEL AND METHOD

3.1 POST-TRAINING SETUP AND CORRECTNESS

We post-train a language model initialized from a base model with parameters θ_{base}. During training, the current parameters θ define the sampling policy π_θ. For a prompt x drawn from the prompt distribution ρ, the model samples a rollout (token sequence) $z \sim \pi_\theta(\cdot \mid x)$, which decodes to a response $y = f(z)$. A ground-truth verifier checks the response: $r(x, y) = \mathbf{1}\{a(y) = a^\star(x)\}$, where $a(y)$ is the extracted answer and $a^\star(x)$ is the reference answer.

Given an inference sampling budget k, let $y_1, \ldots, y_k$ be the responses obtained from k independent rollouts from $\pi_\theta(\cdot \mid x)$. For this candidate pool, define

$$\text{Pass@}\,k(x; y_{1:k}) = \mathbf{1}\left\{\sum_{i=1}^{k} r(x, y_i) \geq 1\right\}. \tag{1}$$

Pass@k indicates whether the sampled pool contains at least one correct response. Using Pass@k to measure repeated-sampling correctness, our goal is to increase the diversity of correct responses without decreasing Pass@k.

3.2 CORRECT MODES AND MODE-WISE LIKELIHOOD OPTIMIZATION

Correct responses to the same prompt may reach the answer using different solution strategies. We say that two correct responses belong to the same *correct mode* if they use the same solution strategy. To make this grouping operational, we use a fixed mode labeler $\phi(x, y) \in \mathcal{M}_x$ on correct responses, where $\mathcal{M}_x$ is the finite set of mode labels for prompt x. The label $\phi(x, y)$ depends only on the prompt and the response. Both ϕ and $\mathcal{M}_x$ are fixed throughout training, so the mode partition does not change with θ.

For the candidate pool $y_{1:k}$, let $\mathcal{C}_k(x; y_{1:k}) = \{\phi(x, y_i) : 1 \leq i \leq k,\ r(x, y_i) = 1\}$ be the set of distinct modes among the correct responses. We define

$$\text{CorrectModes@}\,k(x; y_{1:k}) = |\mathcal{C}_k(x; y_{1:k})|. \tag{2}$$

Thus, Pass@k measures whether the pool contains any correct response, while CorrectModes@k measures the diversity of correct responses. In particular, $\text{Pass@}\,k = \mathbf{1}\{\text{CorrectModes@}\,k \geq 1\}$. Our goal is to increase correct-response diversity without decreasing Pass@k.

We use independently constructed mode labelers for training and evaluation. The training labeler provides the mode labels used by the MILO update below. The evaluation labeler is used only to compute CorrectModes@k on held-out prompts and is never accessed during training. Appendix B.2 gives the construction of both labelers, and appendix C.4 tests sensitivity to the labeling choices.

To achieve the goal above, we define a training objective that controls both total correctness and how correct probability is distributed across modes. For a fixed prompt x and current parameters θ, let $p_m(\theta; x) = \Pr_{z \sim \pi_\theta(\cdot|x)}[r(x, f(z)) = 1,\ \phi(x, f(z)) = m]$ denote the probability of generating a correct response in mode m. The total probability of generating a correct response is $p(\theta; x) = \Pr_{z \sim \pi_\theta(\cdot|x)}[r(x, f(z)) = 1] = \sum_{m \in \mathcal{M}_x} p_m(\theta; x)$. As the model is post-trained, these probabilities change with θ, while the mode-labeling rule ϕ and the label set $\mathcal{M}_x$ remain fixed.

MILO combines pooled and mode-wise log-likelihood terms. For a fixed prompt x, define $J_{\text{Pool}}(\theta; x) = \log p(\theta; x)$ and $J_{\text{Mode}}(\theta; x) = \sum_{m \in \mathcal{M}_x} \log p_m(\theta; x)$. The MILO distributional objective is $J_\lambda(\theta; x) = (1 - \lambda) J_{\text{Pool}}(\theta; x) + \lambda J_{\text{Mode}}(\theta; x)$, where $\lambda \in [0, 1]$ controls the trade-off between total correctness and mode coverage. We average this objective over prompts $x \sim \rho$; Section 4 derives the finite-sample objective corresponding to the update used in training. The pooled term depends only on the total correct probability, whereas the mode-wise term rewards probability assigned to each correct mode.

3.3 FINITE-SAMPLE MILO UPDATE

The objective above contains $\log p_m(\theta; x)$, so it is only defined when every mode has positive probability. In practice, however, MILO is trained from finite rollout groups. The finite-sample update

below, and the objective it estimates as derived in Section 4, remain well defined even when some modes have zero probability.

For each training prompt, we draw N independent rollouts $z_1, \dots, z_N \sim \pi_\theta(\cdot \mid x)$ and set $y_i = f(z_i)$ and $r_i = r(x, y_i)$. Here, N is the training rollout-group size, distinct from the inference sampling budget k. For each correct response, let $m(i) = \phi(x, y_i)$. Let $K = \sum_{i=1}^{N} r_i$ be the total number of correct rollouts, and let $K_m = \sum_{i:r_i=1} \mathbf{1}\{m(i) = m\}$ be the number of correct rollouts in mode m.

MILO weights each rollout's score vector $\nabla_\theta \log \pi_\theta(z_i \mid x)$ with the advantage A_i to form its policy-gradient estimate:

$$A_i = (1-\lambda)\frac{r_i}{K} + \lambda \frac{r_i}{K_{m(i)}} - \frac{1}{N}, \qquad \hat{g}_\lambda = \sum_{i=1}^{N} A_i \nabla_\theta \log \pi_\theta(z_i \mid x). \tag{3}$$

For an incorrect response, both success terms are defined as zero, and when $K = 0$ they are zero for all responses. The quantity $K_{m(i)}$ is evaluated only for correct responses. The coefficients A_i are held fixed when differentiating the sampled log probabilities.

At $\lambda = 0$, the update reduces to our MaxRL pooled-correctness baseline (Tajwar et al., 2026). The difference is how correct rollouts are weighted. The pooled term gives every correct rollout weight $1/K$, so a mode that appears more often receives more total weight. The mode-wise term instead gives a rollout weight $1/K_m$, so all rollouts from the same mode together receive total weight one. For example, if six correct rollouts belong to one mode and two to another, pooled weighting gives the two modes total weights $6/8$ and $2/8$, while mode-wise weighting gives $6(1/6) = 2(1/2) = 1$. Thus, seeing more samples from the same mode does not give that mode more total weight. Since each observed mode contributes total weight one, the scale of the mode-wise term depends on how many modes appear in the rollout group (appendix A.1).

4 Why the update improves mode coverage

We now analyze the MILO update. Throughout this section, we fix a prompt x and its mode labeler, omit $(\theta; x)$ from the notation, and write $M = |\mathcal{M}_x|$. We assume parameter-independent policy support and valid differentiation under the expectation when applying the score identity.

4.1 Finite-sample objective

The distributional objective in Section 3.2 is defined in terms of the probabilities p and p_m, while the update is computed from only N sampled rollouts. We first show that the sampled update is an unbiased gradient estimator of a finite-sample version of the MILO objective.

For $q \in [0, 1]$, define $J^{(N)}(q) = -\sum_{k=1}^{N} (1-q)^k/k$, $J^{(N)}_{\text{Mode}} = \sum_{m \in \mathcal{M}_x} J^{(N)}(p_m)$, and $J^{(N)}_\lambda = (1-\lambda)J^{(N)}(p) + \lambda J^{(N)}_{\text{Mode}}$.

Theorem 1 (Finite-sample MILO objective)**.** *For the MILO update $\hat{g}_\lambda$ in Eq. (3),*

$$\mathbb{E}[\hat{g}_\lambda] = \nabla_\theta J^{(N)}_\lambda.$$

Moreover, for $q > 0$, $J^{(N)}(q) \to \log q$ as $N \to \infty$. Thus, $J^{(N)}_\lambda \to J_\lambda$ when every $p_m > 0$.

Proof. Consider the mode-wise part of the update for a correct mode m. The mode contributes only if it appears among the N rollouts, which happens with probability $1 - (1-p_m)^N$. Conditional on appearing, its rollouts receive total weight one, so their contribution is their average score vector. By the score identity, this average has expectation $\nabla p_m / p_m$. Therefore, for $p_m > 0$, the expected contribution of mode m is $[1 - (1-p_m)^N]\nabla p_m / p_m = \sum_{k=1}^{N} (1-p_m)^{k-1} \nabla p_m = \nabla J^{(N)}(p_m)$. Summing over modes gives $\nabla J^{(N)}_{\text{Mode}}$.

The same argument applied to all correct responses together gives $\nabla J^{(N)}(p)$ for the pooled term. The baseline has zero expectation because $\mathbb{E}[\nabla_\theta \log \pi_\theta(z \mid x)] = \nabla_\theta \sum_z \pi_\theta(z \mid x) = 0$. Combining the three terms gives $\mathbb{E}[\hat{g}_\lambda] = \nabla_\theta J^{(N)}_\lambda$.

Finally, $J^{(N)}(q)$ is the first N terms of the expansion $\log q = -\sum_{k=1}^{\infty}(1-q)^k/k$, so $J^{(N)}(q) \to \log q$ for $q > 0$. Appendix A gives the full derivation, including the case $p_m = 0$. □

The finite-sample objective also gives a useful interpretation of the update in terms of mode coverage. A correct mode m appears at least once among k independent samples with probability $1-(1-p_m)^k$, so $\mathbb{E}[\text{CorrectModes@}\,k] = \sum_{m\in\mathcal{M}_x}[1-(1-p_m)^k]$. It follows that $J^{(N)}_{\text{Mode}}$ differs from $\sum_{k=1}^{N} k^{-1}\mathbb{E}[\text{CorrectModes@}\,k]$ only by a constant. Therefore,

$$\nabla J^{(N)}_{\text{Mode}} = \sum_{k=1}^{N} \frac{1}{k} \nabla \mathbb{E}[\text{CorrectModes@}\,k]. \tag{4}$$

Thus, the finite-sample mode-wise update can also be viewed as jointly optimizing expected correct-mode coverage over sampling budgets $1, \ldots, N$, with weight $1/k$ on budget k.

4.2 The local trade-off with correctness

From the definition of J_λ, the intuition is simple: if the total correct probability p were fixed, moving a small amount of probability mass from a common correct mode to a rarer one would increase J_{Mode} and make distinct correct modes more likely to appear in a sampled pool. In a parameterized model, however, changing the probability assigned to one mode can also change the total correct probability. We therefore study a MILO step at a local correctness optimum. At such a point, the first-order change in p is zero, while the mode-wise objective may still have a nonzero gradient.

Theorem 2 (Local improvement in mode coverage)**.** *Suppose the $p_m(\theta)$ are twice continuously differentiable near an interior local maximizer $\theta^\star$ of p, with $p_m(\theta^\star) > 0$ for every mode. Then $\nabla p(\theta^\star) = 0$. Let $u = \nabla J_{\text{Mode}}(\theta^\star)$. For $\theta(\eta) = \theta^\star + \eta\lambda u$,*

$$J_{\text{Mode}}(\theta(\eta)) = J_{\text{Mode}}(\theta^\star) + \eta\lambda\|u\|^2 + O(\eta^2), \qquad p(\theta(\eta)) = p(\theta^\star) + O(\eta^2).$$

Thus, if $\lambda > 0$ and $u \neq 0$, a sufficiently small MILO step improves the mode-wise objective to first order, while correctness changes only at second order. The same statement holds for the finite-sample objective $J^{(N)}_{\text{Mode}}$, with u replaced by $u_N = \nabla J^{(N)}_{\text{Mode}}(\theta^\star)$.

Proof sketch. At $\theta^\star$, the pooled term has zero gradient because $\nabla J_{\text{Pool}} = \nabla p/p = 0$. Hence the MILO direction is λu. A first-order Taylor expansion of J_{Mode} along this direction gives the first expression. Since $\nabla p(\theta^\star) = 0$, the linear term in the Taylor expansion of p vanishes, giving the second. For the finite-sample objective, the pooled gradient is also a scalar multiple of ∇p, so the same argument applies. Appendix A.3 gives the full quantitative bound. □

By Eq. (4), the finite-sample result also gives a first-order improvement in the weighted expected CorrectModes@k objective optimized by the update. For fixed k, $\mathbb{E}[\text{Pass@}\,k] = 1-(1-p)^k$ is a smooth function of p, so its change is likewise second order near $\theta^\star$. Thus, MILO can improve weighted expected mode coverage while preserving expected Pass@k to first order.

5 Experiments

In this section, we evaluate whether MILO increases CorrectModes@k while preserving pass@k on language models, and whether these gains persist across datasets, sampling budgets, and mode labelers. We further test whether broader mode coverage makes candidate pools more robust to verifier blind spots, by measuring whether they still contain a usable correct response when a verifier rejects some common correct modes (Section 5.6).

5.1 Experimental setup

Training and baselines. Our main experiments post-train Qwen3-4B-Base on GSM8K (Cobbe et al., 2021). We use on-policy REINFORCE based on REINFORCE Leave-One-Out (RLOO) (Ahmadian et al., 2024), replacing its advantage with the MILO advantage in Eq. (3). Each training step samples $N = 8$ rollouts for one prompt, using temperature 1.0, top-$p = 1.0$, and a maximum

generation length of 512 tokens. We train for 500 steps with AdamW at learning rate 10^{-6} and no KL penalty. Unless specified otherwise, MILO uses $\lambda = 0.1$.

We compare against Group Relative Policy Optimization (GRPO) (Shao et al., 2024), MaxRL ($\lambda = 0$), MaxRL+entropy, MaxRL+raw-diversity, UCPO, and a response-count-uniform ablation. All methods use the same training prompts, optimizer, and rollout budget; only MILO uses the mode labels $\phi(x, y)$. Appendix B.1 gives the baseline definitions, coefficients, seed counts, and additional training details; appendix C.2 adds outcome-based exploration and energy-based fine-tuning.

Mode labeling. We implement the fixed mode labeler ϕ from Section 3.2 using a frozen *strategy descriptor*: Qwen2.5-0.5B-Instruct for training and Qwen2.5-1.5B-Instruct for evaluation. Both use greedy decoding and produce at most 24 tokens describing the strategy used by a correct response, such as substitution, elimination, or case analysis. We embed this description using all-MiniLM-L6-v2 and assign it to the nearest prototype in a fixed bank of 256 clusters, producing the mode label $\phi(x, y)$.

The training and evaluation labelers are constructed independently. The training cluster bank is built from Qwen3-1.7B-Base responses on the training split, whereas the evaluation bank is built from untrained Qwen3-4B-Base responses on the test split. Thus, the response bank, strategy descriptor, and cluster prototypes used for evaluation are all fixed independently of MILO training. Appendix B.2 gives the full construction; appendix C.4 reports ablations over the descriptor, embedding, and number of clusters.

Evaluation. Our main evaluation uses the first 128 test prompts, with $n = 96$ responses per prompt at temperature 1.0 and top-$p = 1.0$. We compute CorrectModes@k on the first k responses using the fixed evaluation mode labeler. Our primary coverage metric is CorrectModes@64. Thus, our main evaluation budget $k = 64$ is substantially larger than the training rollout-group size $N = 8$.

We use lowercase pass@k for the expected value of the binary Pass@k defined in Section 3.1. If c of the $n = 96$ sampled responses are correct, we use the estimator $1 - \binom{n-c}{k} / \binom{n}{k}$. We report pass@1 as single-sample accuracy, and pass@16, pass@64, and the area under the pass@k curve (AUP) as repeated-sampling metrics. TopModeShare is the largest mode's share of correct responses in the first 64 samples (zero if none are correct). Reported results are averaged over test prompts and training seeds. For comparisons with MaxRL, we assess non-inferiority using confidence intervals on the difference in pass@k; appendix B.2 gives the full procedure.

5.2 Main comparison on GSM8K

MILO substantially increases correct-mode coverage while preserving repeated-sampling correctness (Fig. 2). At $\lambda = 0.1$, CorrectModes@64 increases from 8.4 for the base model to 22.6, while pass@64 remains nearly unchanged (1.000 vs. 0.997) and pass@1 remains comparable (0.647 vs. 0.639). The six trained baselines reach at most 10.3 correct modes. Relative to MaxRL, the repeated-sampling metrics satisfy non-inferiority at an absolute margin of 0.01; Table 2 reports the confidence intervals and stricter-margin tests. Consistent with the broader coverage, TopModeShare decreases from 0.478 for MaxRL to 0.257 for MILO, indicating that correct responses are less concentrated in a single mode.

The headline result in Fig. 1 also shows that this coverage gain need not come at the expense of single-sample accuracy. With a smaller coverage weight $\lambda = 0.01$, pass@1 improves over the base model while CorrectModes@64 increases from 8.4 to 14.6.

5.3 The effect of λ after full training

We sweep $\lambda \in \{0, 0.01, 0.02, 0.05, 0.1, 0.2, 0.5\}$ in Fig. 3. Across the sweep, pass@64 remains at or above 0.996, while CorrectModes@64 increases from 9.0 at $\lambda = 0$ to 24.1 at $\lambda = 0.5$. At $\lambda = 0.01$, MILO improves both pass@1 and mode coverage over the base model. At $\lambda = 0.1$, pass@64 remains nearly unchanged from $\lambda = 0$ (0.997 vs. 0.998), while pass@1 decreases from 0.753 to 0.639 as mode coverage increases. Larger λ places more weight on the mode-wise objective; in this sweep, coverage increases while single-sample accuracy decreases and repeated-sampling correctness remains nearly unchanged. Theorem 2 describes the effect of a small MILO step at a local correctness optimum; here we measure the resulting trade-off after full post-training.

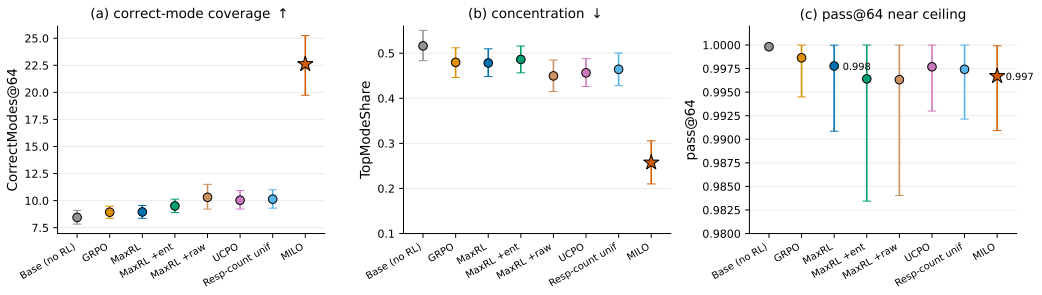


Figure 2: Main comparison on Qwen3-4B-Base GSM8K. **(a)** CorrectModes@64, **(b)** TopModeShare, and **(c)** pass@64. Error bars are 95% hierarchical bootstrap confidence intervals; gray bars show the base model.

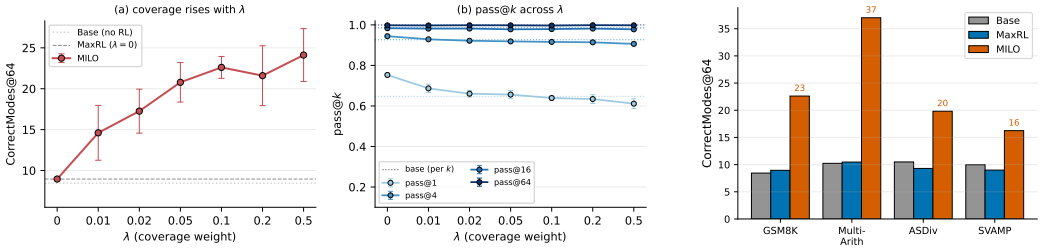


Figure 3: Coverage–accuracy trade-off and cross-dataset replication. **(a)** CorrectModes@64 and **(b)** pass rates across λ on GSM8K. **(c)** CorrectModes@64 across datasets. Dotted lines show the base model, and the dashed line in (a) shows MaxRL. Error bars are ± 1 SEM.

5.4 Generalization across datasets

We next test whether the coverage gain extends beyond GSM8K. We train MILO separately on MultiArith (Roy & Roth, 2015), ASDiv (Miao et al., 2020), and SVAMP (Patel et al., 2021), using the same evaluation protocol. Relative to MaxRL, CorrectModes@64 increases by 1.8–3.5×: 10.5 → 37.0, 9.3 → 19.8, and 9.0 → 16.2, respectively. MILO also maintains pass@64 near the ceiling and lowers TopModeShare on all three datasets.

The gain also extends to additional settings. Appendix C.3 reports full baseline comparisons and an additional MATH result. With the larger Qwen3-8B-Base backbone, $\lambda = 0.5$ increases CorrectModes@64 from 6.3 for matched MaxRL to 30.3 while maintaining pass@64 $= 0.997$ (appendix C.3).

5.5 Robustness of mode coverage

Independent mode labelers. We first test whether the coverage gain depends on the particular mode labeler. We rescore cached responses from six seeds using the main Qwen2.5-1.5B evaluation labeler and two independently constructed labelers based on Gemma-2-9B and Claude Haiku 4.5. MILO has higher CorrectModes@64 than MaxRL under all three labelers (Fig. 4a), although the absolute counts differ because the labelers induce different mode partitions. The gain also persists in separate training runs that vary the strategy descriptor, embedding representation, and number of fixed clusters (appendix C.4). Across these variants, held-out CorrectModes@64 ranges from 21.8 to 26.4, with pass@64 between 0.997 and 0.999. Thus, the improvement is not specific to the labeling choices used in the main experiment.

Labeler-free diversity. We next test whether the result is visible without assigning responses to discrete modes. Among 16 correct responses per prompt, MILO achieves a Vendi score (Friedman & Dieng, 2023) of 3.6, compared with 1.8 for MaxRL and at most 2.2 for the other baselines (Fig. 4b). Vendi measures diversity from embedding similarities rather than mode labels. Lexical metrics give

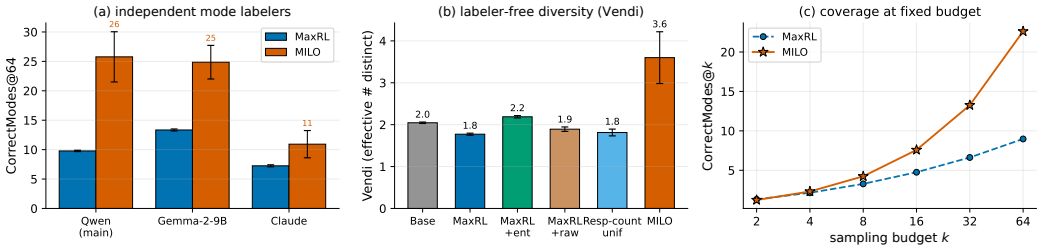


Figure 4: Robustness across evaluation methods and sampling budgets. **(a)** CorrectModes@64 under different mode labelers. **(b)** Vendi score using 16 correct responses per prompt. **(c)** CorrectModes@k as the sampling budget k increases. Error bars are ± 1 SEM.

the same picture: MILO has higher distinct-2 (the fraction of distinct word pairs; Li et al. 2016) and lower self-BLEU (overlap between responses; Zhu et al. 2018) than every baseline. We sample correct responses without replacement and omit prompts with fewer than 16 correct responses; this filter is applied separately to each method. These metrics measure diversity among correct responses, while pass@k separately measures correctness. Appendix C.4 gives the full results.

Sampling budget. Finally, MILO yields more correct modes than MaxRL at every tested budget $k \geq 4$, with a $2.5\times$ gap at $k = 64$ (Fig. 4c). The coverage advantage therefore persists well beyond the training rollout-group size $N = 8$.

5.6 Robustness to verifier blind spots

Finally, we ask whether broader mode coverage helps a candidate pool retain usable correct answers when a verifier has blind spots. For each prompt, we reject each policy's w most common correct modes among its 96 sampled responses, simulating a verifier that fails to recognize those strategies. We measure solve@k, the probability that k candidates contain at least one correct response from an accepted mode. At $w = 0$, solve@k equals pass@k.

With no rejected modes, solve@16 is similar for MILO and MaxRL (0.979 vs. 0.983). As w increases, MILO retains more usable candidates: at $w = 3$, solve@16 is 0.943 versus 0.827. The benefit also appears at other sampling budgets. At $k = 4$, solve@k is 0.761 versus 0.486; at $k = 64$, it is 0.977 versus 0.946. These results show how coverage reduces a pool's reliance on a few common modes.

To isolate the role of concentration, we also reject modes at random from the union of both policies' correct modes, independently of the policy. The methods then perform similarly, with differences within 0.03 across the tested settings. Broader coverage is most useful here when verifier errors concentrate on common modes. Appendix C.5 gives the estimator, full results, and random-rejection control.

6 Conclusion

MILO trains language models to generate a broader set of correct solution strategies by giving each correct mode its own log-likelihood term. For finite rollout groups, the expected mode-wise update is the gradient of a weighted sum of expected CorrectModes@k; the full update also includes a pooled-correctness term. At a local correctness optimum, a small MILO step with $\lambda > 0$ improves this weighted coverage objective to first order when its gradient is nonzero, while correctness changes only at second order. Empirically, λ controls a clear coverage–accuracy trade-off: small values can improve both pass@1 and mode coverage, while larger values produce broader candidate pools with high pass@k. These gains persist across datasets, sampling budgets, and mode labelers. A controlled verifier-blind-spot experiment further illustrates one way broader correct-mode coverage can be useful downstream. MILO currently requires a reliable correctness verifier and a fixed mode labeler; extending it to more open-ended tasks and end-to-end downstream settings is a natural direction for future work.

AI USE STATEMENT

We used generative AI tools as assistants in this work. They helped edit the manuscript for readability and consistency, including the notation and presentation of proofs; identify related literature and check reference metadata; write and debug code for baselines, evaluation, and figures; give feedback on experimental setups, such as hyperparameter ranges for baselines; and format the paper. The authors made all scientific decisions, reviewed and verified all AI-assisted text, code, and references, and take full responsibility for the content of the paper.

REPRODUCIBILITY STATEMENT

The MILO update is given in Eq. (3); in our experiments it replaces the advantage of an on-policy RLOO-style REINFORCE update (Section 5). The setting is defined in Section 3, the theoretical results are stated in Section 4, and complete proofs are given in Appendix A. Table 1 in Appendix B.1 lists all training and evaluation settings, including models, decoding, optimizer, training steps, coefficients, evaluation prompts, seed counts, and hardware; the same appendix defines every baseline, and Appendix C.2 describes the two additional ones. Appendix B.2 describes how the training and evaluation mode labelers are built and gives the estimators for all metrics and the confidence-interval procedure. All datasets and model checkpoints we use are publicly available; the only exception is one of the three independent evaluation labelers in Section 5.5, which uses Claude Haiku 4.5 through its commercial API. Code is not included with this submission; we will open-source the code for training, mode labeling, and evaluation.

REFERENCES

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

Michael Artin. *Algebra*. Pearson, 2nd edition, 2011.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024.

Zhiyu Cao, Kaixin Wu, Mingjie Zhong, Peifeng Li, Can Ye, and Qiaoming Zhu. Hint-guided diversified policy optimization for LLM reasoning. *arXiv preprint arXiv:2606.03021*, 2026.

Fan Chen, Audrey Huang, Noah Golowich, Sadhika Malladi, Adam Block, Jordan T. Ash, Akshay Krishnamurthy, and Dylan J. Foster. The coverage principle: How pre-training enables post-training. In *International Conference on Learning Representations*, 2026.

Zhipeng Chen, Xiaobo Qin, Youbin Wu, Yue Ling, Qinghao Ye, Wayne Xin Zhao, and Guang Shi. Pass@k training for adaptively balancing exploration and exploitation of large reasoning models. *arXiv preprint arXiv:2508.10751*, 2025.

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. *arXiv preprint arXiv:2110.14168*, 2021.

Jean-Antoine Désidéri. Multiple-gradient descent algorithm (MGDA) for multiobjective optimization. *Comptes Rendus Mathematique*, 350(5–6):313–318, 2012.

Paul Erdős. Some remarks on the theory of graphs. *Bulletin of the American Mathematical Society*, 53(4):292–294, 1947. doi: 10.1090/S0002-9904-1947-08785-1.

Shi Feng, Hanlin Zhang, Fan Nie, Sham Kakade, and Yiling Chen. Peer-predictive self-training for language model reasoning. *arXiv preprint arXiv:2604.13356*, 2026.

Dan Friedman and Adji Bousso Dieng. The Vendi score: A diversity evaluation metric for machine learning. *Transactions on Machine Learning Research*, 2023.

Noah Golowich, Fan Chen, Dhruv Rohatgi, Raghav Singhal, Carles Domingo-Enrich, Dylan J. Foster, and Akshay Krishnamurthy. Reject, resample, repeat: Understanding parallel reasoning in language model inference. *arXiv preprint arXiv:2603.07887*, 2026.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645:633–638, 2025. doi: 10.1038/s41586-025-09422-z.

Jubayer Ibn Hamid, Ifdita Hasan Orney, Ellen Xu, Chelsea Finn, and Dorsa Sadigh. Polychromic objectives for reinforcement learning. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=zzTQISAGUp.

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 *International Conference on Learning Representations (ICLR)*, 2026.

Audrey Huang, Adam Block, Dylan J. Foster, Dhruv Rohatgi, Cyril Zhang, Max Simchowitz, Jordan T. Ash, and Akshay Krishnamurthy. Self-improvement in language models: The sharpening mechanism. In *International Conference on Learning Representations*, 2025.

Samy Jelassi, Mujin Kwun, Rosie Zhao, Yuanzhi Li, Nicolo Fusi, Yilun Du, Sham M. Kakade, and Carles Domingo-Enrich. Matching features, not tokens: Energy-based fine-tuning of language models. *arXiv preprint arXiv:2603.12248*, 2026.

Liwei Jiang, Yuanjun Chai, Margaret Li, Mickel Liu, Raymond Fok, Nouha Dziri, Yulia Tsvetkov, Maarten Sap, and Yejin Choi. Artificial hivemind: The open-ended homogeneity of language models (and beyond). In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2025.

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 *International Conference on Learning Representations*, 2024. arXiv:2310.06452.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *International Conference on Learning Representations (ICLR)*, 2023.

Alex Kulesza and Ben Taskar. Determinantal point processes for machine learning. *Foundations and Trends in Machine Learning*, 5(2–3):123–286, 2012.

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

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.

Chenyi Li, Yuan Zhang, Bo Wang, Guoqing Ma, Wei Tang, Haoyang Huang, and Nan Duan. SetPO: Set-level policy optimization for diversity-preserving LLM reasoning. *arXiv preprint arXiv:2602.01062*, 2026.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 110–119, 2016.

Ziniu Li, Congliang Chen, Tian Xu, Zeyu Qin, Jiancong Xiao, Zhi-Quan Luo, and Ruoyu Sun. Preserving diversity in supervised fine-tuning of large language models. In *International Conference on Learning Representations*, 2025.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. Encouraging divergent thinking in large language models through multi-agent debate. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 17889–17904, Miami, Florida, USA, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.992. URL https://aclanthology.org/2024.emnlp-main.992/.

Anamika Lochab, Bolian Li, and Ruqi Zhang. Uniform-correct policy optimization: Breaking RLVR's indifference to diversity. *arXiv preprint arXiv:2605.00365*, 2026.

Shen-yun Miao, Chao-Chun Liang, and Keh-Yih Su. A diverse corpus for evaluating and developing English math word problem solvers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 975–984, 2020. doi: 10.18653/v1/2020.acl-main.92. URL https://aclanthology.org/2020.acl-main.92/.

Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning (ICML)*, pp. 1928–1937, 2016.

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

Jack Parker-Holder, Aldo Pacchiano, Krzysztof M. Choromanski, and Stephen J. Roberts. Effective diversity in population based reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 18050–18062, 2020. arXiv:2002.00632.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2080–2094, 2021. doi: 10.18653/v1/2021.naacl-main.168. URL https://aclanthology.org/2021.naacl-main.168/.

Ruotian Peng, Yi Ren, Zhouliang Yu, Weiyang Liu, and Yandong Wen. Beyond the sampled token: Preserving candidate support in RLVR. *arXiv preprint arXiv:2510.14807*, 2025.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, 2019. doi: 10.18653/v1/D19-1410. URL https://aclanthology.org/D19-1410/.

Subhro Roy and Dan Roth. Solving general arithmetic word problems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1743–1752, 2015. doi: 10.18653/v1/D15-1202. URL https://aclanthology.org/D15-1202/.

Rylan Schaeffer, Joshua Kazdan, John Hughes, Jordan Juravsky, Sara Price, Aengus Lynch, Erik Jones, Robert Kirk, Azalia Mirhoseini, and Sanmi Koyejo. How do large language monkeys get their power (laws)? In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 53132–53176. PMLR, 2025.

Devan Shah, Owen Yang, Daniel Yang, Chongyi Zheng, and Benjamin Eysenbach. Structured response diversity with mutual information. Workshop on Scaling Environments for Agents, NeurIPS 2025, 2025. URL https://neurips.cc/virtual/2025/124504.

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.

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *International Conference on Learning Representations*, 2025.

Yuda Song, Julia Kempe, and Rémi Munos. Outcome-based exploration for LLM reasoning. *arXiv preprint arXiv:2509.06941*, 2025.

Jacob Mitchell Springer, Madhu Advani, Lukas Aichberger, Arwen Bradley, Eran Malach, Omid Saremi, Sinead Williamson, Preetum Nakkiran, Etai Littwin, and Aditi Raghunathan. Annotations mitigate post-training mode collapse. In *Proceedings of the 43rd International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2026.

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. In *Proceedings of the 43rd International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2026.

Christian Walder and Deep Tejas Karkhanis. Pass@k policy optimization: Solving harder reinforcement learning problems. In *Advances in Neural Information Processing Systems*, 2025. URL https://neurips.cc/virtual/2025/poster/117623.

Chaoqi Wang, Yibo Jiang, Chenghao Yang, Han Liu, and Yuxin Chen. Beyond reverse KL: Generalizing direct preference optimization with diverse divergence constraints. In *International Conference on Learning Representations*, 2024.

Siwei Wang, Yifei Shen, Haoran Sun, Shi Feng, Shang-Hua Teng, Li Dong, Yaru Hao, and Wei Chen. Benefits and pitfalls of reinforcement learning for language model planning: A theoretical perspective. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=34a6DfHOUF.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023. arXiv:2203.11171.

Lilian Weng. Exploration strategies in deep reinforcement learning. https://lilianweng.github.io/posts/2020-06-07-exploration-drl/, 2020.

Ronald J. Williams and Jing Peng. Function optimization using connectionist reinforcement learning algorithms. *Connection Science*, 3(3):241–268, 1991.

Tengyang Xie, Dylan J. Foster, Yu Bai, Nan Jiang, and Sham M. Kakade. The role of coverage in online reinforcement learning. *arXiv preprint arXiv:2210.04157*, 2022. URL https://arxiv.org/abs/2210.04157.

Tengyang Xie, Dylan J. Foster, Akshay Krishnamurthy, Corby Rosset, Ahmed Awadallah, and Alexander Rakhlin. Exploratory preference optimization: Harnessing implicit Q*-approximation for sample-efficient RLHF. *arXiv preprint arXiv:2405.21046*, 2024.

Wei Xiong, Hanze Dong, Chenlu Ye, Ziqi Wang, Han Zhong, Heng Ji, Nan Jiang, and Tong Zhang. Iterative preference learning from human feedback: Bridging theory and practice for RLHF under KL-constraint. *arXiv preprint arXiv:2312.11456*, 2023.

An Yang, Anfeng Li, Baosong Yang, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Yingxuan Yang, Chengrui Qu, Muning Wen, Laixi Shi, Ying Wen, Weinan Zhang, Adam Wierman, and Shangding Gu. Understanding agent scaling in LLM-based multi-agent systems via diversity. *arXiv preprint arXiv:2602.03794*, 2026. URL https://arxiv.org/abs/2602.03794.

Yang Yue, 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? *arXiv preprint arXiv:2504.13837*, 2025.

Xiaochen Zhu, Caiqi Zhang, Yizhou Chi, Tom Stafford, Nigel Collier, and Andreas Vlachos. Demystifying multi-agent debate: The role of confidence and diversity. *arXiv preprint arXiv:2601.19921*, 2026. URL https://arxiv.org/abs/2601.19921.

Xuekai Zhu, Daixuan Cheng, Dinghuai Zhang, Hengli Li, Kaiyan Zhang, et al. FlowRL: Matching reward distributions for LLM reasoning. *arXiv preprint arXiv:2509.15207*, 2025. URL https://arxiv.org/abs/2509.15207.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 1097–1100, 2018.

A PROOFS AND ADDITIONAL THEORY

We prove Theorem 1 in appendix A.1 and Theorem 2 in appendix A.3. In between, appendix A.2 shows that when the total correct probability is fixed, J_{Mode} favors a balanced allocation across modes, and that a trade-off with correctness arises only when the parameterization links the mode probabilities. Appendix A.4 gives the bias and variance of the finite-sample update, and appendix C.1 checks both theorems numerically in a setting where every quantity has a closed form.

Throughout this section we fix a prompt x and its mode labeler and use the notation of Sections 3 and 4: a rollout $z \sim \pi_\theta(\cdot \mid x)$ decodes to a response $y = f(z)$; $r(x, y) \in \{0, 1\}$ is the verifier; $\phi(x, y) \in \mathcal{M}_x$ is the mode label of a correct response, with $M = |\mathcal{M}_x|$; p_m is the probability of a correct response in mode m and $p = \sum_m p_m$ the total correct probability; and $J_{\text{Pool}} = \log p$, $J_{\text{Mode}} = \sum_m \log p_m$, $J_\lambda = (1-\lambda) J_{\text{Pool}} + \lambda J_{\text{Mode}}$. Sums over m run over $\mathcal{M}_x$. Let $S(z) = \nabla_\theta \log \pi_\theta(z \mid x)$ be a rollout's score vector, and write $S_i = S(z_i)$. For $q \in [0, 1]$ and an integer $k \geq 1$ we write $h_k(q) = 1 - (1-q)^k$, so that $\text{pass@}k = \mathbb{E}[\text{Pass@}\,k] = h_k(p)$ and $\mathbb{E}[\text{CorrectModes@}\,k] = \sum_m h_k(p_m)$, because mode m appears among k independent samples with probability $h_k(p_m)$.

We assume parameter-independent support and valid differentiation under the expectation, so that for every integrable θ-independent function g,

$$\nabla_\theta \, \mathbb{E}_z[g(z)] = \mathbb{E}_z\big[g(z) \, \nabla_\theta \log \pi_\theta(z \mid x)\big]. \tag{5}$$

Let $r^{(m)}(z) = \mathbf{1}\{r(x, f(z)) = 1,\ \phi(x, f(z)) = m\}$ be the indicator that rollout z is correct and in mode m. It is a deterministic function of z alone, because the labeler is fixed and looks only at (x, y); it satisfies $\sum_m r^{(m)}(z) = r(x, f(z))$, $p_m = \mathbb{E}[r^{(m)}(z)]$, and, by Eq. (5), $\nabla p_m = \mathbb{E}[r^{(m)}(z)\, S(z)]$.

A.1 FINITE-SAMPLE OBJECTIVE

Lemma 1 (Truncated logarithm). *For $q \in (0, 1]$ and $N \geq 1$, let $J^{(N)}(q) = -\sum_{k=1}^{N} (1-q)^k / k$. Then*

$$\frac{d}{dq} J^{(N)}(q) = \sum_{k=1}^{N} (1-q)^{k-1} = \frac{1 - (1-q)^N}{q} = \sum_{k=1}^{N} \frac{1}{k} \, h_k'(q), \qquad J^{(N)}(q) \xrightarrow[N \to \infty]{} \log q.$$

Proof. Differentiating each term gives $\frac{d}{dq}\big[-(1-q)^k/k\big] = (1-q)^{k-1}$, and the finite geometric sum equals $(1-(1-q)^N)/q$. Since $h_k'(q) = k(1-q)^{k-1}$, the same sum is $\sum_k h_k'(q)/k$. The limit is the Mercator series $\log q = -\sum_{k \geq 1} (1-q)^k / k$, which converges for $q \in (0, 1]$. □

Lemma 2 (Counting identity). *For every integer $K \geq 1$, $1/K = \int_0^1 t^{K-1} \, dt$.*

Proof of Theorem 1. Write the MILO update in Eq. (3) as

$$\hat{g}_\lambda = (1-\lambda)\, \hat{g}_{\text{Pool}} + \lambda \, \hat{g}_{\text{Mode}} - \frac{1}{N} \sum_{i=1}^{N} S_i,$$

$$\hat{g}_{\text{Pool}} = \sum_{i:\, r_i = 1} \frac{S_i}{K}, \qquad \hat{g}_{\text{Mode}} = \sum_{i:\, r_i = 1} \frac{S_i}{K_{m(i)}} = \sum_m \hat{g}_m,$$

where $\hat{g}_m = K_m^{-1} \sum_i r_i^{(m)} S_i$ if $K_m \geq 1$ and $\hat{g}_m = 0$ if $K_m = 0$, with $r_i^{(m)} = r^{(m)}(z_i)$ and $K_m = \sum_j r_j^{(m)}$. Empty sums are zero, so $\hat{g}_\lambda$ is well defined when $K = 0$.

Mode-wise part. Fix a mode m and a rollout i with $r_i^{(m)} = 1$. Then $K_m \geq 1$ and $K_m - 1 = \sum_{j \neq i} r_j^{(m)}$, so by Lemma 2,

$$\hat{g}_m = \sum_{i=1}^{N} r_i^{(m)} \, S_i \int_0^1 t^{\,K_m - 1} \, dt = \sum_{i=1}^{N} r_i^{(m)} \, S_i \int_0^1 t^{\sum_{j \neq i} r_j^{(m)}} \, dt.$$

The rollouts are independent and $r_i^{(m)}, S_i$ depend only on z_i, so by Fubini

$$\mathbb{E}[\hat{g}_m] = \sum_{i=1}^{N} \int_0^1 \mathbb{E}\big[r_i^{(m)} S_i\big] \prod_{j \neq i} \mathbb{E}\big[t^{\,r_j^{(m)}}\big]\, dt$$
$$= N\, \nabla p_m \int_0^1 \big(1 - p_m + p_m t\big)^{N-1} dt = \frac{1 - (1-p_m)^N}{p_m}\, \nabla p_m,$$

using $\mathbb{E}[r_i^{(m)} S_i] = \nabla p_m$, $\mathbb{E}[t^{\,r_j^{(m)}}] = 1 - p_m + p_m t$, and the substitution $v = 1 - p_m + p_m t$. By Lemma 1 the prefactor is $\frac{d}{dq} J^{(N)}(q)$ at $q = p_m$, so $\mathbb{E}[\hat{g}_m] = \nabla J^{(N)}(p_m)$ whenever $p_m > 0$. If $p_m = 0$ at an interior parameter point, then $\nabla p_m = 0$ because $p_m \geq 0$ is differentiable and attains its minimum; the mode is never sampled, so $\hat{g}_m = 0$ almost surely and both sides vanish. Summing over m gives $\mathbb{E}[\hat{g}_{\mathrm{Mode}}] = \sum_m \nabla J^{(N)}(p_m) = \nabla J^{(N)}_{\mathrm{Mode}}$.

Pooled part and baseline. Replacing $r_i^{(m)}$ by r_i and K_m by K in the same computation gives $\mathbb{E}[\hat{g}_{\mathrm{Pool}}] = \nabla J^{(N)}(p)$. By Eq. (5) with $g \equiv 1$, $\mathbb{E}[S_i] = 0$, so the baseline term has zero expectation. Hence $\mathbb{E}[\hat{g}_\lambda] = (1-\lambda)\nabla J^{(N)}(p) + \lambda \nabla J^{(N)}_{\mathrm{Mode}} = \nabla J^{(N)}_\lambda$. The limit statement is the last part of Lemma 1. □

Remarks. (i) *Coverage identity.* By Lemma 1, $\nabla J^{(N)}_{\mathrm{Mode}} = \sum_m \sum_{k=1}^{N} \frac{1}{k} \nabla h_k(p_m) = \sum_{k=1}^{N} \frac{1}{k} \nabla\, \mathbb{E}[\mathrm{CorrectModes@}\, k]$, which is Eq. (4); letting $N \to \infty$ gives the same identity for J_{Mode} with an infinite sum when every $p_m > 0$. (ii) *Reductions.* At $\lambda = 0$ the update is the MaxRL update $\sum_{i:r_i=1} S_i/K - \frac{1}{N}\sum_i S_i$. If $M = 1$, then $K_{m(i)} = K$ for every correct rollout, so every λ gives this same update. (iii) *Why the labeler must be fixed.* The factorization above uses that $r_i^{(m)}$ is a function of z_i alone. Our labeler assigns each response to the nearest prototype in a cluster bank built before training (appendix B.2), so the label of a response does not depend on the other responses in its group. Clustering the current rollout group instead would couple the labels and the argument would not apply. (iv) *Total weight per mode.* In $\hat{g}_{\mathrm{Mode}}$ the coefficients of an observed mode sum to $K_m \cdot (1/K_m) = 1$, whereas in $\hat{g}_{\mathrm{Pool}}$ they sum to K_m/K. Repeated samples from a mode refine its average score vector without changing its total weight.

Interpreting the weight λ. The mode-wise term is a sum over modes, not an average, so the effective weight of λ grows with the number of modes. Two facts make this precise. First, when all $p_m > 0$, J_λ has the same gradient direction as the averaged objective $\bar{J}_{\tilde{\lambda}} = (1-\tilde{\lambda}) \log p + \frac{\tilde{\lambda}}{M} \sum_m \log p_m$ with $\tilde{\lambda} = M\lambda/(1 - \lambda + M\lambda)$; both reduce to J_{Pool} at $\lambda = 0$ and when $M = 1$. Second, in a single update with $K > 0$, the mode-wise coefficients sum to λM_{obs} and the pooled coefficients to $1 - \lambda$, where $M_{\mathrm{obs}} \leq \min(N, K)$ is the number of distinct modes observed in the group. For $\lambda < 1$, the ratio $\lambda M_{\mathrm{obs}}/(1-\lambda)$ helps explain why small values of λ (0.01–0.1) already change coverage substantially, and it means that λ is not directly comparable across labelers that produce different numbers of observed modes.

A.2 Mode allocation and the correctness trade-off

Proposition 1 (Allocation of correct probability)**.** *J_{Pool} depends on $(p_m)_{m \in \mathcal{M}_x}$ only through the total p. On the set $\{p_m > 0,\ \sum_m p_m = p\}$ with p fixed, J_{Mode} is strictly Schur-concave (it increases as the allocation becomes more balanced) and is uniquely maximized at $p_m = p/M$. For every $k > 1$, the same allocation uniquely maximizes $\mathbb{E}[\mathrm{CorrectModes@}\, k]$.*

Proof. $J_{\mathrm{Pool}} = \log \sum_m p_m$ is a function of the sum alone. $J_{\mathrm{Mode}} = \sum_m \log p_m$ is a sum of strictly concave functions, hence strictly concave and symmetric, hence strictly Schur-concave. A stationary point of the Lagrangian $\sum_m \log p_m - \mu(\sum_m p_m - p)$ has $1/p_m = \mu$ for all m, so $p_m = p/M$, and strict concavity makes it the unique maximizer. For $k > 1$, h_k is strictly concave on $[0,1]$, so $\sum_m h_k(p_m)$ is strictly Schur-concave as well and Jensen's inequality gives the same maximizer (Boyd & Vandenberghe, 2004). □

When the trade-off appears. If the mode probabilities could be chosen freely, MILO would not lower correctness. Let $F = \{\mathbf{p} \in \mathbb{R}^M_{>0} : \sum_m p_m \le 1\}$, where $1 - \sum_m p_m$ is incorrect mass. For $\mathbf{p} \in F$ with $p = \sum_m p_m < 1$, rescaling to $\mathbf{p}' = \mathbf{p}/p$ stays in F and raises J_λ by $[(1-\lambda) + \lambda M](-\log p) > 0$, so every maximizer of J_λ has $p = 1$. On that face $J_\lambda = \lambda J_{\text{Mode}}$, which for $\lambda > 0$ is uniquely maximized at $p_m = 1/M$ by Proposition 1, while J_0 is constant there. Thus, for $\lambda \in (0, 1]$, the maximizer of J_λ is the balanced allocation with $p = 1$: it lies in the correctness-optimal set of J_0 and, among those points, maximizes expected coverage.

A trade-off arises only when the parameterization links the mode probabilities. In the smallest example, $M = 2$ with a scalar $\alpha \in [0, 1]$, $p_1(\alpha) = a(1-\alpha)$ and $p_2(\alpha) = b\alpha$ with $0 < b < a \le 1$: the total $p(\alpha) = a - (a-b)\alpha$ is maximized at $\alpha = 0$ (all mass on the easier mode), whereas $J_{\text{Mode}} = \log(ab) + \log(1-\alpha) + \log\alpha$ is maximized at $\alpha = \frac{1}{2}$, where $p = \frac{a+b}{2} < a$. The pure mode-wise objective ($\lambda = 1$) therefore lowers p here, and λ tunes how far along this trade-off training moves.

A.3 Proof of Theorem 2 and local refinements

Let $p_m(\theta) > 0$ be twice continuously differentiable near an interior local maximizer $\theta^\star$ of p, so $\nabla p(\theta^\star) = 0$ and $H := \nabla^2 p(\theta^\star) \preceq 0$, and set $u = \nabla J_{\text{Mode}}(\theta^\star)$. Since $\nabla J_{\text{Pool}} = \nabla p / p = 0$ at $\theta^\star$, the MILO direction is $\nabla J_\lambda(\theta^\star) = \lambda u$. Along $\theta(\eta) = \theta^\star + \eta\lambda u$, a first-order expansion gives $J_{\text{Mode}}(\theta(\eta)) = J_{\text{Mode}}(\theta^\star) + \eta\lambda\|u\|^2 + O(\eta^2)$, so $\frac{d}{d\eta} J_{\text{Mode}}\big|_{\eta=0} = \lambda\|u\|^2 > 0$ when $\lambda > 0$ and $u \neq 0$.

For correctness, assume in addition that $\nabla^2 p$ is L-Lipschitz near $\theta^\star$ and let $v = \eta\lambda u$. Taylor's theorem with Lagrange remainder gives $p(\theta^\star + v) = p(\theta^\star) + \frac{1}{2} v^\top H v + R$ with $|R| \le \frac{L}{6}\|v\|^3$, because the linear term vanishes. Since $\theta^\star$ is a local maximizer,

$$0 \le p(\theta^\star) - p(\theta(\eta)) \le \tfrac{1}{2}\eta^2\lambda^2\big(-u^\top H u\big) + \tfrac{L}{6}\eta^3\lambda^3\|u\|^3, \qquad -u^\top H u \ge 0. \tag{6}$$

In particular $\frac{d}{d\eta} p\big|_{\eta=0} = 0$: correctness changes only at second order. Decompose $u = u_0 + u_\perp$ with $u_0 \in \ker H$ and $u_\perp \perp \ker H$. Because H is symmetric, $u^\top H u = u_\perp^\top H u_\perp$, so the second-order correctness cost in Eq. (6) depends only on $u_\perp$, while the first-order gain in J_{Mode}, proportional to $\|u\|^2 = \|u_0\|^2 + \|u_\perp\|^2$, also counts the component u_0 along the *correctness-flat* directions $\ker H$.

For any fixed k, h_k is increasing with $h'_k \le k$, so the mean value theorem and Eq. (6) give $0 \le \text{pass@}k(\theta^\star) - \text{pass@}k(\theta(\eta)) \le k\big[\frac{1}{2}\eta^2\lambda^2(-u^\top H u) + \frac{L}{6}\eta^3\lambda^3\|u\|^3\big]$; expected Pass@$k$ is preserved to first order as well. Finally, for the finite-sample objective, $\nabla J^{(N)}(p) = \frac{1-(1-p)^N}{p}\nabla p = 0$ at $\theta^\star$, so the MILO direction is λu_N with $u_N = \nabla J^{(N)}_{\text{Mode}}(\theta^\star)$, and the same two expansions hold with J_{Mode}, u replaced by $J^{(N)}_{\text{Mode}}, u_N$. By Eq. (4), the first-order gain is then a gain in $\sum_{k=1}^N \frac{1}{k}\mathbb{E}[\text{CorrectModes@}\,k]$. □

Nearby local maxima of J_λ. When $\theta^\star$ is a nondegenerate maximizer of p ($H \prec 0$), the local maximizer of J_λ moves smoothly with λ and the second-order correctness cost has an explicit constant.

Proposition 2. *Under the assumptions of Theorem 2, if $H \prec 0$ and $u \neq 0$, then for λ near 0 there is a C^1 path of local maximizers θ_λ of J_λ with $\theta_0 = \theta^\star$,*

$$\theta_\lambda - \theta^\star = -\lambda\, p\, H^{-1}u + o(\lambda), \qquad p(\theta_\lambda) = p(\theta^\star) - C\lambda^2 + o(\lambda^2), \quad C = -\tfrac{1}{2}p^2\, u^\top H^{-1} u > 0,$$

and $J_{\text{Mode}}(\theta_\lambda) = J_{\text{Mode}}(\theta^\star) - \lambda\, p\, u^\top H^{-1}u + o(\lambda)$, where $-p\, u^\top H^{-1}u > 0$.

Proof. Let $F(\theta, \lambda) = \nabla J_\lambda(\theta)$, so $F(\theta^\star, 0) = 0$. At $(\theta^\star, 0)$, $\partial_\theta F = \nabla^2 \log p = \frac{1}{p}H - \frac{1}{p^2}\nabla p \nabla p^\top = \frac{1}{p}H$, which is invertible, and $\partial_\lambda F = -\nabla \log p + \nabla J_{\text{Mode}} = u$. The implicit function theorem gives a C^1 path with $\frac{d\theta_\lambda}{d\lambda}\big|_0 = -(\partial_\theta F)^{-1}\partial_\lambda F = -pH^{-1}u$. Since $\nabla p(\theta^\star) = 0$, expanding p to second order along the path gives $p(\theta_\lambda) = p + \frac{\lambda^2}{2}p^2(H^{-1}u)^\top H(H^{-1}u) + o(\lambda^2) = p + \frac{\lambda^2}{2}p^2 u^\top H^{-1}u + o(\lambda^2)$, and $H^{-1} \prec 0$ gives $C > 0$. The first-order change of J_{Mode} is $u^\top(-pH^{-1}u)\lambda > 0$. □

When is $u \neq 0$, and how much of it is correctness-flat? Theorem 2 needs $u \neq 0$. At a balanced allocation ($p_m = p/M$ for all m), $u = \sum_m \nabla p_m / p_m = (M/p)\nabla p = 0$ at $\theta^\star$, so an unbalanced allocation is necessary; the next proposition shows it is also sufficient under a rank condition, with a quantitative lower bound.

Proposition 3. *Under the assumptions of Theorem 2, $\sum_m \nabla p_m = \nabla p(\theta^\star) = 0$. Assume $M \geq 2$ and that $\{\nabla p_m\}_m$ spans an $(M-1)$-dimensional space, and let $\sigma^2 > 0$ be the smallest nonzero eigenvalue of the Gram matrix $G_{mn} = \langle \nabla p_m, \nabla p_n \rangle$. Then*

$$\|u\|^2 \;\geq\; \sigma^2 \, M \, \mathrm{Var}_m(1/p_m), \qquad \mathrm{Var}_m(1/p_m) = \tfrac{1}{M} \sum_m \left(\tfrac{1}{p_m} - \bar{w} \right)^2, \quad \bar{w} = \tfrac{1}{M} \sum_m \tfrac{1}{p_m},$$

and the right-hand side is zero if and only if all p_m are equal.

Proof. Let $w = (1/p_1, \ldots, 1/p_M)$, so $u = \sum_m w_m \nabla p_m$ and $\|u\|^2 = w^\top G w$. Since $\sum_m \nabla p_m = 0$, $(G\mathbf{1})_m = \langle \nabla p_m, \sum_n \nabla p_n \rangle = 0$, so $\mathbf{1} \in \ker G$, and the rank assumption makes $\ker G = \mathrm{span}(\mathbf{1})$. Write $w = \bar{w}\mathbf{1} + w'$ with $w' \perp \mathbf{1}$. The component $\bar{w}\mathbf{1}$ contributes $\bar{w} \sum_m \nabla p_m = 0$ to u, hence $\|u\|^2 = w'^\top G w' \geq \sigma^2 \|w'\|^2 = \sigma^2 M \, \mathrm{Var}_m(1/p_m)$. □

Proposition 4. *Under the assumptions of Proposition 3, let P_0 be the orthogonal projection onto $\ker H$, $u_0 = P_0 u$, and $\cos\vartheta = \|u_0\|/\|u\|$ for $u \neq 0$. Moving along u_0 has no second-order correctness cost ($u_0^\top H u_0 = 0$) and raises J_{Mode} at rate*

$$\tfrac{d}{d\eta} J_{\mathrm{Mode}}(\theta^\star + \eta u_0)\big|_{\eta=0} = \|u_0\|^2 = \cos^2\vartheta \, \|u\|^2 \;\geq\; \cos^2\vartheta \, \sigma^2 M \, \mathrm{Var}_m(1/p_m).$$

Proof. $H u_0 = 0$ gives $u_0^\top H u_0 = 0$, so $p(\theta^\star + \eta u_0) = p(\theta^\star) + o(\eta^2)$. Since $\nabla J_{\mathrm{Mode}}(\theta^\star) = u$ and P_0 is an orthogonal projection, $\tfrac{d}{d\eta} J_{\mathrm{Mode}}(\theta^\star + \eta u_0)\big|_0 = \langle u, P_0 u \rangle = \|P_0 u\|^2$. Apply Proposition 3. □

The bound separates two factors: the imbalance of the allocation, through $\mathrm{Var}_m(1/p_m)$, and the fraction $\cos^2\vartheta$ of the squared gradient norm that lies in the correctness-flat subspace. A full MILO step along λu also moves along $u_\perp$, whose second-order cost is the one bounded in Eq. (6). These statements are an instance of improving a secondary objective at a stationary point of a primary one (Désidéri, 2012; Boyd & Vandenberghe, 2004).

A.4 Bias and variance of the finite-sample update

Proposition 5 (Truncation bias)**.** *For each mode with $p_m > 0$, $\nabla \log p_m - \nabla J^{(N)}(p_m) = \frac{(1-p_m)^N}{p_m} \nabla p_m$, so the relative bias of the per-mode gradient is exactly $(1-p_m)^N$, and $0 \leq J^{(N)}(p_m) - \log p_m \leq \frac{(1-p_m)^{N+1}}{(N+1)p_m}$. The relative bias of every mode is at most ϵ once $N \geq \log(1/\epsilon)/p_{\min}$, where $p_{\min} = \min_m p_m$.*

Proof. By Lemma 1, $\nabla J^{(N)}(p_m) = \frac{1-(1-p_m)^N}{p_m} \nabla p_m$ and $\nabla \log p_m = \frac{1}{p_m} \nabla p_m$; subtracting gives the identity. For the objective gap, each tail term satisfies $(1-p_m)^k/k \leq (1-p_m)^k/(N+1)$ for $k > N$, and $\sum_{k>N} (1-p_m)^k = (1-p_m)^{N+1}/p_m$. Finally, $(1-p_m)^N \leq e^{-N p_m} \leq \epsilon$ when $N \geq \log(1/\epsilon)/p_m$. □

Proposition 6 (Conditional variance)**.** *Assume $\|S(z)\| \leq B$. Condition on the realized pattern ω, consisting of the correctness indicators $(r_i)_{i \leq N}$ and the mode labels $\phi(x, y_i)$ of the correct rollouts, that is, which rollouts are correct and which mode each correct rollout uses. Then the per-mode terms $\hat{g}_m$ are conditionally independent and*

$$\mathrm{tr}\,\mathrm{Cov}(\hat{g}_{\mathrm{Mode}} \mid \omega) = \sum_{m : K_m \geq 1} \frac{1}{K_m} \mathrm{tr}\,\mathrm{Cov}\left(S \mid \textit{correct, mode } m\right) \leq \sum_{m : K_m \geq 1} \frac{B^2}{K_m}.$$

If every mode has the same conditional score covariance with positive trace and $K > 0$, then

$$\frac{\mathrm{tr}\,\mathrm{Cov}(\hat{g}_{\mathrm{Mode}} \mid \omega)}{\mathrm{tr}\,\mathrm{Cov}(\hat{g}_{\mathrm{Pool}} \mid \omega)} = K \sum_{m : K_m \geq 1} \frac{1}{K_m} \;\geq\; M_{\mathrm{obs}}^2,$$

with equality if and only if all nonzero K_m are equal (Cauchy–Schwarz).

Proof. Because the labeler is fixed, ω is a coordinate-wise function of $(z_1, \dots, z_N)$, so conditionally on ω the rollouts remain independent, each distributed as π_θ restricted to its label event. Hence $\hat{g}_m = K_m^{-1} \sum_{i \in m} S_i$ is an average of K_m conditionally i.i.d. vectors with covariance $\frac{1}{K_m} \operatorname{Cov}(S \mid \text{correct, mode } m)$; different modes use disjoint rollouts, so their covariances add. The bound uses $\operatorname{tr} \operatorname{Cov}(S \mid \cdot) \le \mathbb{E}[\|S\|^2 \mid \cdot] \le B^2$. Under a common covariance Σ, $\operatorname{tr} \operatorname{Cov}(\hat{g}_{\text{Pool}} \mid \omega) = \operatorname{tr} \Sigma / K$ and $\operatorname{tr} \operatorname{Cov}(\hat{g}_{\text{Mode}} \mid \omega) = \operatorname{tr} \Sigma \sum_m 1/K_m$; the inequality is $(\sum_m 1/K_m)(\sum_m K_m) \ge M_{\text{obs}}^2$. □

A mode observed once contributes up to B^2 to the bound and is not averaged. Rare modes therefore dominate both the truncation bias, through $p_{\min}$ in Proposition 5, and the variance of the mode-wise update; this is the price of giving every observed mode total weight one.

B Experimental Details

B.1 Training Recipe and Baselines

Table 1 lists the shared settings. All GSM8K methods use the same policy, training prompts, optimizer, and rollout budget. MaxRL and the methods built on it, including MILO, use the REINFORCE update with the per-group baseline $-1/N$ (Eq. (3)) and differ in how rollouts are weighted and in any added regularizer or bonus. GRPO instead centers each group's rewards by their mean and divides by their standard deviation, and EBFT replaces the verifier reward (appendix C.2).

Setting	Value
Policy model	Qwen3-4B-Base (main); Qwen3-8B-Base (appendix C.3)
Rollout group size N	8
Training and evaluation decoding	temperature 1.0, top-p 1.0, at most 512 new tokens
Optimizer	AdamW, learning rate 10^{-6} (4B) / 3×10^{-6} (8B), one prompt per step
Training steps	500 (4B) / 1000 (8B)
Training data	GSM8K train split, on-policy; no KL penalty
Reward	ground-truth exact-match verifier $r(x, y)$
MILO weight λ	0.1 (4B default) / 0.5 (8B); sweep in Section 5.3
MaxRL+entropy / +raw-diversity weights	0.01 / 0.1
Cluster bank	256 prototypes per bank; k-means on ℓ_2-normalized all-MiniLM-L6-v2 embeddings (384-d), 4 initializations, seed 0
Strategy descriptor	Qwen2.5-0.5B-Instruct (training) / Qwen2.5-1.5B-Instruct (evaluation); greedy, at most 24 tokens
Evaluation samples per prompt	$n = 96$
Evaluation prompts	the first 128 prompts of the test split for all 4B GSM8K runs, the cross-dataset runs, and MATH; UCPO, EBFT, and outcome-based exploration were evaluated on the first 500 and are reported on the shared first 128; 8B: first 500, except one MILO seed on the first 200
Seeds (4B GSM8K)	MILO 16; MaxRL, UCPO, response-count uniform 6; GRPO, entropy, raw-diversity, EBFT, outcome-based exploration 3; λ sweep 6 per value
Seeds (other)	8B: MILO 8, MaxRL 3; cross-dataset 2–3; mode-partition ablation 2–6
Hardware	one 80GB GPU (A100, H100, or H200) per run

Table 1: Shared training and evaluation settings.

Baselines. MaxRL is the $\lambda = 0$ update. MaxRL+entropy adds the classic entropy regularizer (Williams & Peng, 1991; Mnih et al., 2016) with weight 0.01. MaxRL+raw-diversity adds to the advantage of each correct rollout a bonus of $0.1 \times (1 - \text{mean cosine similarity})$, where the similarity is between the MiniLM embedding of its response and those of the other correct responses in the group (applied when a group has at least two correct rollouts). This is a simple embedding-dispersion

bonus in the spirit of semantic-diversity rewards (Hu et al., 2026). UCPO (Lochab et al., 2026) weights each correct rollout by the inverse of its policy probability. Let q_i be the softmax, over the correct rollouts, of their length-normalized sequence log-probabilities, and let $w_i \propto 1/q_i$ sum to one. The UCPO advantage is then $(1-\tau)r_i/K + \tau r_i w_i - 1/N$. We swept $\tau \in \{0.1, 0.5, 1, 2\}$ on one seed; $\tau = 1$ gave the highest coverage and is used for the six-seed comparison. The response-count-uniform ablation replaces $K_{m(i)}$ in Eq. (3) by $K_{\mathrm{resp}(i)}$, the number of rollouts in the group whose response text is identical to response i. It tests whether spreading weight evenly over distinct responses is enough, or whether the even spread must be over strategies; it coincides with MILO when every mode contains a single distinct response. Outcome-based exploration and energy-based fine-tuning are described in appendix C.2. Only MILO uses the mode labels $\phi(x, y)$.

Uncertainty. Reported values average per-prompt metrics over evaluation prompts and then over training seeds. Confidence intervals for MILO−MaxRL differences use a hierarchical bootstrap that resamples training seeds within each method and resamples the shared evaluation prompts (95%, 4000 replicates). Elsewhere, ± denotes one standard error of the mean (SEM) over seeds.

B.2 Mode labeling and evaluation metrics

Mode labelers. Both the training and the evaluation labeler implement $\phi(x, y)$ in three fixed steps. First, a frozen strategy descriptor produces a short description of how a correct response solves the problem, for example substitution, elimination, or case analysis. Second, the Sentence Transformers checkpoint all-MiniLM-L6-v2[1] (Reimers & Gurevych, 2019; Wang et al., 2020) embeds the description. Third, the response receives the label of the nearest prototype in a cluster bank of $M_{\mathrm{bank}} = 256$ prototypes built before training. The descriptor decodes greedily, so the label is a deterministic function of (x, y), as Theorem 1 requires. These strategy labels group responses by how they solve the problem, which is finer than grouping by shared meaning (Kuhn et al., 2023). The descriptor cost is small: at most 24 tokens per correct rollout, against up to 8×512 tokens generated per rollout group.

Cluster banks. The two banks are built independently. The training bank clusters the descriptions that Qwen2.5-0.5B-Instruct gives for Qwen3-1.7B-Base (Yang et al., 2025) responses to training prompts. The evaluation bank clusters the descriptions that Qwen2.5-1.5B-Instruct gives for responses of the untrained Qwen3-4B-Base to test prompts. Thus the split, the reference policy, the descriptor, and the prototypes all differ between training and evaluation, and no evaluated policy contributes to either bank. The bank size M_{bank} counts prototypes shared across prompts, while the set $\mathcal{M}_x$ in Section 3.2 contains the labels a prompt's correct responses can receive, so $M = |\mathcal{M}_x| \le M_{\mathrm{bank}}$.

Metrics. For each evaluation prompt we draw $n = 96$ responses and compute all metrics from these samples, then average over prompts and seeds. $\mathrm{CorrectModes@}\,k = |\mathcal{C}_k(x; y_{1:k})|$ counts the distinct evaluation-labeler modes among the correct responses in the first k samples (Eq. (2)); our main budget is $k = 64$. If c of the n samples are correct, $\mathrm{pass@}k$ is estimated without bias by $1 - \binom{n-c}{k} / \binom{n}{k}$. The area under the pass@k curve (AUP) is the mean of these estimates over $k = 1, \ldots, n$, which equals the area under the curve on a linear k axis divided by n; it lies in $[0, 1]$. TopModeShare is the fraction of correct responses among the first 64 samples that belong to the most common correct mode, taken as 0 for a prompt with no correct response. MILO is non-inferior to MaxRL at margin ϵ on a repeated-sampling metric (pass@16, pass@64, or AUP) when the lower 95% confidence bound of the MILO−MaxRL difference exceeds $-\epsilon$. Table 2 reports $\epsilon \in \{0.02, 0.01, 0.005\}$, and the main text highlights 0.01. We evaluate on datasets with moderate pass rates because CorrectModes@k carries little information beyond pass@k when almost no samples are correct.

[1] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

C ADDITIONAL RESULTS

C.1 CONTROLLED CHECK OF THE THEORY

We check Theorems 1 and 2 on a softmax policy over eight outcomes, four of which are distinct correct modes ($M = 4$) and four incorrect (Fig. 5). The modes are defined by construction, and every p_m and gradient has a closed form. (a) By Theorem 1, the expected $1/K_m$ update equals $\nabla J^{(N)}_{\mathrm{Mode}}$, so by Proposition 5 its distance to the distributional mode-wise gradient ∇J_{Mode} shrinks at the rate $(1 - p_{\min})^N$. Measured with the mean of 4000 sampled updates at each N, this distance falls from 1.20 at $N = 4$ to 0.007 at $N = 64$, in line with the rate. The pooled $1/K$ update stays at distance about 1.4 for every N because it estimates a different gradient. (b–c) We then take a tightly converged maximizer $\theta^\star$ of p, where $\|\nabla p\| \approx 2 \times 10^{-6}$ and $\|u\|^2 = 12.00$. The MILO step $\theta^\star + \eta\lambda u$ with $\eta = 1$ changes J_{Mode} by $\eta\lambda\|u\|^2$ to within 0.001% for $\lambda \le 0.1$, while p changes by less than 2×10^{-6} in magnitude for $\lambda \le 0.2$. Finally, among policies with $p = 1$, the pooled objective is indifferent to the allocation (Proposition 1), so where gradient ascent on it ends depends on the initialization. Our initialization makes one correct mode the most likely, and gradient ascent on the pooled objective ends with almost all correct mass on that mode (TopModeShare 0.998). MILO ends at the balanced allocation (0.25 per mode), as Proposition 1 predicts.

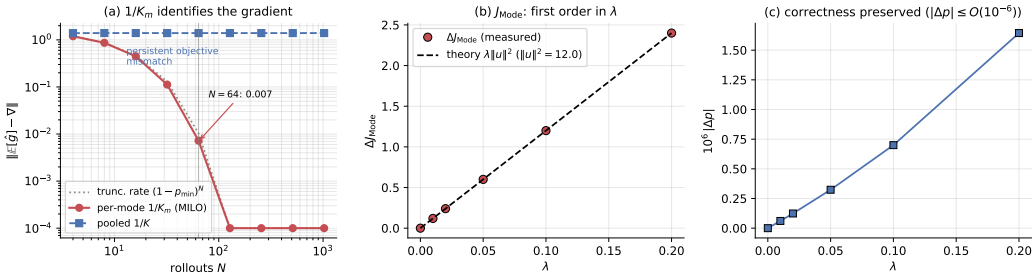


Figure 5: Controlled check with eight outcomes and $M = 4$ correct modes. **(a)** Distance between the expected finite-sample update and the distributional mode-wise gradient as N grows, for the $1/K_m$ (MILO) and $1/K$ (pooled) weightings. **(b)** Change in J_{Mode} after one MILO step from a correctness maximizer, against the predicted first-order term $\eta\lambda\|u\|^2$. **(c)** The corresponding change in the correctness probability p.

C.2 MAIN COMPARISON ON GSM8K

Table 2 gives the values behind Fig. 2. All rows use the same 128 evaluation prompts, so the methods are compared on identical questions. After averaging over seeds, MILO has more correct modes than MaxRL on 125 of the 128 prompts.

Additional baselines: exploration and feature matching. We ran two further published methods under the same recipe to test whether coverage can be raised without mode labels: outcome-based exploration rewards rarely sampled answers during training, and energy-based fine-tuning pulls samples toward a reference solution in feature space while pushing them apart from each other.

Outcome-based exploration (Song et al., 2025) adds an upper-confidence-bound bonus to the MaxRL advantage: $A_i + c\left(\min\{1, 1/\sqrt{\mathrm{count}(x, a_i)}\} - b_0\right)$, where $a_i = a(y_i)$ is the extracted answer and $\mathrm{count}(x, a)$ is the number of times answer a has been sampled for prompt x so far in training. We set $b_0 = 0.5$, give no bonus when the whole group is correct, following the original method, and save the counts with the training checkpoint. The bonus rewards rarely sampled answers and penalizes frequent ones, whether or not they are correct, so a larger c flattens the answer distribution. On one seed, $c \in \{0.1, 0.3, 1.0\}$ gives CorrectModes@64 of 15.3, 22.2, and 22.9 at pass@1 of 0.631, 0.519, and 0.486. Over three seeds at $c = 0.3$, coverage is 22.8 ± 0.3 at pass@1 0.534 ± 0.009 (Table 3). Exploration can therefore match MILO's coverage, but only by giving up about ten points of single-sample accuracy. At matched coverage, its pass@1 is 0.534 at 22.8 modes, against 0.639 for MILO at $\lambda = 0.1$ with 22.6 modes and 0.611 for MILO at $\lambda = 0.5$ with

Method	pass@1	pass@16	pass@64	AUP	CorrectModes @64	TopModeShare↓
Base (no RL)	0.647	0.985	1.000	0.987	8.4	0.516
GRPO	0.742	0.983	0.999	0.988	8.9	0.480
MaxRL	0.753	0.983	0.998	0.988	9.0	0.478
MaxRL+entropy	0.597	0.978	0.996	0.981	9.5	0.486
MaxRL+raw-diversity	0.741	0.982	0.996	0.986	10.3	0.450
UCPO	0.663	0.980	0.998	0.985	10.0	0.456
Response-count uniform	0.750	0.983	0.997	0.987	10.1	0.464
MILO ($\lambda = 0.1$)	0.639	0.979	0.997	0.983	**22.6**	**0.257**
MILO−MaxRL: Δ	–	-0.004	-0.001	-0.005	$+13.7$	-0.221
95% CI	–	$[-.0098, -.0001]$	$[-.005, +.004]$	$[-.009, -.001]$	$[10.7, 16.4]$	$[-.270, -.166]$
margin $\epsilon = 0.02$	–	✓	✓	✓	–	–
margin $\epsilon = 0.01$	–	✓	✓	✓	–	–
margin $\epsilon = 0.005$	–	×	×	×	–	–

Table 2: Main comparison on Qwen3-4B-Base GSM8K (128 evaluation prompts; seeds as in Table 1). The upper block reports means over prompts and seeds. The lower block reports MILO−MaxRL with 95% hierarchical bootstrap intervals over training seeds and shared prompts; non-inferiority holds when the lower bound exceeds $-\epsilon$. The repeated-sampling metrics pass@16, pass@64, and AUP are non-inferior at margins 0.02 and 0.01 but not at 0.005; pass@1 is reported as single-sample accuracy and not tested.

24.1 modes (Section 5.3). At matched pass@1 ≈ 0.63, exploration reaches 15.3 modes against MILO's 22.6.

Energy-based fine-tuning (EBFT) (Jelassi et al., 2026) replaces the verifier reward by feature matching. Let $\psi(y)$ concatenate the ℓ_2-normalized last-token hidden states at 25%, 50%, and 75% of the depth of a frozen copy of the initial policy. Each sampled response receives $R_i = 2\,\psi(y_i)^\top \psi(y^{\text{ref}}) - \frac{2}{N-1}\sum_{j\neq i}\psi(y_i)^\top\psi(y_j)$, where y^{ref} is the dataset's reference solution: the first term pulls samples toward the reference solution's features and the second pushes them apart. We use the RLOO baseline and sweep the optional cross-entropy weight $\gamma \in \{0, 0.03, 0.1\}$ on one seed. Coverage is flat across the sweep (7.7–8.2), so we report $\gamma = 0$ over three seeds. EBFT keeps accuracy without using a verifier (pass@1 0.637 ± 0.003, pass@64 0.997) but leaves coverage at the base level (8.2 ± 0.0, TopModeShare 0.506). Its repulsion term pushes samples apart in feature space, but this does not produce new correct strategies.

Method	pass@1	pass@64	CorrectModes@64	TopModeShare↓
MaxRL	0.753	0.998	9.0 ± 0.1	0.478
EBFT ($\gamma = 0$)	0.637	0.997	8.2 ± 0.0	0.506
Outcome-based exploration ($c = 0.3$)	0.534	0.997	22.8 ± 0.3	0.220
MILO ($\lambda = 0.1$)	0.639	0.997	22.6 ± 1.3	0.257
MILO ($\lambda = 0.5$)	0.611	0.998	24.1 ± 3.2	0.217

Table 3: Exploration and feature-matching baselines on Qwen3-4B-Base GSM8K, same recipe and the same 128 evaluation prompts as Table 2; three seeds for EBFT and outcome-based exploration, six for MaxRL and MILO ($\lambda = 0.5$), sixteen for MILO ($\lambda = 0.1$). Values are means, with $\pm$ SEM over seeds for CorrectModes@64.

C.3 Generalization across datasets and model sizes

Table 4 reports MILO and every baseline in Table 2 except UCPO on MultiArith, ASDiv, and SVAMP (Qwen3-4B-Base, same recipe, 2–3 seeds). Each dataset has its own evaluation labeler, built in the same way as for GSM8K. The trained baselines stay within 8.8–10.5 correct modes; on ASDiv and SVAMP they even cover slightly fewer modes than the untrained base model, consistent with RL reducing the variety of correct solutions (Yue et al., 2025). MILO reaches 37.0, 19.8, and 16.2 modes, with pass@64 within 0.005 of MaxRL on every dataset and TopModeShare reduced by 0.10–0.30.

On MATH (the first 128 MATH-500 test prompts, same recipe), far fewer samples are correct (pass@64 $\approx$ 0.87), and MaxRL and MILO each cover about five correct modes. MILO's gain is small and grows with λ: MaxRL reaches 4.99 ± 0.08 modes, MILO 5.08 ± 0.06 at $\lambda = 0.1$ and 5.7 ± 0.1 at $\lambda = 0.5$ (three seeds each; pass@64 0.870, 0.883, and 0.875).

Dataset	Method	pass@1	pass@64	AUP	CorrectModes@64	TopModeShare↓
MultiArith	Base (no RL)	0.894	1.000	0.999	10.2	0.420
	GRPO	0.945	1.000	0.999	10.1	0.415
	MaxRL	0.946	1.000	0.999	10.5	0.407
	MaxRL+entropy	0.952	1.000	0.999	10.4	0.426
	MaxRL+raw-diversity	0.951	1.000	0.999	10.4	0.421
	Response-count uniform	0.952	1.000	0.999	9.9	0.438
	MILO	0.884	1.000	0.999	**37.0**	**0.106**
ASDiv	Base (no RL)	0.762	0.992	0.987	10.5	0.413
	GRPO	0.811	0.990	0.985	9.1	0.450
	MaxRL	0.812	0.998	0.991	9.3	0.452
	MaxRL+entropy	0.816	0.998	0.992	8.9	0.463
	MaxRL+raw-diversity	0.816	0.996	0.990	8.8	0.452
	Response-count uniform	0.805	0.999	0.992	9.1	0.455
	MILO	0.742	0.993	0.986	**19.8**	**0.302**
SVAMP	Base (no RL)	0.776	0.990	0.984	10.0	0.454
	GRPO	0.843	0.991	0.985	8.9	0.515
	MaxRL	0.835	0.991	0.985	9.0	0.496
	MaxRL+entropy	0.852	0.992	0.986	8.8	0.511
	MaxRL+raw-diversity	0.845	0.996	0.989	8.9	0.508
	Response-count uniform	0.836	0.997	0.990	9.0	0.497
	MILO	0.775	0.995	0.987	**16.2**	**0.392**

Table 4: Cross-dataset replication with Qwen3-4B-Base (128 evaluation prompts per dataset; 3 seeds for MaxRL and MILO, 2–3 for the other trained methods, one evaluation of the base model). Figure 3 summarizes the CorrectModes@64 column.

Larger backbone. We also post-train Qwen3-8B-Base on GSM8K. With the 4B recipe (learning rate 10^{-6}, 500 steps), the 8B MaxRL policy is already concentrated and $\lambda = 0.1$ moves coverage little (7.3 vs. 6.9 modes for MaxRL, three seeds each). We therefore use a longer recipe for both methods, learning rate 3×10^{-6} and 1000 steps, and $\lambda = 0.5$ for MILO. MILO then reaches 30.3 ± 3.6 correct modes over 8 seeds, against 6.3 ± 0.1 for MaxRL over 3 seeds, with pass@64 0.997 vs. 0.996 and TopModeShare 0.185 vs. 0.607 (Table 5). One MILO seed was evaluated on the first 200 test prompts and the other runs on the first 500; on the first 200 prompts, which every run covers, the counts are 30.6 vs. 6.4.

Method	pass@1	pass@64	AUP	CorrectModes@64	TopModeShare↓
MaxRL	0.889	0.996	0.990	6.3 ± 0.1	0.607
MILO ($\lambda = 0.5$)	0.792	0.997	0.990	$\mathbf{30.3 \pm 3.6}$	**0.185**

Table 5: Qwen3-8B-Base on GSM8K with learning rate 3×10^{-6} and 1000 steps for both methods. MILO uses $\lambda = 0.5$ and 8 seeds (one evaluated on the first 200 test prompts, seven on the first 500); MaxRL uses 3 seeds (first 500 prompts). CorrectModes@64 is mean $\pm$ SEM over seeds.

C.4 Robustness of mode coverage

Training-time mode partitions. We change the labeler used *during training* while keeping the evaluation labeler fixed (Table 6): we replace the strategy description by the raw response embedding, change the bank size from $M_{\text{bank}} = 256$ to 64 or 512, and replace the descriptor that builds the training bank, Qwen2.5-0.5B, by Gemma-2-9B or Llama-3.2-3B; the labels during training still come from Qwen2.5-0.5B descriptions. Every variant gives a large gain over MaxRL (21.8–26.4 vs. 9.0 correct modes) at pass@64 between 0.997 and 0.999. Because the mode-wise term is a sum over modes, changing M_{bank} also changes how many modes a rollout group tends to contain, and

hence the effective weight of the mode-wise term (appendix A.1). We use strategy descriptions in the main experiments because the resulting modes are easy to read.

Training-time labeler	seeds	pass@64	CorrectModes@64
MaxRL (no mode labels)	6	0.998	9.0
MILO, strategy descriptor (main)	16	0.997	22.6
MILO, raw response embedding	3	0.999	26.4
MILO, coarser bank ($M_{\text{bank}} = 64$)	2	0.999	26.0
MILO, finer bank ($M_{\text{bank}} = 512$)	2	0.997	25.7
MILO, Gemma-2-9B bank descriptor	6	0.999	21.8
MILO, Llama-3.2-3B bank descriptor	6	0.999	23.3

Table 6: Training-time labeler ablation on Qwen3-4B-Base GSM8K. Coverage is always scored by the fixed evaluation labeler on the same 128 prompts.

Independent evaluation labelers. The main text rescores cached responses from six seeds of MILO and MaxRL with three evaluation labelers (Fig. 4a). CorrectModes@64 is 25.8 vs. 9.8 with the main Qwen2.5-1.5B labeler, 24.9 vs. 13.3 with a Gemma-2-9B labeler, and 10.9 vs. 7.2 with a Claude Haiku 4.5 labeler. The absolute counts depend on how finely each labeler partitions strategies, but MILO is ahead under all three. As an example of what the labeler separates, consider the prompt *"Morisette has 5 apples and 8 oranges; Kael brought twice as many apples and half as many oranges; how many fruits together?"* (answer 27). MILO's correct responses include a direct arithmetic solution ($5+8=13$; $10+4=14$; 27) and an algebraic one that first introduces variables for the two counts. The labeler places these in different modes, whereas a coarser partition might merge them.

Labeler-free diversity. For each method and each prompt with at least 16 correct responses among the 96 samples, we draw 16 correct responses without replacement and compute the Vendi score (Friedman & Dieng, 2023), distinct-2 (Li et al., 2016), and self-BLEU (Zhu et al., 2018) on this fixed-size set. The filter retains 122–123 of the 128 prompts for every method, and no metric uses a mode labeler. We use six seeds for MILO and MaxRL and three for the other methods. MILO scores 3.6 on Vendi against 1.8 for MaxRL and at most 2.2 for the other baselines (Fig. 4b); GRPO, which that panel omits, also scores 1.8. MILO's distinct-2 is 0.45 against at most 0.36, and its self-BLEU is 0.44 against at least 0.54 (Fig. 6). The entropy bonus is the strongest baseline on all three metrics while covering only 9.5 modes, so these diversity metrics and correct-mode coverage measure different things.

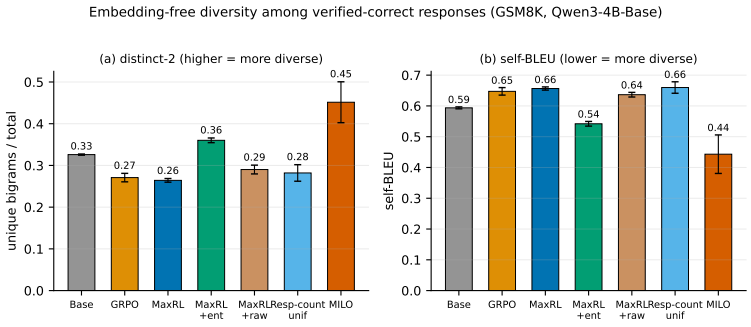


Figure 6: Labeler-free diversity among 16 correct responses per prompt on Qwen3-4B-Base GSM8K. **(a)** distinct-2 (higher is more diverse). **(b)** self-BLEU (lower is more diverse). Error bars are ± 1 SEM over seeds.

C.5 Verifier blind spots

This subsection gives the details behind Section 5.6. For a prompt and a policy, we take the policy's own $n = 96$ samples and treat the correct responses in its w most common correct modes as rejected

by the verifier. If c_w correct responses remain, solve@k is $1 - \binom{n-c_w}{k} / \binom{n}{k}$, the probability that a random subset of k of the 96 samples contains at least one of them; at $w = 0$ this is the pass@k estimator. For the random control, the w rejected modes of a prompt are drawn at random from the correct modes that either policy produces on that prompt in its seed-0 run, and the same rejected set is applied to every run of both policies. Results average over the 128 GSM8K evaluation prompts and over seeds (16 for MILO, 6 for MaxRL). Every 95% confidence interval over seeds (± 1.96 standard errors) has half-width at most 0.05, and at most 0.015 for $k \geq 16$.

Table 7 shows how the two settings differ. When the verifier misses a policy's own dominant strategies, MaxRL loses usable candidates quickly because its correct responses sit in a few modes, while MILO still has correct responses in other modes. Rejecting random modes rarely removes a policy's dominant strategies, so the two policies stay within 0.03 of each other, with MaxRL slightly ahead at $k = 4$ because its pass rate is slightly higher. We have not tested this setting with learned verifiers.

		Own most common modes rejected				Random modes rejected		
Budget	Method	$w = 0$	$w = 1$	$w = 2$	$w = 3$	$w = 1$	$w = 2$	$w = 3$
$k = 4$	MaxRL	0.944	0.791	0.620	0.486	0.939	0.928	0.913
	MILO	0.916	**0.853**	**0.803**	**0.761**	0.912	0.906	0.900
$k = 16$	MaxRL	0.983	0.946	0.896	0.827	0.983	0.981	0.964
	MILO	0.979	**0.968**	**0.956**	**0.943**	0.978	0.977	0.973
$k = 64$	MaxRL	0.998	0.985	0.969	0.946	0.998	0.997	0.984
	MILO	0.997	**0.992**	**0.986**	**0.977**	0.996	0.996	0.995

Table 7: solve@k on GSM8K. Left: the verifier rejects each policy's own w most common correct modes, as in Section 5.6. Right: it rejects w modes drawn at random, the same modes for both policies. At $w = 0$ the verifier is perfect and solve@k equals pass@k.