
Maximum Likelihood Reinforcement Learning

Anonymous Authors¹

Abstract

Maximum likelihood is fundamental to supervised learning but it cannot be directly applied in correctness-based problems with non-differentiable sampling. In these settings, reinforcement learning (RL) is typically used to maximize expected reward. We show that for binary correctness tasks, expected-reward RL is a first-order approximation of the maximum likelihood objective, yielding vanishing learning signal on low-success inputs. We introduce **Maximum Likelihood Reinforcement Learning (MaxRL)**, a compute-indexed family of sampling-based objectives derived from a $\text{pass}@k$ expansion of the likelihood, which interpolates between standard RL and exact maximum likelihood as compute increases. MaxRL admits a simple unbiased policy-gradient estimator whose optimized objective improves with additional compute. Across multiple domains, MaxRL consistently outperforms standard RL and GRPO, achieving higher $\text{pass}@1$ and substantially improved $\text{pass}@k$.

1. Introduction

Maximum likelihood training underlies much of the success of modern machine learning (Krizhevsky et al., 2012; Radford et al., 2018). In fully differentiable settings, optimizing log-likelihood objectives—most commonly through cross-entropy—has reliably translated increases in model capacity, data, and compute into consistent performance improvements. Beyond its empirical effectiveness, maximum likelihood enjoys strong theoretical properties: it defines a strictly proper scoring rule (Good, 1992; Savage, 1971; Gneiting & Raftery, 2007; Waghmare & Ziegel, 2025), yields statistically efficient estimators under standard assumptions (Vaart, 1998; Casella & Berger, 2002; Lehmann & Casella, 2006), and induces gradients that concentrate learning signal on

low-probability or uncertain outcomes through logarithmic weighting (Wang et al., 2020). Together, these properties make maximum likelihood a particularly effective and scalable optimization principle.

Many problems of practical interest share the same correctness-based structure as supervised learning while involving non-differentiable intermediate computations (Sutton et al., 1999; Bengio et al., 2013). Examples include navigation (Thrun et al., 2005; Anderson et al., 2018), program synthesis (Chen et al., 2018; Bunel et al., 2018), structured prediction (Smith, 2011; Mensch & Blondel, 2018), and multi-step reasoning in large language models (Wei et al., 2023), where outputs are generated through sampling procedures that produce trajectories, programs, or chains of thought. In these settings, each input still induces a well-defined notion of success or failure, suggesting an implicit maximum likelihood objective over correct outcomes. However, because the generation process is non-differentiable, this objective cannot be optimized directly, and reinforcement learning, typically via policy-gradient methods that maximize expected reward, serves as the standard surrogate.

Despite its practical utility, reinforcement learning does not generally inherit the optimization behavior of maximum likelihood, particularly in modern settings where generalization and coverage are critical. For example, recent works report that reinforcement learning objectives often induce distribution sharpening and reduced output diversity (Yue et al., 2025; Wu et al., 2026). Together, these observations motivate a closer examination of how different objectives shape learning dynamics by allocating gradient signal across inputs, independent of the particular sampling or optimization algorithm used.

To make this distinction precise, we analyze the two objectives at the *population level*. Let $p_\theta(x)$ denote the probability that a model with parameters θ succeeds on input x . The gradients of the expected-reward objective used in reinforcement learning and the maximum-likelihood objective then take the form

$$\begin{aligned}\nabla_\theta J_{\text{RL}} &= \mathbb{E}_x[\nabla_\theta p_\theta(x)], \\ \nabla_\theta J_{\text{ML}} &= \mathbb{E}_x[\nabla_\theta \log p_\theta(x)] = \mathbb{E}_x\left[\frac{1}{p_\theta(x)} \nabla_\theta p_\theta(x)\right].\end{aligned}$$

At this level of abstraction, the two objectives differ only

¹Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

in how gradient contributions are weighted *across inputs*. Reinforcement learning allocates gradient mass roughly uniformly, causing optimization to be dominated by easy examples and leaving hard inputs with vanishing learning signal. Maximum likelihood instead reweights gradients by inverse success probability, concentrating optimization effort on uncertain and failing cases. This contrast suggests that the training dynamics observed during reinforcement learning are not a consequence of its sampling mechanism, but rather of its *objective function*. This raises a fundamental question:

What objective should reinforcement learning optimize in correctness-based problems?

Historically, this objective mismatch was less visible because classical reinforcement learning was primarily studied in settings where agents interacted with a single environment (Sutton et al., 1998; Todorov et al., 2012; Mnih et al., 2013; 2015) or a small task family, and where expected-reward and likelihood-based objectives differed mainly by scaling and induced similar gradient directions. Modern applications of reinforcement learning to foundation models operate in a different regime, where *generalization* across heterogeneous inputs and *diversity* of correct outputs are central concerns, and where additional rollouts can be obtained by *scaling compute*.

In such settings, we argue that maximum likelihood provides a natural and principled objective for reinforcement learning. To be concrete, we show that the maximum likelihood objective admits an absolutely convergent failure-series (pass@ k) expansion, and that the standard reinforcement learning objective—expected correctness—arises as the *first-order Taylor approximation* of this objective with respect to success probability, thereby discarding higher-order curvature information critical for learning on hard examples. Truncating the series at higher orders yields a compute-controlled family of objectives that interpolate smoothly between expected-reward optimization and exact maximum likelihood. We refer to this framework as **Maximum Likelihood Reinforcement Learning (MAXRL)**. MAXRL retains a REINFORCE-style, sampling-based implementation while progressively recovering the optimization behavior of maximum likelihood as additional compute is allocated.

2. Preliminaries

In this work, we consider reinforcement learning settings that involve *generalization*, where models learn from a set of tasks and are evaluated on a heldout task distribution. We focus on correctness-based problems that can be abstracted as a *binary success or failure* outcome for each input. Formally, let \mathcal{X} and \mathcal{Y} denote the input and output spaces, and

let $x \sim \rho$ be the distribution over tasks. For each input x , we denote $y^*(x) \in \mathcal{Y}$ as the (unique) correct label or answer. Finally, we let the learner be parameterized by θ and denote the predictive distribution induced by the model as $p_\theta(y | x)$, where $p_\theta(\cdot | x) \in \Delta(\mathcal{Y})$ is a conditional probability distribution over outputs for a fixed input x . All logarithms use base e unless stated otherwise.

Latent generation models. In many modern settings, the model does not sample outputs directly from \mathcal{Y} , but instead generates a latent variable $z \in \mathcal{Z}$ according to a conditional distribution $m_\theta(z | x)$. The final output $y \in \mathcal{Y}$ is then obtained via a deterministic decoding function $y = f(z)$, such as parsing a generated program or extracting a boxed answer from a chain of thought. Correctness is evaluated only on the decoded output, i.e., a trajectory z is successful if $f(z) = y^*(x)$. Throughout the paper, expectations with respect to model outputs should be understood as expectations over latent samples $z \sim m_\theta(\cdot | x)$ followed by deterministic decoding.

Pass rate. We define the *pass rate* as the probability that the model produces the correct answer for a fixed input x :

$$p_\theta^{\text{pass}}(x) := p_\theta(y^*(x) | x) = \mathbb{E}_{y \sim p_\theta(\cdot | x)}[\mathbb{I}\{y = y^*(x)\}].$$

Similarly, let $y_1, \dots, y_k \stackrel{\text{i.i.d.}}{\sim} p_\theta(\cdot | x)$. We define *pass@ k* as the probability of at least one correct sample:

$$\text{pass}@k(x) := \mathbb{P}(\exists i \in [k] \text{ s.t. } y_i = y^*(x)).$$

Next, we consider two frameworks for training our models: *maximum likelihood* and *reinforcement learning*.

Maximum likelihood (ML). Implementing the maximum likelihood objective for a binary classification problem yields the following training objective

$$J_{\text{ML}}(\theta) := \mathbb{E}_{x \sim \rho} [\log p_\theta(y^*(x) | x)] = \mathbb{E}_{x \sim \rho} [\log(p_\theta^{\text{pass}}(x))].$$

Reinforcement learning (RL). For a binary correctness based task, we can also define a binary reward function $r(x, y) = \mathbb{I}\{y = y^*(x)\}$, and similarly under the latent variable case define $r(x, z) = \mathbb{I}\{f(z) = y^*(x)\}$. In this binary reward setting, the RL objective becomes (using the latent version without loss of generality):

$$J_{\text{RL}}(\theta) := \mathbb{E}_{x \sim \rho} [\mathbb{E}_{z \sim m_\theta(\cdot | x)} [r(x, z)]] = \mathbb{E}_{x \sim \rho} [p_\theta^{\text{pass}}(x)]. \tag{1}$$

3. Maximum Likelihood Reinforcement Learning (MAXRL)

In this section, we show that reinforcement learning on expected reward optimizes only a low-order approximation of

the likelihood objective: while maximum likelihood objective admits a population-level expansion in terms of pass@ k events, with standard RL optimizing only the first-order term. This suggests a compute-indexed family of objectives that incorporate higher-order terms, converging to ML as more compute is allocated.

3.1. Maclaurin Expansion of Maximum Likelihood

The maximum likelihood objective admits the Maclaurin (failure-series) expansion:

$$\nabla_{\theta} J_{\text{ML}}(x) = \log p = - \sum_{k=1}^{\infty} \frac{(1-p)^k}{k} = - \sum_{k=1}^{\infty} \frac{\text{fail}@k(x)}{k}, \quad (2)$$

where $\text{fail}@k(x) = 1 - \text{pass}@k(x)$ denotes the probability that all k i.i.d. samples from the model fail. Differentiating (2) yields the population-level gradient identity

$$\nabla_{\theta} J_{\text{ML}}(x) = \sum_{k=1}^{\infty} \frac{1}{k} \nabla_{\theta} \text{pass}@k(x) \quad (3)$$

Thus, maximum likelihood optimizes an infinite harmonic mixture of pass@ k gradients, with higher-order terms encoding rare success patterns critical when p is small. In contrast, standard reinforcement learning optimizes only the expected pass@1 objective,

$$\nabla_{\theta} J_{\text{RL}}(x) = \nabla_{\theta} \text{pass}@1(x),$$

corresponding to retaining solely the leading term of (3). From this perspective, reinforcement learning is best understood as a *first-order approximation* of the maximum likelihood objective in correctness space.

3.2. MAXRL Objective Function

Optimizing the full infinite mixture in (3) is infeasible under finite compute, suggesting that maximum likelihood cannot be optimized exactly. In particular, estimating pass@ k gradients for large k requires an increasing number of samples, especially when the pass rate p is small. This motivates approximating the maximum likelihood objective by truncating the expansion (3) at finite order, and then estimating such objective instead.

For a truncation level $T \in \mathbb{N}$, we define the truncated maximum likelihood objective for a fixed input x as

$$J_{\text{MAXRL}}^{(T)}(x) := - \sum_{k=1}^T \frac{(1-p)^k}{k}. \quad (4)$$

Differentiating (4) yields the truncated population gradient

$$\nabla_{\theta} J_{\text{MAXRL}}^{(T)}(x) = \sum_{k=1}^T \frac{1}{k} \nabla_{\theta} \text{pass}@k(x). \quad (5)$$

This defines a compute-indexed hierarchy: $\mathbf{T} = \mathbf{1}$ recovers **RL**, $\mathbf{T} \rightarrow \infty$ recovers **maximum likelihood**, and intermediate T interpolate between them.

Thus, the truncation level T directly controls the order of correctness events that contribute to learning, providing a principled mechanism for trading additional compute for higher-fidelity approximations to the maximum likelihood objective. The remaining question is whether these truncated objectives admit simple, unbiased estimators under finite sampling, a question that we answer affirmatively in the next section.

4. Gradient Estimators for MAXRL

Equation (5) already provides a viable approach for constructing an unbiased estimator: approximate *each* term in the finite series using a pass@ k gradient estimator, as provided in prior work (Walder & Karkhanis, 2025; Chen et al., 2025d). Under this strategy, improvements in pass@ k estimators directly translate into improved estimators for the truncated maximum likelihood objective in Equation (5).

In this work, we instead derive a simpler estimator from a conditional sampling perspective. The key insight is that the maximum likelihood gradient can be expressed as an expectation under the *success-conditioned* distribution:

Theorem 4.1 (Conditional Form of the Maximum Likelihood Gradient). *The gradient of the maximum likelihood objective admits the following conditional expectation representation:*

$$\nabla_{\theta} J_{\text{ML}}(x) = \mathbb{E}[\nabla_{\theta} \log m_{\theta}(z | x) | f(z) = y^*(x)].$$

All proofs in this section can be found in Appendix B. Next, we show how to leverage this result to construct practical gradient estimator for the truncated expansion in Equation (5).

4.1. Empirical Gradient Estimator

Fix an input x and draw N latent trajectories $z_1, \dots, z_N \sim m_{\theta}(\cdot | x)$. Let $r_i := \mathbb{I}\{f(z_i) = y^*(x)\}$ indicate success, $S_i := \nabla_{\theta} \log m_{\theta}(z_i | x)$ denote the score function, and $K := \sum_{i=1}^N r_i$ be the number of successful samples.

Following the conditional expectation form, we average score functions over successful trajectories:

$$\hat{g}_N(x) := \begin{cases} \frac{1}{K} \sum_{i=1}^N r_i S_i, & K \geq 1, \\ 0, & K = 0. \end{cases} \quad (6)$$

We can show that this **conditional estimator** is unbiased for the gradient of the truncated maximum likelihood objective in Equation (5) with truncation level $T = N$:

Table 1. Side-by-side comparison of the REINFORCE estimator and the conditional estimator. While the two estimators differ only in their normalization (N versus K), they are unbiased for fundamentally different objectives.

	REINFORCE	Conditional Estimator
Estimator	$\frac{1}{N} \sum_{i=1}^N r_i S_i$	$\frac{1}{K} \sum_{i=1}^N r_i S_i$
Unbiased for	$\nabla_{\theta} \text{pass@1}(x)$	$\sum_{k=1}^N \frac{1}{k} \nabla_{\theta} \text{pass@k}(x)$

Theorem 4.2 (Estimator–objective equivalence). *The estimator $\hat{g}_N(x)$ is an unbiased estimator for the MAXRL gradient of order $T = N$, i.e.,*

$$\mathbb{E}[\hat{g}_N(x)] = \nabla_{\theta} J_{\text{MAXRL}}^{(N)}(x).$$

Theorem 4.2 reveals an elegant alignment between the conditional estimator in Equation (6) and the truncated Maclaurin estimator in Equation (5). Increasing compute therefore improves the *objective being optimized* with respect to the maximum likelihood objective, not merely the variance of its estimator as in standard REINFORCE-style estimators.

Table 1 compares our estimator with REINFORCE¹, which underlies most RL algorithms. At the estimator level, the difference is simple: both average score functions over sampled trajectories, but REINFORCE normalizes by total samples N while MaxRL normalizes by successful samples K . This difference in normalization determines the objective each estimator is unbiased for.

Consequently, increasing samples N has different effects: REINFORCE reduces variance of a fixed objective (pass@1), while MaxRL increases the approximation order to maximum likelihood. Additional compute thus improves the *objective itself* for MAXRL, not just estimation quality.

4.2. Variance Reduction via Control Variates

Like REINFORCE, the estimator (6) can exhibit high variance when successful samples K is small. Policy-gradient baselines are typically introduced to reduce variance without changing the expected gradient. However, standard arguments for policy-gradient baselines are not directly applicable in this setting, as the estimator normalizes by the random quantity K (which depends on all samples) rather than a fixed sampling budget.

¹Modern policy-gradient methods such as PPO (Schulman et al., 2017) introduce additional mechanisms (clipping, importance weight truncation) that trade bias for robustness. In the on-policy setting, these reduce to REINFORCE, our canonical baseline. GRPO is a notable exception, which we discuss further in Section 5.

We instead use a simple zero-mean control variate: the unconditional average score

$$V_N := \frac{1}{N} \sum_{i=1}^N \nabla_{\theta} \log m_{\theta}(z_i | x),$$

which satisfies $\mathbb{E}[V_N] = 0$. Subtracting V_N preserves unbiasedness while reducing variance:

$$\tilde{g}_N(x) = \frac{1}{K} \sum_{i=1}^N r_i S_i - \frac{1}{N} \sum_{i=1}^N S_i = \sum_{i=1}^N \left(\frac{r_i}{K} - \frac{1}{N} \right) S_i, \tag{7}$$

with the convention that the first term is zero when $K = 0$. In practice, we drop both terms when $K = 0$, as this variant is simpler and performs better empirically; for reference, we present the final pseudocode in Section C.

5. A Unifying Weight-Function View

In this section, we provide a unified view of all objectives, including Maximum likelihood, MAXRL, classical reinforcement learning, and GRPO. Specifically, their population-level gradients admit a weighted representation

$$\nabla_{\theta} J = \mathbb{E}_{x \sim \rho} [w(p_{\theta}(x)) \nabla_{\theta} p_{\theta}(x)], \tag{8}$$

where $p_{\theta}(x) = p_{\theta}^{\text{pass}}(x)$ and the weighting function $w(p)$ determines how learning signal is allocated across inputs of different difficulty. Figure 1 shows $w(p)$ for each objective (proof in Appendix B).

This weight perspective also provides a useful reinterpretation of GRPO (Shao et al., 2024). Although GRPO is heuristically motivated by Z-normalization using the empirical standard deviation, such normalization induces a fundamentally different population-level objective than REINFORCE, a conclusion also reached by recent work (Davis & Recht, 2025; Liu et al., 2025b; Xiong et al., 2025b). Relative to standard expected-reward optimization, GRPO upweights low-pass-rate inputs approximately as $1/\sqrt{p}$ when p is small, placing it between classical reinforcement learning and maximum likelihood. However, increasing compute via additional sampling under GRPO does not yield a better approximation to the maximum likelihood objective, as the induced population loss is fundamentally distinct. Moreover, as shown in Figure 1, the GRPO weighting function *inverts* for sufficiently large pass rates, increasing as $p \rightarrow 1$, unlike likelihood-based objectives. Consequently, GRPO assigns increased weight to very easy inputs when they are present, in contrast to the other formulations.²

Figure 1 shows that as T increases, MaxRL approaches ML weighting in the low-pass regime while remaining bounded

²We conjecture that this inversion may contribute to distribution sharpening when datasets contain a substantial fraction of overly easy inputs, and leave a detailed analysis to future work.

Table 2. Population-level weighting functions $w(p)$.

	RL	GRPO	MAXRL (T)	ML
$w(p)$	1	$\frac{1}{\sqrt{p(1-p)}}$	$\frac{1 - (1-p)^T}{p}$	$\frac{1}{p}$

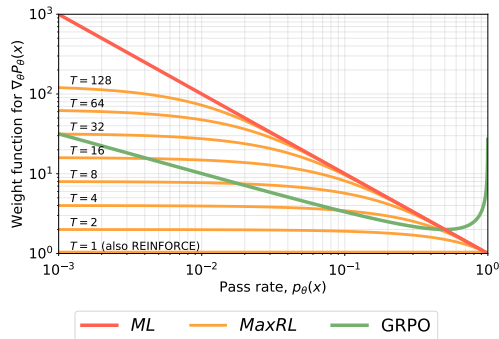


Figure 1. Population-level weighting functions $w(p)$ as a function of pass rate p . Truncated objectives $\log_T p_\theta$ interpolate between (REINFORCE) and maximum likelihood as T increases.

at moderate p . The key distinction among objectives is *how strongly they emphasize hard, low-probability inputs*. MaxRL uniquely recovers ML weighting with increased compute, aligning RL with likelihood-based optimization.

6. Experiments

In this section, we empirically evaluate MAXRL on non-differentiable correctness-based tasks.

We begin in Section 6.1 with a controlled setting where exact maximum likelihood optimization is possible, allowing direct comparison with MAXRL as compute increases. We then study non-differentiable correctness-based tasks in two regimes: (i) an *effectively infinite-data* setting with continually novel tasks (Section 6.2), and (ii) a *data-scarce* setting with a fixed training dataset (Section 6.3). Finally, in Section 6.4, we evaluate billion-parameter reasoning models on mathematical problem-solving tasks, testing whether the benefits of MAXRL extend to large-scale LLM training.

Because we compare training objectives rather than algorithms, all methods are trained on-policy. We compare against REINFORCE with a leave-one-out baseline (RLOO) (Ahmadian et al., 2024) and Group Relative Policy Optimization (GRPO) (Shao et al., 2024).

6.1. Comparisons with Exact Likelihood

As a first step, we evaluate how closely MAXRL approximates *exact* maximum likelihood in a setting where likelihood optimization is directly available. We compare three objectives: (i) reinforcement learning on expected reward,

(ii) MAXRL, and (iii) exact maximum likelihood training. We consider a standard image classification task, where maximum likelihood corresponds to minimizing cross-entropy. The reinforcement learning reward is defined as 1 if the predicted class matches the ground-truth label and 0 otherwise.

We instantiate this comparison on ImageNet (Deng et al., 2009) using a ResNet-50 trained under each objective; full experimental details are provided in Appendix D. Figure 2 summarizes the results. REINFORCE (with a standard baseline) fails to achieve meaningful improvements in accuracy even as the per-input sampling budget increases, whereas exact maximum likelihood training yields steady gains.

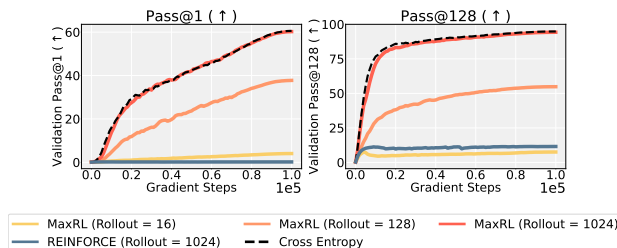


Figure 2. (ImageNet) Comparison of training dynamics under exact maximum likelihood, MAXRL, and REINFORCE in a controlled image classification setting. With sufficient rollouts, MAXRL closely matches cross-entropy training, while REINFORCE fails to make progress from low initial pass rates.

In contrast, MAXRL is trained on the same samples and observes the same sparse set of successful trajectories as REINFORCE, but makes more effective use of this limited learning signal through likelihood-inspired reweighting. As the compute increases by means of higher rollout counts, MAXRL improves consistently and closely tracks exact maximum likelihood.

6.2. Infinite Data Regime

Table 3. Performance comparison across methods in maze.

Method	Pass@1	Pass@128	Pass@256
GRPO	39.6	42.4	43.0
GRPO (with entropy bonus)	47.2	53.4	54.0
PKPO ($T = 16$) (Walder & Karkhanis, 2025)	74.5	77.6	77.9
SELF (Nguyen et al., 2025b)	46.1	86.3	87.5
Differential Smoothing (Gai et al., 2025)	50.0	57.8	58.7
MaxRL	84.4	92.0	94.3

To study training with continually fresh data, we construct a procedurally generated maze-navigation environment. Each training input is a newly generated maze, and the model never encounters the same maze twice during training; multiple valid solution paths may exist for a given maze. We reserve a held-out set of 256 mazes for evaluation and apply a brief supervised pretraining phase—uniformly across all methods—to ensure a non-zero initial pass rate. Full task details are provided in Appendix E.1.

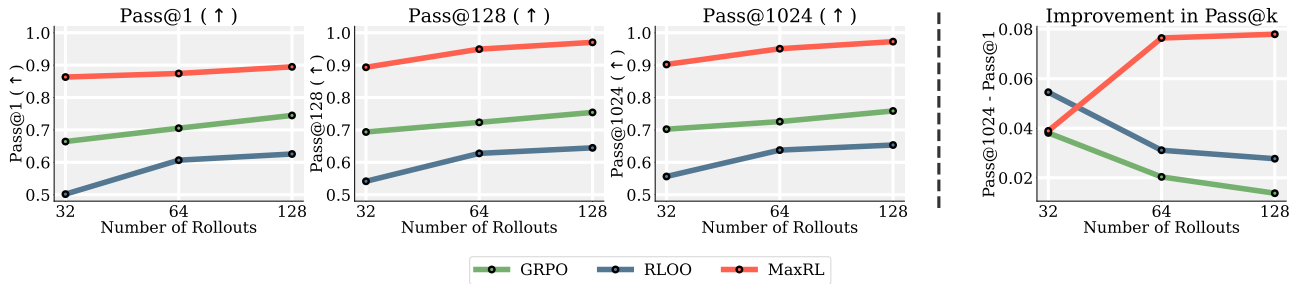


Figure 3. (Maze) Scaling behavior with increasing rollouts per prompt.

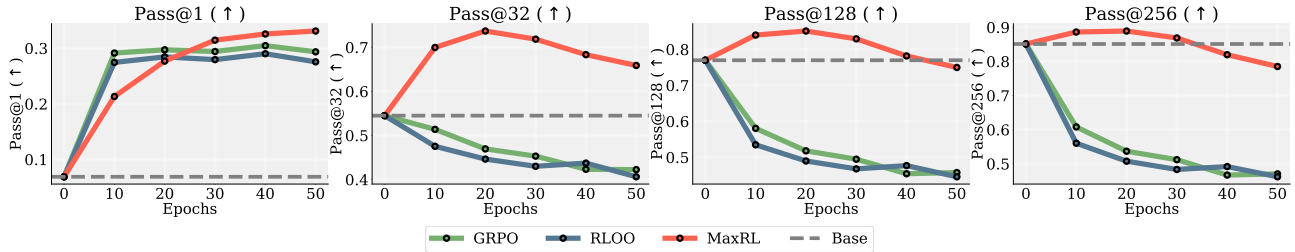


Figure 4. (GMS8K) Training dynamics on GSM8K with a fixed dataset and increasing training compute. MAXRL shows slower initial gains but sustained improvement and substantially less pass@k degradation compared to GRPO and REINFORCE.

We train a lightweight transformer model (Vaswani et al., 2017) with approximately 3M parameters and simulate extended training by running 9K RL steps with up to 128 rollouts per prompt, varying the number of rollouts to control compute. We report performance after 9K steps in Figure 3 as a function of rollouts per prompt.

All three objectives—REINFORCE with a leave-one-out baseline (RLOO), GRPO, and MAXRL—improve upon the base model. However, performance is clearly stratified: MAXRL consistently outperforms GRPO, which in turn outperforms RLOO, indicating progressively more effective use of additional compute. Table 3 and Figure 18 show comparison with additional baselines in this setting.

6.3. Data-Scarce Regime

Table 4. Performance comparison across methods on GSM8K.

Method	Pass@1	Pass@128	Pass@1024
GRPO (Shao et al., 2024)	29.3	45.8	48.8
RLOO (Ahmadian et al., 2024)	27.5	44.6	48.5
GRPO (with entropy bonus)	31.1	48.1	51.6
PKPO (T = 16) (Walder & Karkhanis, 2025)	30.7	67.2	75.9
Differential Smoothing (Gai et al., 2025)	31.4	48.5	52.3
MaxRL	33.2	75.0	83.4

We next consider a data-scarce regime in which models are trained for many epochs over a fixed dataset until they reach peak performance. Unlike the infinite-data setting in Section 6.2, this regime exposes differences in how objectives allocate learning signal under repeated training.

We train a SmoLLM2-360M-Instruct model (Allal et al., 2025) on GSM8K (Cobbe et al., 2021), consisting of 7,473

grade-school math problems, for up to 50 epochs. We report training dynamics in Figure 4; additional details are provided in Appendix E.

All methods improve upon the base model, with the familiar stratification: MAXRL outperforms GRPO, which in turn outperforms RLOO. However, their training dynamics differ substantially. Both RLOO and GRPO rapidly peak in pass@1 performance—around 10 epochs—and exhibit pronounced pass@k degradation under continued training, consistent with prior observations of distribution sharpening (Yue et al., 2025).

In contrast, MAXRL shows slower initial gains but sustained improvement. Pass@1 overtakes competing methods at approximately 30 epochs and continues to increase through the end of training, while pass@k remains substantially healthier and exceeds the base model for a large portion of training. This behavior suggests that preserving higher-order correctness provides a continued source of learning signal under repeated exposure to the same data. For comparison with additional baselines, refer to Table 4.

6.4. Large Reasoning Model Training

We next demonstrate that the benefits of MAXRL extend to large-scale LLM reasoning training. We train Qwen3-1.7B-Base and Qwen3-4B-Base models on POLARIS-53K (An et al., 2025), a dataset of approximately 50K mathematical reasoning prompts.

We evaluate on four standard math benchmarks: AIME 2025, BeyondAIME (ByteDance-Seed, 2025), MATH-

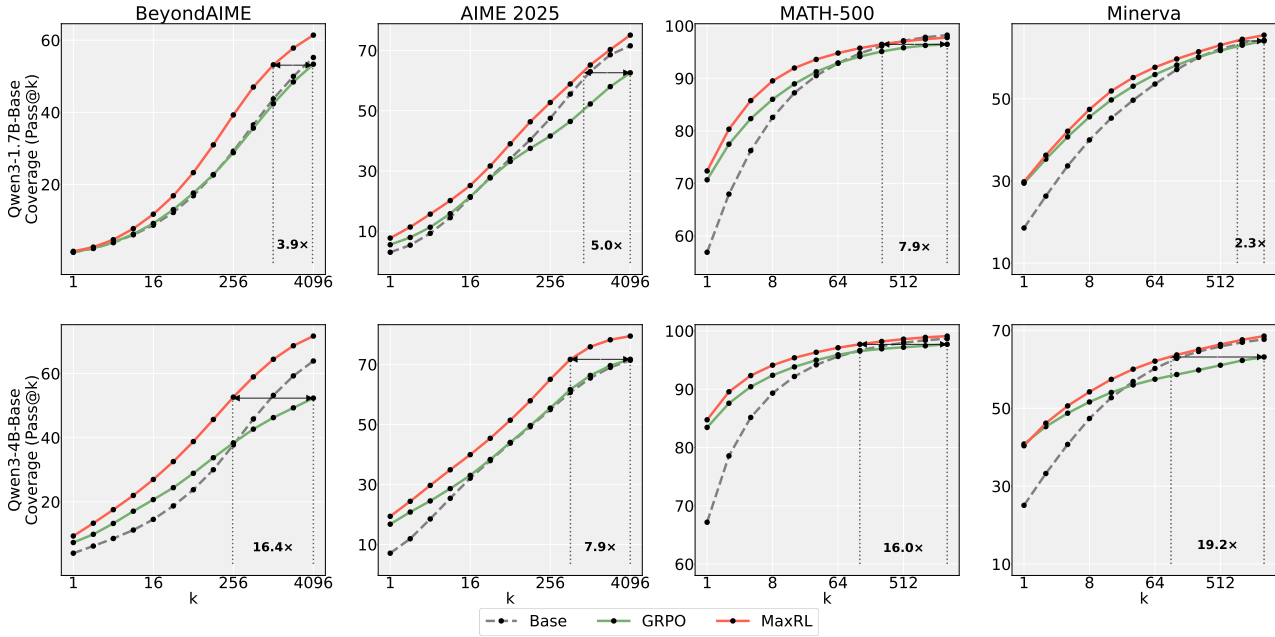


Figure 5. (Qwen3 training results) Evaluation of final checkpoints from training Qwen3-1.7B-Base and Qwen3-4B-Base models. MAXRL match or outperform GRPO in all 4 evaluation datasets and shows little to no degradation at coverage (pass@k) for very high k values. We also note the increase in inference efficiency: MAXRL can provide 2.3x - 19.2x speedup compared to GRPO while generating multiple samples with a perfect verifier and maintains similar or better pass@1 performance.

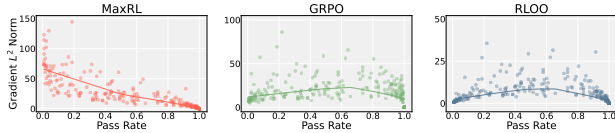


Figure 6. (Gradient norm analysis) To compare different objectives qualitatively, we show a scatter plot of gradient L^2 norm vs pass rate over individual prompts. We use Qwen2.5-1.5B-Instruct on MATH-500 dataset for this analysis. MAXRL generate larger gradient norms over prompts with close to 0 pass rates.

500 (Hendrycks et al., 2021; Lightman et al., 2023), and Minerva (Lewkowycz et al., 2022). We compare against GRPO, a widely used baseline for large-scale reasoning. All methods are trained under the same compute budget (256 prompts per batch, 16 rollouts per prompt, and 1000 RL steps); additional details are provided in Appendix E.3.

Figure 5 summarizes the results. Across both model sizes, MAXRL consistently Pareto dominates GRPO, achieving higher pass@1 while simultaneously improving pass@k. Consistent with prior work (Yue et al., 2025; Wu et al., 2026), GRPO exhibits pronounced pass@k degradation at larger k. In contrast, MAXRL improves pass@k relative to both the pretrained base model and the GRPO-trained checkpoint in 7 out of 8 evaluation settings. Improved pass@k directly translates into inference efficiency under repeated sampling. As shown in Figure 5, MAXRL achieves comparable or better pass@k using up to 20x fewer samples than

GRPO, yielding substantial practical savings at inference time.

Analysis of Training Dynamics. As shown in Figure 6, MAXRL assigns larger gradient norms to difficult prompts than GRPO and RLOO, leading to a higher fraction of training prompts with non-zero pass rate over training (Figure 7). Additional analyses are provided in Appendix I.

7. Related Works

Supervised training vs reinforcement learning. Supervised learning and reinforcement learning (RL) are complementary but fundamentally different paradigms. Supervised training is stable, sample-efficient, and well-calibrated within the training distribution (Ng & Jordan, 2001), but it is limited by the quality and scope of available data and cannot directly optimize non-differentiable objectives such as correctness or preferences. In contrast, RL — typically via policy gradients (Williams, 1992; Sutton et al., 1999; 1998; Schulman et al., 2017; Guo et al., 2017) — can optimize such objectives directly and improve performance beyond available demonstrations by having access to interactions with an environment and the resulting reward-based feedback. Although recent work has reframed RL objectives as supervised ones (Rafailov et al., 2023), on-policy learning, characteristic of online RL algorithms, appears crucial for optimal performance (Tajwar et al., 2024; Xu et al., 2024).

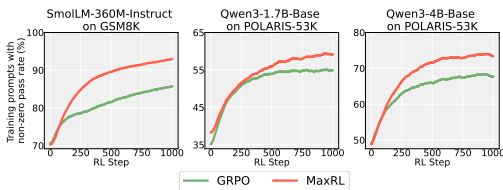


Figure 7. (Training dynamics comparison) Fraction of prompts where the model generates at least one correct rollout (out of 128, 16, and 16 rollouts for SmolLM-360M-Instruct, Qwen3-1.7B-Base, and Qwen3-4B-Base, respectively) during training. MAXRL consistently produces at least one correct rollout for more prompts across all settings, demonstrating its effectiveness at extracting more learning signal from the training dataset.

Modern foundation model training often combines supervised learning on human data with subsequent RL (Ouyang et al., 2022). Unlike these approaches, we assume no access to high-quality demonstrations or a stronger model, and instead study a purely interactive RL setting that nonetheless optimizes an objective that mimics cross-entropy. We discuss additional related works in Appendix A.

Training LLMs for strong reasoning abilities. Reinforcement learning from verifiable rewards (RLVR), where LLMs receive reward from a ground truth verifier instead of using a trained reward model, has emerged as the dominant paradigm for instilling strong reasoning capabilities into LLMs (OpenAI et al., 2024; Guo et al., 2025; Team et al., 2025; Lambert et al., 2025; Yang et al., 2025). Whereas supervised training learns better behavior from fixed static datasets, reinforcement learning uses policy gradient algorithms (e.g., PPO (Schulman et al., 2017), GRPO (Shao et al., 2024), RLOO (Ahmadian et al., 2024)) to learn from self-generated responses and non-differentiable rewards. However, these algorithms and their variants (Zheng et al., 2025; Liu et al., 2025b; MiniMax et al., 2025) optimize expected reward or pass rate and only differs in how the advantage or off-policy updates are calculated. In contrast, the goal of our work is to propose a fundamentally different objective for RL training.

RL training causes distribution sharpening. Despite its usefulness, questions remain on whether RLVR teaches LLMs fundamentally new behavior/skills, or simply sharpens existing good behavior from the pretrained model. Prior works (Liu et al., 2025b; Zhao et al., 2025; AI et al., 2025) demonstrated that certain reasoning skills like reflection already exist in the pretrained model, and Gandhi et al. (2025) shows that good reasoning behaviors learned from pretraining is crucial for the success of RLVR in the post-training phase. More recently, studies (Yue et al., 2025; Dang et al., 2025; Wu et al., 2026) found that RLVR decreases the model’s diversity by reducing pass@k. In our paper, we confirm these findings and attribute this to the RL

objective itself — we demonstrate that optimizing expected reward tends to marginalize learning signal from harder prompts, which results in distribution sharpening.

Learning to solve hard problems. Due to RLVR’s shrinking of model coverage, significant attention has been drawn to new RL algorithms mitigating pass@k collapse. Approaches range from directly optimizing for pass@k during training (Walder & Karkhanis, 2025; Tang et al., 2025) to employing exploration bonuses in RL (Song et al., 2025; Tuyls et al., 2025). We show in our work that pass@k optimization objectives are a special case of our objective, since it optimizes an infinite harmonic series of pass@k objectives. On the other hand, the other works carry the fundamental limitation of RLVR of maximizing expected reward or pass rate over a batch of prompts, which we demonstrate to have vanishing gradient for prompts with low pass rate. This issue is also recognized by Nguyen et al. (2025b), which introduces selective learning only on prompts where greedy response fails, but unlike us, does not weigh prompts differently based on their pass rate. One exception is the recent work, Xiong et al. (2025b) — like us, they suggest to investigate non-linear functions of the pass rate. Although our papers have similar motivation, the development of them is different. Our work establishes connections between a cross-entropy-like objective and pass@k optimization, which then motivates our practical objectives. Empirically, our work focuses on an on-policy validation of the theoretical ideas we introduce, and provide didactic experiments where the compute scaling can be thoroughly investigated, whereas Xiong et al. (2025b)’s effort instead focuses on adaptive rollout budget allocation rules, similar to Yao et al. (2025), and thus offers a complementary investigation. We discuss additional related works in Appendix A.

8. Conclusion

In this work, we introduced MAXRL, a framework for reinforcement learning that models maximum likelihood as the training objective in non-differentiable binary reward settings. We showed that MAXRL approaches maximum likelihood in differentiable settings as compute increases, and that in non-differentiable settings it offers key advantages over traditional RL—scaling more effectively with additional compute while exhibiting substantially less diversity reduction. More broadly, our results suggest that some limitations attributed to reinforcement learning with foundation models arise from objective choice rather than optimization or sampling. Our work currently assumes a binary reward setting and does not directly extend to continuous or arbitrarily valued rewards. Generalizing MAXRL to continuous rewards, multi-turn reinforcement learning, and off-policy settings such as PPO-style training are promising directions for future work.

Impact Statement

This paper presents work whose goal is to advance the field of machine learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

- Afriyie, J. K., Tawiah, K., Pels, W. A., Addai-Henne, S., Dwamena, H. A., Owiredu, E. O., Ayeh, S. A., and Eshun, J. A supervised machine learning algorithm for detecting and predicting fraud in credit card transactions. *Decision Analytics Journal*, 6:100163, 2023. ISSN 2772-6622. doi: <https://doi.org/10.1016/j.dajour.2023.100163>. URL <https://www.sciencedirect.com/science/article/pii/S2772662223000036>.
- Ahmadian, A., Cremer, C., Gallé, M., Fadaee, M., Kretutzer, J., Pietquin, O., Üstün, A., and Hooker, S. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms, 2024. URL <https://arxiv.org/abs/2402.14740>.
- AI, E., :, Shah, D. J., Rushton, P., Singla, S., Parmar, M., Smith, K., Vanjani, Y., Vaswani, A., Chaluvaraju, A., Hojel, A., Ma, A., Thomas, A., Polloreno, A., Tanwer, A., Sibai, B. D., Mansingka, D. S., Shivaprasad, D., Shah, I., Stratos, K., Nguyen, K., Callahan, M., Pust, M., Iyer, M., Monk, P., Mazarakis, P., Kapila, R., Srivastava, S., and Romanski, T. Rethinking reflection in pre-training, 2025. URL <https://arxiv.org/abs/2504.04022>.
- Allal, L. B., Lozhkov, A., Bakouch, E., Blázquez, G. M., Penedo, G., Tunstall, L., Marafioti, A., Kydlíček, H., Lajarín, A. P., Srivastav, V., Lochner, J., Fahlgren, C., Nguyen, X.-S., Fourier, C., Burtenshaw, B., Larcher, H., Zhao, H., Zakka, C., Morlon, M., Raffel, C., von Werra, L., and Wolf, T. Smollm2: When smol goes big – data-centric training of a small language model, 2025. URL <https://arxiv.org/abs/2502.02737>.
- An, C., Xie, Z., Li, X., Li, L., Zhang, J., Gong, S., Zhong, M., Xu, J., Qiu, X., Wang, M., and Kong, L. Polaris: A post-training recipe for scaling reinforcement learning on advanced reasoning models, 2025. URL <https://hkunlp.github.io/blog/2025/Polaris>.
- Anderson, P., Wu, Q., Teney, D., Bruce, J., Johnson, M., Sünderhauf, N., Reid, I., Gould, S., and van den Hengel, A. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments, 2018. URL <https://arxiv.org/abs/1711.07280>.
- Anonymous. Entropy-preserving reinforcement learning. In *Submitted to The Fourteenth International Conference on Learning Representations*, 2025a. URL <https://openreview.net/forum?id=E8MR8jgEeZ>. under review.
- Anonymous. Exploratory memory-augmented LLM agent via hybrid on- and off-policy optimization. In *Submitted to The Fourteenth International Conference on Learning Representations*, 2025b. URL <https://openreview.net/forum?id=UOzxviKVFO>. under review.
- Auer, P., Cesa-Bianchi, N., and Fischer, P. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47:235–256, 2002.
- Belkhal, S., Cui, Y., and Sadigh, D. Data quality in imitation learning, 2023. URL <https://arxiv.org/abs/2306.02437>.
- Bellemare, M. G., Srinivasan, S., Ostrovski, G., Schaul, T., Saxton, D., and Munos, R. Unifying count-based exploration and intrinsic motivation, 2016. URL <https://arxiv.org/abs/1606.01868>.
- Bellman, R. A markovian decision process. *Journal of Mathematics and Mechanics*, 6(5):679–684, 1957. ISSN 00959057, 19435274. URL <http://www.jstor.org/stable/24900506>.
- Bengio, Y., Léonard, N., and Courville, A. Estimating or propagating gradients through stochastic neurons for conditional computation, 2013. URL <https://arxiv.org/abs/1308.3432>.
- Bojarski, M., Testa, D. D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., Zhang, X., Zhao, J., and Zieba, K. End to end learning for self-driving cars, 2016. URL <https://arxiv.org/abs/1604.07316>.
- Boucher, R. H., Semeraro, O., and Mathelin, L. Evidence on the regularisation properties of maximum-entropy reinforcement learning, 2025. URL <https://arxiv.org/abs/2501.17115>.
- Bunel, R., Hausknecht, M., Devlin, J., Singh, R., and Kohli, P. Leveraging grammar and reinforcement learning for neural program synthesis, 2018. URL <https://arxiv.org/abs/1805.04276>.
- Burda, Y., Edwards, H., Storkey, A., and Klimov, O. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018.
- ByteDance-Seed. Beyondaime: Advancing math reasoning evaluation beyond high school olympiads. [<https://huggingface.co/datasets/ByteDance-Seed/BeyondAIME>]

- (<https://huggingface.co/datasets/ByteDance-Seed/BeyondAIME>), 2025.
- Casella, G. and Berger, R. L. *Statistical inference*, volume 2. Duxbury Pacific Grove, CA, 2002.
- Chang, J. D., Zhan, W., Oertell, O., Brantley, K., Misra, D., Lee, J. D., and Sun, W. Dataset reset policy optimization for rlhf, 2024. URL <https://arxiv.org/abs/2404.08495>.
- Chen, J. C.-Y., Peng, B. X., Choubey, P. K., Huang, K.-H., Zhang, J., Bansal, M., and Wu, C.-S. Nudging the boundaries of llm reasoning, 2025a. URL <https://arxiv.org/abs/2509.25666>.
- Chen, M., Tworek, J., Jun, H., Yuan, Q., de Oliveira Pinto, H. P., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., Ray, A., Puri, R., Krueger, G., Petrov, M., Khlaaf, H., Sastry, G., Mishkin, P., Chan, B., Gray, S., Ryder, N., Pavlov, M., Power, A., Kaiser, L., Bavarian, M., Winter, C., Tillet, P., Such, F. P., Cummings, D., Plappert, M., Chantzis, F., Barnes, E., Herbert-Voss, A., Guss, W. H., Nichol, A., Paino, A., Tezak, N., Tang, J., Babuschkin, I., Balaji, S., Jain, S., Saunders, W., Hesse, C., Carr, A. N., Leike, J., Achiam, J., Misra, V., Morikawa, E., Radford, A., Knight, M., Brundage, M., Murati, M., Mayer, K., Welinder, P., McGrew, B., Amodei, D., McCandlish, S., Sutskever, I., and Zaremba, W. Evaluating large language models trained on code, 2021. URL <https://arxiv.org/abs/2107.03374>.
- Chen, P., Li, X., Li, Z., Yin, W., Chen, X., and Lin, T. Exploration vs exploitation: Rethinking rlvr through clipping, entropy, and spurious reward, 2025b. URL <https://arxiv.org/abs/2512.16912>.
- Chen, X., Liu, C., and Song, D. Towards synthesizing complex programs from input-output examples, 2018. URL <https://arxiv.org/abs/1706.01284>.
- Chen, X., Lu, J., Kim, M., Zhang, D., Tang, J., Piché, A., Gontier, N., Bengio, Y., and Kamaloo, E. Self-evolving curriculum for llm reasoning, 2025c. URL <https://arxiv.org/abs/2505.14970>.
- Chen, Z., Qin, X., Wu, Y., Ling, Y., Ye, Q., Zhao, W. X., and Shi, G. Pass@k training for adaptively balancing exploration and exploitation of large reasoning models, 2025d. URL <https://arxiv.org/abs/2508.10751>.
- Cheng, D., Huang, S., Zhu, X., Dai, B., Zhao, W. X., Zhang, Z., and Wei, F. Reasoning with exploration: An entropy perspective, 2025a. URL <https://arxiv.org/abs/2506.14758>.
- Cheng, Z., Hao, S., Liu, T., Zhou, F., Xie, Y., Yao, F., Bian, Y., Zhuang, Y., Dey, N., Zha, Y., Gu, Y., Zhou, K., Wang, Y., Li, Y., Fan, R., She, J., Gao, C., Saparov, A., Li, H., Killian, T. W., Yurochkin, M., Liu, Z., Xing, E. P., and Hu, Z. Revisiting reinforcement learning for llm reasoning from a cross-domain perspective, 2025b. URL <https://arxiv.org/abs/2506.14965>.
- Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems, 2021. URL <https://arxiv.org/abs/2110.14168>.
- Codevilla, F., Müller, M., López, A., Koltun, V., and Dosovitskiy, A. End-to-end driving via conditional imitation learning, 2018. URL <https://arxiv.org/abs/1710.02410>.
- Dai, R., Song, L., Liu, H., Liang, Z., Yu, D., Mi, H., Tu, Z., Liu, R., Zheng, T., Zhu, H., and Yu, D. Cde: Curiosity-driven exploration for efficient reinforcement learning in large language models, 2025. URL <https://arxiv.org/abs/2509.09675>.
- Dang, X., Baek, C., Kolter, J. Z., and Raghunathan, A. Assessing diversity collapse in reasoning. In *Scaling Self-Improving Foundation Models without Human Supervision*, 2025. URL <https://openreview.net/forum?id=AMiKsHLjQh>.
- Davis, D. and Recht, B. What is the objective of reasoning with reinforcement learning?, 2025. URL <https://arxiv.org/abs/2510.13651>.
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.
- Dong, X., Cheng, J., and Zhang, X. S. Maximum entropy reinforcement learning with diffusion policy, 2025. URL <https://arxiv.org/abs/2502.11612>.
- Editya, A. S., Alamin, M. M., Pramana, A. L., and Kurniati, N. Fraud classification in online payments using supervised machine learning algorithms. *Green Intelligent Systems and Applications*, 5(1):40–50, Mar. 2025. doi: 10.53623/gisa.v5i1.552. URL <https://tecnicoscientifica.com/journal/gisa/article/view/552>.
- Eysenbach, B. and Levine, S. Maximum entropy rl (provably) solves some robust rl problems, 2022. URL <https://arxiv.org/abs/2103.06257>.

- 550 Eysenbach, B., Gupta, A., Ibarz, J., and Levine, S. Diversity
551 is all you need: Learning skills without a reward function.
552 *arXiv preprint arXiv:1802.06070*, 2018.
- 553
554 Fortunato, M., Azar, M. G., Piot, B., Menick, J., Osband, I.,
555 Graves, A., Mnih, V., Munos, R., Hassabis, D., Pietquin,
556 O., Blundell, C., and Legg, S. Noisy networks for ex-
557 ploration, 2017. URL [https://arxiv.org/abs/
558 1706.10295](https://arxiv.org/abs/1706.10295).
- 559
560 Gai, J., Zeng, G., Zhang, H., and Raghunathan, A. Differ-
561 ential smoothing mitigates sharpening and improves llm
562 reasoning, 2025. URL [https://arxiv.org/abs/
563 2511.19942](https://arxiv.org/abs/2511.19942).
- 564
565 Gandhi, K., Chakravarthy, A., Singh, A., Lile, N., and
566 Goodman, N. D. Cognitive behaviors that enable self-
567 improving reasoners, or, four habits of highly effec-
568 tive stars, 2025. URL [https://arxiv.org/abs/
569 2503.01307](https://arxiv.org/abs/2503.01307).
- 570
571 Ged, F. and Veiga, M. H. Matryoshka policy gradient for
572 entropy-regularized rl: Convergence and global optimal-
573 ity, 2024. URL [https://arxiv.org/abs/2303.
574 12785](https://arxiv.org/abs/2303.12785).
- 575
576 Gneiting, T. and Raftery, A. E. Strictly proper scoring
577 rules, prediction, and estimation. *Journal of the American
578 Statistical Association*, 102(477):359–378, 2007. doi:
579 [10.1198/016214506000001437](https://doi.org/10.1198/016214506000001437). URL [https://doi.
580 org/10.1198/016214506000001437](https://doi.org/10.1198/016214506000001437).
- 581
582 Good, I. J. Rational decisions. *Journal of the Royal Statisti-
583 cal Society: Series B (Methodological)*, 14(1):107–114,
584 1992.
- 585
586 Gulcehre, C., Paine, T. L., Srinivasan, S., Konyushkova, K.,
587 Weerts, L., Sharma, A., Siddhant, A., Ahern, A., Wang,
588 M., Gu, C., Macherey, W., Doucet, A., Firat, O., and
589 de Freitas, N. Reinforced self-training (rest) for language
590 modeling, 2023. URL [https://arxiv.org/abs/
591 2308.08998](https://arxiv.org/abs/2308.08998).
- 592
593 Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On
594 calibration of modern neural networks, 2017. URL
595 <https://arxiv.org/abs/1706.04599>.
- 596
597 Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R.,
598 Zhu, Q., Ma, S., Wang, P., Bi, X., et al. Deepseek-rl: In-
599 centivizing reasoning capability in llms via reinforcement
600 learning. *arXiv preprint arXiv:2501.12948*, 2025.
- 601
602 GX-Chen, A., Prakash, J., Guo, J., Fergus, R., and Ran-
603 ganath, R. Kl-regularized reinforcement learning is de-
604 signed to mode collapse, 2025. URL [https://arxiv.
org/abs/2510.20817](https://arxiv.org/abs/2510.20817).
- Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-
critic: Off-policy maximum entropy deep reinforcement
learning with a stochastic actor, 2018. URL [https:
//arxiv.org/abs/1801.01290](https://arxiv.org/abs/1801.01290).
- Hao, Z., Wang, H., Liu, H., Luo, J., Yu, J., Dong, H., Lin,
Q., Wang, C., and Chen, J. Rethinking entropy interven-
tions in rlvr: An entropy change perspective, 2025. URL
<https://arxiv.org/abs/2510.10150>.
- He, H. and Lab, T. M. Defeating nondeterminism in llm
inference, September 2025.
- Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart,
S., Tang, E., Song, D., and Steinhardt, J. Measuring math-
ematical problem solving with the math dataset, 2021.
URL <https://arxiv.org/abs/2103.03874>.
- Hester, T., Vecerik, M., Pietquin, O., Lanctot, M., Schaul,
T., Piot, B., Horgan, D., Quan, J., Sendonaris, A., Dulac-
Arnold, G., Osband, I., Agapiou, J., Leibo, J. Z., and
Grusl, A. Deep q-learning from demonstrations, 2017.
URL <https://arxiv.org/abs/1704.03732>.
- Ho, J. and Ermon, S. Generative adversarial imitation learn-
ing, 2016. URL [https://arxiv.org/abs/1606.
03476](https://arxiv.org/abs/1606.03476).
- Houthoofd, R., Chen, X., Chen, X., Duan, Y., Schulman,
J., De Turck, F., and Abbeel, P. Vime: Variational
information maximizing exploration. In Lee, D.,
Sugiyama, M., Luxburg, U., Guyon, I., and Garnett,
R. (eds.), *Advances in Neural Information Process-
ing Systems*, volume 29. Curran Associates, Inc.,
2016. URL [https://proceedings.neurips.
cc/paper_files/paper/2016/file/
abd815286ba1007abfbb8415b83ae2cf-Paper.
pdf](https://proceedings.neurips.cc/paper_files/paper/2016/file/abd815286ba1007abfbb8415b83ae2cf-Paper.pdf).
- Hu, J., Liu, M., Lu, X., Wu, F., Harchaoui, Z., Diao, S.,
Choi, Y., Molchanov, P., Yang, J., Kautz, J., and Dong, Y.
Brorl: Scaling reinforcement learning via broadened ex-
ploration, 2025. URL [https://arxiv.org/abs/
2510.01180](https://arxiv.org/abs/2510.01180).
- Huang, W., Zhang, Q., Fang, Y., Liang, J., Rong, X., Yao,
H., Wan, G., Liang, K., He, W., Li, M., Rutkowski, L.,
Ye, M., Du, B., and Tao, D. Mapo: Mixed advantage pol-
icy optimization, 2025. URL [https://arxiv.org/
abs/2509.18849](https://arxiv.org/abs/2509.18849).
- Ishfaq, H., Lan, Q., Xu, P., Mahmood, A. R., Precup,
D., Anandkumar, A., and Azizzadenesheli, K. Prov-
able and practical: Efficient exploration in reinforce-
ment learning via langevin monte carlo, 2024a. URL
<https://arxiv.org/abs/2305.18246>.

- 605 Ishfaq, H., Tan, Y., Yang, Y., Lan, Q., Lu, J., Mahmood,
606 A. R., Precup, D., and Xu, P. More efficient randomized
607 exploration for reinforcement learning via approximate
608 sampling, 2024b. URL [https://arxiv.org/abs/
609 2406.12241](https://arxiv.org/abs/2406.12241).
- 610
611 Ishfaq, H., Wang, G., Islam, S. N., and Precup, D. Langevin
612 soft actor-critic: Efficient exploration through uncertainty-
613 driven critic learning, 2025. URL [https://arxiv.
614 org/abs/2501.17827](https://arxiv.org/abs/2501.17827).
- 615
616 Ito, K. and Kashima, K. Risk-sensitive control as infer-
617 ence with rényi divergence. In *The Thirty-eighth Annual
618 Conference on Neural Information Processing Systems*,
619 2024. URL [https://openreview.net/forum?
620 id=LUIXdWn6Z5](https://openreview.net/forum?id=LUIXdWn6Z5).
- 621
622 Jain, T., Garg, P., Chalil, N., Sinha, A., Verma, V. K.,
623 and Gupta, R. Sms spam classification using machine
624 learning techniques. In *2022 12th International Confer-
625 ence on Cloud Computing, Data Science & Engineer-
626 ing (Confluence)*, pp. 273–279, 2022. doi: 10.1109/
627 Confluence52989.2022.9734128.
- 628
629 Jamil, M., Mihajloska Trpcheska, H., Popovska-Mitrovikj,
630 A., Dimitrova, V., and Creutzburg, R. Advancing image
631 spam detection: Evaluating machine learning models
632 through comparative analysis. *Applied Sciences*, 15(11),
633 2025. ISSN 2076-3417. doi: 10.3390/app15116158.
634 URL [https://www.mdpi.com/2076-3417/15/
635 11/6158](https://www.mdpi.com/2076-3417/15/11/6158).
- 636
637 Khatri, D., Madaan, L., Tiwari, R., Bansal, R., Duvvuri,
638 S. S., Zaheer, M., Dhillon, I. S., Brandfonbrener, D., and
639 Agarwal, R. The art of scaling reinforcement learning
640 compute for llms, 2025. URL [https://arxiv.org/
641 abs/2510.13786](https://arxiv.org/abs/2510.13786).
- 642
643 Kingma, D. P. and Ba, J. Adam: A method for stochastic op-
644 timization, 2017. URL [https://arxiv.org/abs/
645 1412.6980](https://arxiv.org/abs/1412.6980).
- 646
647 Kornblith, S., Lee, H., Chen, T., and Norouzi, M. Demystify-
648 ing loss functions for classification, 2021. URL [https:
649 //openreview.net/forum?id=jNTEyScgSw8](https://openreview.net/forum?id=jNTEyScgSw8).
- 650
651 Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet
652 classification with deep convolutional neural networks.
653 *Advances in neural information processing systems*, 25,
654 2012.
- 655
656 Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu,
657 C. H., Gonzalez, J. E., Zhang, H., and Stoica, I. Efficient
658 memory management for large language model serving
659 with pagedattention. In *Proceedings of the ACM SIGOPS
29th Symposium on Operating Systems Principles*, 2023.
- Lambert, N., Morrison, J., Pyatkin, V., Huang, S., Ivison, H.,
Brahman, F., Miranda, L. J. V., Liu, A., Dziri, N., Lyu, S.,
Gu, Y., Malik, S., Graf, V., Hwang, J. D., Yang, J., Bras,
R. L., Taffjord, O., Wilhelm, C., Soldaini, L., Smith, N. A.,
Wang, Y., Dasigi, P., and Hajishirzi, H. Tulu 3: Pushing
frontiers in open language model post-training, 2025.
URL <https://arxiv.org/abs/2411.15124>.
- Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-
based learning applied to document recognition. *Pro-
ceedings of the IEEE*, 86(11):2278–2324, 1998. doi:
10.1109/5.726791.
- Lehmann, E. and Casella, G. *Theory of Point Estima-
tion*. Springer Texts in Statistics. Springer New York,
2006. ISBN 9780387227283. URL [https://books.
google.com/books?id=4f24CgAAQBAJ](https://books.google.com/books?id=4f24CgAAQBAJ).
- Levine, S. Reinforcement learning and control as prob-
abilistic inference: Tutorial and review, 2018. URL
<https://arxiv.org/abs/1805.00909>.
- Levine, S. and Koltun, V. Guided policy search. In Das-
gupta, S. and McAllester, D. (eds.), *Proceedings of the
30th International Conference on Machine Learning*, vol-
ume 28 of *Proceedings of Machine Learning Research*,
pp. 1–9, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.
URL [https://proceedings.mlr.press/v28/
levine13.html](https://proceedings.mlr.press/v28/levine13.html).
- Lewkowycz, A., Andreassen, A., Dohan, D., Dyer, E.,
Michalewski, H., Ramasesh, V., Slone, A., Anil, C.,
Schlag, I., Gutman-Solo, T., Wu, Y., Neyshabur, B.,
Gur-Ari, G., and Misra, V. Solving quantitative reason-
ing problems with language models, 2022. URL
<https://arxiv.org/abs/2206.14858>.
- Li, C., Liu, K., and Liu, S. A survey of loss functions
in deep learning. *Mathematics*, 13(15), 2025. ISSN
2227-7390. doi: 10.3390/math13152417. URL [https:
//www.mdpi.com/2227-7390/13/15/2417](https://www.mdpi.com/2227-7390/13/15/2417).
- Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker,
B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and
Cobbe, K. Let’s verify step by step, 2023. URL [https:
//arxiv.org/abs/2305.20050](https://arxiv.org/abs/2305.20050).
- Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez,
T., Tassa, Y., Silver, D., and Wierstra, D. Continuous
control with deep reinforcement learning, 2015. URL
<https://arxiv.org/abs/1509.02971>.
- Liu, M., Diao, S., Lu, X., Hu, J., Dong, X., Choi, Y., Kautz,
J., and Dong, Y. Prorl: Prolonged reinforcement learning
expands reasoning boundaries in large language mod-
els, 2025a. URL [https://arxiv.org/abs/2505.
24864](https://arxiv.org/abs/2505.24864).

- 660 Liu, Z., Chen, C., Li, W., Qi, P., Pang, T., Du, C., Lee,
661 W. S., and Lin, M. Understanding r1-zero-like training:
662 A critical perspective, 2025b. URL <https://arxiv.org/abs/2503.20783>.
- 664 Loshchilov, I. and Hutter, F. Sgdr: Stochastic gradient
665 descent with warm restarts, 2017. URL <https://arxiv.org/abs/1608.03983>.
- 668 Loshchilov, I. and Hutter, F. Decoupled weight decay regu-
669 larization, 2019. URL <https://arxiv.org/abs/1711.05101>.
- 672 Mao, A., Mohri, M., and Zhong, Y. Cross-entropy loss
673 functions: Theoretical analysis and applications, 2023.
674 URL <https://arxiv.org/abs/2304.07288>.
- 676 Mensch, A. and Blondel, M. Differentiable dynamic pro-
677 gramming for structured prediction and attention, 2018.
678 URL <https://arxiv.org/abs/1802.03676>.
- 679 Millidge, B., Tschantz, A., Seth, A. K., and Buckley, C. L.
680 On the relationship between active inference and control
681 as inference, 2020. URL <https://arxiv.org/abs/2006.12964>.
- 684 MiniMax, :, Chen, A., Li, A., Gong, B., Jiang, B., Fei, B.,
685 Yang, B., Shan, B., Yu, C., Wang, C., Zhu, C., Xiao, C.,
686 Du, C., Zhang, C., Qiao, C., Zhang, C., Du, C., Guo, C.,
687 Chen, D., Ding, D., Sun, D., Li, D., Jiao, E., Zhou, H.,
688 Zhang, H., Ding, H., Sun, H., Feng, H., Cai, H., Zhu,
689 H., Sun, J., Zhuang, J., Cai, J., Song, J., Zhu, J., Li, J.,
690 Tian, J., Liu, J., Xu, J., Yan, J., Liu, J., He, J., Feng, K.,
691 Yang, K., Xiao, K., Han, L., Wang, L., Yu, L., Feng, L.,
692 Li, L., Zheng, L., Du, L., Yang, L., Zeng, L., Yu, M., Tao,
693 M., Chi, M., Zhang, M., Lin, M., Hu, N., Di, N., Gao,
694 P., Li, P., Zhao, P., Ren, Q., Xu, Q., Li, Q., Wang, Q.,
695 Tian, R., Leng, R., Chen, S., Chen, S., Shi, S., Weng, S.,
696 Guan, S., Yu, S., Li, S., Zhu, S., Li, T., Cai, T., Liang, T.,
697 Cheng, W., Kong, W., Li, W., Chen, X., Song, X., Luo,
698 X., Su, X., Li, X., Han, X., Hou, X., Lu, X., Zou, X.,
699 Shen, X., Gong, Y., Ma, Y., Wang, Y., Shi, Y., Zhong,
700 Y., Duan, Y., Fu, Y., Hu, Y., Gao, Y., Fan, Y., Yang, Y.,
701 Li, Y., Hu, Y., Huang, Y., Li, Y., Xu, Y., Mao, Y., Shi,
702 Y., Wenren, Y., Li, Z., Li, Z., Tian, Z., Zhu, Z., Fan, Z.,
703 Wu, Z., Xu, Z., Yu, Z., Lyu, Z., Jiang, Z., Gao, Z., Wu,
704 Z., Song, Z., and Sun, Z. Minimax-m1: Scaling test-time
705 compute efficiently with lightning attention, 2025. URL
706 <https://arxiv.org/abs/2506.13585>.
- 707
708 Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A.,
709 Antonoglou, I., Wierstra, D., and Riedmiller, M. Play-
710 ing atari with deep reinforcement learning, 2013. URL
711 <https://arxiv.org/abs/1312.5602>.
- 712
713 Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Ve-
714 nness, J., Bellemare, M. G., Graves, A., Riedmiller, M.,
Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie,
C., Sadik, A., Antonoglou, I., King, H., Kumaran, D.,
Wierstra, D., Legg, S., and Hassabis, D. Human-level
control through deep reinforcement learning. *Nature*, 518
(7540):529–533, February 2015. ISSN 00280836. URL
<http://dx.doi.org/10.1038/nature14236>.
- Motwani, S. R., Ivanova, A., Cai, Z., Torr, P., Islam, R.,
Shah, S., de Witt, C. S., and London, C. h1: Bootstrap-
ping llms to reason over longer horizons via reinforce-
ment learning, 2025. URL <https://arxiv.org/abs/2510.07312>.
- Nair, A., McGrew, B., Andrychowicz, M., Zaremba, W.,
and Abbeel, P. Overcoming exploration in reinforcement
learning with demonstrations, 2018. URL <https://arxiv.org/abs/1709.10089>.
- Ng, A. and Jordan, M. On discriminative vs. genera-
tive classifiers: A comparison of logistic regression
and naive bayes. In Dietterich, T., Becker, S., and
Ghahramani, Z. (eds.), *Advances in Neural Informa-
tion Processing Systems*, volume 14. MIT Press,
2001. URL https://proceedings.neurips.cc/paper_files/paper/2001/file/7b7a53e239400a13bd6be6c91c4f6c4e-Paper.pdf.
- Nguyen, M. N., Baker, A., Neo, C., Roush, A., Kirsch,
A., and Shwartz-Ziv, R. Turning up the heat: Min-p
sampling for creative and coherent llm outputs, 2025a.
URL <https://arxiv.org/abs/2407.01082>.
- Nguyen, P. M., La, C. D., Nguyen, D. M. H., Chawla, N. V.,
Nguyen, B. T., and Doan, K. D. The reasoning bound-
ary paradox: How reinforcement learning constrains lan-
guage models, 2025b. URL <https://arxiv.org/abs/2510.02230>.
- Niculescu-Mizil, A. and Caruana, R. Predicting good prob-
abilities with supervised learning. In *Proceedings of
the 22nd International Conference on Machine Learning*,
ICML '05, pp. 625–632, New York, NY, USA, 2005. As-
sociation for Computing Machinery. ISBN 1595931805.
doi: 10.1145/1102351.1102430. URL <https://doi.org/10.1145/1102351.1102430>.
- O’Donoghue, B., Osband, I., and Ionescu, C. Making
sense of reinforcement learning and probabilistic in-
ference, 2020. URL <https://arxiv.org/abs/2001.00805>.
- Olmo, T., :, Ettinger, A., Bertsch, A., Kuehl, B., Graham,
D., Heineman, D., Groeneveld, D., Brahman, F., et al.
Olmo 3, 2025. URL <https://arxiv.org/abs/2512.13961>.

- 715 OpenAI, Jaech, A., Kalai, A., Lerer, A., Richardson, A., El-
716 Kishky, A., Low, A., Helyar, A., Madry, A., Beutel, A.,
717 Carney, A., et al. Openai o1 system card. *arXiv preprint*
718 *arXiv:2412.16720*, 2024.
- 719 Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright,
720 C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K.,
721 Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L.,
722 Simens, M., Askell, A., Welinder, P., Christiano, P., Leike,
723 J., and Lowe, R. Training language models to follow
724 instructions with human feedback, 2022. URL <https://arxiv.org/abs/2203.02155>.
- 725
726
727 Pang, R. Y., Yuan, W., Cho, K., He, H., Sukhbaatar, S.,
728 and Weston, J. Iterative reasoning preference optimiza-
729 tion, 2024. URL [https://arxiv.org/abs/2404.](https://arxiv.org/abs/2404.19733)
730 [19733](https://arxiv.org/abs/2404.19733).
- 731 Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T.
732 Curiosity-driven exploration by self-supervised predic-
733 tion. In *International conference on machine learning*,
734 pp. 2778–2787. PMLR, 2017.
- 735 Pathak, D., Gandhi, D., and Gupta, A. Self-supervised
736 exploration via disagreement. In *International conference*
737 *on machine learning*, pp. 5062–5071. PMLR, 2019.
- 738
739 Pomerleau, D. A. Alvin: an autonomous land vehicle in a
740 neural network. In *Proceedings of the 2nd International*
741 *Conference on Neural Information Processing Systems*,
742 NIPS’88, pp. 305–313, Cambridge, MA, USA, 1988.
743 MIT Press.
- 744 Prim, R. C. Shortest connection networks and some gen-
745 eralizations. *The Bell System Technical Journal*, 36(6):
746 1389–1401, 1957.
- 747
748 Qu, Y., Setlur, A., Smith, V., Salakhutdinov, R., and Kumar,
749 A. How to explore to scale rl training of llms on hard prob-
750 lems? [urlhttps://blog.ml.cmu.edu/2025/11/26/how-to-](https://blog.ml.cmu.edu/2025/11/26/how-to-explore-to-scale-rl-training-of-llms-on-hard-problems)
751 [explore-to-scale-rl-training-of-llms-on-hard-problems](https://blog.ml.cmu.edu/2025/11/26/how-to-explore-to-scale-rl-training-of-llms-on-hard-problems),
752 2025a. CMU MLD Blog.
- 753
754 Qu, Y., Singh, A., Lee, Y., Setlur, A., Salakhutdinov, R.,
755 Finn, C., and Kumar, A. Rlad: Training llms to discover
756 abstractions for solving reasoning problems, 2025b. URL
757 <https://arxiv.org/abs/2510.02263>.
- 758
759 Qwen, :, Yang, A., Yang, B., Zhang, B., Hui, B., Zheng,
760 B., Yu, B., Li, C., et al. Qwen2.5 technical report, 2025.
761 URL <https://arxiv.org/abs/2412.15115>.
- 762
763 Radford, A., Narasimhan, K., Salimans, T., and Sutskever,
764 I. Improving language understanding by generative pre-
765 training. 2018.
- 766 Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and
767 Sutskever, I. Language models are unsupervised multitask
768 learners. 2019.
- 769 Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning,
C. D., and Finn, C. Direct preference optimization: Your
language model is secretly a reward model, 2023. URL
<https://arxiv.org/abs/2305.18290>.
- Rawlik, K., Toussaint, M., and Vijayakumar, S. On stochas-
tic optimal control and reinforcement learning by approx-
imate inference. *Proceedings of Robotics: Science and*
Systems VIII, 2012.
- Rosenblatt, F. The perceptron: a probabilistic
model for information storage and organization in
the brain. *Psychological review*, 65 6:386–408,
1958. URL [https://api.semanticscholar.](https://api.semanticscholar.org/CorpusID:12781225)
[org/CorpusID:12781225](https://api.semanticscholar.org/CorpusID:12781225).
- Ross, S. and Bagnell, D. Efficient reductions for imita-
tion learning. In Teh, Y. W. and Titterton, M. (eds.),
Proceedings of the Thirteenth International Conference
on Artificial Intelligence and Statistics, volume 9 of
Proceedings of Machine Learning Research, pp. 661–
668, Chia Laguna Resort, Sardinia, Italy, 13–15 May
2010. PMLR. URL [https://proceedings.mlr.](https://proceedings.mlr.press/v9/ross10a.html)
[press/v9/ross10a.html](https://proceedings.mlr.press/v9/ross10a.html).
- Ross, S., Gordon, G. J., and Bagnell, J. A. A reduction of
imitation learning and structured prediction to no-regret
online learning, 2010. URL [https://arxiv.org/](https://arxiv.org/abs/1011.0686)
[abs/1011.0686](https://arxiv.org/abs/1011.0686).
- Rumelhart, D. E., Hinton, G. E., and Williams, R. J. Learn-
ing representations by back-propagating errors. *Nature*,
323(6088):533–536, 1986. doi: 10.1038/323533a0.
- Savage, L. J. Elicitation of personal probabilities
and expectations. *Journal of the American Statisti-*
cal Association, 66(336):783–801, 1971. ISSN
01621459, 1537274X. URL [http://www.jstor.](http://www.jstor.org/stable/2284229)
[org/stable/2284229](http://www.jstor.org/stable/2284229).
- Schaul, T., Borsa, D., Modayil, J., and Pascanu, R. Ray
interference: a source of plateaus in deep reinforcement
learning, 2019. URL [https://arxiv.org/abs/](https://arxiv.org/abs/1904.11455)
[1904.11455](https://arxiv.org/abs/1904.11455).
- Schmidhuber, J. Curious model-building control systems. In
Proc. international joint conference on neural networks,
pp. 1458–1463, 1991.
- Schmidhuber, J. Gödel machines: Fully self-referential
optimal universal self-improvers. In *Artificial general*
intelligence, pp. 199–226. Springer, 2007.
- Schulman, J., Levine, S., Moritz, P., Jordan, M. I., and
Abbeel, P. Trust region policy optimization, 2015. URL
<https://arxiv.org/abs/1502.05477>.

- Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms, 2017. URL <https://arxiv.org/abs/1707.06347>.
- Setlur, A., Yang, M. Y. R., Snell, C., Greer, J., Wu, I., Smith, V., Simchowitz, M., and Kumar, A. e3: Learning to explore enables extrapolation of test-time compute for llms, 2025. URL <https://arxiv.org/abs/2506.09026>.
- Shafayat, S., Tajwar, F., Salakhutdinov, R., Schneider, J., and Zanette, A. Can large reasoning models self-train?, 2025. URL <https://arxiv.org/abs/2505.21444>.
- Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y. K., Wu, Y., and Guo, D. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL <https://arxiv.org/abs/2402.03300>.
- Sharma, A., Gu, S., Levine, S., Kumar, V., and Hausman, K. Dynamics-aware unsupervised discovery of skills. *arXiv preprint arXiv:1907.01657*, 2019.
- Sheng, G., Zhang, C., Ye, Z., Wu, X., Zhang, W., Zhang, R., Peng, Y., Lin, H., and Wu, C. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv:2409.19256*, 2024.
- Smith, N. A. *Linguistic Structure Prediction*. Morgan & Claypool Publishers, 1st edition, 2011. ISBN 1608454053.
- Song, Y., Kempe, J., and Munos, R. Outcome-based exploration for llm reasoning, 2025. URL <https://arxiv.org/abs/2509.06941>.
- Stojanovski, Z., Stanley, O., Sharratt, J., Jones, R., Adefioye, A., Kaddour, J., and Köpf, A. Reasoning gym: Reasoning environments for reinforcement learning with verifiable rewards, 2025. URL <https://arxiv.org/abs/2505.24760>.
- Su, J., Lu, Y., Pan, S., Murtadha, A., Wen, B., and Liu, Y. Roformer: Enhanced transformer with rotary position embedding, 2021. URL <https://arxiv.org/abs/2104.09864>.
- Sutton, R. S. Learning to predict by the methods of temporal differences. *Machine Learning*, 1988.
- Sutton, R. S., Barto, A. G., et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.
- Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.
- Tajwar, F., Singh, A., Sharma, A., Rafailov, R., Schneider, J., Xie, T., Ermon, S., Finn, C., and Kumar, A. Preference fine-tuning of llms should leverage suboptimal, on-policy data, 2024. URL <https://arxiv.org/abs/2404.14367>.
- Tajwar, F., Jiang, Y., Thankaraj, A., Rahman, S. S., Kolter, J. Z., Schneider, J., and Salakhutdinov, R. Training a generally curious agent, 2025. URL <https://arxiv.org/abs/2502.17543>.
- Tang, Y., Zheng, K., Synnaeve, G., and Munos, R. Optimizing language models for inference time objectives using reinforcement learning, 2025. URL <https://arxiv.org/abs/2503.19595>.
- Tarbouriech, J., Lattimore, T., and O’Donoghue, B. Probabilistic inference in reinforcement learning done right, 2023. URL <https://arxiv.org/abs/2311.13294>.
- Team, K., Du, A., Gao, B., Xing, B., Jiang, C., Chen, C., Li, C., Xiao, C., Du, C., et al. Kimi k1.5: Scaling reinforcement learning with llms, 2025. URL <https://arxiv.org/abs/2501.12599>.
- Terven, J., Cordova-Esparza, D.-M., Romero-González, J.-A., Ramírez-Pedraza, A., and Chávez-Urbiola, E. A. A comprehensive survey of loss functions and metrics in deep learning. *Artificial Intelligence Review*, 58 (7), April 2025. ISSN 1573-7462. doi: 10.1007/s10462-025-11198-7. URL <http://dx.doi.org/10.1007/s10462-025-11198-7>.
- Thompson, W. R. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3-4):285–294, 1933.
- Thrun, S., Burgard, W., and Fox, D. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005. ISBN 0262201623.
- Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, 2012. doi: 10.1109/IROS.2012.6386109.
- Tuyls, J., Foster, D. J., Krishnamurthy, A., and Ash, J. T. Representation-based exploration for language models: From test-time to post-training, 2025. URL <https://arxiv.org/abs/2510.11686>.
- Vaart, A. W. v. d. *Asymptotic Statistics*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 1998.

- 825 Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones,
826 L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. At-
827 tention is all you need. *Advances in neural information*
828 *processing systems*, 30, 2017.
- 829
- 830 Vendrow, J., Vendrow, E., Beery, S., and Madry, A. Do large
831 language model benchmarks test reliability?, 2025. URL <https://arxiv.org/abs/2502.03461>.
- 832
- 833 Waghmare, K. and Ziegel, J. Proper scoring rules for es-
834 timation and forecast evaluation, 2025. URL <https://arxiv.org/abs/2504.01781>.
- 835
- 836
- 837 Walder, C. and Karkhanis, D. Pass@k policy optimization:
838 Solving harder reinforcement learning problems, 2025.
839 URL <https://arxiv.org/abs/2505.15201>.
- 840
- 841 Wang, Q., Ma, Y., Zhao, K., and jie Tian, Y. A compre-
842 hensive survey of loss functions in machine learning. *Annals*
843 *of Data Science*, 9:187 – 212, 2020.
- 844
- 845 Wang, S., Yu, L., Gao, C., Zheng, C., Liu, S., Lu, R., Dang,
846 K., Chen, X.-H., Yang, J., Zhang, Z., Liu, Y., Yang, A.,
847 Zhao, A., Yue, Y., Song, S., Yu, B., Huang, G., and Lin,
848 J. Beyond the 80/20 rule: High-entropy minority tokens
849 drive effective reinforcement learning for LLM reason-
850 ing. In *The Thirty-ninth Annual Conference on Neural*
851 *Information Processing Systems*, 2025. URL <https://openreview.net/forum?id=yfcpdY4gMP>.
- 852
- 853
- 854 Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang,
855 S., Chowdhery, A., and Zhou, D. Self-consistency im-
856 proves chain of thought reasoning in language mod-
857 els, 2023a. URL <https://arxiv.org/abs/2203.11171>.
- 858
- 859
- 860 Wang, Y., Ivison, H., Dasigi, P., Hessel, J., Khot, T., Chandu,
861 K. R., Wadden, D., MacMillan, K., Smith, N. A., Beltagy,
862 I., and Hajishirzi, H. How far can camels go? exploring
863 the state of instruction tuning on open resources, 2023b.
864 URL <https://arxiv.org/abs/2306.04751>.
- 865
- 866 Watkins, C. J. C. H. and Dayan, P. Q-learning. *Ma-*
867 *chine Learning*, 8(3–4):279–292, 1992. doi: 10.1007/
868 BF00992698. URL [https://doi.org/10.1007/
869 BF00992698](https://doi.org/10.1007/BF00992698).
- 870
- 871 Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter,
872 B., Xia, F., Chi, E., Le, Q., and Zhou, D. Chain-of-
873 thought prompting elicits reasoning in large language
874 models, 2023. URL [https://arxiv.org/abs/
875 2201.11903](https://arxiv.org/abs/2201.11903).
- 876
- 877 Williams, R. J. Simple statistical gradient-following algo-
878 rithms for connectionist reinforcement learning. *Machine*
879 *learning*, 8:229–256, 1992.
- Wu, F., Xuan, W., Lu, X., Liu, M., Dong, Y., Harchaoui, Z.,
and Choi, Y. The invisible leash: Why rlvr may or may
not escape its origin, 2026. URL [https://arxiv.
org/abs/2507.14843](https://arxiv.org/abs/2507.14843).
- Xiong, W., Yao, J., Xu, Y., Pang, B., Wang, L., Sahoo,
D., Li, J., Jiang, N., Zhang, T., Xiong, C., and Dong, H.
A minimalist approach to llm reasoning: from rejection
sampling to reinforce, 2025a. URL [https://arxiv.
org/abs/2504.11343](https://arxiv.org/abs/2504.11343).
- Xiong, W., Ye, C., Liao, B., Dong, H., Xu, X., Monz,
C., Bian, J., Jiang, N., and Zhang, T. Reinforce-ada:
An adaptive sampling framework under non-linear rl ob-
jectives, 2025b. URL [https://arxiv.org/abs/
2510.04996](https://arxiv.org/abs/2510.04996).
- Xu, S., Fu, W., Gao, J., Ye, W., Liu, W., Mei, Z., Wang,
G., Yu, C., and Wu, Y. Is dpo superior to ppo for llm
alignment? a comprehensive study, 2024. URL [https://arxiv.org/abs/
2404.10719](https://arxiv.org/abs/2404.10719).
- Yang, A., Yang, B., Hui, B., Zheng, B., Yu, B., Zhou, C.,
Li, C., Li, C., Liu, D., Huang, F., Dong, G., Wei, H.,
Lin, H., Tang, J., Wang, J., Yang, J., Tu, J., Zhang, J.,
Ma, J., Yang, J., Xu, J., Zhou, J., Bai, J., He, J., Lin,
J., Dang, K., Lu, K., Chen, K., Yang, K., Li, M., Xue,
M., Ni, N., Zhang, P., Wang, P., Peng, R., Men, R., Gao,
R., Lin, R., Wang, S., Bai, S., Tan, S., Zhu, T., Li, T.,
Liu, T., Ge, W., Deng, X., Zhou, X., Ren, X., Zhang,
X., Wei, X., Ren, X., Liu, X., Fan, Y., Yao, Y., Zhang,
Y., Wan, Y., Chu, Y., Liu, Y., Cui, Z., Zhang, Z., Guo,
Z., and Fan, Z. Qwen2 technical report, 2024a. URL
<https://arxiv.org/abs/2407.10671>.
- Yang, A., Zhang, B., Hui, B., Gao, B., Yu, B., Li, C., Liu,
D., Tu, J., et al. Qwen2.5-math technical report: Toward
mathematical expert model via self-improvement, 2024b.
URL <https://arxiv.org/abs/2409.12122>.
- Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng,
B., Yu, B., Gao, C., Huang, C., Lv, C., Zheng, C., Liu,
D., Zhou, F., Huang, F., Hu, F., Ge, H., Wei, H., Lin,
H., Tang, J., Yang, J., Tu, J., Zhang, J., Yang, J., Yang,
J., Zhou, J., Zhou, J., Lin, J., Dang, K., Bao, K., Yang,
K., Yu, L., Deng, L., Li, M., Xue, M., Li, M., Zhang,
P., Wang, P., Zhu, Q., Men, R., Gao, R., Liu, S., Luo,
S., Li, T., Tang, T., Yin, W., Ren, X., Wang, X., Zhang,
X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Zhang, Y., Wan,
Y., Liu, Y., Wang, Z., Cui, Z., Zhang, Z., Zhou, Z., and
Qiu, Z. Qwen3 technical report, 2025. URL [https://arxiv.org/abs/
2505.09388](https://arxiv.org/abs/2505.09388).
- Yang, K., Tao, J., Lyu, J., and Li, X. Exploration and
anti-exploration with distributional random network dis-
tillation, 2024c. URL [https://arxiv.org/abs/
2401.09750](https://arxiv.org/abs/2401.09750).

- 880 Yao, J., Hao, Y., Zhang, H., Dong, H., Xiong, W., Jiang, N.,
881 and Zhang, T. Optimizing chain-of-thought reasoners via
882 gradient variance minimization in rejection sampling and
883 rl, 2025. URL [https://arxiv.org/abs/2505.](https://arxiv.org/abs/2505.02391)
884 02391.
- 885
886 Yu, Q., Zhang, Z., Zhu, R., Yuan, Y., Zuo, X., Yue, Y.,
887 Dai, W., Fan, T., Liu, G., Liu, L., Liu, X., Lin, H., Lin,
888 Z., Ma, B., Sheng, G., Tong, Y., Zhang, C., Zhang, M.,
889 Zhang, W., Zhu, H., Zhu, J., Chen, J., Chen, J., Wang, C.,
890 Yu, H., Song, Y., Wei, X., Zhou, H., Liu, J., Ma, W.-Y.,
891 Zhang, Y.-Q., Yan, L., Qiao, M., Wu, Y., and Wang, M.
892 Dapo: An open-source llm reinforcement learning system
893 at scale, 2025a. URL [https://arxiv.org/abs/](https://arxiv.org/abs/2503.14476)
894 2503.14476.
- 895
896 Yu, Z., Su, W., Tao, L., Wang, H., Singh, A., Yu, H., Wang,
897 J., Gao, H., Yuan, W., Weston, J., Yu, P., and Xu, J.
898 Restrain: From spurious votes to signals – self-driven rl
899 with self-penalization, 2025b. URL [https://arxiv.](https://arxiv.org/abs/2510.02172)
900 [org/abs/2510.02172](https://arxiv.org/abs/2510.02172).
- 901
902 Yuan, L., Chen, W., Zhang, Y., Cui, G., Wang, H., You, Z.,
903 Ding, N., Liu, Z., Sun, M., and Peng, H. From $f(x)$ and
904 $g(x)$ to $f(g(x))$: Llms learn new skills in rl by composing
905 old ones, 2025. URL [https://arxiv.org/abs/](https://arxiv.org/abs/2509.25123)
906 2509.25123.
- 907
908 Yue, Y., Chen, Z., Lu, R., Zhao, A., Wang, Z., Yue, Y.,
909 Song, S., and Huang, G. Does reinforcement learn-
910 ing really incentivize reasoning capacity in llms beyond
911 the base model?, 2025. URL [https://arxiv.org/](https://arxiv.org/abs/2504.13837)
912 [abs/2504.13837](https://arxiv.org/abs/2504.13837).
- 913
914 Zelikman, E., Wu, Y., Mu, J., and Goodman, N. D. Star:
915 Bootstrapping reasoning with reasoning, 2022. URL
916 <https://arxiv.org/abs/2203.14465>.
- 917
918 Zeng, W., Huang, Y., Liu, Q., Liu, W., He, K., Ma, Z.,
919 and He, J. Simplerl-zoo: Investigating and taming
920 zero reinforcement learning for open base models in
921 the wild, 2025. URL [https://arxiv.org/abs/](https://arxiv.org/abs/2503.18892)
922 2503.18892.
- 923
924 Zhang, C., Sheng, G., Liu, S., Li, J., Feng, Z., Liu,
925 Z., Liu, X., Jia, X., Peng, Y., Lin, H., and Wu,
926 C. A framework for training large language mod-
927 els for code generation via proximal policy optimiza-
928 tion, 2024. URL [https://i.cs.hku.hk/~cwu/](https://i.cs.hku.hk/~cwu/papers/gmsheng-NL2Code24.pdf)
929 [papers/gmsheng-NL2Code24.pdf](https://i.cs.hku.hk/~cwu/papers/gmsheng-NL2Code24.pdf). PDF avail-
930 able online.
- 931
932 Zhang, C., Neubig, G., and Yue, X. On the interplay of pre-
933 training, mid-training, and rl on reasoning language mod-
934 els, 2025. URL [https://arxiv.org/abs/2512.](https://arxiv.org/abs/2512.07783)
07783.
- Zhang, L., Wang, S., and Liu, B. Deep learning for
sentiment analysis : A survey, 2018. URL [https:](https://arxiv.org/abs/1801.07883)
[//arxiv.org/abs/1801.07883](https://arxiv.org/abs/1801.07883).
- Zhang, S., Dong, L., Li, X., Zhang, S., Sun, X., Wang, S.,
Li, J., Hu, R., Zhang, T., Wu, F., and Wang, G. Instruction
tuning for large language models: A survey, 2023. URL
<https://arxiv.org/abs/2308.10792>.
- Zhang, T. Statistical analysis of some multi-category large
margin classification methods. *J. Mach. Learn. Res.*, 5:
1225–1251, December 2004. ISSN 1532-4435.
- Zhao, R., Metereez, A., Kakade, S., Pehlevan, C., Jelassi, S.,
and Malach, E. Echo chamber: Rl post-training amplifies
behaviors learned in pretraining, 2025. URL [https:](https://arxiv.org/abs/2504.07912)
[//arxiv.org/abs/2504.07912](https://arxiv.org/abs/2504.07912).
- Zheng, C., Liu, S., Li, M., Chen, X.-H., Yu, B., Gao, C.,
Dang, K., Liu, Y., Men, R., Yang, A., Zhou, J., and
Lin, J. Group sequence policy optimization, 2025. URL
<https://arxiv.org/abs/2507.18071>.
- Ziebart, B. D., Maas, A., Bagnell, J. A., and Dey, A. K.
Maximum entropy inverse reinforcement learning. In
*Proceedings of the 23rd National Conference on Artificial
Intelligence - Volume 3, AAAI'08*, pp. 1433–1438. AAAI
Press, 2008. ISBN 9781577353683.

A. Extended Related Works

Cross-entropy objective. Cross-entropy is a widely used objective in machine learning due to its simplicity and favorable theoretical properties. In particular, cross-entropy is a strictly proper scoring rule, meaning that the expected loss is uniquely minimized by the true probability distribution, which encourages statistically calibrated predictions (Good, 1992; Gneiting & Raftery, 2007; Savage, 1971; Waghmare & Ziegel, 2025). As a result, cross-entropy (or log-loss) often yields consistent estimators for classification and tends to generalize well in practice (Ng & Jordan, 2001; Zhang, 2004). However, more recent work has shown that models trained to maximize log-likelihood can still overfit and exhibit miscalibration, motivating post-hoc techniques such as temperature scaling (Niculescu-Mizil & Caruana, 2005; Guo et al., 2017). Moreover, the unbounded nature of cross-entropy and its sensitivity to small perturbations in the predicted distribution suggest that alternative strictly proper scoring rules may be more suitable in certain settings (Kornblith et al., 2021). Because cross-entropy is extensively studied, we refer interested readers to Mao et al. (2023); Li et al. (2025); Terven et al. (2025) for a comprehensive review.

Supervised training vs reinforcement learning. Supervised learning has been the go-to training paradigm in machine learning, beginning with early “learning with a teacher” neural network based systems such as the perceptron (Rosenblatt, 1958) and later becoming practical with backpropagation-based neural network training (Rumelhart et al., 1986). It has been used to tackle a broad range of problems, from financial fraud detection (Afriyie et al., 2023; Editya et al., 2025), to sentiment analysis (Zhang et al., 2018) and spam detection (Jain et al., 2022; Jamil et al., 2025). More recently, supervised training has been used in modern image classification systems (Lecun et al., 1998; Krizhevsky et al., 2012) to achieve strong performance. In modern foundation models, “pretraining” is typically done by supervised learning via minimizing cross-entropy over next-token prediction (Radford et al., 2018; 2019) on large corpus of text, followed by additional supervised training on high quality human written demonstrations (Ouyang et al., 2022; Zhang et al., 2023) to teach models how to respond to prompts, often known as instruction-tuning. In sequential decision making domains, supervised training appears as behavior cloning from expert demonstration, used in early autonomous driving and robotic systems (Pomerleau, 1988; Bojarski et al., 2016; Codevilla et al., 2018). However, supervised learning in sequential decision making breaks the i.i.d. assumption since the learned policy’s actions affect which states it visits during execution, causing compounding errors over time (Ross & Bagnell, 2010; Belkhale et al., 2023). The classic no-regret reductions line (Dagger) makes this explicit and addresses drift by iteratively querying the expert on states visited by the learner, turning the sequential problem into an online supervised learning loop with improved guarantees (Ross et al., 2010).

In such scenarios, reinforcement learning is an attractive alternative paradigm that formalizes learning under delayed, sparse, and evaluative feedback in Markov decision processes (MDPs), with foundational roots in dynamic programming and MDP theory (Bellman, 1957). Mechanistically, the key contrast with supervised learning / behavior cloning is that supervised learning assumes (or benefits strongly from) an i.i.d. dataset of correct targets under a fixed data distribution, whereas RL’s data distribution is policy-induced and nonstationary, and gradients arise from credit assignment through rewards rather than direct target labels. This mismatch shows up starkly in sequential prediction/imitation: naive behavior cloning trains on expert state distributions but at test time visits states induced by its own errors, causing compounding error (a form of distribution shift / “state drift”). Classical RL algorithms include temporal-difference learning (Sutton, 1988; Sutton et al., 1998) and value-based control such as Q-learning (Watkins & Dayan, 1992), while policy-gradient methods (Williams, 1992; Sutton et al., 1999) directly optimize expected return via likelihood-ratio gradients (REINFORCE) and later stabilized large-scale learning through variants like trust-region policy optimization (TRPO) (Schulman et al., 2015) and off-policy actor-critic methods for continuous control (e.g., DDPG (Lillicrap et al., 2015)) and maximum-entropy actor-critic (e.g., SAC (Haarnoja et al., 2018)). Empirically, deep RL’s modern resurgence is often associated with representation learning + RL on high-dimensional inputs (e.g., DQN (Mnih et al., 2013)).

A large body of work blends supervised and RL to get the best of both: (i) Imitation + online correction methods like DAgger (Ross et al., 2010) explicitly combine supervised learning with interactive data collection to mitigate distribution shift; (ii) Inverse RL / MaxEnt IRL reframes imitation as learning a reward/cost model that explains expert behavior, with maximum-entropy formulations giving a principled probabilistic objective (Ziebart et al., 2008); (iii) Adversarial imitation (GAIL) (Ho & Ermon, 2016) avoids explicit reward learning by matching occupancy measures via a GAN-like discriminator, typically trained with policy optimization; (iv) Learning from demonstrations in deep RL injects supervised losses and/or demonstration replay into RL to improve exploration and sample efficiency—e.g., DQfD (Hester et al., 2017) combines TD learning with supervised large-margin imitation terms, and demonstration-augmented continuous-control methods address sparse-reward exploration failures (Nair et al., 2018); and (v) Trajectory-optimization-guided policy learning (guided policy search (Levine & Koltun, 2013)) explicitly produces supervised targets for a policy network from trajectory optimization /

990 local controllers, bridging optimal control, RL, and supervised regression. In modern LLM alignment, the same hybrid
991 template appears as “Supervised fine-tuning + preference-based RL”: InstructGPT (Ouyang et al., 2022) first performs
992 supervised fine-tuning on demonstrations and then applies RL from human feedback (RLHF), while more recent approaches
993 like Direct Preference Optimization (DPO) (Rafailov et al., 2023) recast parts of RLHF into a supervised-style classification
994 objective — illustrating an active trend of recovering supervised-like training signals even when the underlying goal is
995 preference/reward optimization.

996 Finally, control as inference is also a closely related topic (Millidge et al., 2020; O’Donoghue et al., 2020; Rawlik et al.,
997 2012; Ito & Kashima, 2024; Tarbouriech et al., 2023), and we point the reader to Levine (2018) for more details.
998
999

1000 **Training LLMs for strong reasoning abilities.** A few different approaches for post-training have demonstrated success,
1001 including supervised fine-tuning on human-crafted high quality demonstrations (Wang et al., 2023b), iterative supervised
1002 training on self-generated good quality responses (Zelikman et al., 2022; Gulcehre et al., 2023), reinforcement learning
1003 from a learned reward model on human preferences (Ouyang et al., 2022), and more recently preference-based contrastive
1004 learning (Rafailov et al., 2023; Pang et al., 2024). In our work, we focus on recovering the cross-entropy based classification
1005 objective in an RL training pipeline, fundamentally differing from the prior works. Since the recent advent of RLVR,
1006 multiple followup works have studied the RLVR pipeline (Zeng et al., 2025; Liu et al., 2025b; Khatri et al., 2025) and
1007 proposed alternative algorithms such as Dr.GRPO (Liu et al., 2025b), DAPO (Yu et al., 2025a), GSPO (Zheng et al., 2025)
1008 and CISPO (MiniMax et al., 2025), RAFT (Xiong et al., 2025a). The idea of normalizing advantages by mean reward,
1009 similar to ours, has been explored in (Huang et al., 2025), but whereas we normalize advantage by group mean reward
1010 (mean reward over the rollouts associated with a particular prompt), Huang et al. (2025) normalizes by the batch mean
1011 reward (mean reward over all prompts in a batch of policy gradient updates). Finally, recent work such as Zhang et al.
1012 (2025) has also studied how RL training is influenced by pretraining and midtraining in toy didactic settings, establishing
1013 the importance of good pretraining/midtraining for the success of RL, similar to Gandhi et al. (2025).
1014

1015 **On exploration for reinforcement learning for LLMs.** Exploration, or taking actions to discover new information, is a
1016 widely studied topic in reinforcement learning. A closely related topic is *curiosity*, where an agent seeks new information
1017 about its environment via interactions. *Intrinsic motivation* is a popular notion for curiosity, where the agent is driven
1018 by an exploration bonus that is not necessarily related to the task to be achieved (Schmidhuber, 1991; 2007). Followup
1019 works have built on this notion to mitigate problems of sparse reward (reward is observed at a very belated phase of
1020 interactions) or no reward at all (Pathak et al., 2017; 2019; Eysenbach et al., 2018; Burda et al., 2018; Sharma et al., 2019;
1021 Yang et al., 2024c; Houthoof et al., 2016). Count-based bonuses have also been introduced as a way of computing intrinsic
1022 motivation (Bellemare et al., 2016). Prompt-level reweighting of gradients has also been studied (Yu et al., 2025b), though
1023 under a different context (self-training) and a different weighting mechanism. Finally, adding noise to network parameters
1024 or optimization has been another line of work to improve exploration during RL training (Fortunato et al., 2017; Ishfaq et al.,
1025 2024b;a; 2025). Maximum entropy RL, the principle where one attempts to recover an agent that achieves high reward but
1026 is as stochastic as possible, can be seen as another attempt at solving exploration for classical RL (Haarnoja et al., 2018;
1027 Boucher et al., 2025; Eysenbach & Levine, 2022; Dong et al., 2025). In summary, exploration-exploitation tradeoff (Sutton,
1028 1988; Auer et al., 2002; Thompson, 1933) has been a crucial topic for ensuring RL agents’ success.

1029 More recently, exploration has emerged as an important topic for building modern LLM based systems. There are two
1030 types of exploration to consider. The first is *inference-time exploration*, where an agent has to efficiently gather information
1031 during deployment by strategically choosing its interactions with its environment, Tajwar et al. (2025) is an important
1032 work in this line of research. More importantly, pass@k degradation (mode collapse) during RLVR (Yue et al., 2025;
1033 Wu et al., 2026; GX-Chen et al., 2025) has prompted research into *train-time exploration*, where the challenge is to go
1034 beyond the pretrained model’s capabilities and discover new knowledge. Primary approaches include directly optimizing
1035 for pass@k (Walder & Karkhanis, 2025; Tang et al., 2025; Chen et al., 2025d), curriculum learning (Tajwar et al., 2025;
1036 Chen et al., 2025c; Setlur et al., 2025; Motwani et al., 2025), learning from additional hints or abstractions (Qu et al.,
1037 2025b; Chen et al., 2025a; Anonymous, 2025b), increasing number of rollouts to prevent RL gains from saturating (Hu
1038 et al., 2025), employing data curation algorithm to redirect effort to problems with low success rate (Nguyen et al., 2025b),
1039 leveraging expert guidance (Chang et al., 2024; Qu et al., 2025a), or differential smoothing by penalizing entropy on low
1040 reward trajectories and encouraging entropy on high reward trajectories. Entropy based bonuses to encourage exploration
1041 during RL training (Hao et al., 2025; Chen et al., 2025b; Cheng et al., 2025a; Wang et al., 2025; Anonymous, 2025a; Ged &
1042 Veiga, 2024) is another popular line of work for improving exploration. A few modern approaches for exploration bonus
1043 utilized for LLM training are Song et al. (2025); Tuyls et al. (2025). The idea of curiosity-driven exploration from classical
1044

RL discussed above has also been adopted for LLMs (Dai et al., 2025). Although some works have reported pass@k degradation during RL training, others have found the opposite results. For example, ProRL (Liu et al., 2025a) has shown that RL training on a mixture of reasoning puzzles (Stojanovski et al., 2025) can improve pass@k on a heldout reasoning task. Similarly, Yuan et al. (2025) has shown that LLMs can learn new skills via RL by composing old ones, showing the promise of going beyond pre-training knowledge, and Cheng et al. (2025b) also found pass@k to improve, particularly on tasks less likely to appear during the pre-training stage. Ray interference (Schaul et al., 2019) has been proposed as an explanation for the observed pass@k degradation. Overall, this line of research remains important as focus moves to LLMs discovering new information during RL training and it is therefore an ongoing field of research.

B. Theoretical Results

Here we present the proofs of theorems mentioned in the main paper. First we restate and prove Theorem 4.1.

Theorem B.1 (Restatement of Theorem 4.1). *The gradient of the maximum likelihood objective admits the following conditional expectation representation:*

$$\nabla_{\theta} J_{\text{ML}}(x) = \mathbb{E}[\nabla_{\theta} \log m_{\theta}(z | x) | f(z) = y^*(x)].$$

Proof. Recall the standard REINFORCE identity for the gradient of the pass rate:

$$\nabla_{\theta} p_{\theta}^{\text{pass}}(x) = \nabla_{\theta} \mathbb{E}_{z \sim m_{\theta}(\cdot | x)}[\mathbb{I}\{f(z) = y^*(x)\}] = \mathbb{E}_{z \sim m_{\theta}(\cdot | x)}[\mathbb{I}\{f(z) = y^*(x)\} \nabla_{\theta} \log m_{\theta}(z | x)].$$

The gradient of the maximum likelihood objective is:

$$\nabla_{\theta} J_{\text{ML}}(x) = \nabla_{\theta} \log p_{\theta}^{\text{pass}}(x) = \frac{\nabla_{\theta} p_{\theta}^{\text{pass}}(x)}{p_{\theta}^{\text{pass}}(x)} = \frac{\mathbb{E}_{z \sim m_{\theta}(\cdot | x)}[\mathbb{I}\{f(z) = y^*(x)\} \nabla_{\theta} \log m_{\theta}(z | x)]}{\mathbb{E}_{z \sim m_{\theta}(\cdot | x)}[\mathbb{I}\{f(z) = y^*(x)\}]}.$$

By the definition of conditional expectation for an event A with $\mathbb{P}(A) > 0$:

$$\mathbb{E}[X | A] = \frac{\mathbb{E}[X \cdot \mathbb{I}_A]}{\mathbb{P}(A)}.$$

Letting $X = \nabla_{\theta} \log m_{\theta}(z | x)$ and $A = \{z : f(z) = y^*(x)\}$, and noting that $p_{\theta}^{\text{pass}}(x) = \mathbb{P}(A)$, we obtain:

$$\nabla_{\theta} J_{\text{ML}}(x) = \mathbb{E}[\nabla_{\theta} \log m_{\theta}(z | x) | f(z) = y^*(x)].$$

□

Next, we restate and prove Theorem 4.2.

Theorem B.2 (Restatement of Theorem 4.2). *The estimator $\hat{g}_N(x)$ is an unbiased estimator for the MAXRL gradient of order $T = N$, i.e.,*

$$\mathbb{E}[\hat{g}_N(x)] = \nabla_{\theta} J_{\text{MAXRL}}^{(N)}(x).$$

Proof. Conditioned on $K \geq 1$, the successful samples are i.i.d. draws from the success-conditioned distribution, so by Theorem 4.1:

$$\mathbb{E}[\hat{g}_N(x) | K \geq 1] = \nabla_{\theta} \log p_{\theta}^{\text{pass}}(x).$$

Since $\hat{g}_N(x) = 0$ when $K = 0$:

$$\mathbb{E}[\hat{g}_N(x)] = \nabla_{\theta} \log p_{\theta}^{\text{pass}}(x) \cdot \mathbb{P}(K \geq 1) = \nabla_{\theta} \log p_{\theta}^{\text{pass}}(x) \cdot \text{pass}@N(x).$$

Writing $p = p_{\theta}^{\text{pass}}(x)$ and using $\text{pass}@k(x) = 1 - (1 - p)^k$:

$$\frac{\nabla_{\theta} p}{p} \cdot (1 - (1 - p)^N) = \nabla_{\theta} p \sum_{k=1}^N (1 - p)^{k-1} = \sum_{k=1}^N \frac{1}{k} \nabla_{\theta} \text{pass}@k(x) = \nabla_{\theta} J_{\text{MAXRL}}^{(N)}(x),$$

where the second equality uses $\nabla_{\theta} \text{pass}@k(x) = k(1 - p)^{k-1} \nabla_{\theta} p$.

□

Proposition B.3. For MAXRL with order T , we can rewrite it as

$$\nabla_{\theta} J_{\text{MAXRL}}^{(T)} = \mathbb{E}_{x \sim \rho} [w(p_{\theta}(x)) \nabla_{\theta} p_{\theta}(x)],$$

where

$$w_T(p) = \sum_{k=1}^T (1-p)^{k-1} = \frac{1 - (1-p)^T}{p}.$$

Proof. From Equation (5), we have:

$$\nabla_{\theta} J_{\text{MAXRL}}^{(T)}(x) = \sum_{k=1}^T \frac{1}{k} \nabla_{\theta} \text{pass}@k(x).$$

Using $\text{pass}@k(x) = 1 - (1-p)^k$ where $p = p_{\theta}^{\text{pass}}(x)$:

$$\nabla_{\theta} \text{pass}@k(x) = k(1-p)^{k-1} \nabla_{\theta} p.$$

Substituting:

$$\nabla_{\theta} J_{\text{MAXRL}}^{(T)}(x) = \sum_{k=1}^T \frac{1}{k} \cdot k(1-p)^{k-1} \nabla_{\theta} p = \left(\sum_{k=1}^T (1-p)^{k-1} \right) \nabla_{\theta} p = w_T(p) \nabla_{\theta} p_{\theta}^{\text{pass}}(x).$$

Taking the expectation over $x \sim \rho$ completes the proof. □

C. Algorithm Pseudocode

Algorithm 1 Practical Implementation of MAXRL

- 1: **Input:** Batch of training tasks B , number of rollouts K , latent generation model m_{θ}
- 2: **for** each training example $(x_i, y^*(x_i)) \in B$ **do**
- 3: Sample K rollouts $y^1, \dots, y^K \stackrel{\text{i.i.d.}}{\sim} m_{\theta}(\cdot|x_i)$
- 4: Calculate binary reward $r_j = \mathbb{I}[y^j = y^*(x_i)]$
- 5: Calculate mean reward $\hat{\mu} = \frac{1}{K} \sum_{j=1}^K r_j$
- 6: Set advantage

$$A(y_j) = \begin{cases} 0, & \text{if } \hat{\mu} = 0 \\ \frac{r_j - \hat{\mu}}{\hat{\mu}}, & \text{otherwise} \end{cases}$$

- 7: **end for**
 - 8: Take a policy gradient step
-

Section C shows the practical algorithm we use for running MAXRL. The most important part is the modified advantage computation, and the rest follows standard policy gradient training like GRPO/RLOO.

D. Additional Details on ImageNet Experiments

D.1. Training Procedure

Let \mathcal{X} be the input space and \mathcal{Y} be the label space. Let π_θ denote our model: given an input image $x \in \mathcal{X}$, $\pi_\theta(y|x)$ is model’s predicted probability of image x belonging to class $y \in \mathcal{Y}$. For an input image and label pair $(x, y^*(x))$, the cross-entropy loss is:

$$\mathcal{L}_{\text{CE}}(x, y^*(x); \pi_\theta) = -\log \pi_\theta(y^*(x)|x)$$

On the other hand, the corresponding RL objective for the same pair is:

$$\mathcal{L}_{\text{RL}}(x, y^*; \pi_\theta) = -\mathbb{E}_{y \sim \pi_\theta(\cdot|x)}[-\log \pi_\theta(y|x) \cdot \hat{A}(y|x)]$$

where the expectation is computed using Monte-Carlo sampling K rollouts of y from $\pi_\theta(\cdot|x)$. GRPO, REINFORCE and MAXRL vary only in the calculation of the advantage $A(y|x)$. Concretely, let $y^{(1)}, \dots, y^{(K)}$ be our K rollouts, sampled from the conditional probability distribution $\pi_\theta(\cdot|x)$. We operate under a binary reward setting, meaning the reward function $r(x, y)$ is:

$$r(x, y) = \mathbb{I}[y = y^*(x)] = \begin{cases} 1, & \text{if } y = y^*(x) \\ 0, & \text{otherwise} \end{cases}$$

Given this reward, we calculate advantage under GRPO, REINFORCE and MAXRL as follows:

$$\hat{A}_{\text{GRPO}}(x, y) = \frac{r(x, y) - \hat{\mu}}{\hat{\sigma}}$$

$$\hat{A}_{\text{REINFORCE}}(x, y) = r(x, y) - \hat{\mu}$$

$$\hat{A}_{\text{MAXRL}}(x, y) = \frac{r(x, y) - \hat{\mu}}{\hat{\mu}}$$

where $\hat{\mu} = \frac{\sum_{i=1}^K r(x, y^{(i)})}{K}$, $\hat{\sigma} = \sqrt{\frac{\sum_{i=1}^K (r(x, y^{(i)}) - \hat{\mu})^2}{K}}$ is the mean and standard deviation of rewards of the sampled rollouts.

Finally, at each training step, a batch of (input image, label) pairs are collected from the training dataset. The above computation gives us per (input image, label) loss, we average them over all the pairs in a given batch to calculate the final loss which is then used to update the model via gradient descent.

D.2. Training Hyperparameters

We use the following set of hyperparameters in all ImageNet experiments:

- **Batch size:** 256
- **Number of epochs:** 20
- **Optimizer:** SGD with momentum 0.9, no Nesterov momentum, initial learning rate 0.1. We run a sweep over the learning rate over 0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 0.7, and 1.0. We find the standard learning rate, 0.1, generally works well for all objectives, and report that in our experiments.
- **Learning rate scheduler:** Cosine scheduler (Loshchilov & Hutter, 2017), with linear warmup for the first epoch.

- **Image augmentations:** No augmentations are used for evaluation, we only resize each image to 224x224 and normalize the images by mean and standard deviation of pixel values. For training, in addition to the same resizing and normalizing steps, we also add a random horizontal flip (with probability 0.5) and a random resized crop to 224 (with scale (0.08, 1.0)).
- **Number of rollouts, K:** This is usually varied for different experiments.

All training is done on single L40S GPUs for 15 hours.

D.3. Equivalence of Validation Top-1 Accuracy and Majority Voting Accuracy

In this section, we discuss the validation top-1 accuracy metric, which is the traditional metric used in image classification. Formally, validation accuracy for a single image and label pair $(x, y^*(x))$ is defined as:

$$\text{Accuracy}(x, y^*(x); \pi_\theta) = \mathbb{I} \left[\arg \max_{y \in \mathcal{Y}} \pi_\theta(y|x) = y^*(x) \right] = \begin{cases} 1, & \text{if } \arg \max_{y \in \mathcal{Y}} \pi_\theta(y|x) = y^*(x) \\ 0, & \text{otherwise} \end{cases}$$

which is then averaged over all validation examples for the final metric. In other words, validation accuracy is the same as majority voting accuracy (Wang et al., 2023a) in traditional LLM chain-of-thought reasoning tasks.

D.4. Pass@k Calculation

To calculate pass@k from a generative model, one usually samples $T \geq k$ rollouts from the model, calculate success or failure from each of them, and then uses an appropriate statistical estimator for pass@k (Chen et al., 2021; Yue et al., 2025). However, since there is no latent reasoning process involved in our didactic ImageNet experiments and since we can directly calculate the model likelihood of label $y \in \mathcal{Y}$ for an input image $x \in \mathcal{X}$, namely $\pi_\theta(y|x)$, we can also analytically compute pass@k without sampling as well. Formally, in all ImageNet experiments, we calculate pass@k for an example (image, label) pair $(x, y^*(x))$ as follows:

$$\text{Pass@k}(x, y^*(x); \pi_\theta) = 1 - (1 - \pi_\theta(y^*(x)|x))^k$$

The average pass@k is then obtained by averaging the above quantity over all example pairs in the validation dataset.

D.5. Gradient Norm Analysis

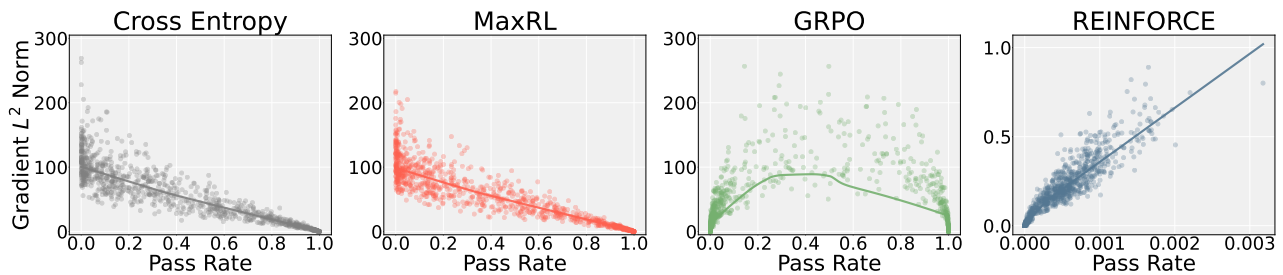


Figure 8. (ImageNet Gradient Norm Analysis) Scatter plot, where each point has the pass rate (model’s predicted probability of the correct class) of a particular image in the x-axis, and gradient L^2 norm for that image in the y-axis, for 1000 randomly selected images from the ImageNet validation dataset after 1500 steps of training on a ResNet-50 model. Cross Entropy and MAXRL have similar scatter plot: with high gradient norm for hard inputs (pass rate close to 0) and lower gradient norm for the easier ones (pass rate close to 1). In contrast, highest gradient norm for GRPO is on medium difficulty (pass rate close to 0.5) inputs, with hard inputs having very low gradient norm. Finally, REINFORCE fails to produce any significant gradient norm and its pass rate is confined below 0.003 after 1500 steps, demonstrating its difficulty to learn in this setting.

Figure 8 shows the correlation between gradient norm and pass rate (model’s predicted probability of the correct class) for a particular image on different objectives. We see that cross-entropy and MAXRL have similar scatter plot: with high gradient norm for hard inputs (pass rate close to 0) and lower gradient norm for the easier ones (pass rate close to 1). In contrast,

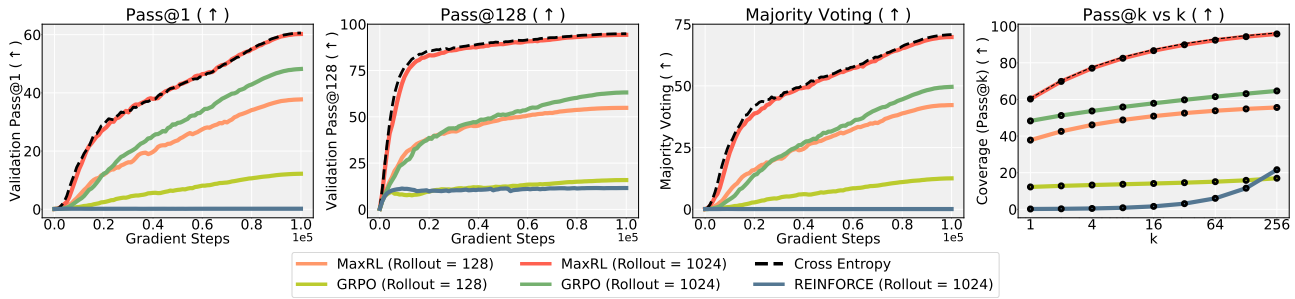


Figure 9. (Additional ImageNet results) On the didactic image classification setting, MAXRL outperforms and scales better than GRPO with additional compute, and approaches the same performance as maximum likelihood training via cross-entropy given sufficient number of rollouts (≥ 1024). Note that REINFORCE remains flat, since the initial model’s pass rate is low ($\sim 0.1\%$) and REINFORCE fails to generate significant gradient signal (Figure 8). From left, the plots show Pass@1, Pass@128, Majority Voting Accuracy (equivalent to traditional validation top-1 accuracy in image classification, see Appendix D.3), and coverage of the final checkpoint, respectively.

highest gradient norm for GRPO is on medium difficulty (pass rate close to 0.5) inputs, with hard inputs having very low gradient norm. Finally, REINFORCE fails to produce any significant gradient norm compared to the other objectives and its pass rate is confined below 0.003 after 1500 steps, demonstrating its difficulty to learn in this setting. This is also reflected in our other results, where REINFORCE does not show any signs of learning. We attribute this to the very low gradient norm: since the randomly initialized model has pass rate 0.001 in expectation over all inputs, REINFORCE fails to produce sufficiently large gradients during training and therefore stalls in model improvement. One caveat: REINFORCE’s failure maybe due to us training the model from scratch — on a pretrained model, it indeed produces gradients but still shows poor gradient norm on hard inputs (see Figure 6).

D.6. More Experimental Results

Here we present additional experimental results. In particular, (1) we compare against GRPO with varying number of rollouts, (2) record additional metrics such as majority voting accuracy (i.e., validation top-1 accuracy), and (3) show the resulting coverage (pass@k vs k) from different objectives. Figure 9 records our findings: MAXRL outperforms and scales better than GRPO with additional compute. While GRPO improves performance if given more compute unlike REINFORCE, it remains suboptimal compared to MAXRL and supervised cross-entropy training. Moreover, both GRPO and REINFORCE exhibit worse coverage as their pass@k values are significantly lower compared to MAXRL, corroborating our experiments from other sections.

E. Details on Other Training Settings

E.1. Maze

E.1.1. MODEL ARCHITECTURE

We adopt a lightweight decoder-only Transformer model following the Qwen2 architecture (Yang et al., 2024a), with a total of approximately $3M$ parameters. The model consists of 4 Transformer layers, each using full self-attention. The hidden size is set to 256, with an intermediate (feed-forward) dimension of 1024, and 4 attention heads per layer. We use grouped query attention with 2 key-value heads. The model employs RMSNorm with $\sigma = 1 \times 10^{-6}$ and uses the SiLU activation function in the feed-forward networks. Rotary positional embeddings (RoPE) (Su et al., 2021) are applied with $\theta = 1,000,000$, and the maximum sequence length is 512 tokens. The vocabulary size is 32 tokens, and input and output embeddings are tied. The model is trained and evaluated using bfloat16 precision, with attention dropout set to 0. The architecture follows a standard causal language modeling setup with autoregressive decoding.

E.1.2. TASK DESCRIPTION

Mazes are procedurally generated using Prim’s algorithm (Prim, 1957), and task difficulty is controlled by the grid size. We use a symbolic tokenization to represent both the maze layout and the navigation policy, with tokens drawn from a small, discrete vocabulary.

The input sequence describes a two-dimensional grid in row-major order. Each cell is represented by a single token indicating its type (e.g., WALL, PATH, START, or GOAL). Rows are separated by a dedicated NEWLINE token, and the entire grid is delimited by special boundary tokens marking the beginning (GRID_START) and end (GRID_END) of the grid description. Following the maze specification, the model autoregressively generates a sequence of navigation actions drawn from a fixed action vocabulary (e.g., directional moves) and terminates by a DONE token.

Below, we provide an example data instance following this format.

7*7 Maze Example Model Input and Output Format

Input:

```
<bos> GRID_START WALL WALL WALL WALL WALL WALL WALL WALL NEWLINE WALL START WALL PATH PATH
PATH WALL NEWLINE WALL PATH WALL PATH WALL WALL WALL NEWLINE WALL PATH PATH PATH PATH
PATH WALL NEWLINE WALL PATH WALL WALL WALL PATH WALL NEWLINE WALL PATH WALL PATH PATH
GOAL WALL NEWLINE WALL WALL WALL WALL WALL WALL WALL NEWLINE GRID_END PATH_START
```

Output:

```
RIGHT RIGHT RIGHT RIGHT DOWN DOWN DOWN DOWN DONE <eos>
```

For reference, we also visualize one typical successful trajectory and one representative failed prediction in Fig. 10.

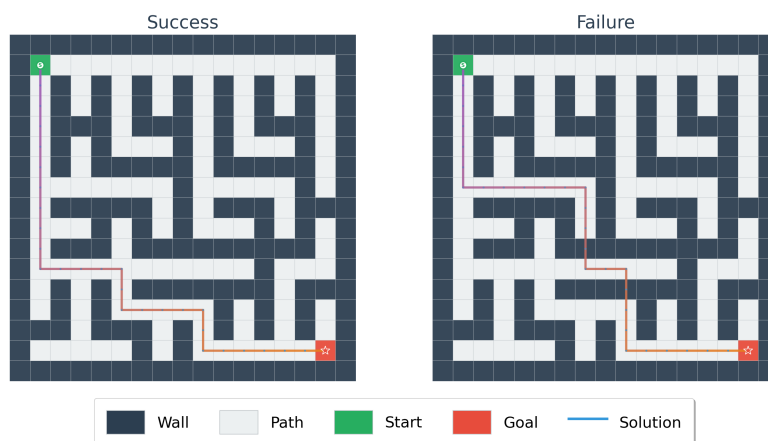


Figure 10. (Maze Data Visualization) The left plot shows a successful navigation trajectory, while the right plot illustrates a failure case produced by the trained model, where the generated action sequence deviates from the correct path before reaching the goal.

E.1.3. TRAINING SETUPS

To ensure sufficient task complexity and rigorous evaluation, we construct a training set of 1 million distinct 17×17 mazes and a test set of 256 non-overlapping samples. We first pretrain the model from scratch, where it is trained to follow a provided ground-truth trajectory for each maze. During SFT, we use a learning rate of 5×10^{-4} with the AdamW optimizer (Kingma & Ba, 2017; Loshchilov & Hutter, 2019) and train for 1,500 steps with a batch size of 32. This pretraining stage initializes the model with the basic output format required for representing maze-solving trajectories. Subsequently, we perform reinforcement learning (RL) training. By default, we use a data batch size of 32, a rollout number of 128, a learning rate of 1×10^{-4} . We update the model parameters only once per RL step (fully on-policy setting (Tajwar et al., 2024)) to ensure all trajectories are on-policy, with 20,000 training steps. On this basis, we compare the performance of different RL algorithms.

In addition, we fix the data batch size to 256 and vary the rollout number from 4, 8, 16, 32, 64 to 128 when illustrating the scaling trends of GRPO and our MaxRL algorithm. Each training run uses 4 RTX 4090 GPUs and is conducted for at least 9,000 steps. Given the small number of model parameters, the model is no longer memory-bound, so we modified the rollout engine (instead of using the default vLLM (Kwon et al., 2023) engine) in the `verl` (Zhang et al., 2024; Sheng et al., 2024) codebase to increase sampling parallelism and speed up training.

E.2. GSM8K

E.2.1. PROMPT TEMPLATE

We conduct reinforcement finetuning for the SmoLLM2-360M-Instruct model (Allal et al., 2025) on GSM8K (Cobbe et al., 2021) training set for 1500 steps. We use GSM8K-Platinum (Vendrow et al., 2025), a revised version of the original GSM8K test that is non-overlapping with the GSM8K train dataset, for evaluation. An example of prompt and the model’s response are shown below. In particular, we take each individual problem, append it with “Let’s think step by step and output the final answer within `\boxed{}`.” and process the string through the SmoLLM2-360M-Instruct model’s tokenizer chat-template to obtain the final prompt for the model.

SmolLM2-360M-Instruct

```
<|im.start|>system
You are a helpful AI assistant named SmoLLM, trained by Hugging Face<|im.end|>
<|im.start|>user
Jeannie hikes the 12 miles to Mount Overlook at a pace of 4 miles per hour, and then
returns at a pace of 6 miles per hour. How long did her hike take, in hours? Let’s
think step by step and output the final answer within \boxed{}. <|im.end|>
<|im.start|>assistant
To find the time it takes for Jeannie to hike the entire trip, we need to calculate her
total travel time and then divide it by the rate at which she travels.
Jeannie hikes the 12 miles at a pace of 4 miles per hour. So, the time taken is 12
miles / 4 miles per hour = 3 hours.
Next, she returns at a pace of 6 miles per hour. So, the time taken to return is 12
miles / 6 miles per hour = 2 hours.
Therefore, the total time taken is 3 hours + 2 hours = 5 hours.
So, the final answer is 5 hours. \boxed{5} <|im.end|>
```

E.2.2. HYPERPARAMETERS

Detailed hyperparameter setup is provided in Table 5.

Table 5. Training hyperparameters for SmoLLM2-360M-Instruct on GSM8K.

Training hyperparameters for SmoLLM2-360M-Instruct			
Parameter	Value	Parameter	Value
Base model	SmoLLM2-360M-Instruct	Training set	GSM8K
Test set	GSM8K	Prompts per batch	256
Generations per prompt	128	Grad update per RL step	1
Max prompt length	512	Max response len	2048
Learning rate	1×10^{-5}	Training Steps	1500
KL coeff	0.0	Entropy coeff	0.0
Rollout temp	1.0	Validation top_p	0.95
Validation temp	0.6	Device	8 × Nvidia GH200

E.3. Qwen3 Training

E.3.1. PROMPT TEMPLATE

We use the Qwen-math template (Yang et al., 2024a; Qwen et al., 2025; Yang et al., 2024b) for formatting our prompts. We show an example prompt (Yu et al., 2025a) after formatting through our template below. In particular, we take each individual problem, append it with “\nPlease reason step by step, and put your final answer within `\boxed{\}`.” and process the string through the SmoLLM2-360M-Instruct model’s tokenizer chat-template to obtain the final prompt for the model.

Qwen Math Prompt Template

```
<|im_start|>system
Please reason step by step and put the final answer in \boxed{\}. <|im_end|>
<|im_start|>user
Denote by  $S(n)$  the sum of the digits of the positive integer  $n$ . Find all the solutions
of the equation  $n(S(n) - 1) = 2010$ . Let’s think step by step and output the final answer
within \boxed{\}. <|im_end|>
<|im_start|>assistant
```

E.3.2. HYPERPARAMETERS

Next, we describe the default hyperparameters for our training setup. Since there are many possible alternatives to handle off-policy updates and corresponding importance ratio (Schulman et al., 2017; Shao et al., 2024; Zheng et al., 2025; MiniMax et al., 2025; Yu et al., 2025a), to keep things simple, we choose to train in the fully on-policy setup, meaning we have no importance ratio or associated clipping. Similarly, to avoid tuning additional hyperparameters for each algorithm, following Olmo et al. (2025), we remove KL penalty and also entropy bonus in our default training comparison. Note: we train with GRPO and entropy bonus as a baseline in our SmoLLM2-360M-Instruct training on GSM8K, results are recorded in Table 4: MAXRL outperform this variant, showing that entropy bonus does not fully mitigate issues resulting from GRPO though it can slightly mitigate it, as also observed by Yue et al. (2025).

We generate all training rollouts using temperature 1.0, and do not use special sampling techniques. Similarly, we also do not use any adaptive sampling (Yu et al., 2025a) or fixes for inference-training logit mismatch (He & Lab, 2025; Khatri et al., 2025) Finally, for evaluation, we follow the same protocol as Yue et al. (2025), and we run inference with temperature 0.6, top-p sampling parameter 0.95, no top-k or min-p sampling (Nguyen et al., 2025a).

Table 6 shows our default hyperparameter setting.

Table 6. Training hyperparameters for Qwen3-1.7B-Base and Qwen3-4B-Base training.

Training hyperparameters for Qwen3-1.7B-Base and Qwen3-4B-Base			
Parameter	Value	Parameter	Value
Base model	Qwen3-1.7B-Base, Qwen3-4B-Base	Prompts per batch	256
Generations per prompt	16	Grad update per RL step	1
Max prompt length	1024	Max response len	4096
Learning rate	1×10^{-6}	Training Steps	1000
KL coeff	0.0	Entropy coeff	0.0
Rollout temp	1.0	Validation top-p	0.95
Validation temp	0.6	Device	32 \times Nvidia H200

F. General RL Training Objective

Our discussion here follows that of (Shafayat et al., 2025). For continuity with existing literature, we use slightly different notations from the rest of the paper for this section. Let x represent a prompt, and let $y \sim \pi(\cdot|x)$ represent sequence of tokens autoregressively sampled from the language model π conditioned on the prompt x . Let π_θ be the current policy, and $\pi_{\theta_{\text{old}}}$ be an older policy (from earlier iterations in training) used for data generation. In our implementation (based on ver1 (Zhang et al., 2024; Sheng et al., 2024)), we use the following general RL objective:

$$\mathcal{J}(\theta) = \mathbb{E}_{x \sim \mathcal{D}, \{y_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(\cdot|x)} \left[\frac{1}{T} \sum_{i=1}^G \sum_{t=1}^{|y_i|} \min \left(w_{i,t}(\theta) \hat{A}_{i,t}, \text{clip}(w_{i,t}(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_{i,t} \right) \right]$$

where T is the total number of tokens generated in the entire batch, π_θ represents the current LLMs autoregressive probability distribution, $\pi_{\theta_{\text{old}}}$ denote the behavior policy/data generation policy’s probability distribution, $w_{i,t}(\theta)$ is the importance ratio, defined as:

$$w_{i,t}(\theta) = \frac{\pi_\theta(y_{i,t} | x, y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t} | x, y_{i,<t})}$$

Since we operate fully on-policy, i.e., one RL step per one batch of generated rollouts, this is always one in our experiments, and the clipping parameter ϵ has no effect on our training. $\hat{A}_{i,t}$ represents the advantage for the t -th token in the sequence y_i . The same advantage defined at a sequence level is applied to each token in the sequence, so henceforth we will drop the t from the notation as well.

The main difference between GRPO (Shao et al., 2024), RLOO (Ahmadian et al., 2024) and MAXRL come from their use of different advantage functions. RLOO objective uses the following advantage function:

$$\frac{1}{G} \sum_{i=1}^G [R(y_{(i)}, x) - \frac{1}{G-1} \sum_{j \neq k} R(y_{(j)}, x)]$$

whereas GRPO uses the following advantage function:

$$\hat{A}_i = \frac{r(x, y_i) - \text{mean}(\{r(x, y_i)\}_{i=1}^G)}{\text{std}(\{r(x, y_i)\}_{i=1}^G) + \epsilon}$$

where ϵ is a small number (1×10^{-6}) added to avoid division by zero. Finally, the advantage for MAXRL is follows:

$$\hat{A}_i = \frac{r(x, y_i) - \text{mean}(\{r(x, y_i)\}_{i=1}^G)}{\text{mean}(\{r(x, y_i)\}_{i=1}^G) + \epsilon}$$

Here G is the number of online samples generated. RLOO, GRPO and MAXRL create a dynamic baseline for each sample without needing a separate value function (unlike PPO (Schulman et al., 2017)), effectively estimating the expected return on-the-fly during training. Not having a value networks makes the training much simpler for all three algorithms.

G. Results on an Additional Benchmark: AIME 2024

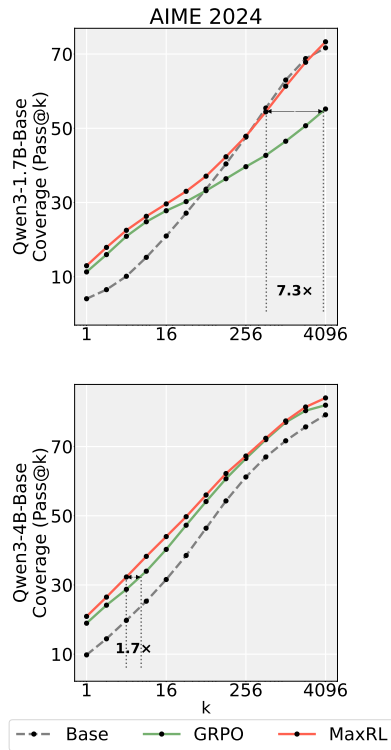


Figure 11. (Evaluation of Qwen3 model training on AIME 2024) Here we report performance on an additional benchmark, AIME 2024. MAXRL outperform both base model and GRPO on both Qwen3-1.7B-Base and Qwen3-4B-Base, leading to 7.3x and 1.7x maximum inference efficiency respectively.

H. More on MAXRL Extracting Better Learning Signal During Training

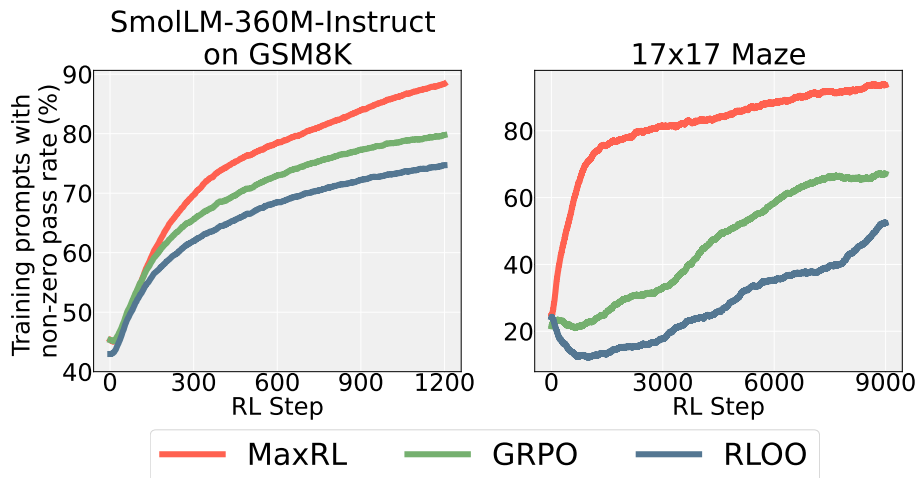


Figure 12. (Fraction of training tasks with non-zero pass rate) Similar to Figure 7, we also record the fraction of training tasks where the model generates at least one correct rollout on maze and SmolLM2-360M-Instruct training on GSM8K. We see the same trends as Figure 7, and MAXRL consistently outperforms both GRPO and RLOO, demonstrating MAXRL ability to generate better learning signal during training, as tasks with zero pass-rate contributes no gradients.

I. Additional Results on Qwen3-4B-Base

I.1. Majority Voting Performance

Table 7. (Majority Voting Performance Comparison on Qwen3-4B-Base) We compare the performance of MAXRL in terms of majority voting against the pretrained base model and GRPO.

	AIME 2024 (majority@4096)	AIME 2025 (majority@4096)	BeyondAIME (majority@4096)	MATH-500 (majority@2048)	Minerva (majority@2048)
Base	23.3	23.3	7.0	69.8	18.8
GRPO	23.3	23.3	7.0	72.4	27.2
MaxRL	26.7	26.7	14.0	74.0	28.7

Here we present comparisons across one other metric, majority voting (Wang et al., 2023a), a commonly used verifier free method for scaling test-time compute, where we generate N i.i.d. rollouts from the model for a single task x , group the responses by the final answer, and take the most frequent answer as our outcome. Table 7 shows our results across all five benchmarks, we outperform both the pre-trained base model and GRPO trained model on majority voting across all benchmarks.

I.2. Training Dynamics

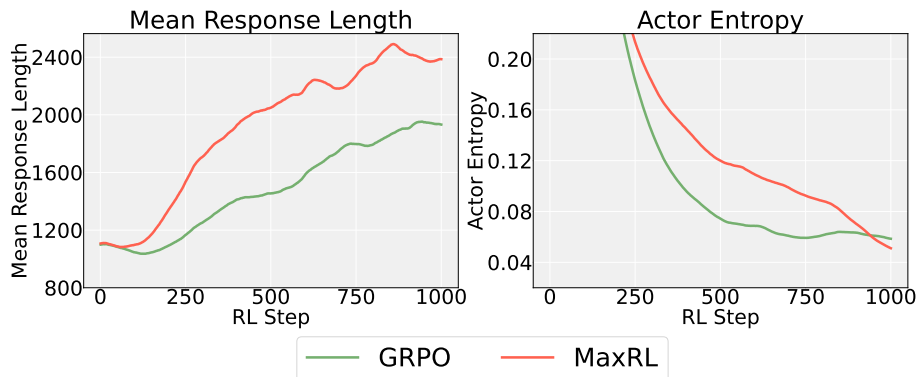


Figure 13. (Additional training dynamics metrics for Qwen3-4B-Base) We show comparison between GRPO and MAXRL in terms of mean response length and entropy of the actor during training for the Qwen3-4B-Base model. MAXRL generally produces longer chains-of-thought, and also retains higher actor entropy during training.

I.3. Validation Accuracy During Training

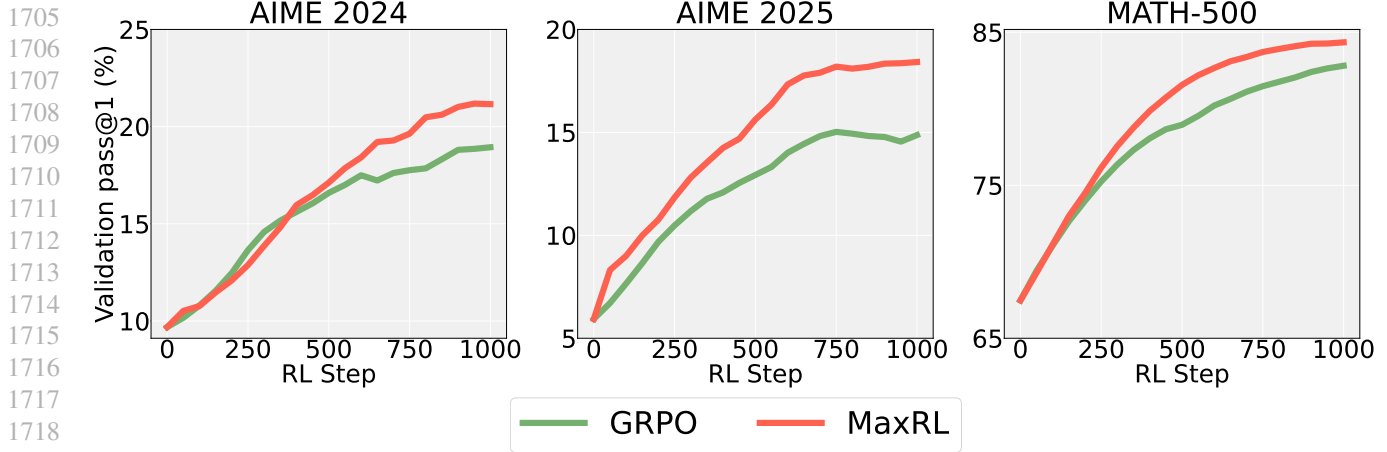


Figure 14. (Qwen3-4B-Base validation pass@1 during training) Pass@1 (estimated using 32 samples) during training of Qwen3-4B-Base, on 3 different evaluation dataset. MAXRL consistently outperform GRPO during training.

J. Additional Results on Qwen3-1.7B-Base

J.1. Validation Accuracy During Training

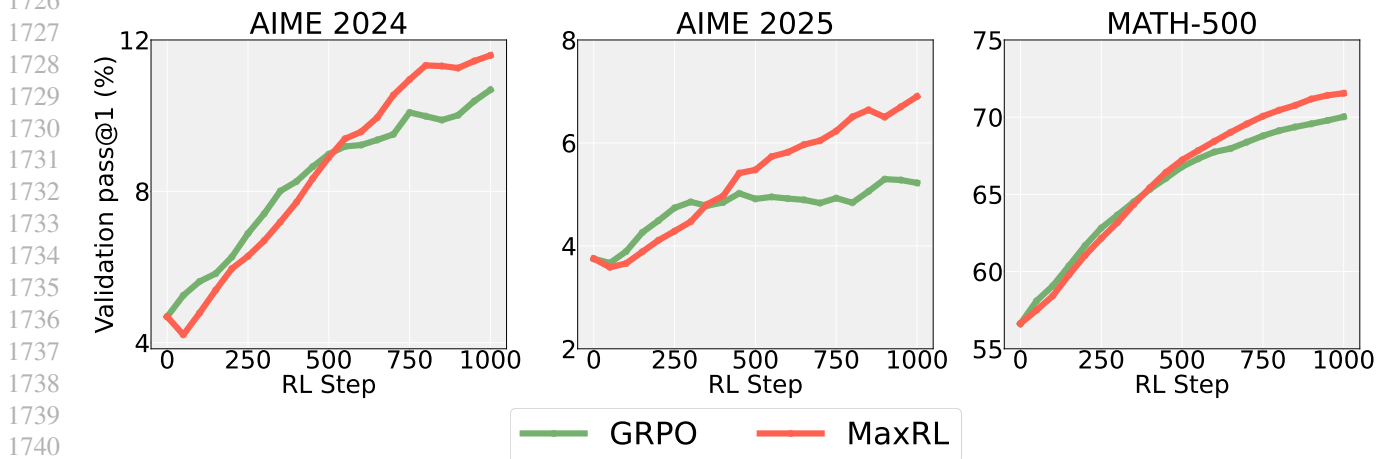


Figure 15. (Qwen3-1.7B-Base validation accuracy during intermediate training) We record validation pass@1 (using mean over 32 rollouts per prompt) over AIME 2024, AIME 2025 and MATH-500 during Qwen3-1.7B-Base model training. Similar to Figure 4, we observe that MAXRL initially trail behind GRPO at pass@1, but catches up with extended training and then converges to a higher value.

J.2. Training Dynamics

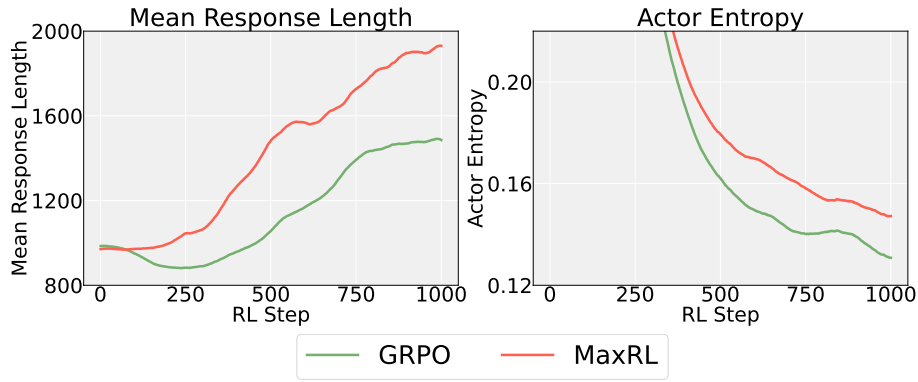


Figure 16. (Additional training dynamics metrics for Qwen3-1.7B-Base) We show comparison between GRPO and MAXRL in terms of mean response length and entropy of the actor during training for the Qwen3-1.7B-Base model. MAXRL generally produces longer chains-of-thought, and also retains higher actor entropy during training.

K. Additional Experimental Results on Maze

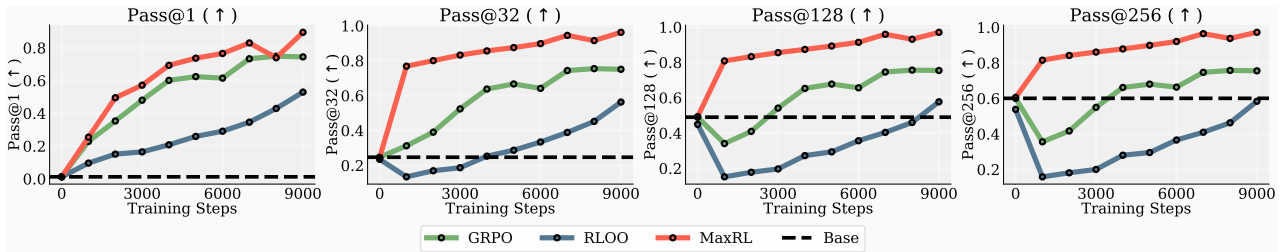


Figure 17. (Infinite training compute in maze experiment) We investigate how different objectives perform when we train a 3M model to solve 17x17 maze puzzles. MAXRL performs significantly better compared with GRPO and REINFORCE in Pass@1, Pass@32, Pass@128 and Pass@256. These results signify MAXRL’s effectiveness in computation scaling during RL.

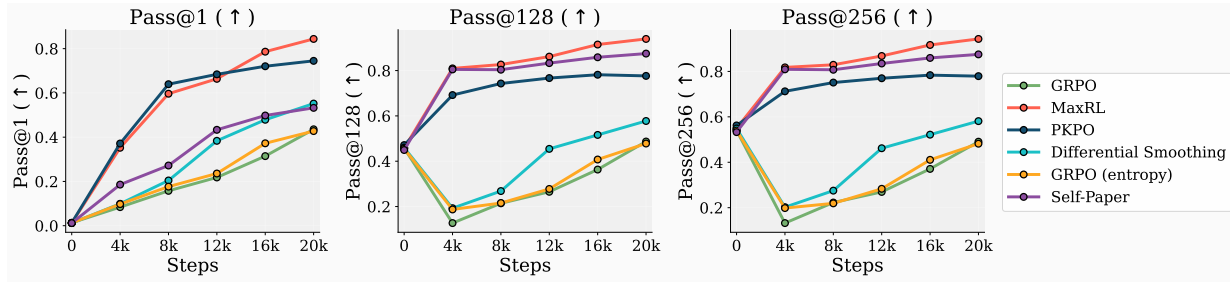


Figure 18. (Training curves compared with other baselines.) We compare MaxRL with other RL algorithms, including entropy regularization, PKPO, Differential Smoothing and SELF. MaxRL significantly outperforms other methods.