UNCALIBRATED REASONING: GRPO INDUCES OVERCONFIDENCE FOR STOCHASTIC OUTCOMES

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning (RL) has proven remarkably effective at improving the accuracy of language models in verifiable and deterministic domains like mathematics. However, it is unclear if current RL methods are similarly effective at optimizing language models for stochastic settings, such as scientific experimentation and model uncertainty estimation. Here, we demonstrate that Group Relative Policy Optimization (GRPO) induces overconfident probability predictions for categorical stochastic outcomes, while Proximal Policy Optimization (PPO) and REINFORCE Leave-One-Out (RLOO) yield well-calibrated models. We show that removing group standard normalization in GRPO fixes its miscalibration and provide a theoretical explanation for why normalization causes overconfidence. Our results provide new evidence against the use of standard normalization in GRPO and help pave the way for applications of RL for reasoning language models beyond deterministic domains.

1 INTRODUCTION

Reinforcement learning (RL) has achieved remarkable success at improving the accuracy of language models in verifiable domains like mathematics and coding (OpenAI, 2024; Shao et al., 2024; Kimi Team, 2025). In particular, recent success has been achieved by optimizing language models to generate chain-of-thought text before responding to a prompt (often called "reasoning") with supervision from a verifier. Current research has focused primarily on domains where proposed answers are deterministically correct or incorrect.

We posit that an important next step for the reasoning RL paradigm is to expand to domains with verifiable yet stochastic answers. For example, scientific experiments, which are subject to random variation, could serve as powerful verifiers for optimizing language models beyond current written knowledge. Scientific reasoning models trained in this manner could support hypothesis generation, experimental design, and decision making through both their predictions and generated reasoning traces. Other potentially impactful settings for training reasoning models with stochastic outcomes include model alignment, which considers human behaviors and preferences (Ziegler et al., 2020; Ouyang et al., 2022), and model uncertainty estimation, which is important for high-stakes decision making and can be framed as modeling the probability that a prediction is correct (Band et al., 2024; Stangel et al., 2025; Damani et al., 2025).

In this paper, we examine whether three popular algorithms for reasoning RL in deterministic domains, namely GRPO (Shao et al., 2024), PPO (Schulman et al., 2017), and RLOO (Kool et al., 2019; Ahmadian et al., 2024), are also effective in settings with binary stochastic outcomes. Through applications to synthetic data, real-world scientific experiments, and medical question-answering, we demonstrate that models trained to predict outcome probabilities with a log-likelihood reward using GRPO make highly overconfident predictions, while models optimized with PPO and RLOO are relatively well calibrated (Fig. 1 and 2). We find that GRPO can be modified for better calibration by removing the group standard normalization term and provide a theoretical justification for why normalization causes overconfidence. In sum, our results provide new evidence against the use of standard normalization in GRPO, highlight the value of unbiasedness as a design principle for policy gradients, and help support future applications of reasoning RL beyond deterministic domains.

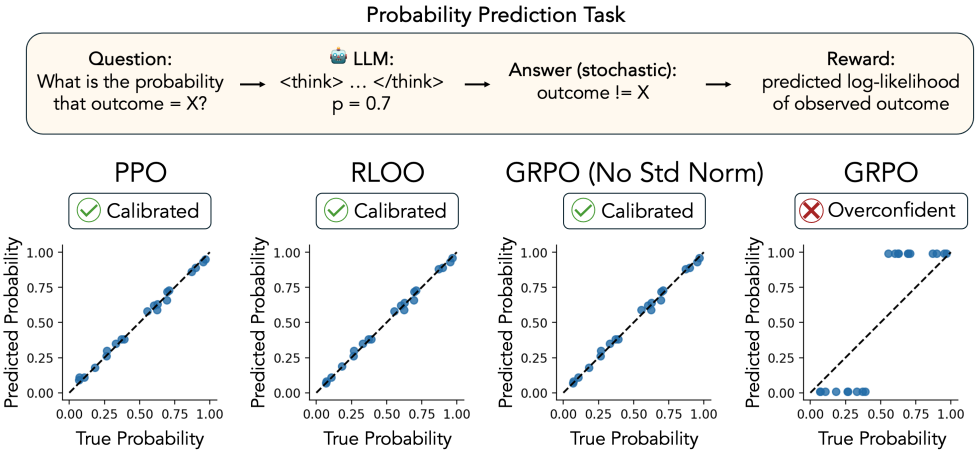

Figure 1: Group standard normalization in GRPO induces overconfident predictions of stochastic outcome probabilities. **Top:** Probability prediction task. **Bottom:** Synthetic data experiment results. Models trained with PPO, RLOO, and GRPO with no standard normalization are well calibrated, while models trained with GRPO are extremely overconfident.

2 PRELIMINARIES

RL with Language Models Reinforcement learning methods cast autoregressive language models as stochastic policies π_θ that specify actions (selecting new tokens) based on the current state (the prompt and prior generated tokens). We consider a setting with outcome supervision, where the goal is to maximize the expected reward received from a verifier that scores the correctness of a response given the ground-truth answer. While current work focuses primarily on settings with deterministic answers, we consider answers that may be stochastic conditional on the prompt.

Value and Advantage Functions The state value function $V^\pi(s)$ is defined as the expected reward from following policy π from state s, and the state-action value function $Q^\pi(s, a)$ is defined as the expected reward of following policy π from state s when the next action is set to be a. The advantage function $A^\pi(s, a) = Q^\pi(s, a) - V^\pi(s)$ is the expected increase in reward from selecting a as the next action from state s relative to an action sampled from π.

Policy Gradients Policy gradient methods optimize policy π_θ by directly estimating the gradient of the expected reward with respect to policy parameters. Let q be a prompt, a be the true answer, $\mathbf{o} = (o_1, ..., o_t)$ be a sequence of response tokens, and $\mathrm{r}(\mathbf{o}, a)$ be the final reward received from the verifier. From the policy gradient theorem (Sutton et al., 1999)

$$\hat{g}^{\mathrm{PG}} = \hat{\mathbb{E}}_{\substack{q\sim p(Q),\ a\sim p(A|q),\\ \mathbf{o}\sim\pi_\theta(O|q)}} \left[\sum_{t=1}^{|\mathbf{o}|} \nabla_\theta \log \pi_\theta(o_t|s_t) \left(\mathrm{r}(\mathbf{o}, a) - b(s_t)\right) \right] \tag{1}$$

is an unbiased estimate of the policy gradient, where $\hat{\mathbb{E}}$ is an empirical sample mean, $s_t := (q, o_{<t})$ is the state at step t (the prompt and prior tokens), and baseline $b(s_t)$ is a function of the current state. A common choice is $b(s_t) = \hat{V}(s_t)$, which makes the baselined reward equivalent to an estimate of the advantage $\hat{A}(s_t, o_t)$. The policy gradient estimator can be interpreted as as increasing the probability of actions with above average expected rewards and decreasing the probability of actions with below average expected rewards.

Each of the three algorithms considered in this paper (GRPO, PPO, and RLOO) are policy gradient methods. We discuss the different strategies these methods take for advantage estimation and deviations from the policy gradient estimator $\hat{g}^{\mathrm{PG}}$ below.

Advantage Estimation for Policy Gradients Consider sampling G responses from a single prompt, and let $\mathbf{r} = (r_1, ..., r_G)$ be the rewards for these responses. Let $\hat{A}_{i,t}$ be the estimated advantage for token t in response i. PPO, RLOO, and GRPO then have the following advantage estimators:

Algorithm	Advantage estimator $\hat{A}_{i,t}$	Unbiased PG?
PPO	$r_i - \hat{V}_\psi(s_{i,t})$	Yes
RLOO	$r_i - \text{mean}(\mathbf{r}_{j \neq i})$	Yes
GRPO	$\frac{r_i - \text{mean}(\mathbf{r})}{\text{std}(\mathbf{r}) + \epsilon}$	No
GRPO (No Std Norm)	$r_i - \text{mean}(\mathbf{r})$	No (proportional)

PPO uses Generalized Advantage Estimation (GAE) (Schulman et al., 2018) and learns an explicit model of the value function $\hat{V}_\psi$ as a baseline (we focus on the unbiased variant of GAE). To avoid the computational costs associated with learning an explicit value model, RLOO and GRPO instead compute a Monte Carlo estimate of the value using multiple responses generated from the same prompt. Specifically, RLOO subtracts the mean reward from the other sampled responses, yielding an unbiased advantage estimate, while GRPO subtracts the mean reward from all responses and divides by the standard deviation, which is biased. We also consider a variant of GRPO without standard normalization which yields a policy gradient estimate that is proportional to an unbiased estimate (this modification was proposed as part of the Dr. GRPO algorithm (Liu et al., 2025)). We note that RLOO and GRPO uses the same advantage estimate for each token, which can be interpreted as casting question answering as a bandit problem where generating the full response corresponds to a single action.

Clipped Policy Gradients The primary contribution of PPO was to introduce a clipped policy gradient estimator to stabilize training when performing multiple gradient updates on a single batch of rollouts (at the cost of introducing bias). The clipped estimator is

$$\hat{g}_t^{\text{clip}} = \nabla_\theta \hat{\mathbb{E}}_{\substack{q \sim p(Q) \\ \mathbf{o} \sim \pi_{\theta_{old}}(O|q)}} \min \left[\frac{\pi_\theta(o_t|q, o_{<t})}{\pi_{\theta_{old}}(o_t|q, o_{<t})} \hat{A}_t, \text{clip} \left(\frac{\pi_\theta(o_t|q, o_{<t})}{\pi_{\theta_{old}}(o_t|q, o_{<t})}, 1 - \epsilon, 1 + \epsilon \right) \hat{A}_t \right]$$

When applied on-policy, $\pi_\theta = \pi_{\theta_{old}}$ and the clipped estimator reduces to the vanilla policy gradient. The clipped policy gradient is also used in GRPO and can be applied with any of the advantage estimators discussed above.

3 Experiments

3.1 Problem Statement

We consider the following **probability prediction task**: for prompt q and categorical answer $a \in \{1, ..., K\}$, predict the probability distribution over possible values for a. Training data consists of question-answer pairs $(q_i, a_i)|_{i=1}^N$, where observed answers a_i are sampled from some unknown probability distribution $a_i \sim p(A|q_i)$. We compare the performance of RL algorithms on this task using the log-likelihood of the observed answer under the model predicted probability as the reward function.

3.2 Metrics

We evaluate model predictions for both calibration and classification performance. To measure calibration, we visualize reliability plots and compute the Expected Calibration Error (ECE). ECE is computed by binning predicted probabilities (we use 10 bins) and computing the average difference between the frequency of positive instances and mean predicted probability in each bin, weighted by the number of points. We measure classification performance with both the Area Under the Receiver Operator Characteristic (AUROC) and accuracy of the maximum likelihood choice.

Dataset	Algorithm	ECE (↓)	AUROC (↑)	Acc. (↑)
Synthetic Data	GRPO	0.239	0.75	0.75
	GRPO (No Std.)	0.002	0.82	0.75
	RLOO	0.002	0.82	0.75
	PPO	0.005	0.82	0.75
CRISPR Screen	GRPO	0.292	0.69	0.67
	GRPO (No Std.)	0.036	0.72	0.68
	RLOO	0.040	0.72	0.68
	PPO	0.038	0.72	0.67
MedMCQA	GRPO	0.117	0.80	0.58
	GRPO (No Std.)	0.013	0.81	0.59
	RLOO	0.009	0.81	0.59
	PPO	0.020	0.80	0.58

Table 1: Evaluation metrics from probability prediction experiments. Across applications to synthetic data and real-world biological experiments, we find that GRPO achieves poor ECE and AUROC relative to GRPO without standard normalization, RLOO, and PPO. All algorithms perform nearly identically on accuracy with predicted probabilities thresholded at 0.5, which does not require well-calibrated predictions.

3.3 Experiment 1: Synthetic Data

We begin by characterizing the behavior of each RL algorithm in a synthetic data experiment with known ground-truth probabilities.

Data We simulate a dataset of 10,000 (q_i, c_i, a_i) triples, representing questions, categories, and binary answers. Questions are randomly assigned to one of 20 random categories. For each category, a true category answer rate is sampled from a uniform distribution: $p_1, ..., p_{20} \sim \text{Uniform}(0, 1)$. Answers are then sampled from the true answer rate for the question category: $a_i|q_i, c_i \sim \text{Bernoulli}(p_{c_i})$.

Model We define a minimal "language model" that enables us to examine the behavior of each RL algorithm in a simplified setting (we verify that these behaviors generalize to real language models in the next two experiments). Specifically, the minimal model samples a single token representing the predicted probability given a question, parameterized as a categorical distribution $p_\theta(a_i = 1|q_i) = p_\theta(a_i = 1|c_i)$ using a learnable parameter for each category / probability token pair. We use a vocabulary of 99 tokens representing probabilities between 0.01 and 0.99. For experiments with PPO, we define a value model that predicts $\hat{V}(q_i) = \psi_{c_i}$, where ψ_c is a learnable parameter for each category.

Optimization We optimize models using PPO, RLOO, GRPO, and GRPO without standard normalization both on-policy and off-policy (1 and 10 gradient updates per rollout, respectively). Off-policy models are optimized with the clipped policy gradient estimator, and we consider clipping thresholds of 0.2 and 0.001 to assess the effects of different clipping rates.

Results Across all settings, we find that GRPO yields highly overconfident probability predictions: models optimized with GRPO converge to predict the minimum available probability for categories with true probability < 0.5 and the maximum available probability for categories with true probability > 0.5 (Fig. 1). In contrast, GRPO without standard normalization, PPO, and RLOO all yield well-calibrated predictions (Fig. 1). These observations are reflected GRPO's poor ECE (0.24 vs <0.01) and AUROC (0.75 vs 0.82) relative to the other algorithms (Tbl. 1). We observe that all considered algorithms perform equivalently on thresholded accuracy, which does not require calibrated predictions. We also obtain nearly identical results when training on-policy and off-policy, even when introducing high clipping rates, which suggests that the clipped policy gradient estimator does not introduce a systematic bias for probability prediction (Appendix Tbl. 2 and Fig. 6).

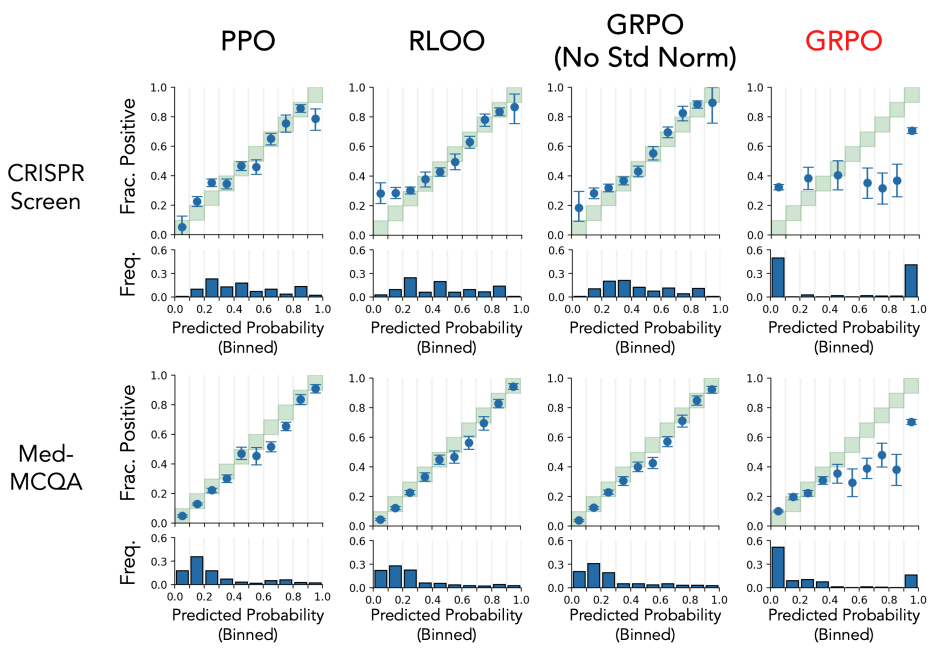

Figure 2: Predictions from the CRISPR screen and MedMCQA experiments after reinforcement learning with Qwen3-4B. In both settings, we observe that models optimized with PPO, RLOO, and GRPO without standard normalization achieve fairly well-calibrated predictions for held-out queries, while models optimized with GRPO make highly overconfident probability predictions. Error bars represent 95% confidence intervals.

3.4 Experiment 2: Scientific Experiment Preidction (CRISPR Screen)

Next, we evaluate if the conclusions from the synthetic data experiments hold when optimizing a large language model, Qwen3-4B (Yang et al., 2025), to predict outcome probabilities in real-world biological experiments.

Data In recent years, Perturb-seq (Dixit et al., 2016) has emerged as a powerful experimental technique for identifying causal effects on cells, a key question in drug discovery, cell engineering, and basic biology research. CRISPR perturb-seq experiments involve perturbing genes with CRISPR and measuring the effect of those perturbations on gene expression counts for each gene in individual cells (which can be interpreted as a broad measurement of cell state). For this experiment, we convert a large perturb-seq dataset from Replogle et al. (2022) into a binary task: for a given perturbed gene and target gene expression phenotype, predict the probability that the perturbed gene has a strong effect on the phenotype (full preprocessing details in Appendix A.5). We sample a balanced dataset of positive and negative instances for the final dataset and generate validation and test splits with held-out perturbations.

Model We optimize Qwen3-4B to predict the probability that a perturbed gene has a strong effect on a target phenotype. The model is prompted to predict the probability as a percentage between 1 and 99 (full prompt in Appendix A.6).

Optimization We optimize models with PPO, RLOO, GRPO, and GRPO without standard normalization using Verl (Sheng et al., 2024). Each algorithm is applied off-policy (8 updates per sampled training batch) with the clipped policy gradient estimator. Models are trained for 16 epochs with train batch size 512 and 4 rollouts per sample (details in Appendix A.7).

Results Consistent with the synthetic data experiment, we find that optimization with GRPO results in highly overconfident probability predictions (ECE=0.29), while GRPO with no standard normalization, PPO, and RLOO yield well-calibrated models (ECE$\leq$0.04, Fig. 2 and Tbl. 1). GRPO again performs poorly on AUROC (0.69 vs 0.72 from the other algorithms) and all models are

similarly accurate. We also find that the clipped policy gradient, which was used for all models in this experiment, did not cause biased probability predictions.

3.5 Experiment 3: Question Answering Uncertainty Estimation (Med-MCQA)

Next, we analyze the performance of RL algorithms under a multi-class uncertainty estimation task with a multiple choice QA dataset.

Data: We use the MedMCQA dataset (Pal et al.), which consists of exam questions pulled from medical entrance exams in India. There are four possible answers provided for each question.

Model and Optimization: We optimize Qwen3-4B to predict the probability that each multiple-choice option is correct using GRPO, GRPO without standard normalization, RLOO, and PPO (full prompt and experiment details in Appendix A.8 and A.9).

Results: Consistent with the prior experiments, we find that GRPO yields poorly calibrated and overconfident probability predictions in the multi-class uncertainty estimation setting (ECE=0.117) while the other algorithms are relatively well calibrated (ECE ≤ 0.02) (Tbl. 1, Fig. 2).

4 Theoretical Analysis

Finally, we analyze why standard normalization in GRPO induces overconfident predictions. Recall that GRPO reinforces actions based on their estimated advantage: actions that have large advantages are made more likely, while actions with negative advantages are made less likely. We will show that standard normalization causes GRPO to overestimate the advantage of overconfident predictions, resulting in overconfident policies (Fig. 3).

In Appendix A.1, we derive expressions for the expected advantage estimates from GRPO with and without standard normalization. Let q be a prompt with stochastic answers $a \sim \text{Categorical}(\mathbf{p})$, where $\mathbf{p} = (p_1, ..., p_k)$ are the true answer probabilities. Let $\hat{\mathbf{p}} = (\hat{p}_1, ..., \hat{p}_k)$ be the predicted answer probabilities, and let $\mathrm{r}(\hat{\mathbf{p}}, a)$ be a reward function such as the log-likelihood reward $\mathrm{r}(\hat{\mathbf{p}}, a) = \sum_{i=1}^{k} \mathbf{1}[a = i] \log \hat{p}_i$. The true advantage for prediction $\hat{\mathbf{p}}$ is then

$$A(q, \hat{\mathbf{p}}) = \sum_{i=1}^{k} p_i(\mathrm{r}(\hat{\mathbf{p}}, i) - \mu_i)$$

where $\mu_i = \mathbb{E}_{\hat{\mathbf{p}}' \sim \pi_\theta(q)} [\mathrm{r}(\hat{\mathbf{p}}', i)]$ is the expected reward under predictions sampled from the policy if the answer is i. We show that the expected advantage estimate for GRPO without standard normalization is

$$\mathbb{E}\left[\hat{A}^{\text{NO-STD}}(q, \hat{\mathbf{p}})\right] = \frac{G-1}{G} A(q, \hat{\mathbf{p}}) \propto A(q, \hat{\mathbf{p}})$$

This means that the policy gradients using GRPO without standard normalization are approximately unbiased (up to a constant factor), consistent with the calibrated predictions we observed experimentally. In contrast, the advantage estimate for GRPO is approximately

$$\mathbb{E}\left[\hat{A}^{\text{STD}}(q, \hat{\mathbf{p}})\right] \approx \sum_{i=1}^{k} \frac{1}{\sigma_i + \epsilon} p_i(\mathrm{r}(\hat{\mathbf{p}}, i)) - \mu_i)$$

where $\sigma_i = \mathbb{E}_{\hat{\mathbf{p}}^{(1)}, ..., \hat{\mathbf{p}}^{(G)}} \left[\text{std}(\mathrm{r}(\hat{\mathbf{p}}^{(1)}, i), ..., \mathrm{r}(\hat{\mathbf{p}}^{(G)}, i))\right]$ is the expected standard deviation of the G group rewards sampled from the prompt if the true answer is i. We observe that the approximate GRPO advantage expression closely resembles the true advantage with the addition of $\frac{1}{\sigma_i + \epsilon}$ coefficients, which introduce a policy-dependent bias that we analyze empirically.

In Fig. 3, we visualize empirical estimates of the expected advantage for GRPO with and without standard normalization with binary answers and a log-likelihood reward (estimation details in A.2). The predicted probability on the x-axis is defined as the predicted probability that $a = 1$. Under a uniform policy, the advantage estimates from both methods closely approximate the true advantage

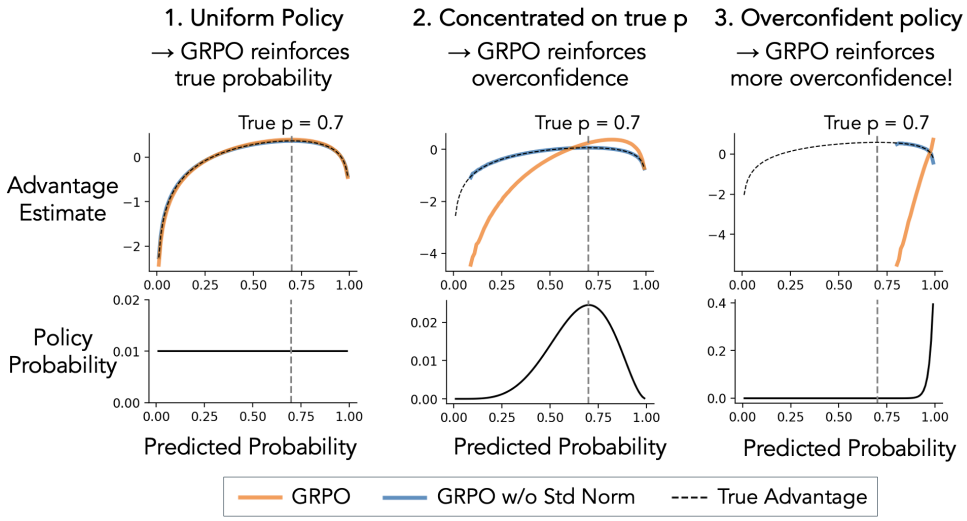

Figure 3: Bias in GRPO advantage estimates explains overconfident predictions. Advantages are computed with a log-likelihood reward. **Left:** Under a uniform policy, both GRPO and GRPO without standard normalization closely approximate the true advantages. **Middle:** Under a policy concentrated on the true probability, GRPO overestimates the advantage of overconfident predictions. **Right:** As the policy becomes increasingly overconfident, GRPO increasingly overestimates the advantage of more overconfident predictions. This pattern creates a positive feedback loop towards increasingly overconfident predictions consistent with our experimental observations.

(left column). As the policy begins to concentrate around the true probability, we observe that GRPO starts to overestimate the advantage of overconfident predictions, while the unnormalized estimates remains accurate (center column). This will cause GRPO to reinforce overconfident predictions more strongly than the true probability, resulting in overconfident policies. Finally, we observe that under a very overconfident policy, GRPO's advantage estimates will have an even more extreme bias towards overconfident predictions, while GRPO without standard normalization remains approximately unbiased (right column). These observations are consistent with our approximate GRPO advantage expression: as the policy concentrates above 0.5, σ_0 becomes larger than σ_1, resulting in a reduced weight on the penalty for overconfident predictions (Appendix Fig. 5).

To summarize, group standard normalization in GRPO's advantage estimates creates a policy-dependent bias that pushes policies towards overconfident predictions. While our analysis focused on a log-likelihood reward, we also consider rewards based on other strictly proper scoring rules in Appendix A.3.

5 Discussion

Many important tasks, from scientific experimentation to uncertainty estimation, require reasoning about the likelihood of stochastic outcomes. We showed that reasoning language models optimized to predict the probability of binary stochastic outcomes from samples with GRPO are highly overconfident, while models optimized with PPO and RLOO are well calibrated (all using a log-likelihood reward). We identified a bias in GRPO's advantage estimate due to group standard normalization as the relevant difference between these algorithms and provided a theoretical explanation for why normalization causes overconfidence. We also found that using the clipped policy gradient introduced by PPO did not impact calibration in our experiments.

Our results fit into a broader set of findings that biased policy gradients can lead to unexpected behavior for reasoning language models. For example, Liu et al. (2025) introduce Dr. GRPO, a modification of GRPO designed to eliminate terms that introduce bias. They propose to remove length normalization, which they find biases models to longer outputs, and to remove group standard

normalization, which they interpret as a question-level difficulty bias. Our work identifies a novel negative impact of standard normalization in GRPO and supports unbiasedness as a useful design principle for policy gradient methods in reasoning RL.

We note that there are other possible framings of the outcome probability task explored in this paper. For example, one could directly estimate the probability of stochastic outcomes and train models to accurately predict these continuous values. While summarizing uncertainty can be useful, this approach requires having robust probability estimates ahead of time, which may be unavailable or model dependent, and limits the opportunity for the reasoning model to learn to make more precise estimates. Alternatively, one can train only on deterministic tasks and hope for transfer to stochastic settings (for example, we observe better than random zero-shot predictions on the CRISPR task in Appendix Fig. 8), though this limits the available data and tasks for training models. Overall, we believe that modeling stochastic outcomes from observed samples is an important capability for reasoning RL and that it is useful to characterize algorithms for this setting.

Finally, we presented an initial application of RL to train reasoning models directly from noisy biological experiments. While we found that RL can yield calibrated predictions for held-out experiments, these predictions do not necessarily reflect rigorous reasoning about uncertainty in the model's chain-of-thought. The development of new methods to train models to reason rigorously about uncertainty in science is an exciting future direction.

6 Reproducibility Statement

The code and data required to reproduce experiments and figures are provided in supplementary materials.

References

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to Basics: Revisiting REINFORCE Style Optimization for Learning from Human Feedback in LLMs, February 2024.

Neil Band, Xuechen Li, Tengyu Ma, and Tatsunori Hashimoto. Linguistic Calibration of Long-Form Generations, June 2024.

Mehul Damani, Isha Puri, Stewart Slocum, Idan Shenfeld, Leshem Choshen, Yoon Kim, and Jacob Andreas. Beyond Binary Rewards: Training LMs to Reason About Their Uncertainty, July 2025.

Atray Dixit, Oren Parnas, Biyu Li, Jenny Chen, Charles P. Fulco, Livnat Jerby-Arnon, Nemanja D. Marjanovic, Danielle Dionne, Tyler Burks, Raktima Raychowdhury, Britt Adamson, Thomas M. Norman, Eric S. Lander, Jonathan S. Weissman, Nir Friedman, and Aviv Regev. Perturb-Seq: Dissecting Molecular Circuits with Scalable Single-Cell RNA Profiling of Pooled Genetic Screens. *Cell*, 167(7):1853–1866.e17, December 2016. ISSN 1097-4172. doi: 10.1016/j.cell.2016.11.038.

Kimi Team. Kimi k1.5: Scaling Reinforcement Learning with LLMs, June 2025.

Wouter Kool, Herke van Hoof, and Max Welling. Buy 4 REINFORCE Samples, Get a Baseline for Free! April 2019.

Dylan Kotliar, Adrian Veres, M Aurel Nagy, Shervin Tabrizi, Eran Hodis, Douglas A Melton, and Pardis C Sabeti. Identifying gene expression programs of cell-type identity and cellular activity with single-cell rna-seq. *eLife*, 8:e43803, jul 2019. ISSN 2050-084X. doi: 10.7554/eLife.43803. URL https://doi.org/10.7554/eLife.43803.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding R1-Zero-Like Training: A Critical Perspective, March 2025.

OpenAI. Learning to reason with LLMs. https://openai.com/index/learning-to-reason-with-llms/, September 2024.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, March 2022.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. MedMCQA: A Large-scale Multi-Subject Multi-Choice Dataset for Medical domain Question Answering. In *Proceedings of the Conference on Health, Inference, and Learning*, pp. 248–260. PMLR. URL https://proceedings.mlr.press/v174/pal22a.html.

Joseph M. Replogle, Reuben A. Saunders, Angela N. Pogson, Jeffrey A. Hussmann, Alexander Lenail, Alina Guna, Lauren Mascibroda, Eric J. Wagner, Karen Adelman, Gila Lithwick-Yanai, Nika Iremadze, Florian Oberstrass, Doron Lipson, Jessica L. Bonnar, Marco Jost, Thomas M. Norman, and Jonathan S. Weissman. Mapping information-rich genotype-phenotype landscapes with genome-scale perturb-seq. *Cell*, 185(14):2559–2575.e28, 2022. ISSN 0092-8674. doi: https://doi.org/10.1016/j.cell.2022.05.013. URL https://www.sciencedirect.com/science/article/pii/S0092867422005979.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms, August 2017.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-Dimensional Continuous Control Using Generalized Advantage Estimation, October 2018.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models, April 2024.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024.

Paul Stangel, David Bani-Harouni, Chantal Pellegrini, Ege Özsoy, Kamilia Zaripova, Matthias Keicher, and Nassir Navab. Rewarding Doubt: A Reinforcement Learning Approach to Calibrated Confidence Expression of Large Language Models, May 2025.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy Gradient Methods for Reinforcement Learning with Function Approximation. In *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 Technical Report, May 2025. URL http://arxiv.org/abs/2505.09388.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-Tuning Language Models from Human Preferences, January 2020.

A Appendix

A.1 Analysis of Bias in GRPO Advantage Estimates

Let q be a prompt with stochastic answers $a \sim \text{Categorical}(\mathbf{p})$, where $\mathbf{p} = (p_1, ..., p_k)$ are the true answer probabilities. Let $\hat{\mathbf{p}} = (\hat{p}_1, ..., \hat{p}_k)$ be the predicted answer probabilities from policy π_θ. Let $\mathrm{r}(\hat{\mathbf{p}}, a)$ be a reward function from a proper scoring rule such as the log-likelihood reward $\mathrm{r}(\hat{p}, a) = \sum_{i=1}^{k} \mathbf{1}[a = i] \log \hat{p}_i$ or Brier score $\mathrm{r}(\hat{\mathbf{p}}, a) = \sum_{i=1}^{k} (\mathbf{1}[a = i] - \hat{p}_i)^2$. Proper scoring rules have the property that the expected value is maximized by the true probability and have been shown to be effective rewards for training calibrated classifiers (Band et al., 2024).

The true advantage estimate for prompt q and prediction $\hat{p}$ is

$$\begin{aligned} A(q, \hat{\mathbf{p}}) &= Q^\pi(q, \hat{\mathbf{p}}) - V^\pi(q) \\ &= \mathbb{E}_{a \sim p(A|q)}[\mathrm{r}(\hat{\mathbf{p}}, a)] - \mathbb{E}_{a \sim p(A|q), \hat{\mathbf{p}}' \sim \pi_\theta(q)}[\mathrm{r}(\hat{\mathbf{p}}', a)] \\ &= \sum_{i=1}^{k} p_i \mathrm{r}(\hat{\mathbf{p}}, i) - \mathbb{E}_{\hat{\mathbf{p}}' \sim \pi_\theta(q)} \sum_{i=1}^{k} p_i \mathrm{r}(\hat{\mathbf{p}}', i) \\ &= \sum_{i=1}^{k} p_i (\mathrm{r}(\hat{\mathbf{p}}, i) - \mu_i) \end{aligned}$$

where $\mu_i = \mathbb{E}_{\hat{\mathbf{p}}' \sim \pi_\theta(q)} [\mathrm{r}(\hat{\mathbf{p}}', i)]$ is the expected reward under predictions sampled from the policy if the answer is i.

Next, we compare the advantage estimates from GRPO (Shao et al., 2024) to the true advantage to characterize any biases. Let $\hat{\mathbf{p}}^{(1)}, ..., \hat{\mathbf{p}}^{(G)} \sim \pi_\theta(q)$ be a group of G predictions sampled from the same prompt. Without loss of generality, we will set the index of the prediction whose advantage we are estimating to be 1. We see that the expected advantage for GRPO without standard normalization is

$$\begin{aligned} \mathbb{E}_{\substack{a \sim p(A|q), \\ \hat{\mathbf{p}}^{(2)}, ..., \hat{\mathbf{p}}^{(G)} \sim \pi_\theta(q)}} \left[\hat{A}^{\text{NO-STD}}(q, \hat{\mathbf{p}}) \right] &= \mathbb{E}_{a, \hat{\mathbf{p}}^{(2)}, ..., \hat{\mathbf{p}}^{(G)}} \left[\mathrm{r}(\hat{\mathbf{p}}^{(1)}, a) - \text{mean}(\mathrm{r}(\hat{\mathbf{p}}^{(1)}, a), ..., \mathrm{r}(\hat{\mathbf{p}}^{(G)}, a)) \right] \\ &= \mathbb{E}_{a, \hat{\mathbf{p}}^{(2)}, ..., \hat{\mathbf{p}}^{(G)}} \left[\mathrm{r}(\hat{\mathbf{p}}^{(1)}, a) - \frac{1}{G} \left(\mathrm{r}(\hat{\mathbf{p}}^{(1)}, a) + \sum_{j=2}^{G} \mathrm{r}(\hat{\mathbf{p}}^{(j)}, a) \right) \right] \\ &= \mathbb{E}_a \left[\left(\mathrm{r}(\hat{\mathbf{p}}^{(1)}, a) - \frac{1}{G} \mathrm{r}(\hat{\mathbf{p}}^{(1)}, a) \right) - \frac{G-1}{G} \mathbb{E}_{\hat{\mathbf{p}}' \sim \pi_\theta(q)} \mathrm{r}(\hat{\mathbf{p}}', a) \right] \\ &= \frac{G-1}{G} \mathbb{E}_a \left[\mathrm{r}(\hat{\mathbf{p}}^{(1)}, a) - \mathbb{E}_{\hat{\mathbf{p}}'}[\mathrm{r}(\hat{\mathbf{p}}', a)] \right] \\ &= \frac{G-1}{G} A(q, \hat{\mathbf{p}}) \end{aligned}$$

We see that the estimate is proportional to the true advantage, though it is attenuated by a factor of $\frac{1}{G}$. A fully unbiased estimate can be achieved with the advantage from RLOO (Kool et al., 2019), which excludes $\hat{p}_i$ from the mean baseline.

Finally, we consider the expected GRPO advantage estimate with standard normalization. We define $\sigma_i = \mathbb{E}_{\hat{\mathbf{p}}^{(1)}, ..., \hat{\mathbf{p}}^{(G)}} \left[\text{std}(\mathrm{r}(\hat{\mathbf{p}}^{(1)}, i), ..., \mathrm{r}(\hat{\mathbf{p}}^{(G)}, i)) \right]$ as the expected standard deviation of the G group rewards sampled from the prompt if the true answer is i. We make the following simplifying assumptions in our approximation of the advantage: we assume that G is large so that $\mathbb{E}_{\hat{\mathbf{p}}^{(2)}, ..., \hat{\mathbf{p}}^{(G)} \sim \pi_\theta(q)} \left[\text{std}(\mathrm{r}(\hat{\mathbf{p}}^{(1)}, i)) \right] \approx \sigma_i$ and ignore the dependency between the mean and standard deviation of group rewards. With these simplifications, we have:

$$\mathbb{E}_{\substack{a\sim p(A|q),\\ \hat{\mathbf{p}}^{(2)},\ldots,\hat{\mathbf{p}}^{(G)}\sim\pi_\theta(q)}}\left[\hat{A}^{\text{STD}}(q,\hat{\mathbf{p}})\right] = \mathbb{E}_{a,\hat{\mathbf{p}}^{(2)},\ldots,\hat{\mathbf{p}}^{(G)}}\left[\frac{\mathrm{r}(\hat{\mathbf{p}}^{(1)},a) - \text{mean}\left(\mathrm{r}(\hat{\mathbf{p}}^{(1)},a),\ldots,\mathrm{r}(\hat{\mathbf{p}}^{(G)},a)\right)}{\text{std}\left(\mathrm{r}(\hat{\mathbf{p}}^{(1)},a),\ldots,\mathrm{r}(\hat{\mathbf{p}}^{(G)},a)\right)+\epsilon}\right]$$
$$\approx p_1\frac{\mathrm{r}(\hat{\mathbf{p}}^{(1)},1)-\mu_1}{\sigma_1+\epsilon} + \ldots + p_k\frac{\mathrm{r}(\hat{\mathbf{p}},k)-\mu_k}{\sigma_k+\epsilon}$$
$$= \sum_{i=1}^{k}\frac{1}{\sigma_i+\epsilon}p_i(\mathrm{r}(\hat{\mathbf{p}},i)) - \mu_i)$$

We see that the approximate expected advantage estimate from GRPO has the same weighted reward terms as the true advantage with the addition of new $\frac{1}{\sigma_i+\epsilon}$ coefficients. These coefficients make the GRPO advantage estimate biased in a policy-dependent manner, which is analyzed in Figures 3, 4, and 5.

A.2 Advantage Empirical Estimate Details

We compute the empirical GRPO advantage estimates in Fig. 3 with binary outcomes for a log-likelihood reward using group size $G = 1000$, true probability $p = 0.7$, and 100,000 samples of $(\hat{p}_1,\ldots,\hat{p}_G) \sim \pi_\theta$. Policies are categorical distributions over predicted probabilities $(0.01, 0.02, \ldots, 0.99)$, where categorical log probs are set by discretizing Beta distributions (Beta$(1,1)$, Beta$(5.7,3)$, Beta$(50,1)$). Empirical advantage estimates are plotted for predictions with at least 1,000 observed samples. True advantages are computed exactly.

A.3 GRPO Bias with Other Rewards

While our analysis in the main text focused on a log-likelihood reward, prior work has found that optimization based on other proper scoring rules (which are maximized in expectation by the true probability) can yield well-calibrated classifiers (Band et al., 2024). We show a similar pattern of GRPO advantage estimate biases with binary outcomes using a reward based on the Brier score $\mathrm{r}(\hat{\mathrm{p}},\mathrm{a}) = -(a-\hat{p})^2$ in Fig. 4. We hypothesize that GRPO will yield overconfident predictions for rewards based on strictly proper scoring rules more generally because they are strictly concave, which should lead to similar changes in σ_1 and σ_2 as the policy changes, but we leave a formal characterization as out of scope for this paper.

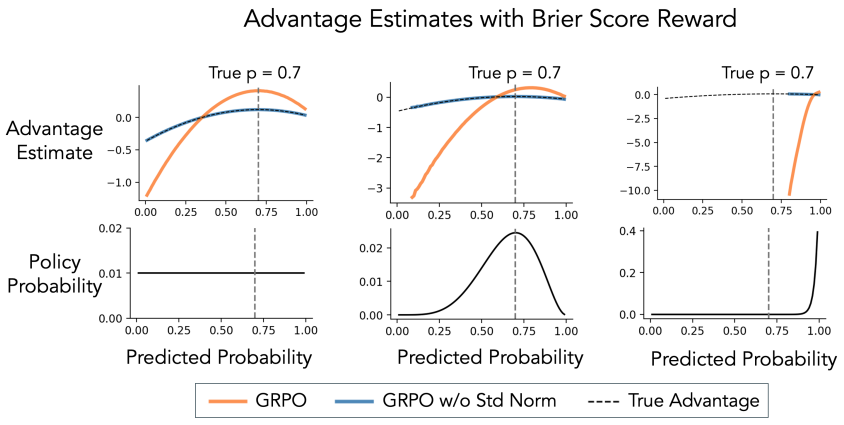

Figure 4: Analysis of advantage estimates with a reward based on the Brier score. We observe a similar pattern of overestimated advantages for overconfident probabilities as observed with a log-likelihood in Fig. 3

.

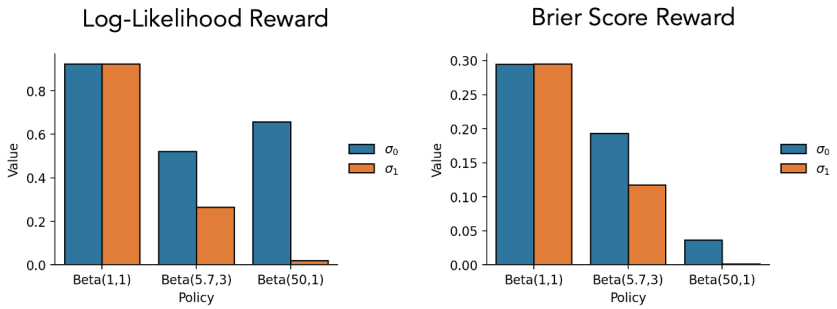

Figure 5: Empirical estimates of σ_0 and σ_1 (standard deviation of rewards within groups for answers 0 and 1) for the three policies in Figures 3 and 4. As the policies concentrate on predictions greater than 0.5, σ_0 becomes larger than σ_1.

A.4 Synthetic Data Experiment Extended Results

Algorithm	Grad Steps / Rollout	ϵ_{clip}	ECE	AUROC	Accuracy
GRPO	1	NA	0.239	0.750	0.751
GRPO	10	0.200	0.239	0.751	0.751
GRPO	10	0.001	0.239	0.751	0.751
GRPO (No Std)	1	NA	0.002	0.823	0.751
GRPO (No Std)	10	0.200	0.005	0.823	0.751
GRPO (No Std)	10	0.001	0.005	0.823	0.751
PPO	1	NA	0.005	0.823	0.751
PPO	10	0.200	0.008	0.823	0.751
PPO	10	0.001	0.008	0.823	0.751
RLOO	1	NA	0.002	0.823	0.751
RLOO	10	0.200	0.004	0.823	0.751
RLOO	10	0.001	0.004	0.823	0.751

Table 2: Extended results from synthetic data experiments. We observe that results are consistent between experiments with a single update per rollout and multiple updates per rollout with a clipped policy gradient estimates, even with low clipping thresholds that encourage high clipping rates.

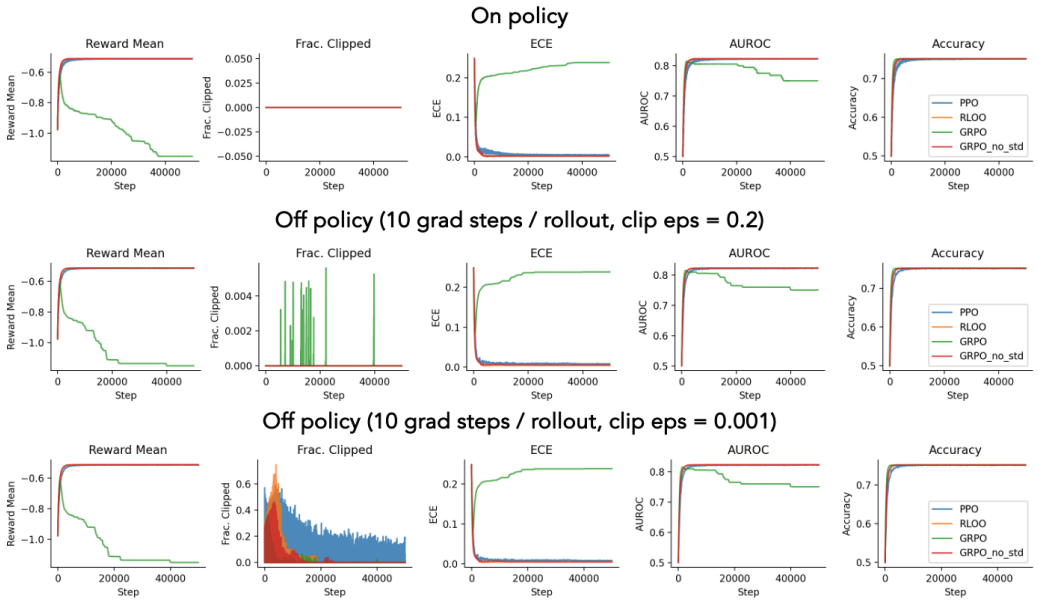

Figure 6: Synthetic data training metrics.

A.5 CRISPR Experiment Data Processing

CRISPR perturb-seq screens involve perturbing individual genes with CRISPR (which modulates the expression of a target gene) and measuring the effect of the perturbation on RNA transcript counts for all genes in individual cells cell. We use the essential gene CRISPRi (CRISPR interference) perturb-seq screen in K562 cells from Replogle et al. (2022) for our experiment. The dataset contains CRISPRi perturbations, which lower gene expression, that target approximately 2,000 unique genes. We apply consensus non-negative matrix factorization (cNMF) (Kotliar et al., 2019) to infer 50 aggregate transcriptional target phenotypes and select the top 15 marker genes for each phenotype as defined by the cNMF method to describe each phenotype. We estimate the effect size of each perturbation on each phenotype as the difference in mean phenotype values for cells that received the perturbation and control cells. To define perturbations with strong effects ("hits"), we fit a cluster model on the perturbation effect sizes for each phenotype and select perturbations that are highly unlikely in the control cluster. Specifically, we fit a Gaussian Mixture Model on the effect sizes for each phenotype (number of clusters between 1-4, selected based on Bayesian Information Criterion) and select perturbations with $<1\%$ chance under the cluster closest to zero as strong effects. To construct a balanced dataset, we select an equal number of perturbations that are most likely under the control cluster as non-hits. We note that the dataset is naturally very imbalanced (hits are relatively rare for most phenotypes) but choose to work with a balanced dataset for simplicity as our primary focus is understanding the behavior of RL algorithms with stochastic outcomes.

A.6 CRISPR Task Prompt

Experiment Prediction Prompt

```
I am planning a perturb-seq screen and plan to assess effects of
↪  perturbations on a phenotype with the following marker genes:
↪  ```{pheno_markers}```.

How likely is a CRISPRi perturbation applied to {pert} to have a
↪  strong effect on this phenotype? Respond with probability from
↪  1-99, representing 1% to 99% chance of a strong effect. Enclose
↪  your answer in <answer> </answer> tags.
```

`pheno_markers` is a list of 15 marker genes for the phenotype, and `pert` is the gene perturbed by the CRISPR perturbation. We also considered prompts that specified the overall frequency of hits in the dataset, but found that this reduced the zero-shot model performance.

A.7 CRISPR Experiment Details

Models were trained with a log-likelihood reward, with a minimum reward of $\log 0.01$ for outputs that do not match the required format (corresponding to the worst possible reward given the prediction range of 0.01-0.99). Each model was trained with batch size 512, group size 4, max response length 2048, mini-batch size (batches for gradient updates within each rollout) of 64, learning rate 1e-6, and KL loss coefficient of 0.001. For PPO, the critic is trained with mini-batch size 64 and learning rate 1e-5. We train all models without length normalization as discussed in Liu et al. (2025) to avoid a length bias. Models were trained for 16 epochs with Verl (Sheng et al., 2024). For PPO, RLOO, and GRPO with no standard normalization, we select the checkpoint with the best validation reward for evaluation (epoch 15 / step 180 for all three). We use the same checkpoing from the GRPO run for consistency (validation reward begins dropping early and we want to understand what predictions it converges to) (Fig. 7). We generate 4 samples per prompt for test set evaluation and drop samples with no valid prediction (at most one sample of 5608 predictions for each trained models).

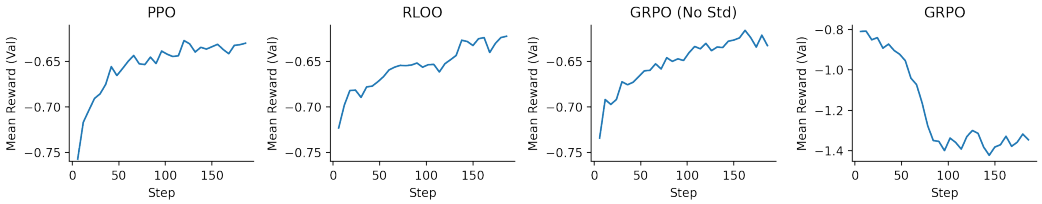

Figure 7: CRISPR experiment prediction task validation set rewards during training.

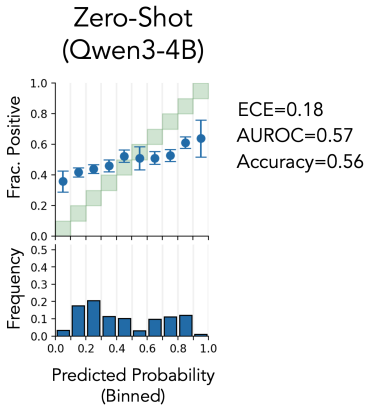

Figure 8: Zero shot predictions on CRISPR task test set with Qwen3-4B.

A.8 MedMCQA Task Prompt

Experiment Prediction Prompt

```
Predict the probability that each answer to the following multiple
↪  choice question is correct. Reason carefully about the
↪  probability of each answer, and make sure to be preciseif
↪  possible, predict the exact probability.
```

```
Formatting: Enclose your answer with <answer> <\answer> tags. Write
↪  each probability with up to two decimal places. None of your
↪  predictions can have value exactly equal to 0 or 1. Format your
↪  answer as a comma separated list for the probability of options
↪  A,B,C,D. For example, a valid answer is
↪  `<answer>0.28,0.61,0.04,0.07</answer>`.

Reminders: Your probabilities must sum to 1, and 0.00 and 1.00 are
↪  not valid responses!

Question: {question}

A: {opa}
B: {opb}
C: {opc}
D: {opd}
```

A.9 MEDMCQA EXPERIMENT DETAILS

Models were trained with a log-likelihood reward, with a minimum reward of $\log 0.001$ for outputs that do not match the required format. We reuse the training setup from the CRISPR experiment A.7, with the changes of using a smaller batch size (256) in order to reduce the computational cost. Each model was trained for 204 steps before evaluation on the validation set.

B LLM USAGE

The most prominent usage of LLM's in writing this paper was to search for related literature (request ChatGPT to find papers related to specific topics). ChatGPT was also used to as a general purpose search and question-answering tool, for example to for latex formatting and to refine mathematical notation.