BEYOND THE STABILITY-EXPLORATION DILEMMA: ENVIRONMENTAL REGULARIZATION FOR LLM POLICY OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Policy optimization (PO) has advanced Large Language Models (LLMs), yet training remains constrained by a stability–exploration trade-off. We analyze the coupling between the input environment and the policy in LLM RL, and *decouple* parameter regularization from the optimization objective by moving regularization to the input side. Concretely, we propose **Environment-Regularized Policy Optimization (ERPO)**, instantiated with **Query-KL (QKL)**, which penalizes the KL divergence between the evolving *query* distribution and a fixed reference. By regularizing the input (query) distribution rather than the action (response) distribution, QKL indirectly controls policy drift induced by environmental shift while preserving exploration. To avoid premature convergence, we introduce a *query-weighted advantage* that reweights updates according to estimated query prevalence, reducing estimator variance and improving robustness. Across diverse mathematical reasoning benchmarks, ERPO achieves KL control comparable to methods with explicit policy regularization, while delivering stronger final performance and smoother training dynamics. Temperature-swept sampling further indicates more stable long-horizon behavior. These results suggest that making the input environment a first-class object—via QKL and query-weighted advantage—is a principled and practical route to improve the stability–exploration trade-off in PO for LLMs.

1 INTRODUCTION

Background and challenge. Policy optimization (PO) methods have become the de facto recipe for post-training large language models (LLMs), spanning trust-region style updates (TRPO/PPO) and preference-based objectives (DPO) together with broader RLHF/RLAIF variants (Schulman et al., 2015c; 2017b; Ouyang et al., 2022b; Bai et al., 2022; Rafailov et al., 2023a). Despite impressive progress in mathematical reasoning and beyond, practitioners still face a persistent dilemma: *how to trade off training stability against effective exploration*. In long-horizon runs, optimization noise and distribution shift tend to accumulate, leading to oscillations and occasional collapses.

Instability from the input side. We argue that a key—and under-controlled—source of instability is *environment non-stationarity* induced by the **query distribution**. During RL fine-tuning, the inputs used for training are sampled from a mechanism that *co-evolves* with the policy (e.g., active data selection, prompt generators, curriculum schedulers). As the policy changes, the conditional likelihood of future prompts also shifts, altering the effective training environment and amplifying gradient variance. This mirrors classic RL settings in which either the initial-state distribution or the transition kernel drifts over time; non-stationary and robust RL therefore advocate explicit distributional control (Padakandla, 2021; Iyengar, 2005; Nilim & El Ghaoui, 2005). A related lesson from imitation learning is that policy updates induce covariate (state) shift, motivating interactive data aggregation such as DAgger/AggreVaTe (Ross et al., 2011; Ross & Bagnell, 2014).

Limitations of action-only regularization. Recent LLM work has started to surface the role of prompt distributions. EVA frames open-ended alignment as a two-player game in which a creator evolves the prompt distribution while a solver learns on it, implicitly regularizing prompt shift;

Align-Pro gives a principled objective to optimize a prompter distribution with explicit KL terms (Ye et al., 2024; Trivedi et al., 2025). Complementary strands stabilize optimization or reweight data from the policy side (e.g., StablePrompt, WPO), yet they do not directly constrain the environmental statistics over queries (Kwon et al., 2024; Zhou et al., 2024). In contrast, mainstream PO for RLHF focuses on action regularization via a Policy-KL budget to an SFT reference (Schulman et al., 2015c; 2017b; Ouyang et al., 2022b), leaving the input/query process comparatively unconstrained. Empirically, even under a fixed Policy-KL budget, the input environment keeps drifting: the batch-estimated Query-KL rises steadily throughout training while the Policy-KL on responses remains nearly flat (Figure 1). This demonstrates that constraining only the action distribution fails to stabilize the input/query process, leaving environment non-stationarity unaddressed.

In this paper. We treat queries as part of the environment and make environment statistics a first-class object in the training objective. We introduce **Query-KL regularization (QKL)**, a plug-in penalty on the divergence between the current empirical query sampler and a chosen reference sampler, explicitly limiting inter-round drift of the training environment while leaving the action space free to explore. In parallel, we propose a lightweight query reweighting scheme that reduces estimator variance and improves robustness under high-temperature decoding—where LLMs are especially sensitive to the long tail of decoding distributions (Holtzman et al., 2020; Wang et al., 2023). Both components are model- and optimizer-agnostic and drop into PPO/DPO-style implementations with minimal changes. Figure 2 sketches ERPO: on top of GRPO we replace the usual Policy-KL with a pre-computed Query-KL, and during advantage computation we weight the within-query samples by the query's occurrence probability, yielding an environment-aware update while preserving action-side exploration.

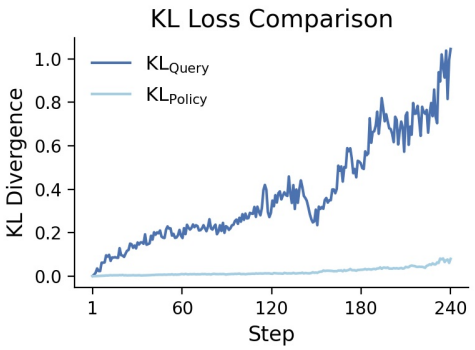

Figure 1: KL losses during GRPO training. The Query-KL (dark) rises while the Policy-KL (light) stays low, showing action-only KL does not stabilize the query process.

Contributions. We make four main contributions. (1) **Query-environment control:** We treat queries as part of the environment and stabilize training by combining *Query-KL (QKL)* to bound query drift with batch self-normalized query weights to reduce variance and tame high-temperature behavior. (2) **Drop-in practicality:** The method is optimizer-agnostic and adds only a QKL term plus per-batch reweighting on top of GRPO/PPO-style pipelines with minimal changes. (3) **Stability evaluation:** We assess RL stability via *multi-temperature* sampling paired with a *multi-metric* suite (Pass@k, Pass@1, Avg@k), enabling comprehensive capability and robustness evaluation. (4) **Empirical gains:** Across diverse reasoning benchmarks, the approach consistently improves accuracy.

2 Related Works

2.1 Reinforcement Learning with Verifiable Rewards (RLVR)

Reinforcement Learning with Verifiable Rewards represents a paradigm shift from traditional RLHF approaches by leveraging automatically verifiable outcomes rather than human preference annotations. This approach is particularly powerful for domains where ground truth can be objectively determined, such as mathematical reasoning, code generation, and logical problem solving. Models like AlphaCode (Li et al., 2022) and recent mathematical reasoning (Jeannotte & Kieran, 2017; Xia et al., 2025) systems leverage execution results and correctness verification as direct reward signals, eliminating the need for expensive human annotation.

Process Reward Models (PRMs) have emerged as a sophisticated extension of RLVR, where intermediate steps in reasoning processes are evaluated and rewarded based on their correctness (Uesato et al., 2022; Lightman et al., 2023). Recent developments include tool-augmented reasoning

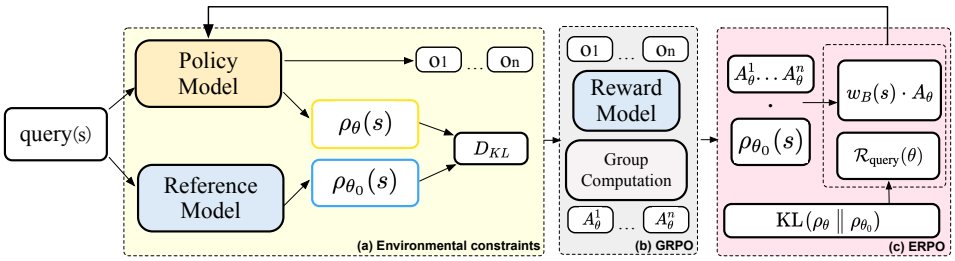

Figure 2: **The Proposed ERPO Overview.** (a) For each query, the policy and reference induce current and reference query samplers, and we pre-compute a Query-KL to penalize environment drift. (b) For each query, the policy samples a response group scored by the reward model to produce the standard GRPO learning signal. (c) On top of GRPO we replace response-KL with pre-computed Query-KL and weight within-query advantages by the query's occurrence probability, yielding an environment-aware update.

systems (Schick et al., 2023) and self-verification approaches (Kojima et al., 2022), which combine language models with external verification tools to enable automatic reward computation for broader task domains.

While RLVR provides scalable and consistent training signals compared to subjective human preferences, it introduces unique challenges in handling high variance from sparse rewards and potential reward hacking behaviors. These stability issues motivate the need for robust training methodologies that can effectively leverage verifiable rewards while maintaining training stability.

2.2 Reinforcement Learning Stability in Language Model Training

The stability of reinforcement learning algorithms in language model training has become a critical research area due to unique challenges posed by discrete action spaces, large parameter spaces, and complex reward landscapes (Sutton et al., 1998). Recent works have identified specific stability issues including reward hacking (Gao et al., 2023) and the alignment tax problem (Dai et al., 2025), where policy optimization can degrade downstream performance while improving target metrics. Distribution shift during training has been recognized as a fundamental source of instability in policy gradient methods (Reddy et al., 2020). In language model contexts, this manifests as shifts in the query distribution during training, leading to high variance in gradient estimates and potential policy collapse (Wen et al., 2024). Existing approaches primarily focus on action-space regularization through trust region methods (Schulman et al., 2015b) and KL divergence penalties between current and reference policies.

Despite progress in understanding RL stability issues, there remains a notable gap in explicitly managing the input query distribution during training. Most current approaches focus on output regularization rather than addressing environmental shifts at the input level, leaving query distribution management as an underexplored avenue for improving training stability.

3 Preliminaries

Setting and notation. Queries $s \in \mathcal{S}$ and responses $a \in \mathcal{A}$ are generated by a single model with parameters θ, which induces a *query distribution* $\rho_\theta(s)$ and a *response policy* $\pi_\theta(a \mid s)$. For each s, define

$$\bar{g}_\theta(s) \triangleq \mathbb{E}_{a \sim \pi_\theta(\cdot|s)}[g(a)], \tag{1}$$

and the training objective

$$J(\theta) = \mathbb{E}_{s \sim \rho_\theta}[\bar{g}_\theta(s)] = \sum_s \rho_\theta(s)\, \bar{g}_\theta(s). \tag{2}$$

We can evaluate the sequence log-likelihood

$$\ell_\theta(s) = \log p_\theta(s) = \sum_{t=1}^{T(s)} \log p_\theta(x_t \mid x_{<t}), \tag{3}$$

and in fully online generation take $\rho_\theta(s) \equiv p_\theta(s)$.

Policy-gradient (PG) family. The PG identity (Sutton et al., 1999; Williams, 1992) gives

$$\nabla_\theta J(\theta) = \mathbb{E}_{s\sim\rho_\theta,\ a\sim\pi_\theta}\big[A_\theta(s,a)\,\nabla_\theta \log \pi_\theta(a \mid s)\big], \quad A_\theta(s,a) = g(a) - b_\theta(s). \tag{4}$$

TRPO/PPO stabilize updates via KL trust regions or ratio clipping (Schulman et al., 2015a; 2017a). Large-scale LLM alignment employs PG-style pipelines (RLHF/Instruction tuning) (Ziegler et al., 2019; Ouyang et al., 2022a) and classification-style surrogates such as DPO (Rafailov et al., 2023b). We will instantiate experiments with *GRPO* (Shao et al., 2024), which computes a group-relative advantage from K sampled responses:

$$A_\theta^{\mathrm{grp}}\big(s, a^{(k)}\big) = g\big(a^{(k)}\big) - \frac{1}{K}\sum_{j=1}^{K} g\big(a^{(j)}\big). \tag{5}$$

Query-level KL regularization. KL control is standard for trust regions (Schulman et al., 2015a; 2017a). We apply forward KL (relative to reference θ_0) on the *query distribution* to constrain environmental drift, while not applying KL on the action layer, thus preserving exploration on the response end.

4 METHOD

4.1 BATCH SELF-NORMALIZED QUERY-LEVEL REWEIGHTING

Our goal is to construct a low-variance outer reweighting of the objective in equation 2 without relying on exponential (log-normal) importance ratios. Given a mini-batch $B = \{s_i\}_{i=1}^m$ drawn from a proposal $q(s)$ (a tractable distribution used to sample candidate states for training), we build a *batch self-normalized substitute distribution* on B using a monotone, tempered score $r(\cdot)$ computed with **stop-gradient**:

$$\mu_B(s_i) \;=\; \frac{r(s_i)}{Z_B}, \qquad Z_B \;=\; \sum_{j=1}^{m} r(s_j), \qquad r_\theta(s) \;\triangleq\; \frac{1}{-\ell_\theta(s)}\,(>0), \tag{6}$$

where $\ell_\theta(s) = \log p_\theta(s) < 0$ is the sequence log-likelihood from equation 3. This defines the batch objective

$$J_B(\theta) \;:=\; \sum_{i=1}^{m} \mu_B(s_i)\,\bar{g}_\theta(s_i) \;=\; \frac{1}{Z_B}\sum_{i=1}^{m} r_\theta(s_i)\,\bar{g}_\theta(s_i), \tag{7}$$

where $\bar{g}_\theta(s)$ is the per-query expected return in equation 1. Because r_θ and Z_B are treated as constants (stop-grad),

$$\nabla_\theta J_B(\theta) \;=\; \frac{1}{Z_B}\sum_{i=1}^{m} r_\theta(s_i)\,\nabla_\theta \bar{g}_\theta(s_i). \tag{8}$$

Scale-invariant surrogate. For any batch constant $c_B > 0$ independent of θ, define

$$\widetilde{J}_B(\theta) \;:=\; \sum_{i=1}^{m} c_B\, r_\theta(s_i)\,\bar{g}_\theta(s_i)\,. \tag{9}$$

Then $\nabla_\theta \widetilde{J}_B(\theta) = c_B Z_B\,\nabla_\theta J_B(\theta)$, so $\widetilde{J}_B$ and J_B share the same gradient direction (differ only by a positive scale). This allows us to replace μ_B by an *unnormalized* but scale-adjusted weight.

Closed-form query weight. Let $\bar{\ell}_B = \frac{1}{m}\sum_{j=1}^{m} \ell_\theta(s_j)$ (<0) and choose $c_B = -\bar{\ell}_B > 0$. Define the per-query outer weight

$$w_B(s_i) \;:=\; c_B\, r_\theta(s_i) \;=\; \frac{-\bar{\ell}_B}{-\ell_\theta(s_i)} \;=\; \frac{\bar{\ell}_B}{\ell_\theta(s_i)}\,(>0). \tag{10}$$

Replacing the outer expectation in equation 2 by a Monte Carlo sum with these weights yields the *query-reweighted* objective

$$\widehat{J}(\theta) \;:=\; \frac{1}{m}\sum_{s\in B} w_B(s)\,\bar{g}_\theta(s) \quad \text{with} \quad w_B(s) \text{ detached from gradients.} \tag{11}$$

The choice $r_\theta(s) = 1/(-\ell_\theta(s))$ preserves the likelihood ordering yet compresses the dynamic range compared with $\exp(\ell_\theta)$, reducing variance while keeping weights positive. (Optionally one may replace ℓ_θ by its length-normalized version $\bar{\ell}_\theta = \ell_\theta/T(s)$; we keep the notation w_B unchanged.)

4.2 Stabilizing the environment via a query-level KL

Because the query distribution ρ_θ co-evolves with the policy, we constrain its drift using a *query-level* forward KL to a fixed or slowly updated reference θ_0:

$$\mathcal{R}_{\text{query}}(\theta) \;:=\; \mathrm{KL}\big(\rho_\theta \,\big\|\, \rho_{\theta_0}\big) \;=\; \mathbb{E}_{s\sim\rho_\theta}\left[\log \rho_\theta(s) - \log \rho_{\theta_0}(s)\right]. \tag{12}$$

This penalizes forgetting probability mass under ρ_{θ_0} while *not* imposing any action-level KL, thus preserving response-side exploration. In practice we estimate the gradient of equation 12 by Monte Carlo over queries drawn from ρ_{θ_0} (or a cached pool), using $\nabla_\theta[-\log \rho_\theta(s)]$; the additive constant $\mathbb{E}_{s\sim\rho_\theta}[\log \rho_{\theta_0}(s)]$ is dropped.

4.3 Final loss and PG-compatible surrogate

Combining the query-reweighted objective equation 11 with the query-level KL gives the loss we minimize

$$\mathcal{L}(\theta) \;:=\; -\frac{1}{m}\sum_{s\in B} w_B(s)\,\bar{g}_\theta(s) \;+\; \alpha\,\mathcal{R}_{\text{query}}(\theta), \tag{13}$$

with trade-off parameter $\alpha > 0$. To instantiate equation 13 with any policy-gradient (PG) algorithm, we use the surrogate

$$\mathcal{L}_{\text{PG-family}}(\theta) \;:=\; -\frac{1}{m}\sum_{s\in B} w_B(s)\,\frac{1}{K}\sum_{a\in\mathcal{G}(s)} u_\theta(s,a)\,A^\star_\theta(s,a) \;+\; \alpha\,\mathcal{R}_{\text{query}}(\theta), \tag{14}$$

where $\mathcal{G}(s)$ is the set of K responses sampled for s, $A^\star_\theta$ is the algorithm-specific advantage (e.g., A_θ for REINFORCE; A^{grp}_θ in equation 5 for GRPO), and $u_\theta(s,a)$ is the algorithm-specific action weight ($u_\theta \equiv 1$ for REINFORCE/GRPO; clipped ratios for PPO). Our contribution is orthogonal: we *replace the per-query outer weight by $w_B(s)$ and add the query-KL term.* During backpropagation $w_B(s)$ is treated as a constant (stop-grad).

4.4 Instantiation on ERPO

For experiments we instantiate equation 14 with a GRPO-style group-relative baseline. For each s we sample $\mathcal{G}(s) = \{a^{(k)}\}_{k=1}^K$, compute the group-relative advantage using equation 5, take $u_\theta \equiv 1$, and optimize

$$\mathcal{L}_{\text{ERPO}}(\theta) \;:=\; -\frac{1}{m}\sum_{s\in B} w_B(s)\,\frac{1}{K}\sum_{a\in\mathcal{G}(s)} A^{\text{grp}}_\theta(s,a)\,\log \pi_\theta(a\mid s) \;+\; \alpha\,\mathcal{R}_{\text{query}}(\theta). \tag{15}$$

5 Experiments

5.1 Experimental Setup

Training We conduct experiments on mathematical reasoning tasks using Level 3–5 problems from the MATH dataset (Hendrycks et al., 2021), totaling approximately 8.5K examples. These are used to evaluate our proposed ERPO method, in comparison with the vanilla GRPO baseline. As described in Appendix A, the model must wrap its intermediate reasoning in `<think></think>` tags, and place the final answer inside `\boxed {}`.

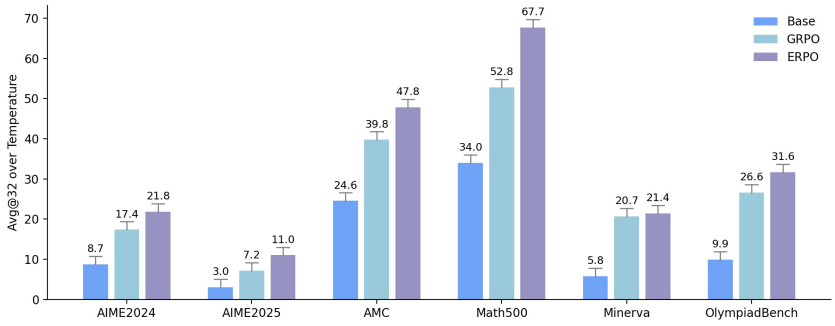

Figure 3: Avg@32 over Sampling Temperatures on Mathematical Reasoning Tasks

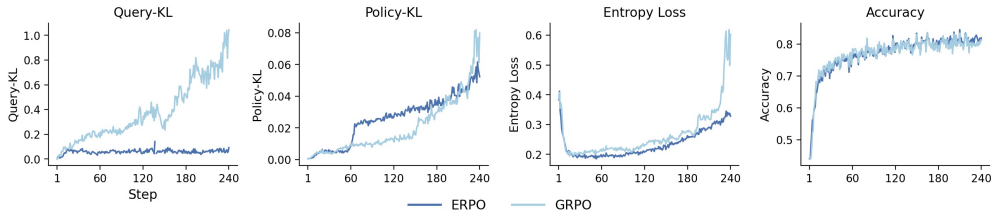

Figure 4: Training Dynamics on ERPO

Evaluation We follow standard practice and assess performance on six widely used benchmarks: AIME24, AIME25, AMC, MATH500 (Hendrycks et al., 2021), Minerva (Lewkowycz et al., 2022), and OlympiadBench (He et al., 2024). Prior work typically reports Avg@K (Yu et al., 2025), Pass@1 (Liu et al., 2025), and Pass@K (Hao et al., 2025) after RLVR training, often without specifying or controlling the inference-time sampling temperature. This omission can substantially affect reported performance and render results across studies not directly comparable. In preliminary experiments, we found that inference-time sampling temperature has a significant impact on performance, and that the effect intensifies as training progresses. To control for this factor, we fix the number of training steps across all models and evaluate at temperatures from 0.1 to 1.5; performance is then aggregated over this range.

Implementation Details We conduct all experiments using the EasyR1 framework (Zheng et al., 2025), training the Qwen2.5-Math model (Yang et al., 2024) with both GRPO and ERPO algorithms. Following prior work (Liu et al., 2025), we set the maximum sequence length to 3K tokens. For each problem, we sample eight responses at an inference temperature of 1.0. The rollout batch size is set to 512, and the update batch size to 128, for a total of 240 training steps. Token-level loss is applied throughout training. To ensure a fair comparison, we adopt the default KL divergence coefficient of 0.01.

5.2 Main Results

Figure 3 summarizes Avg@32 accuracy on six mathematical reasoning benchmarks, averaged over sampling temperatures from 0.1 to 1.5. ERPO consistently outperforms GRPO, with gains of up to 14.9% and an overall average improvement of 6.2%, highlighting its enhanced capability. Table 1 presents the detailed results for each benchmark, grouped by evaluation metric (e.g., Pass@1, Pass@K).

For both GRPO and ERPO, the prompts are identical to those used during training, whereas the Qwen base model adopts the default configuration from Dr.GRPO (Liu et al., 2025) to ensure optimal performance. Consistent with the aggregated results in Figure 3, ERPO surpasses GRPO across all evaluation metrics, achieving improvements of 6.2% in Avg@32, 3.64% in Pass@32, and 5.69% in Pass@1.

Table 1: Performance comparison across mathematical reasoning benchmarks. Best results per column are highlighted in bold.

Method	*AIME24*	*AIME25*	*AMC*	*MATH500*	*Minerva*	*Olympiad*	Avg.
Mean Avg@32							
Base	0.087	0.030	0.246	0.340	0.058	0.099	0.143
GRPO	0.174	0.072	0.398	0.528	0.207	0.266	0.274
ERPO	**0.218**	**0.110**	**0.478**	**0.677**	**0.214**	**0.316**	**0.336**
Mean Pass@32							
Base	0.373	0.206	0.674	0.764	0.349	0.411	0.463
GRPO	0.471	0.287	0.768	0.850	**0.516**	0.558	0.575
ERPO	**0.509**	**0.342**	**0.820**	**0.904**	0.500	**0.593**	**0.611**
Mean Pass@1							
Base	0.090	0.038	0.264	0.342	0.062	0.099	0.149
GRPO	0.169	0.084	0.398	0.533	0.201	0.263	0.275
ERPO	**0.207**	**0.091**	**0.477**	**0.679**	**0.217**	**0.320**	**0.332**

5.3 Training Dynamics

Figure 4 illustrates the training dynamics of the ERPO method. For both approaches, the sampling accuracy on the training set remains largely consistent; however, their divergence from the reference model exhibits markedly different trajectories.

In GRPO, constraints are imposed on the action distribution, causing the query distribution to drift away from the reference model at a substantially faster rate. Consequently, the KL divergence at the query level is an order of magnitude greater than at the policy level.

This imbalance leads to pronounced discrepancies in performance between the training and evaluation datasets. In contrast, ERPO applies constraints directly to the query distribution and adjusts the loss according to the probability of the given problem. This design both limits the degree of divergence from the reference model during training and, by leveraging the independence between the problem and the response, allows unconstrained exploration at the policy level. As a result, ERPO achieves superior generalization performance on general problems.

5.4 Analysis

Ablation Study We conduct ablation studies on the MATH500 benchmarks to assess reasoning efficiency. Table 2 summarizes the results for several commonly used sampling temperatures. Figure 5 further provides the complete performance–temperature variation curves across different experimental settings, along with the corresponding training dynamics.

Without modifying other hyperparameters, replacing the policy-based KL divergence with query-based KL divergence yields the best overall performance[1], with an average improvement of 15.9% over GRPO. In contrast, the policy-based KL divergence shows larger fluctuations (see Figure 5(a)) and its effectiveness diminishes under high-temperature sampling. We attribute this to the model learning from all queries without distinction, making it more sensitive to outlier data and more susceptible to noise, thereby reducing its generalization ability. Figure 5(d) of Figure 5 illustrates the policy distribution shift with and without $w_B(s)$, showing that variance-reduced sampling effectively constrains divergence from the reference model during training, leading to improved generalization.

[1]We also experimented with completely removing all KL divergence constraints, which resulted in the training process failing to converge.

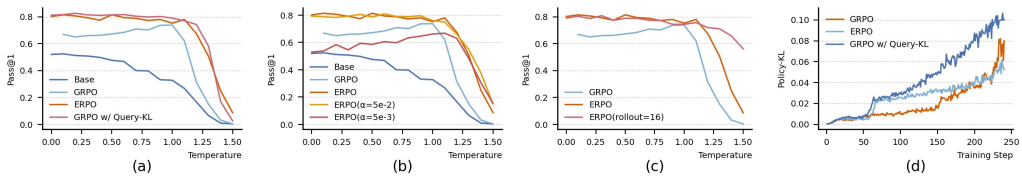

Figure 5: Pass@1 accuracy and training dynamics under different settings: (a)–(c) Model performance at various temperatures on MATH500; (d) Policy-KL divergence variation with GRPO using only Query-KL.

Different regularization strengths α also exert a significant influence on performance. As the constraint strength increases (e.g., $\alpha = 5 \times 10^{-2}$), the model achieves further improvements in overall performance (see Table 2). It is worth noting that we did not conduct an exhaustive search for the optimal α; instead, we retained the default value to ensure a relatively fair comparison.

Table 2: Performance Comparison Under Different Experimental settings

Method	KL Type	α	$w_B(s)$	Rollout Count	Temperature Setting				
					0.1	0.6	1.0	1.5	Mean
Baseline	—	—	—	—	0.524	0.468	0.328	0.004	0.331
GRPO	Policy	1×10^{-2}	—	8	0.668	0.684	0.738	0.004	0.533
	Query	1×10^{-2}	—	8	**0.816**	**0.816**	**0.790**	0.026	<u>0.692</u>
ERPO	Query	5×10^{-3}	✓	8	0.538	0.606	0.662	<u>0.154</u>	0.549
		1×10^{-2}	✓	8	0.794	0.806	0.752	0.086	0.678
		5×10^{-2}	✓	8	0.788	<u>0.810</u>	<u>0.760</u>	0.150	<u>0.692</u>
		1×10^{-2}	✓	16	<u>0.804</u>	0.788	0.744	**0.562**	**0.746**

Note: Best results per column are highlighted in **bold**, second-best results are <u>underlined</u>. Mean represents the average performance across all temperature settings (0.1–1.5). The $w_B(s)$ column indicates whether bias weighting is applied (✓) or not (—).

Rollouts We also analyze the effect of the number of samples per query. By increasing the sampling number to 16, we achieve the best performance, with the average Pass@1 rising to 74.6%. A higher sampling count also significantly improves sampling stability at high temperatures (see Table 2), without a noticeable increase in divergence from the reference model. Moreover, increasing the sampling count facilitates ERPO-based models in acquiring the correct reasoning format more effectively. [2]

Long-term Training To assess the stability of long-term RL training, we scale the training steps up to 1K and monitor changes in model performance over time. As shown in the figure 6, GRPO remains stable for sampling temperatures below 1.0 until approximately 240 steps (epoch=15). However, a pronounced performance degradation is first observed in the high-temperature sampling regime after 400 steps, and subsequently propagates to encompass sampling across all temperatures as the steps increase.

In contrast, ERPO exhibits a modest performance decline; however, the overall deterioration is substantially smaller, and its performance even improves within the high-temperature range. Figure 6 presents the complete training trajectories for both GRPO and ERPO. Although ERPO is not entirely immune to the collapse phenomenon that may occur during extended training—manifested as a sudden increase in entropy and a loss of sampling capability—it consistently outperforms vanilla GRPO and achieves a comparable degree of policy distribution constraint without relying on an explicit policy-based KL divergence term.

[2] Across multiple experiments, the GRPO method consistently failed to capture the desired output format. Consequently, for all experiments, we report only the answer accuracy.

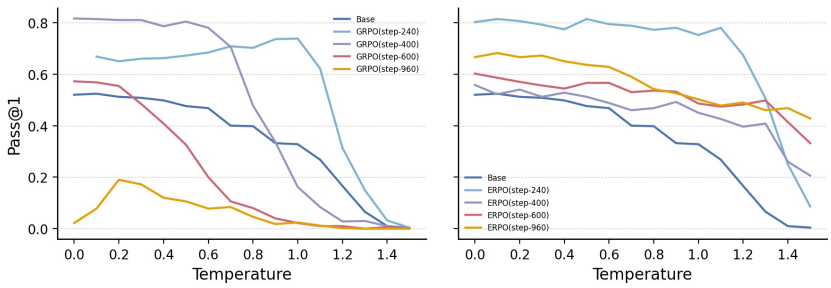

Figure 6: Performance Variation Across Training Steps

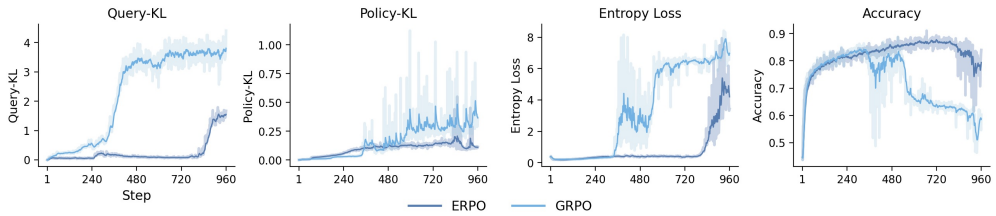

Figure 7: Training Dynamics on Long-term RL

6 Conclusion

By analyzing the coupling between the environment and the policy space in large language models, we decouple parameter regularization from the optimization objective during training. Specifically, we employ query-level KL divergence to indirectly constrain the distance between the policy model and the reference model. To prevent the model from prematurely converging to suboptimal solutions, we weight the advantage by the occurrence probability of each query. Experiments across multiple mathematical reasoning benchmarks demonstrate that the proposed ERPO method can achieve comparable KL divergence control without explicit policy regularization, while delivering superior performance. Furthermore, by sampling at different temperatures, we examine the evolution of sampling capability over long-term RL training, providing additional evidence of ERPO's stability during training.

Reproducibility Statement

We use open-source datasets for both training and testing, and conduct all experiments on an NVIDIA A100 GPU cluster. The complete environment configuration and step-by-step instructions for reproducing our results are openly available at: `https://anonymous.4open.science/r/ERPO-5B0C/`

References

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022. URL `https://arxiv.org/abs/2204.05862`.

Juntao Dai, Taiye Chen, Yaodong Yang, Qian Zheng, and Gang Pan. Mitigating reward overoptimization in rlhf via behavior-supported regularization. *arXiv preprint arXiv:2503.18130*, 2025.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pp. 10835–10866. PMLR, 2023.

Yaru Hao, Li Dong, Xun Wu, Shaohan Huang, Zewen Chi, and Furu Wei. On-policy rl with optimal reward baseline. *arXiv preprint arXiv:2505.23585*, 2025.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*, 2024.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Math. *Cornell University - arXiv,Cornell University - arXiv*, Mar 2021.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations (ICLR)*, 2020. URL https://arxiv.org/abs/1904.09751.

Garud N. Iyengar. Robust dynamic programming. *Mathematics of Operations Research*, 30(2): 257–280, 2005. doi: 10.1287/moor.1040.0129.

Doris Jeannotte and Carolyn Kieran. A conceptual model of mathematical reasoning for school mathematics. *Educational Studies in mathematics*, 96(1):1–16, 2017.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.

Minchan Kwon, Gaeun Kim, Jongsuk Kim, Haeil Lee, and Junmo Kim. Stableprompt: Automatic prompt tuning using reinforcement learning for large language model. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9868–9884, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.551. URL https://aclanthology.org/2024.emnlp-main.551/.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857, 2022.

Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, R'emi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, et al. Competition-level code generation with alphacode. *Science*, 378(6624):1092–1097, 2022.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2023.

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. *arXiv preprint arXiv:2503.20783*, 2025.

Arnab Nilim and Laurent El Ghaoui. Robust control of markov decision processes with uncertain transition matrices. *Operations Research*, 53(5):780–798, 2005. doi: 10.1287/opre.1050.0216.

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. *arXiv preprint arXiv:2203.02155*, 2022a. URL https://arxiv.org/abs/2203.02155.

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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022b. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf.

Sindhu Padakandla. A survey of reinforcement learning algorithms for dynamically varying environments. *ACM Computing Surveys*, 54(6):127:1–127:25, 2021. doi: 10.1145/3459991. URL https://dl.acm.org/doi/10.1145/3459991.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023a. URL https://arxiv.org/abs/2305.18290.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*, 2023b. URL https://papers.nips.cc/paper/2023/hash/a85b405ed65c6477a4fe8302b5e06ce7-Abstract-Conference.html.

Siddharth Reddy, Anca Dragan, Sergey Levine, Shane Legg, and Jan Leike. Learning human objectives by evaluating hypothetical behavior. In *International conference on machine learning*, pp. 8020–8029. PMLR, 2020.

Stephane Ross and J. Andrew Bagnell. Reinforcement and imitation learning via interactive no-regret learning. *arXiv preprint arXiv:1406.5979*, 2014. URL https://arxiv.org/abs/1406.5979.

Stephane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 15 of *Proceedings of Machine Learning Research*, pp. 627–635, Fort Lauderdale, FL, USA, Apr 2011. PMLR. URL https://proceedings.mlr.press/v15/ross11a.html.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551, 2023.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 1889–1897. PMLR, 2015a. URL https://proceedings.mlr.press/v37/schulman15.html.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. PMLR, 2015b.

John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1889–1897, Lille, France, 2015c. PMLR. URL https://proceedings.mlr.press/v37/schulman15.html.

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

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

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

Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

Richard S. Sutton, David A. McAllester, Satinder P. Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems 12 (NIPS 1999)*, 1999. URL https://papers.nips.cc/paper/1999/hash/464d828b85b0bed98e80ade0a5c43b0f-Abstract.html.

Prashant Trivedi, Souradip Chakraborty, Avinash Reddy, Vaneet Aggarwal, Amrit Singh Bedi, and George K. Atia. Align-pro: A principled approach to prompt optimization for llm alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2025. URL https://arxiv.org/abs/2501.03486.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process-and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://arxiv.org/abs/2203.11171.

Jiaxin Wen, Ruiqi Zhong, Akbir Khan, Ethan Perez, Jacob Steinhardt, Minlie Huang, Samuel R Bowman, He He, and Shi Feng. Language models learn to mislead humans via rlhf. *arXiv preprint arXiv:2409.12822*, 2024.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992. doi: 10.1007/BF00992696. URL https://link.springer.com/article/10.1007/BF00992696.

Shijie Xia, Xuefeng Li, Yixin Liu, Tongshuang Wu, and Pengfei Liu. Evaluating mathematical reasoning beyond accuracy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 27723–27730, 2025.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

Ziyu Ye, Rishabh Agarwal, Tianqi Liu, Rishabh Joshi, Sarmishta Velury, Quoc V. Le, Qijun Tan, and Yuan Liu. Scalable reinforcement post-training beyond static human prompts: Evolving alignment via asymmetric self-play. *arXiv preprint arXiv:2411.00062*, 2024. URL https://arxiv.org/abs/2411.00062.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Yaowei Zheng, Junting Lu, Shenzhi Wang, Zhangchi Feng, Dongdong Kuang, and Yuwen Xiong. Easyr1: An efficient, scalable, multi-modality rl training framework. https://github.com/hiyouga/EasyR1, 2025.

Wenxuan Zhou, Ravi Agrawal, Shujian Zhang, Sathish Reddy Indurthi, Sanqiang Zhao, Kaiqiang Song, Silei Xu, and Chenguang Zhu. Wpo: Enhancing rlhf with weighted preference optimization. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 8328–8340, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.475. URL https://aclanthology.org/2024.emnlp-main.475/.

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. *arXiv preprint arXiv:1909.08593*, 2019. URL https://arxiv.org/abs/1909.08593.

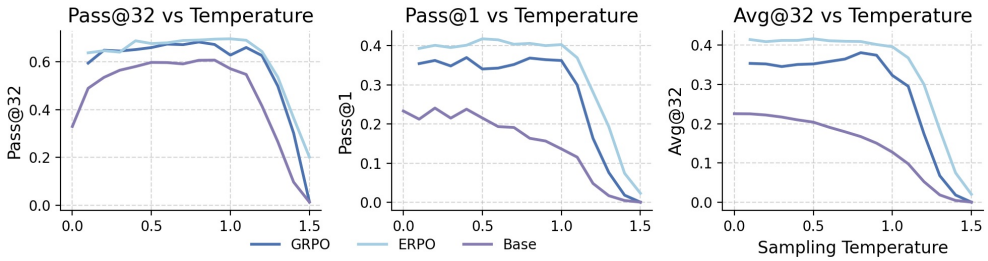

Figure 8: Variation of Metrics with Temperature

A Prompt

```
{{ content | trim }} You FIRST think about the reasoning process
as an internal monologue and then provide the final answer.  The
reasoning process MUST BE enclosed within <think> </think> tags.
The final answer MUST BE put in \boxed {}.
```

B Variation of Metrics with Temperature

Figure 8 illustrates the model performance across different evaluation metrics and sampling temperatures. Our approach reduces the performance gap between different sampling temperatures, while increasing the likelihood of sampling correct outputs.

C Usage of Large Language Models

In this work, we leveraged large language model to assist in the writing process by polishing the language. The LLM provided grammatical refinement, rephrased ambiguous expressions, and enhanced overall readability, while all technical content and claims remain the sole responsibility of the authors.