Connecting Low-Rank Adapters and Policy Stability in GRPO Fine-Tuning

Antonin Rottman [1] **Francesco Tonin** [2] **Yongtao Wu** [2] **Toshiaki Koike-Akino** [3] **Volkan Cevher** [2]

Abstract

Low-Rank Adaptation (LoRA) is widely used for parameter-efficient reinforcement learning fine-tuning of large language models (LLMs), often together with an explicit Kullback-Leibler (KL) penalty toward a reference policy. We study whether the low-rank constraint itself can restrict parameter trajectories and limit policy drift during Group Relative Policy Optimization (GRPO). In a simplified single-layer setting, we derive a rank-dependent upper bound on the KL divergence between reference and updated policies, providing a mechanistic explanation for how LoRA can constrain policy shift. Empirically, in short-horizon GRPO fine-tuning of several 1B–3B LLM families on reasoning tasks, we observe that KL-free LoRA preserves evaluation accuracy while reducing training time by avoiding reference-policy evaluations. Across LoRA ranks, policy divergence increases with rank, supporting the qualitative prediction of the analysis. These exploratory results suggest that low-rank parameterizations can contribute to policy stability in reinforcement learning fine-tuning, though broader studies across larger scales, longer horizons, and varied hyperparameters are needed.

1. Introduction

While large language models (LLMs) exhibit strong general abilities, their performance on multi-step reasoning remains limited (Patil & Jadon, 2025). Recent studies therefore explore reinforcement learning (RL) fine-tuning methods that optimize models via structured preferences rather than next-token prediction. Among these, Group Relative Policy Optimization (GRPO) extends Proximal Policy Optimization (PPO) (Schulman et al., 2017) and Direct Preference Optimization (DPO) (Rafailov et al., 2023) by learning from groupwise preferences over multiple model completions, and chains of thought (Shao et al., 2024).

Policy optimization approaches typically regularize the updated policy toward a reference using a Kullback-Leibler (KL) penalty (Schulman et al., 2015) to stabilize training with favorable error-averaging properties and linear horizon bounds (Vieillard et al., 2020), at the cost of increased computation due to multiple policy evaluations (Stiennon et al., 2020). Hu et al. (2022) introduced Low-Rank Adaptation (LoRA) which has been incorporated into RL pipelines to reduce memory and computational overhead (Wang et al., 2025), yet most existing implementations still retain the expensive KL regularization computation (Santacroce et al., 2023; Li et al., 2025). Several recent works report empirical observations suggesting that removing explicit KL penalties during LoRA training may not substantially affect final task performance (Yu et al., 2025; Liu et al., 2025; Li et al., 2025; Sun et al., 2023). However, a principled understanding of the mechanisms driving these KL-free methods – and controlled empirical comparisons isolating this effect – remain largely absent.

In this exploratory study, we analyze the behavior of KL-free LoRA training in the 1B–3B parameter regime for short-horizon GRPO. We develop a theoretical heuristic for how the low-rank parameterization constrains policy updates, and provide preliminary empirical evidence supporting this mechanism. We summarize our contributions as follows. First, we derive an explicit upper bound on the KL divergence induced by LoRA updates, showing that it scales with the adapter rank. Specifically, we prove that the KL divergence between policies is bounded by a function that grows with the LoRA rank r (Theorem 4.6). While derived in a highly simplified setting, this bound provides a mechanistic intuition for how low-rank constraints might limit policy drift during training. Secondly, we show empirically that policy divergence scales with the LoRA rank and run controlled GRPO fine-tuning experiments on Gemma,

[1]École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland [2]Laboratory for Information and Inference Systems (LIONS), École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland [3]Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, USA. Correspondence to: Antonin Rottman <antonin.rottman@gmail.com>.

ColorAI Workshop at the 43rd International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026. Copyright 2026 by the author(s).

Llama, and Qwen models in the 1B–3B regime. These short-horizon experiments suggest that low-rank constraints can provide sufficient regularization, unlike full fine-tuning, allowing KL-free LoRA training to preserve reasoning performance in this specific regime while reducing average training time.

2. Related Work

Parameter-efficient fine-tuning methods have become standard for adapting LLMs, offering significant reductions in memory and compute while maintaining competitive performance. Among these, LoRA (Hu et al., 2022) is particularly popular: it constrains weight updates to a low-rank subspace via a reparameterization. Instead of updating the full weight matrix $\mathbf{W_0} \in \mathbb{R}^{d\times k}$, it injects trainable matrices $\mathbf{A}$ and $\mathbf{B}$ such that $\Delta\mathbf{W} = \mathbf{BA}$, with $\mathbf{B} \in \mathbb{R}^{d\times r}$, $\mathbf{A} \in \mathbb{R}^{r\times k}$, and $r \ll \min(d, k)$.

Recent work has explored reinforcement-style fine-tuning to enhance the reasoning capabilities of LLMs. PPO, which requires a separate value model, is prone to instability and requires carefully tuned regularization (Schulman et al., 2017). Earlier approaches relied on KL penalties to constrain policy deviation from a reference model, stabilizing training but increasing computational cost (Schulman et al., 2015; Stiennon et al., 2020). Recent extensions include GRPO (Shao et al., 2024), which learns from groupwise preference signals. Several works have targeted computational overheads in GRPO. Yu et al. (2025); Liu et al. (2025) heuristically relax or remove the KL term in specific settings, though without theoretical justification or empirical evidence in general cases. Additional work has targeted the computational overhead of sampling multiple completions per prompt: Lin et al. (2025) propose pruning low-advantage completions, which dynamically reallocates GPU capacity, achieving substantial training speedups while preserving accuracy.

LoRA is widely adopted in RL fine-tuning of LLMs to reduce memory and computational costs (e.g., (Wang et al., 2025; Sun et al., 2023; Santacroce et al., 2023; Li et al., 2025)). Schulman & Lab (2025) demonstrate across policy gradient experiments that LoRA matches full fine-tuning even at rank 1, arguing on information-theoretic grounds that RL updates require very low representational capacity. They ablate LoRA rank and its effect on reward, but never consider LoRA as an implicit regularizer or compare it to other regularization methods. Wang et al. (2025) additionally shows that, even with extremely small LoRA rank, LoRA-based RL follows the reward dynamics of full-parameter GRPO with KL regularization across multiple architectures. Sun et al. (2023) empirically observe a high win rate without KL regularization in a PPO setting in Llama experiments, but provided no theoretical explanation, nor any systematic study of LoRA's regularization effect and the impact of rank. Li et al. (2025) provide limited empirical evidence in restricted settings of fine-tuning KL-free GRPO under LoRA for computational reasons, but do not analyze how the LoRA rank influences policy divergence nor provide a theoretical characterization of this effect or systematic experimental comparisons.

GRPO extends PPO by learning from groupwise preferences over multiple model completions. For each prompt, GRPO samples a group of G outputs and computes advantages relative to the group mean, eliminating the need for a separate value network. The objective is:

$$\mathcal{L}_{\text{GRPO}} = -\mathbb{E}_{\mathbf{x},\{\mathbf{y}_g\}_{g=1}^{G}\sim\boldsymbol{\pi}_{\theta_{\text{ref}}}}\Bigg[\frac{1}{G}\sum_{g=1}^{G}\min\Bigg(\frac{\boldsymbol{\pi}_\theta(\mathbf{y}_g|\mathbf{x})}{\boldsymbol{\pi}_{\theta_{\text{ref}}}(\mathbf{y}_g|\mathbf{x})}\hat{A}_g, \text{clip}\left(\frac{\boldsymbol{\pi}_\theta(\mathbf{y}_g|\mathbf{x})}{\boldsymbol{\pi}_{\theta_{\text{ref}}}(\mathbf{y}_g|\mathbf{x})}, 1-\epsilon, 1+\epsilon\right)\hat{A}_g\Bigg) - \beta D_{\text{KL}}(\boldsymbol{\pi}_\theta\|\boldsymbol{\pi}_{\theta_{\text{ref}}})\Bigg],$$

where $\hat{A}_g = r(\mathbf{x}, \mathbf{y}_g) - \frac{1}{G}\sum_{g'=1}^{G} r(\mathbf{x}, \mathbf{y}_{g'})$ is the advantage computed from reward $r(\cdot)$, ϵ is the clipping parameter, and β controls KL regularization toward a reference policy $\boldsymbol{\pi}_{\text{ref}}$.

3. Problem Setting

We now analyze the effect of low-rank updates on model behavior in LoRA adaptation of Transformer-based language models. Following the approach of Ren & Sutherland (2025), we model the update dynamics using a single-layer neural network as a tractable proxy for full Transformer training. While LoRA is applied to linear projection layers in practice (e.g., queries, keys, values, and output representations), the multi-layer composition and autoregressive softmax objective make direct analysis intractable; our simplified setting is intended to isolate the rank-dependent structure of LoRA updates and provide qualitative intuition rather than quantitative predictions for real LLM training. We consider binary classification with a sigmoid activation and binary cross-entropy (BCE) loss. We analyze the effect of Stochastic Gradient Descent (SGD) updates on LoRA parameters by considering randomly sampled training pairs $(\mathbf{x}_i, \mathbf{y}_i)_{i=1}^{t}$. We initialize the model with parameters $(\mathbf{A}_0, \mathbf{B}_0)$ and study how gradient steps influence the model's prediction on an evaluation input. The policy is parameterized by weights $\boldsymbol{\theta}_i$ at step i:

$$\boldsymbol{\pi}_{\theta_i}(\mathbf{x}_i) = \sigma\big(\mathbf{z}_i(\mathbf{x}_i)\big), \qquad \mathbf{z}_i(\mathbf{x}_i) := \mathbf{W}_0\mathbf{x}_i + \mathbf{B}_i\mathbf{A}_i\mathbf{x}_i,$$

where $\mathbf{x}_i \in \mathbb{R}^k$ is the input, $\mathbf{W}_0$ the fixed base weight, $\mathbf{A}_i, \mathbf{B}_i$ the trainable LoRA parameters, $\mathbf{z}_i(\mathbf{x}_i)$ the logits at step i and $\sigma(\cdot)$ the sigmoid function.

In our binary classification problem, we define BCE loss to

Table 1. Summary of notation.

Symbol	Dimension	Description
$\mathbf{W}_0$	$\mathbb{R}^{d\times k}$	Frozen pretrained weight matrix
$\mathbf{B}_i$	$\mathbb{R}^{d\times r}$	LoRA matrix at policy update step i
$\mathbf{A}_i$	$\mathbb{R}^{r\times k}$	LoRA matrix at policy update step i
$\mathbf{z}_i(\mathbf{x})$	$\mathbb{R}^d$	Logits: $\mathbf{z}_i(\mathbf{x}) = (\mathbf{W}_0 + \mathbf{B}_i\mathbf{A}_i)\mathbf{x}$
$\boldsymbol{\pi}_{\theta_i}(\mathbf{x})$	$\mathbb{R}^d$	Policy: $\sigma(\mathbf{W}_0\mathbf{x} + \mathbf{B}_i\mathbf{A}_i\mathbf{x})$
$\mathbf{y}_i$	$\mathbb{R}^d$	Ground-truth binary labels
$\mathbf{g}_i$	$\mathbb{R}^d$	$\mathbf{g}_i := \nabla_{\mathbf{z}_i(\mathbf{x})}\mathcal{L}_{\text{BCE}}(\boldsymbol{\pi}_{\theta_i}(\mathbf{x}), \mathbf{y}_i)$
$\mathbf{x}_e$	$\mathbb{R}^k$	Evaluation sample
$\mathbf{x}_i$	$\mathbb{R}^k$	Training sample at iteration
i	–	Index of the current SGD update step
t	–	Index of the final SGD update step

minimize:

$$\mathcal{L}_{\text{BCE}}\left(\boldsymbol{\pi}_{\theta_i}(\mathbf{x}_i), \mathbf{y}_i\right) := -\mathbf{y}_i^\top \log\left(\boldsymbol{\pi}_{\theta_i}(\mathbf{x}_i)\right) - \left(\mathbf{1} - \mathbf{y}_i\right)^\top \log\left(\mathbf{1} - \boldsymbol{\pi}_{\theta_i}(\mathbf{x}_i)\right).$$

The next section considers the learning dynamics of LoRA (Ren & Sutherland, 2025) to characterize the influence on the policy. Specifically, we study the variation in the policy $\boldsymbol{\pi}_{\theta_0}(\mathbf{x}_e)$ induced by SGD steps. We ask the fundamental question:

How does this update influence the model's behavior, as measured by the change in the policy $\Delta \log \boldsymbol{\pi}_{\theta_i}(\mathbf{x}_e) = \log \boldsymbol{\pi}_{\theta_i}(\mathbf{x}_e) - \log \boldsymbol{\pi}_{\theta_0}(\mathbf{x}_e)$ for an arbitrary input $\mathbf{x}_e$?

In the next section, we derive an explicit expression for this shift and characterize its structure and magnitude.

4. Theoretical Motivation in a Simplified Setting

We start by considering the SGD updates with learning rate η on the model parameters. Since the base weights $\mathbf{W}_0$ are frozen, we have $\nabla_{\mathbf{W}_0}\mathcal{L}_{\text{BCE}} = 0$. We derive the gradients of the binary cross-entropy loss with respect to $\mathbf{B}_i$ and $\mathbf{A}_i$, which parameterize the low-rank update. The accumulated differences in parameter weights after t SGD steps are denoted as:

$$\begin{aligned}
\boldsymbol{\Delta}\mathbf{A}_t = \mathbf{A}_t - \mathbf{A}_0 &= -\eta \sum_{i=0}^{t-1} \nabla_{[\mathbf{A}_i]_{l,m}} \mathcal{L}_{\text{BCE}}(\boldsymbol{\pi}_{\theta_i}(\mathbf{x}_i), \mathbf{y}_i) \\
&= -\eta \sum_{i=0}^{t-1} \operatorname{tr}\Big[\Big[\frac{\partial \mathcal{L}_{\text{BCE}}}{\partial \mathbf{z}_i(\mathbf{x}_i)}\Big]^\top \frac{\partial \mathbf{z}_i(\mathbf{x}_i)}{\partial [\mathbf{A}_i]_{l,m}}\Big] \\
&= -\eta \sum_{i=0}^{t-1} \operatorname{tr}\Big[\mathbf{g}_i^\top \mathbf{B}_i \frac{\partial \mathbf{A}_i}{\partial [\mathbf{A}_i]_{l,m}} \mathbf{x}_i\Big] \\
&= -\eta \sum_{i=0}^{t-1} \operatorname{tr}\big[\mathbf{e}_l^\top \mathbf{B}_i^\top \mathbf{g}_i \mathbf{x}_i^\top \mathbf{e}_m\big] = -\eta \sum_{i=0}^{t-1} \big[\mathbf{B}_i^\top \mathbf{g}_i \mathbf{x}_i^\top\big]_{l,m},
\end{aligned} \tag{1a}$$

$$\begin{aligned}
\boldsymbol{\Delta}\mathbf{B}_t = \mathbf{B}_t - \mathbf{B}_0 &= -\eta \sum_{i=0}^{t-1} \nabla_{[\mathbf{B}_i]_{l,m}} \mathcal{L}_{\text{BCE}}(\boldsymbol{\pi}_{\theta_i}(\mathbf{x}_i), \mathbf{y}_i) \\
&= -\eta \sum_{i=0}^{t-1} \operatorname{tr}\Big[\Big[\frac{\partial \mathcal{L}_{\text{BCE}}}{\partial \mathbf{z}_i(\mathbf{x}_i)}\Big]^\top \frac{\partial \mathbf{z}_i(\mathbf{x}_i)}{\partial [\mathbf{B}_i]_{l,m}}\Big] \\
&= -\eta \sum_{i=0}^{t-1} \operatorname{tr}\Big[\mathbf{g}_i^\top \frac{\partial \mathbf{B}_i}{\partial [\mathbf{B}_i]_{l,m}} \mathbf{A}_i \mathbf{x}_i\Big] \\
&= -\eta \sum_{i=0}^{t-1} \operatorname{tr}\big[\mathbf{e}_l^\top \mathbf{g}_i \mathbf{x}_i^\top \mathbf{A}_i^\top \mathbf{e}_m\big] = -\eta \sum_{i=0}^{t-1} \big[\mathbf{g}_i \mathbf{x}_i^\top \mathbf{A}_i^\top\big]_{l,m},
\end{aligned} \tag{1b}$$

where $\mathbf{e}_m$ is the m-th basis vector where the m-th entry is 1 and all other entries are 0.

We study the perturbation induced by SGD updates over t steps. After the update, evaluating the model on $\mathbf{x}_e$ allows us to view the change in the logits as a perturbation induced by the optimizer steps on $\mathbf{A}_0, \mathbf{B}_0$. The perturbation on the logit is due to the perturbation on the LoRA weights as follows:

$$\begin{aligned}
\mathbf{z}_t(\mathbf{x}_e) &= \mathbf{W}_0\mathbf{x}_e + (\mathbf{B}_0 + \boldsymbol{\Delta}\mathbf{B}_t)(\mathbf{A}_0 + \boldsymbol{\Delta}\mathbf{A}_t)\mathbf{x}_e \\
&= \underbrace{(\mathbf{W}_0 + \mathbf{B}_0\mathbf{A}_0)\mathbf{x}_e}_{\mathbf{z}_0(\mathbf{x}_e)} + \boldsymbol{\Delta}\mathbf{z}_t(\mathbf{x}_e),
\end{aligned} \tag{2}$$

where the update term can be further expressed with (Eq. (1a)) and (Eq. (1b)) as:

$$\begin{aligned}
\boldsymbol{\Delta}\mathbf{z}_t(\mathbf{x}_e) = -\eta \sum_{i=0}^{t-1} \Big(&\mathbf{B}_0\mathbf{B}_i^\top \mathbf{g}_i \mathbf{x}_i^\top \mathbf{x}_e + \mathbf{g}_i \mathbf{x}_i^\top \mathbf{A}_i^\top \mathbf{A}_0 \mathbf{x}_e \\
&- \eta \sum_{j=0}^{t-1} \mathbf{g}_i \mathbf{x}_i^\top \mathbf{A}_i^\top \mathbf{B}_j^\top \mathbf{g}_j \mathbf{x}_j^\top \mathbf{x}_e\Big).
\end{aligned}$$

To relate changes in parameter space to shifts in the output distribution, we begin by establishing a connection between the policy difference and the corresponding change in logits. We write the log-ratio shift vector $\Delta \log \boldsymbol{\pi}_{\theta_i}(\mathbf{x}_e) \in \mathbb{R}^d$ in coordinate-wise update as follows:

$$\Delta \log \boldsymbol{\pi}_{\theta_t}(\mathbf{x}_e) := \log \boldsymbol{\pi}_{\theta_t}(\mathbf{x}_e) - \log \boldsymbol{\pi}_{\theta_0}(\mathbf{x}_e), \tag{3}$$

where $\boldsymbol{\pi}_{\theta_t}(\mathbf{x}_e)$ is the updated policy and $\boldsymbol{\pi}_{\theta_0}(\mathbf{x}_e)$ is the reference policy. Then, we obtain the following first-order characterizations.

Proposition 4.1 (Policy Variation from Parameters)**.** *Let t SGD updates induce a change in the model parameters from*

$(\mathbf{A}_0, \mathbf{B}_0)$ *to* $(\mathbf{A}_t, \mathbf{B}_t)$. *Then, under a first-order approximation at the reference policy, with constant* η*, the change in the policy logits at* $\mathbf{x}_e$ *is:*

$$\Delta \log \boldsymbol{\pi}_{\theta_t}(\mathbf{x}_e) = \operatorname{diag}(\mathbf{1} - \boldsymbol{\pi}_{\theta_0}(\mathbf{x}_e))\boldsymbol{\Delta}\mathbf{z}_t(\mathbf{x}_e) + \mathcal{O}(\eta^2). \tag{4}$$

Proof. Substituting Eq. (2) for $\mathbf{z}_t(\mathbf{x}_e)$ and applying a first-order Taylor expansion of $\log \sigma(\mathbf{z}_t(\mathbf{x}_e))$ around the reference logits $\mathbf{z}_0(\mathbf{x}_e)$ results in

$$\begin{aligned}\Delta \log \boldsymbol{\pi}_{\theta_t}(\mathbf{x}_e) &= \log \sigma\big(\mathbf{z}_0(\mathbf{x}_e) + \boldsymbol{\Delta}\mathbf{z}_t(\mathbf{x}_e)\big) - \log \sigma\big(\mathbf{z}_0(\mathbf{x}_e)\big) \\ &= \operatorname{diag}\Big(\tfrac{d}{d\mathbf{z}} \log \sigma(\mathbf{z})\big|_{z=\mathbf{z}_0(\mathbf{x}_e)}\Big)\boldsymbol{\Delta}\mathbf{z}_t(\mathbf{x}_e) + \mathcal{O}(\|\boldsymbol{\Delta}\mathbf{z}_t\|^2) \\ &= \operatorname{diag}\big(\mathbf{1} - \boldsymbol{\pi}_{\theta_0}(\mathbf{x}_e)\big)\boldsymbol{\Delta}\mathbf{z}_t(\mathbf{x}_e) + \mathcal{O}(\eta^2).\end{aligned}$$

□

We now derive an expression for the KL divergence between $\boldsymbol{\pi}_t(\mathbf{x}_e)$ and $\boldsymbol{\pi}_0(\mathbf{x}_e)$ using only the log-ratio shift vector $\Delta \log \boldsymbol{\pi}_{\theta_i}(\mathbf{x}_e)$.

Proposition 4.2 (Exponential Perturbation)**.** *Expressing the KL divergence in terms of the perturbations and the reference policy,*

$$\begin{aligned}&D_{\mathrm{KL}}\big(\boldsymbol{\pi}_{\theta_t}(\mathbf{x}_e) \,\|\, \boldsymbol{\pi}_{\theta_0}(\mathbf{x}_e)\big) \\ &\quad = \mathbb{E}_{j\sim\boldsymbol{\pi}_{\theta_0}(\mathbf{x}_e)}\left[\mathrm{e}^{\Delta \log \boldsymbol{\pi}_{\theta_t,j}(\mathbf{x}_e)} \cdot \Delta \log \boldsymbol{\pi}_{\theta_t,j}(\mathbf{x}_e)\right].\end{aligned}$$

Proof. Using Eq. (3), it follows that $\boldsymbol{\pi}_{\theta_t}(\mathbf{x}_e) = \boldsymbol{\pi}_{\theta_0}(\mathbf{x}_e) \cdot \mathrm{e}^{\Delta \log \boldsymbol{\pi}_{\theta_t}(\mathbf{x}_e)}$. Substituting into the KL definition and taking expectation over $\boldsymbol{\pi}_{\theta_0}$ gives

$$\begin{aligned}&D_{\mathrm{KL}}\big(\boldsymbol{\pi}_{\theta_t}(\mathbf{x}_e) \,\|\, \boldsymbol{\pi}_{\theta_0}(\mathbf{x}_e)\big) \qquad (5)\\ &\quad = \sum_{j=1}^{d} \boldsymbol{\pi}_{\theta_0} \mathrm{e}^{\Delta \log \boldsymbol{\pi}_{\theta_t}(\mathbf{x}_e)} \Delta \log \boldsymbol{\pi}_{\theta_t}(\mathbf{x}_e) \\ &\quad = \mathbb{E}_{j\sim\boldsymbol{\pi}_{\theta_0}(\mathbf{x}_e)}\left[\mathrm{e}^{\Delta \log \boldsymbol{\pi}_{\theta_t,j}(\mathbf{x}_e)} \cdot \Delta \log \boldsymbol{\pi}_{\theta_t,j}(\mathbf{x}_e)\right].\end{aligned}$$

□

We introduce our main result below, where $\lesssim$ denotes *with high probability*.

Assumption 4.3. Let $\mathbf{A}_0$, $\mathbf{B}_0$ be random with independent, mean-zero, unit-variance, sub-Gaussian entries.

Assumption 4.4. Let $\mathbf{x_i}$ and $\mathbf{x_e}$ have independent, mean-zero, unit-variance, sub-Gaussian entries.

Assumption 4.5. Let the gradient norm be bounded by G: $\|g_i\|_\infty \leq G$ as is done in prior theoretical studies on the learning dynamics of LLMs (Ren & Sutherland, 2025).

Theorem 4.6 (KL Divergence Bound under Low-Rank Updates)**.** *The KL divergence between the reference* $\boldsymbol{\pi}_{\theta_0}(\mathbf{x}_e)$ *and updated* $\boldsymbol{\pi}_{\theta_t}(\mathbf{x}_e)$ *policies is bounded by:*

$$\begin{aligned}&D_{\mathrm{KL}}\big(\boldsymbol{\pi}_{\theta_t}(\mathbf{x}_e) \,\|\, \boldsymbol{\pi}_{\theta_0}(\mathbf{x}_e)\big) \\ &\leq \|\Delta \log \boldsymbol{\pi}_{\theta_t}(\mathbf{x}_e)\|_\infty \cdot e^{\left(\|\Delta \log \boldsymbol{\pi}_{\theta_t}(\mathbf{x}_e)\|_\infty\right)} + \mathcal{O}(\eta^2) \\ &\leq \mu(r) \cdot e^{\mu(r)} \qquad (6)\\ &\lesssim C(2\sqrt{r} + \sqrt{d} + \sqrt{k})^2 e^{\left(C(2\sqrt{r}+\sqrt{d}+\sqrt{k})^2\right)} + \mathcal{O}(\eta^2) \qquad (7)\end{aligned}$$

$$\textit{with } C := \eta\sqrt{k}Gt(1 + \eta\sqrt{k}G)^t \tag{8}$$

$$\textit{and } \mu(r) := \sqrt{rk}\big(\|\mathbf{B}_0\|_2 \cdot \|\Delta\mathbf{A}_t\|_2 + \|\mathbf{A}_0\|_2 \cdot \|\Delta\mathbf{B}_t\|_2\big).$$

Proof. The KL divergence can be expressed as: $D_{\mathrm{KL}}\big(\boldsymbol{\pi}_{\theta_t}(\mathbf{x}_e) \,\|\, \boldsymbol{\pi}_{\theta_0}(\mathbf{x}_e)\big) = \mathbb{E}_{j\sim\boldsymbol{\pi}_{\theta_0}(\mathbf{x}_e)}\left[e^{\delta_j} \cdot \delta_j\right]$, where $\delta_j := \Delta \log \boldsymbol{\pi}_{\theta_t,j}(\mathbf{x}_e)$. Since $|e^{\delta_j}\delta_j| \leq \|\boldsymbol{\delta}\|_\infty e^{\|\boldsymbol{\delta}\|_\infty}$ pointwise (for $\delta_j > 0$ both factors are dominated by their supremum; for $\delta_j < 0$, $e^{\delta_j} < 1$ so $|e^{\delta_j}\delta_j| \leq |\delta_j| \leq \|\boldsymbol{\delta}\|_\infty \leq \|\boldsymbol{\delta}\|_\infty e^{\|\boldsymbol{\delta}\|_\infty}$), taking expectations over $\boldsymbol{\pi}_{\theta_0}$ yields:

$$D_{\mathrm{KL}}(\boldsymbol{\pi}_{\theta_t} \| \boldsymbol{\pi}_{\theta_0}) \leq \|\Delta \log \boldsymbol{\pi}_{\theta_t}(\mathbf{x}_e)\|_\infty \cdot e^{\left(\|\Delta \log \boldsymbol{\pi}_{\theta_t}(\mathbf{x}_e)\|_\infty\right)}. \tag{9}$$

Let j index a row of the output; we now upper bound the individual coordinate $|\Delta z_t(\mathbf{x_e})_j|$, which contributes to the shift in $\Delta \log \boldsymbol{\pi}_{\theta_t}$.

$$\begin{aligned}|[\Delta\mathbf{z}_t(\mathbf{x}_e)]_j| &= |\mathbf{b}_j^\top \Delta\mathbf{A}_t\mathbf{x}_e + \mathbf{e}_j^\top \Delta\mathbf{B}_t\mathbf{A}_0\mathbf{x}_e| \\ &\leq |\mathbf{b}_j^\top \Delta\mathbf{A}_t\mathbf{x}_e| + |\mathbf{e}_j^\top \Delta\mathbf{B}_t\mathbf{A}_0\mathbf{x}_e|. \qquad (10)\end{aligned}$$

Here $\mathbf{b}_j^\top$ denotes the j-th row of $\mathbf{B}_0$, and $\mathbf{e}_j^\top$ is the standard basis row vector. We now bound each term separately. Each LoRA update matrix $\Delta\mathbf{A}_t$ and $\Delta\mathbf{B}_t$ has the form: $\Delta\mathbf{A}_t = -\eta\sum_{i=0}^{t-1}\mathbf{B}_i^\top\mathbf{g}_i\mathbf{x}_i^\top$, $\Delta\mathbf{B}_t = -\eta\sum_{i=0}^{t-1}\mathbf{g}_i\mathbf{x}_i^\top\mathbf{A}_i^\top$. Each term in the sum is an outer product, so $\Delta\mathbf{A}_t$ and $\Delta\mathbf{B}_t$ each have rank at most r after t steps. Using the inequalities: $\|\mathbf{M}\mathbf{v}\|_2 \leq \|\mathbf{M}\|_{\mathrm{F}} \cdot \|\mathbf{v}\|_2$, $\|\mathbf{M}\|_{\mathrm{F}} \leq \sqrt{r} \cdot \|\mathbf{M}\|_2 \rightarrow \|\mathbf{M}\mathbf{v}\|_2 \leq \sqrt{r} \cdot \|\mathbf{M}\|_2 \cdot \|\mathbf{v}\|_2$ and under Assumption 4.4, $\|\mathbf{x_e}\|_2 \lesssim \sqrt{k}$ (Thm 3.1.1 in Vershynin (2018)):

$$|\mathbf{e}_j^\top \Delta\mathbf{B}_t\mathbf{A}_0\mathbf{x}_e| \leq \|\Delta\mathbf{B}_t\mathbf{A}_0\mathbf{x}_e\|_2 \leq \sqrt{rk}\|\mathbf{A}_0\|_2\|\Delta\mathbf{B}_t\|_2, \tag{11a}$$

$$|\mathbf{b}_j^\top \Delta\mathbf{A}_t\mathbf{x}_e| \leq \|\mathbf{b}_j\Delta\mathbf{A}_t\mathbf{x}_e\|_2 \leq \sqrt{rk}\|\mathbf{B}_0\|_2\|\Delta\mathbf{A}_t\|_2. \tag{11b}$$

Define $\mu(r) := \sqrt{rk}\big(\|\mathbf{B}_0\|_2 \cdot \|\Delta\mathbf{A}_t\|_2 + \|\mathbf{A}_0\|_2 \cdot \|\Delta\mathbf{B}_t\|_2\big)$, from Eqs. (10), (11), then with Eqs. (4), (9):

$$D_{\mathrm{KL}}\big(\boldsymbol{\pi}_{\theta_t}(\mathbf{x}_e) \,\|\, \boldsymbol{\pi}_{\theta_0}(\mathbf{x}_e)\big) \leq \mu(r) \cdot e^{\mu(r)}.$$

We derive a bound independent of the operator norms and apply the reverse triangle inequality to Eq.(1) at step i, $\|\Delta\mathbf{A}_i\|_2 = \|\mathbf{A}_i - \mathbf{A}_0\|_2 \geq \big|\|\mathbf{A}_i\|_2 - \|\mathbf{A}_0\|_2\big|$:

$$\|\mathbf{A}_0\|_2 - \|\Delta\mathbf{A}_i\|_2 \leq \|\mathbf{A}_i\|_2 \leq \|\mathbf{A}_0\|_2 + \|\Delta\mathbf{A}_i\|_2. \quad (12)$$

Focusing on the upper bound of Eq. (12), and applying the same argument to $\|\mathbf{B}_i\|_2$:

$$\|\mathbf{A}_i\|_2 + \|\mathbf{B}_i\|_2 \leq \|\mathbf{A}_0\|_2 + \|\mathbf{B}_0\|_2 + \|\Delta\mathbf{A}_i\|_2 + \|\Delta\mathbf{B}_i\|_2.$$

Define $s_i := \|\mathbf{A}_i\|_2 + \|\mathbf{B}_i\|_2$, $s_0 := \|\mathbf{A}_0\|_2 + \|\mathbf{B}_0\|_2$ and add the inequalities of $\|\Delta\mathbf{A}_i\|_2$ and $\|\Delta\mathbf{B}_i\|_2$:

$$s_i \leq s_0 + \eta\sqrt{k}G\sum_{j=0}^{i-1} s_j, \quad i = 0, 1, \ldots, t-1.$$

This is precisely the form of the discrete Grönwall inequality (Clark, 1987). Applying the Grönwall lemma:

$$s_i \leq s_0 \prod_{j=0}^{i-1}(1 + \eta\sqrt{k}G) = s_0(1 + \eta\sqrt{k}G)^i.$$

Therefore, for $i < t$:

$$\|\mathbf{A}_i\|_2 + \|\mathbf{B}_i\|_2 \leq (\|\mathbf{A}_0\|_2 + \|\mathbf{B}_0\|_2) \cdot (1 + \eta\sqrt{k}G)^t.$$

Under Assumption 4.5, $\|\mathbf{g}_i\|_2 \leq G$, and under Assumption 4.4, $\|\mathbf{x}_i\|_2 \lesssim \sqrt{k}$.

$$\|\mathbf{\Delta A_t}\|_2 \leq \eta\sqrt{k}Gt\sum_{i=0}^{t-1}\|\mathbf{B}_i\|_2$$

$$\|\mathbf{\Delta B_t}\|_2 \leq \eta\sqrt{k}Gt\sum_{i=0}^{t-1}\|\mathbf{A}_i\|_2.$$

We can further upper bound each LoRA weight:

$$\|\mathbf{\Delta A_t}\|_2, \|\mathbf{\Delta B_t}\|_2 \leq \eta\sqrt{k}Gt(\|\mathbf{A}_0\|_2 + \|\mathbf{B}_0\|_2) \cdot (1 + \eta\sqrt{k}G)^t.$$

Under Assumption 4.3, by Theorem 4.3.3 in Vershynin (2018):

$$\|\mathbf{A}_0\|_2 \lesssim \sqrt{r} + \sqrt{k}, \quad \|\mathbf{B}_0\|_2 \lesssim \sqrt{d} + \sqrt{r}.$$

$$\begin{aligned}|[\Delta\mathbf{z}_t(\mathbf{x}_e)]_j| &\leq \sqrt{rk}\,(\|\mathbf{B}_0\|_2\|\Delta\mathbf{A}_t\|_2 + \|\mathbf{A}_0\|_2\|\Delta\mathbf{B}_t\|_2) \\ &\leq \eta\sqrt{k}Gt(1 + \eta\sqrt{k}G)^t \cdot (\|\mathbf{A}_0\|_2 + \|\mathbf{B}_0\|_2)^2 \\ &\lesssim \eta\sqrt{k}Gt(1 + \eta\sqrt{k}G)^t \cdot (2\sqrt{r} + \sqrt{d} + \sqrt{k})^2.\end{aligned}$$

And finally replacing into Eq. (9):

$$\begin{aligned}&D_{\mathrm{KL}}(\boldsymbol{\pi}_{\theta_t}\|\boldsymbol{\pi}_{\theta_0}) \\ &\quad \lesssim C(2\sqrt{r} + \sqrt{d} + \sqrt{k})^2 e^{\left(C(2\sqrt{r}+\sqrt{d}+\sqrt{k})^2\right)} + \mathcal{O}(\eta^2),\end{aligned}$$

with $C := \eta\sqrt{k}Gt(1 + \eta\sqrt{k}G)^t$. □

Theorem 4.6 follows from a first-order Taylor expansion around the reference policy. It serves as a heuristic motivation rather than a rigorous multi-layer guarantee, characterizing the induced policy shift only in the local regime where higher-order terms remain negligible. In this setting, the bound in Eq. (6) offers an interpretable link between LoRA rank and KL growth. Moreover, Eq. (7) isolates the dominant term through the factor $e^{C(2\sqrt{r}+\sqrt{d}+\sqrt{k})^2}$, making explicit how policy shift depends on the rank. As the policy drifts further from the reference, higher-order corrections may contribute meaningfully, and the bound in Eq. (6) becomes less descriptive. Since $C := \eta\sqrt{k}Gt(1+\eta\sqrt{k}G)^t$ in Eq. (8) depends only on known hyperparameters (learning rate η, step count t, and the clipping-controlled gradient bound G) the expression can be used to guide practical choices of η, t, and clipping so as to keep the exponential term controlled and ensure that rank remains a meaningful predictor of KL throughout training.

5. Scope and Limitations of the Theoretical Analysis

Theorem 4.6 is derived in a deliberately simplified setting: a single-layer network trained with binary cross-entropy and SGD on a binary classification task. This is a common approach in the theoretical study of LLM fine-tuning, where the complexity of full Transformer training makes rigorous results difficult to obtain directly. Ren & Sutherland (2025) adopt the same simplification – analyzing learning dynamics through single-layer, first-order Taylor expansion frameworks – and explicitly acknowledge that the fixed neural tangent kernel assumption may be too strong for full LLM fine-tuning, yet show that the resulting framework still yields qualitatively accurate predictions about alignment and preference tuning behavior. Zhang et al. (2025) similarly ground their LoRA convergence guarantees in linear and single-layer nonlinear models before arguing that the structural insights transfer to practice, and Dayi & Chen (2024) analyze a two-layer teacher-student setup for rank-1 LoRA to study convergence, again relying on architectural simplification to make the analysis tractable.

Several concrete gaps exist between our setting and real LLM GRPO fine-tuning that the reader should keep in mind. First, our single-layer analysis does not account for the *composition of layers*: in practice, LoRA layers are applied to earlier blocks in a deep Transformer. These early low-rank perturbations compound and can be amplified through subsequent nonlinear operations, potentially producing a much larger effective policy shift at the output than our single-layer theory captures (Sun et al., 2023). Second, we analyze a sigmoid activation with BCE loss, whereas autoregressive LLMs use a softmax output with token-level cross-entropy; while the first-order structure of the KL bound is analogous,

the precise constants and interaction with vocabulary size differ. Third, our analysis uses SGD with a fixed learning rate, while practice uses AdamW (Loshchilov & Hutter, 2019) with adaptive per-parameter step sizes, which changes the effective trajectory of the LoRA parameters in ways not captured by our Grönwall-based norm bounds. Fourth, the bound in Eq. (7) is derived in the local first-order regime and becomes vacuous when the policy has drifted substantially from the reference.

We therefore present Theorem 4.6 as providing *qualitative intuition and a structural explanation* for why low-rank constraints tend to limit policy drift, rather than as a quantitative prediction for multi-layer Transformer GRPO. The key structural insight – the $\sqrt{r}$ factor in $\mu(r)$ – identifies the mechanism: low-rank updates confine parameter changes to a low-dimensional subspace, and the resulting logit perturbations scale with the rank of that subspace. Whether this mechanism is sufficient to eliminate the need for explicit KL regularization in practice is an empirical question, which we address in Section 6.

6. Experiments

We now examine the empirical implications of our theoretical analysis. We compare LoRA-based GRPO models trained with and without KL regularization, evaluating both policy divergence and task performance to assess LoRA's effectiveness as an implicit regularizer. Our results suggest that, in the short-horizon 1B–3B setting studied here, LoRA can provide sufficient regularization without explicit KL penalties.

To examine how the rank of LoRA adapters affects policy divergence, we conduct controlled GRPO fine-tuning experiments on Gemma, Llama, and Qwen models in the 1B–3B regime on instruction-tuned checkpoints. For each rank setting, we also compare training runs with and without KL regularization in order to isolate its influence on accuracy and training dynamics. Models are trained with GRPO using AdamW (Loshchilov & Hutter, 2019) on the widely used GSM8K (Cobbe et al., 2021) dataset. We evaluate performance on the GSM8K evaluation split, GPQA (Rein et al., 2024), and MATH-500 (Hendrycks et al., 2021), which are established mathematical reasoning benchmarks (Shao et al., 2024).

Training hyperparameters are kept consistent across all model families. Preliminary experiments indicated that the standard learning rate of 5×10^{-6} allowed all the models to reach a stable reward signal within 300 steps. The batch size is 32 and the models were trained on a single Tesla V100 32GB GPU. We use $\beta = 0.05$ for the KL penalty, aligned with prior GRPO work (Shao et al., 2024). Following Schulman & Lab (2025), who demonstrate that applying LoRA to

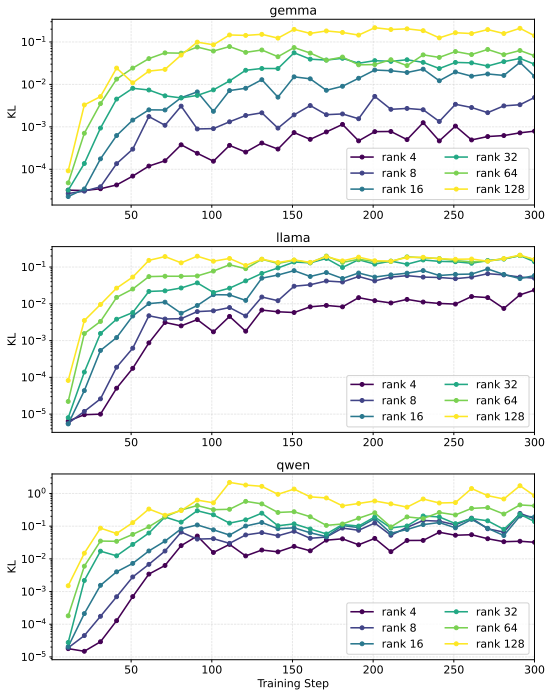


Figure 1. Mean KL divergence during GRPO training across LoRA ranks for Gemma, Llama, and Qwen. KL grows with rank. Higher ranks allow greater policy deviation.

all layers is both sufficient and preferable for policy gradient fine-tuning, we apply LoRA adapters to all linear projection layers. Training is performed using a single random seed due to compute constraints. Runtime is measured as wall-clock training time in minutes for the full 300-step run on the same hardware; the variation across ranks reflects differences in the number of trainable parameters and memory bandwidth rather than convergence speed. For the KL penalty, we use the K_3 estimator (the default implementation in Hugging Face TRL), added directly to the loss. As recently demonstrated by Shah et al. (2025), while K_3 is an unbiased estimator of the KL divergence value, placing it directly in the loss objective yields biased gradient updates with respect to the true reverse KL divergence. Although this gradient bias can induce training instabilities, our study focuses on the relationship between LoRA and the standard, widely adopted KL penalty used in practice for GRPO; we therefore utilize this common formulation and leave the evaluation of alternative, unbiased gradient estimators to future work.

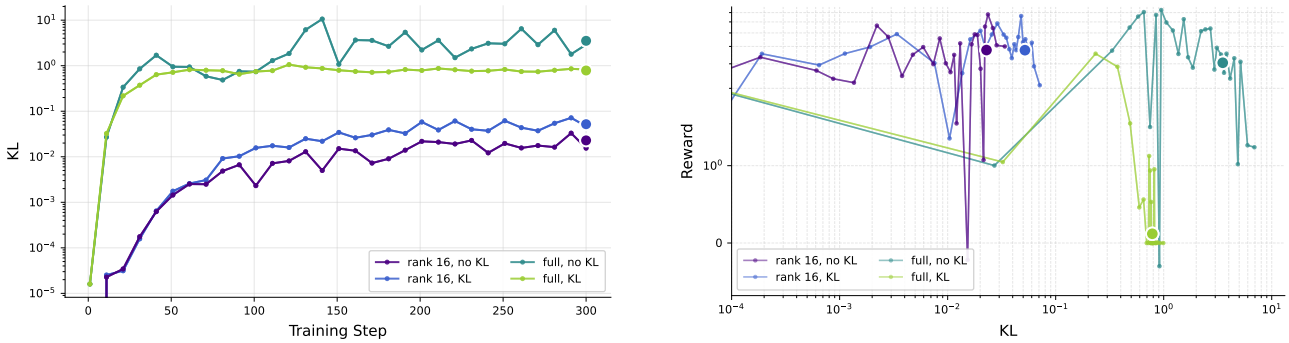
To assess how the LoRA rank influences policy divergence and downstream reasoning performance, we perform GRPO fine-tuning across three LLM families: Gemma (Gemma Team, 2025), Llama (Grattafiori et al., 2024), and Qwen (Yang et al., 2025), using the GSM8K (Cobbe et al., 2021), GPQA (Rein et al., 2024), and MATH-500 (Hendrycks et al., 2021) benchmarks. Each model is trained with and without explicit KL regularization, holding all hyperparam-

Table 2. Qwen2.5-3B-Instruct GRPO results with/without KL regularization across ranks. ΔAverage denotes the mean difference (w/o KL − with KL). Removing KL reduces average runtime while maintaining accuracy.

KL	r	GPQA (↑)	GSM8K (↑)	MATH-500 (↑)	Runtime (min) (↓)
w/o	4	**35.35**	75.97	67.60	139.17
	8	31.31	75.59	65.40	119.64
	16	30.30	77.48	63.40	109.61
	32	30.81	77.26	66.20	112.33
	64	34.85	**77.56**	66.60	106.95
	128	32.83	**77.56**	65.00	104.02
w	4	34.85	77.33	64.80	147.64
	8	33.84	76.35	67.20	130.30
	16	25.25	76.95	65.60	134.14
	32	25.76	77.26	64.80	125.40
	64	30.81	76.27	**67.80**	112.79
	128	31.82	77.48	64.60	87.01
Δ *Average*		+2.19	-0.04	-0.10	-6.76

Table 3. Results on Gemma-3-1B-it.

KL	r	GPQA (↑)	GSM8K (↑)	MATH-500 (↑)	Runtime (min) (↓)
w/o	4	29.29	48.22	33.80	342.85
	8	30.30	48.52	32.60	328.48
	16	24.75	49.13	31.20	321.71
	32	24.24	50.42	31.40	342.89
	64	24.24	51.02	31.20	344.69
	128	27.27	**52.31**	32.60	332.16
w	4	23.74	48.60	31.40	343.24
	8	24.75	48.37	31.80	353.41
	16	25.25	50.42	**34.20**	346.22
	32	**33.33**	48.82	31.80	364.34
	64	25.76	48.07	31.20	366.42
	128	31.31	46.40	32.80	350.97
Δ *Average*		-0.68	+1.49	-0.07	-18.30



(a) KL Divergence vs. Step

(b) Reward vs. KL Divergence

Figure 2. Effect of LoRA and explicit KL regularization on policy divergence and performance under GRPO (Gemma-3-1B-it). (a) KL divergence over training steps for LoRA (rank 16) vs. full fine-tuning, with and without explicit KL regularization. (b) Average reward vs. KL over the last 50 steps for each configuration. The reward model is shaped, including format and correctness components.

eters fixed to isolate the effect of LoRA rank.

Across all models, we observe a consistent growing trend of KL divergence with LoRA rank, as illustrated in Figure 1. The curves exhibit a near log-linear trend on a semilog scale: small ranks ($r \leq 16$) tightly constrain the policy shift, whereas larger ranks ($r \geq 64$) produce an order-of-magnitude increase in KL. This pattern aligns with the theoretical insight that the effective stability region of the updates expands with the rank-dependent geometry of LoRA updates. Despite the increase in KL with rank, task performance remains stable across all configurations. As shown in Tables 2–4, accuracy differences between runs with and without KL regularization are minor. This suggests that LoRA may provide sufficient regularization to maintain pol-

Table 4. Results on Llama-3.2-3B-Instruct.

KL	r	GPQA (↑)	GSM8K (↑)	MATH-500 (↑)	Runtime (min) (↓)
w/o	4	26.77	68.23	43.60	176.26
	8	26.26	68.08	41.80	172.80
	16	30.30	68.23	43.60	161.34
	32	**32.83**	68.99	42.20	151.89
	64	31.82	**69.83**	**44.80**	159.62
	128	26.26	68.01	42.40	152.48
w	4	27.27	68.76	42.00	182.12
	8	30.30	68.16	43.60	193.30
	16	27.27	68.39	41.80	189.47
	32	28.28	67.48	43.60	173.53
	64	26.77	67.78	**44.80**	181.67
	128	29.29	69.60	43.60	173.11
Δ *Average*		+0.84	+0.20	-0.17	-20.13

icy stability during GRPO fine-tuning without an explicit KL term, while also reducing computational cost on average. Across models, mean runtime decreases by roughly 7–20% without KL, reflecting the omitted reference-policy computations and implementation-level differences in memory and compute across ranks.

To disentangle the contributions of low-rank constraints and explicit KL regularization, we conduct a controlled comparison on Gemma-3-1B-it with four configurations: LoRA (rank 16) and full fine-tuning, each with and without KL regularization. Fig. 2a shows that the low-rank constraint is the dominant factor controlling KL magnitude. Both the rank-16 runs remain an order of magnitude below their full fine-tuning counterparts throughout training, regardless of KL regularization. In contrast, the explicit KL penalty produces only a modest downward shift in both settings, where the LoRA stabilization can operate independently of explicit regularization in this controlled setting. Fig. 2b reveals a striking difference in how the two methods respond to removing KL regularization. In full fine-tuning, final rewards drop to zero without KL but remain at 3.0 with KL. However, LoRA rank 16 shows no such degradation: performance with and without KL regularization remains nearly identical at just below 4.0. This suggests that overall the low-rank constraint itself can provide regularization for fine-tuning.

These results support the qualitative intuition from Section 5 in our LLM experiments: the LoRA rank restrains the scale of policy divergence, and the theoretical $\sqrt{r}$ dependence holds qualitatively in full model training. The preserved accuracy and absence of reward collapse in unregularized runs supports our main result that LoRA acts as an implicit policy regularizer. Our comparison on Gemma-3-1B-it shows this mechanism operates independently: the low-rank constraint has a larger effect on KL magnitude than the explicit KL penalty in this ablation, and prevents the reward collapse observed in unregularized full fine-tuning. Within this scope, our results offer theoretical motivation and preliminary practical evidence that low-rank adaptation can reduce the need for explicit KL penalties in GRPO. Our empirical results are limited to 1B–3B models, 300-step GRPO runs on GSM8K, fixed hyperparameters, and a single random seed. Our findings should be read as preliminary evidence of stability without KL regularization, not superiority. Broader hyperparameter sweeps, longer horizons, larger models, and multi-task training are needed to assess generality.

7. Conclusion

This work establishes a theoretical and empirical connection between the low-rank geometry of LoRA and policy drift in reinforcement-style fine-tuning. We derive a rank-dependent upper bound on the KL divergence in a simplified theoretical setting using a single layer network with sigmoid activation and BCE loss, to give intuition on how policy updates constrained to a rank-r subspace might influence policy drift. This analysis follows standard practice in the literature (Ren & Sutherland, 2025; Zhang et al., 2025) and provides qualitative intuition for the empirical stability of LoRA-based reinforcement fine-tuning methods in short-horizon training. In controlled experiments on 1B–3B parameter models across three LLM families, LoRA-based GRPO appears to preserve accuracy and avoid reward collapse without the KL penalty over 300-step runs on GSM8K. Beyond its parameter efficiency, our results suggest that LoRA may also act as an implicit regularizer, limiting policy drift. Extending these findings to larger models, longer training horizons, and more diverse task distributions is an important direction for future work to determine if our findings could generalize.

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.

Acknowledgments

LIONS-EPFL was supported by the Swiss National Science Foundation (SNSF) under grant numbers 2000-1-240094 and 200021_205011; project ID # 37 as part of the Swiss AI Initiative, through a grant from the ETH Domain and computational resources provided by the Swiss National Supercomputing Centre (CSCS) under the Alps infrastructure; the Army Research Office and was accomplished under Grant Number W911NF-24-1-0048; Hasler Foundation Program: Hasler Responsible AI (project number 21043). This work was supported under project ID 114 as part of the Swiss AI Initiative, through a small grant from the ETH Domain and computational resources provided by the Swiss National Supercomputing Centre (CSCS) under the Alps infrastructure.

References

Clark, D. S. Short proof of a discrete Gronwall inequality. *Discrete Applied Mathematics*, 1987.

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. *arXiv preprint arXiv:2110.14168*, 2021.

Dayi, A. K. and Chen, S. Gradient dynamics for low-rank fine-tuning beyond kernels. *arXiv preprint, arXiv:2411.15385*, 2024.

Gemma Team. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Grattafiori, A. et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the MATH dataset. *NeurIPS*, 2021.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022.

Li, X., Li, Z., Kosuga, Y., and Bian, V. Optimizing safe and aligned language generation: A multi-objective GRPO approach. *arXiv preprint arXiv:2503.21819*, 2025.

Lin, Z., Lin, M., Xie, Y., and Ji, R. CPPO: Accelerating the training of group relative policy optimization-based reasoning models. In *NeurIPS*, 2025.

Liu, Z., Chen, C., Li, W., Qi, P., Pang, T., Du, C., Lee, W. S., and Lin, M. Understanding R1-Zero-Like training: A critical perspective. In *COLM*, 2025.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *ICLR*, 2019.

Patil, A. and Jadon, A. Advancing reasoning in large language models: Promising methods and approaches. *arXiv preprint arXiv:2502.03671*, 2025.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct Preference Optimization: Your language model is secretly a reward model. In *NeurIPS*, 2023.

Rein, D., Hou, B. L., Stickland, A. C., Petty, J., Pang, R. Y., Dirani, J., Michael, J., and Bowman, S. R. GPQA: A graduate-level Google-proof Q&A benchmark. In *First Conference on Language Modeling*, 2024.

Ren, Y. and Sutherland, D. J. Learning dynamics of LLM finetuning. In *ICLR*, 2025.

Santacroce, M., Lu, Y., Yu, H., Li, Y., and Shen, Y. Efficient RLHF: Reducing the memory usage of PPO. *arXiv preprint arXiv:2309.00754*, 2023.

Schulman, J. and Lab, T. M. LoRA without regret. *Thinking Machines Lab: Connectionism*, 2025. https://thinkingmachines.ai/blog/lora/.

Schulman, J., Levine, S., Abbeel, P., Jordan, M. I., and Moritz, P. Trust region policy optimization. In *ICML*, 2015.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Shah, V., Obando-Ceron, J., Jain, V., Bartoldson, B., Kailkhura, B., Mittal, S., Berseth, G., Castro, P. S., Bengio, Y., Malkin, N., et al. A comedy of estimators: On KL regularization in RL training of LLMs. *arXiv preprint arXiv:2512.21852*, 2025.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. F. Learning to summarize with human feedback. In *NeurIPS*, 2020.

Sun, S., Gupta, D., and Iyyer, M. Exploring the impact of low-rank adaptation on the performance, efficiency, and regularization of RLHF. *arXiv preprint arXiv:2309.09055*, 2023.

Vershynin, R. *High-dimensional probability: An introduction with applications in data science*. Cambridge university press, 2018.

Vieillard, N., Kozuno, T., Scherrer, B., Pietquin, O., Munos, R., and Geist, M. Leverage the average: an analysis of KL regularization in RL. In *NeurIPS*, 2020.

Wang, S., Asilis, J., Akgül, Ö. F., Bilgin, E. B., Liu, O., and Neiswanger, W. Tina: Tiny reasoning models via LoRA. *arXiv preprint arXiv:2504.15777*, 2025.

Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Yu, Q., Zhang, Z., Zhu, R., Yuan, Y., Zuo, X., YuYue, Dai, W., Fan, T., Liu, G., Liu, J., Liu, L., Liu, X., Lin, H., Lin, Z., Ma, B., Sheng, G., Tong, Y., Zhang, C., Zhang, M., Zhang, R., Zhang, W., Zhu, H., Zhu, J., Chen, J., Chen, J., Wang, C., Yu, H., Song, Y., Wei, X., Zhou, H., Liu, J., Ma, W.-Y., Zhang, Y.-Q., Yan, L., Wu, Y., and Wang, M. DAPO: An open-source LLM reinforcement learning system at scale. In *NeurIPS*, 2025.

Zhang, Y., Liu, F., and Chen, Y. LoRA-One: One-step full gradient could suffice for fine-tuning large language models, provably and efficiently. *arXiv preprint arXiv:2502.01235*, 2025.