Stabilizing Policy Optimization via Logits Convexity

Hongzhan Chen[1,2] **Tao Yang**[3*] **Yuhua Zhu**[1] **Shiping Gao**[1]
Xiaojun Quan[4*] **Ting Yao**[3]
[1]School of Computer Science and Engineering, Sun Yat-sen University
[2] Shanghai Innovation Institute [3] Wechat Search, Tencent Inc, China.
[4] Shenzhen Loop Area Institute
chenhzh59@mail2.sysu.edu.cn xiaojunquan@slai.edu.cn

Abstract

While reinforcement learning (RL) has been central to the recent success of large language models (LLMs), RL optimization is notoriously unstable, especially when compared to supervised fine-tuning (SFT). In this work, we investigate the stability gap between SFT and RL from a gradient-based perspective, and show that the convexity of the SFT loss with respect to model logits plays a key role in enabling stable training. Our theoretical analysis demonstrates that this property induces favorable gradient directionality during optimization. In contrast, Proximal Policy Optimization (PPO), a widely adopted policy gradient algorithm utilizing a clipped surrogate objective, lacks this stabilizing property. Motivated by this observation, we propose **Logits Convex Optimization (LCO)**, a simple yet effective policy optimization framework that aligns the learned policy with an optimal target derived from the original RL objective, leveraging logit-level convexity to ensure a well-behaved loss geometry. Extensive experiments across multiple model families show that our LCO framework consistently improves training stability and outperforms conventional RL methods on a broad range of benchmarks.

1 Introduction

Reinforcement learning (RL) has become a cornerstone for aligning large language models (LLMs) with human preferences [Ouyang et al., 2022, Bai et al., 2022] and enhancing complex capabilities such as reasoning [Guo et al., 2025, Yang et al., 2025a]. Despite these advances, RL training often suffers from inherent instability [Rafailov et al., 2024b]. Existing approaches attempt to address this issue through variance reduction in advantage estimation [Schulman et al., 2015b], clipping strategies that constrain parameter updates [Schulman et al., 2017, Yu et al., 2025], and KL-based penalties that regulate policy shifts [Ouyang et al., 2022, Shao et al., 2024]. Although these methods mitigate instability to some extent, it remains a persistent challenge in LLM optimization, and its underlying causes are still not fully understood [Team et al., 2025, Zhu et al., 2025].

In this work, we adopt a gradient-centric perspective and explore whether intrinsic properties of the loss landscape contribute to RL instability in LLM training. Under this lens, we find a contrast between RL and supervised fine-tuning (SFT). In widely used RL algorithms, such as Proximal Policy Optimization (PPO) [Schulman et al., 2017], the clipped surrogate objective (i.e., the importance sampling ratio) often produces volatile gradients, even when standard stabilization techniques like clipping or KL regularization are applied (Figure 1 (a)). These fluctuations can trigger excessively large parameter updates, potentially causing irreversible training collapse (Figure 1 (b)). In contrast, SFT typically follows a more stable optimization trajectory. This raises a fundamental question: *What explains the superior training stability of SFT compared to RL methods?*

*Corresponding author

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

By investigating the geometric properties of the optimization landscape, we identify a property termed *logits convexity*, defined as local convexity at the logits of LLMs. Our theoretical analysis in Section 4.3 demonstrates that logits convexity facilitates favorable gradient behavior during optimization. Specifically, under the assumption of the near-linear property of over-parameterized models, this property guarantees that the parameter-space gradient remains directionally aligned with the path toward the near-optimal parameters, as established in Proposition 4.3. This provides a guarantee that gradient descent is not misled by spurious stationary points that may arise from the loss landscape. While SFT loss exhibits logits convexity, which ensures well-behaved gradient updates, RL objectives such as PPO [Schulman et al., 2017] lack this property and are therefore susceptible to highly turbulent gradient dynamics that undermine training stability.

Building on these theoretical insights, we propose **Logits Convex Optimization (LCO)**, an RL optimization framework that preserves logits convexity and promotes stable training for LLMs. LCO reformulates the RL task as an optimal target matching problem, while remaining mathematically consistent with the original PPO objective that maximizes the expected advantage. Under this framework, we explore two primary approaches to improve the optimization landscape. First, we develop two direct alignment approaches (LCO-MSE and LCO-LCH) that target the optimal logits, ensuring strong convexity and enabling rapid convergence. Second, we examine a distribution-based approach (LCO-KLD) that aligns with the optimal distribution via forward KL divergence to ensure probabilistic consistency. By targeting the same optimal solution through a more well-behaved loss landscape, LCO produces reliable policy gradient updates and performance improvements (Figure 1 (c) and (d)).

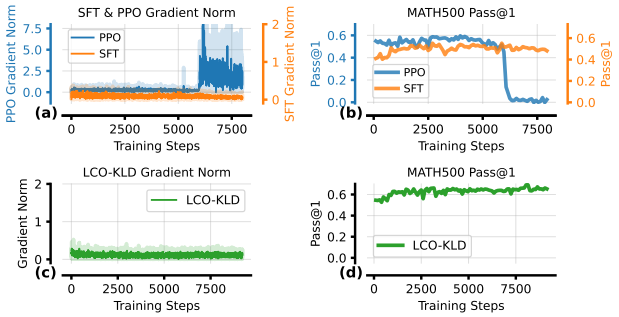


Figure 1: **(a)** Gradient norm during training for SFT and PPO. **(b)** Pass@1 results of SFT and PPO on the MATH500 benchmark. **(c)** Gradient norm during training for LCO-KLD. **(d)** Pass@1 results of LCO-KLD on MATH500.

Empirical results across mathematical reasoning, reading comprehension, and instruction-following tasks show that LCO outperforms standard RL baselines in both stability and performance. Moreover, our analysis yields several key findings. **First**, we identify a primary source of training instability when using surrogate objectives: excessively large gradient norms arising from negative samples within non-convex loss regions. **Second**, we reveal that sampled actions with low probability can cause sudden spikes in gradient updates, which affect the stability of methods such as PPO. **Third**, we show that the LCO objectives yield stable gradient updates that progressively diminish as training nears convergence, thereby mitigating instability in RL training.

2 Preliminary

2.1 Problem Setup

We model autoregressive text generation as a Markov decision process (MDP). At time step t, the state s_t consists of the prompt tokens together with all tokens generated up to step t. An action $a_t \in \mathcal{V}$ corresponds to selecting the next token from the vocabulary $\mathcal{V}$. Given state s_t at time step t, the policy π_θ defines a distribution over actions: $\pi_\theta(a_t|s_t) = \frac{\exp z_\theta(s_t,a_t)}{\sum_{a'_t \in \mathcal{V}} \exp z_\theta(s_t,a'_t)}$ where $z_\theta(s_t, a_t)$ is the logit associated with action a_t at state s_t, parameterized by θ.

2.2 Supervised Fine-Tuning

Supervised fine-tuning (SFT) trains the language model to maximize the likelihood of target (ground-truth) tokens a_t given the preceding context s_t. The objective is defined as:

$$\mathcal{L}_{\text{SFT}} = -\mathbb{E}_t \left[\log \pi_\theta(a_t|s_t)\right], \tag{1}$$

where the expectation $\mathbb{E}_t$ indicates the empirical average over a finite batch of samples.

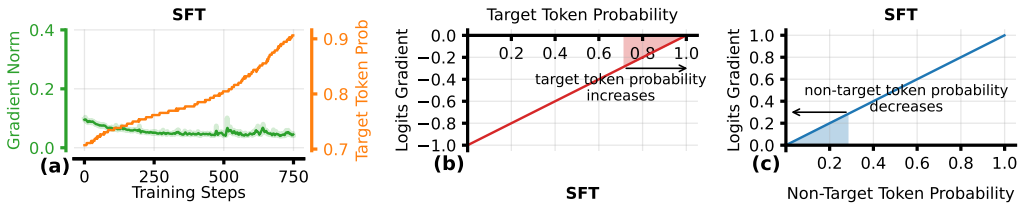


Figure 2: Training dynamics of SFT. **(a)** Gradient norm decreases as training progresses while target token probability on training sample increases. The magnitudes of both target token logit gradient **(b)** and the non-target token logit gradient **(c)** diminish as training progresses.

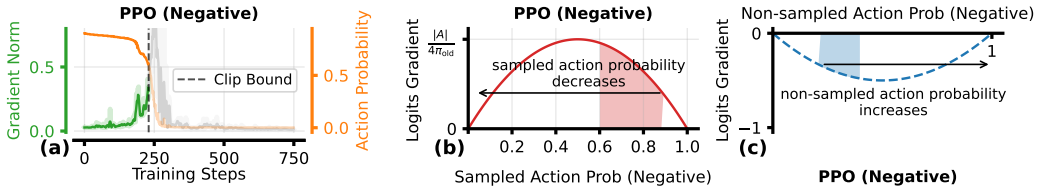


Figure 3: Training dynamics of PPO for actions with negative advantages. **(a)** Gradient norm increases before being clipped, while sampled action probabilities decrease. The magnitudes of both sampled action logit gradient **(b)** and non-sampled action logit gradient **(c)** increase before being clipped. The results show a correlation between gradients of parameter-space and logit-space.

2.3 Proximal Policy Optimization

Proximal Policy Optimization (PPO) [Schulman et al., 2017] is a widely used policy-gradient method for reinforcement learning. A common PPO-style objective augments the expected advantage with a KL penalty that constrains the policy π_θ to remain close to a behavioral policy π_{old}:

$$\max_{\pi_\theta} \mathbb{E}_{a_t \sim \pi_\theta(\cdot|s_t)}[A(s_t, a_t)] - \beta \mathbb{D}_{\text{KL}}(\pi_\theta(\cdot|s_t) \| \pi_{\text{old}}(\cdot|s_t)), \tag{2}$$

where $A(s_t, a_t)$ denotes the advantage function and $\beta > 0$ controls the strength of the KL regularization. Rather than optimizing this KL-regularized objective directly, PPO introduces a clipped surrogate objective that heuristically constrains policy updates by limiting change in action probabilities between successive policies.

$$\mathcal{L}_{\text{PPO}} = -\mathbb{E}_t[\min(r_t(\theta)A(s_t, a_t), \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)A(s_t, a_t))], \tag{3}$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\text{old}}(a_t|s_t)}$, clip$(\cdot)$ is the clip function and the ϵ is the clip range. This clipped surrogate objective prevents excessively large policy updates, thereby improving training stability compared to standard policy gradient methods.

3 Gradient Dynamics

In this section, we empirically analyze the distinct gradient dynamics of $\mathcal{L}_{\text{SFT}}$ and $\mathcal{L}_{\text{PPO}}$, and demonstrate how these dynamics influence logits gradients and training stability.

3.1 Gradient Dynamics of SFT

We analyze the gradient behavior of SFT starting from the standard negative log-likelihood objective in Equation (1). At a time step t, let a_t denote the target (or ground-truth) token and a'_t be an arbitrary token in the vocabulary $\mathcal{V}$. The derivative of $\mathcal{L}_{\text{SFT}}$ with respect to the logit $z_\theta(s_t, a'_t)$ is:

$$\frac{\partial \mathcal{L}_{\text{SFT}}}{\partial z_\theta(s_t, a'_t)} = \pi_\theta(a'_t|s_t) - \mathbb{I}_{a_t = a'_t}, \tag{4}$$

where $\mathbb{I}_{a_t = a'_t}$ is the indicator (derivation is in Appendix E.1). The logit gradient of a target token a_t is $\pi_\theta(a_t|s_t) - 1$, which pushes the model to increase the target logit. While for a non-target

token a'_t ($a_t \neq a'_t$), the logit gradient equals $\pi_\theta(a'_t|s_t)$, which pushes the model to decrease the non-target logit. Figure 2 (a) illustrates the overall gradient dynamics, while Figures 2 (b) and (c) depict the logit gradient dynamics. During training, target token probabilities increase and gradient norms decrease. We observe a parallel trend in the logit gradients: as target token probabilities asymptotically increase and the probabilities of non-target tokens vanish, the corresponding logit gradient magnitudes diminish correspondingly, reflecting convergence. This behavior aligns with the intuition that as the model nears optimality, the parameter updates naturally become smaller.

3.2 Gradient Dynamics of PPO

The clipping mechanism in PPO restricts parameter updates by nullifying the gradient when the ratio $r_t(\theta)$ moves outside a predefined interval. Specifically, the gradient is non-zero only when the following condition is satisfied: ($A(s_t, a_t) > 0$ and $r_t(\theta) < 1 + \epsilon$) or ($A(s_t, a_t) < 0$ and $r_t(\theta) > 1 - \epsilon$). If this condition is not met, the objective enters the clipped region, and the gradient becomes zero. Consequently, we focus our analysis on the active region where the gradient is non-zero, and examine the dynamics of the gradient. At time step t, let a_t be the sampled action and a'_t be an arbitrary action in the vocabulary $\mathcal{V}$. Under this setting, the gradient of $\mathcal{L}_{\text{PPO}}$ with respect to the logit $z_\theta(s_t, a'_t)$ is given by (derivation is in Appendix E.2):

$$\frac{\partial \mathcal{L}_{\text{PPO}}}{\partial z_\theta(s_t, a'_t)} = \frac{A(s_t, a_t)}{\pi_{\text{old}}(a_t|s_t)} \pi_\theta(a_t|s_t)(\pi_\theta(a'_t|s_t) - \mathbb{I}_{a_t = a'_t}). \tag{5}$$

We focus on the case where $A(s_t, a_t) < 0$ (for $A(s_t, a_t) > 0$, see Figure 7 in the Appendix). Figure 3 (a) shows that gradient norm exhibits an increase before being clipped. A similar phenomenon can also be observed in logit gradients (Figure 3 (b) and (c)). **A counterintuitive phenomenon emerges: as training progresses, the loss decreases while the gradient norms grow**. A comparison of the gradient norms for positive advantages reveals that updates are largely dominated by actions with negative advantages due to their disproportionately larger gradient magnitudes. This dominance results in a systemic reduction of action probabilities in subsequent sampling. Such gradient spikes typically manifest for actions with intermediate probabilities (e.g., near 0.5), triggering massive parameter updates that destabilize the training process. Furthermore, our analysis indicates that without the clipping mechanism, the gradient norm would exhibit initial increases followed by decreases, aligning closely with the dynamics of the logit gradients.

4 On the Convexity of Logits

Drawing upon the aforementioned analysis of gradient dynamics, we propose *Logits Convex Optimization* (*LCO*), a general training paradigm designed to stabilize the training of LLMs in reinforcement learning. The fundamental principle of LCO is to reformulate the complex RL task as a supervised alignment problem toward an optimal target derived from the original objective. We start from the standard regularized RL objective in Eq. (2). Under this objective, the optimal policy π^* that maximizes Equation (2) at time step t has the following form, as established in [Peng et al., 2019]:

$$\pi^*(a_t|s_t) = \frac{\pi_{\text{old}}(a_t|s_t) \exp\left[\frac{A(s_t,a_t)}{\beta}\right]}{\sum_{a'_t \in \mathcal{V}} \pi_{\text{old}}(a'_t|s_t) \exp\left[\frac{A(s_t,a'_t)}{\beta}\right]}, \tag{6}$$

where π_{old} is the behavioral policy and $A(s_t, a_t)$ is the advantage function. A specific solution for the optimal logits $z^*(s_t, a_t)$ that induces π^* is obtained by direct advantage-based adjustment:

$$z^*(s_t, a_t) = z_{\text{old}}(s_t, a_t) + \frac{A(s_t, a_t)}{\beta}, \tag{7}$$

where $z_{\text{old}}(s_t, a_t)$ is the logit of behavioral policy π_{old} corresponding to the action a_t and state s_t. We provide detailed derivation in Appendix F. The general objective of LCO is to fit the parameterized policy π_θ to these optimal targets (either the optimal distribution π^* or the optimal logits z^*). Under this framework, we propose two distinct implementation strategies: regression-based alignment and distribution-based alignment.

4.1 LCO Objectives

Regression-based LCO Objectives Following the characterization of the optimal logits in Equation (7), we first define the LCO objective as the minimization of the discrepancy between the target optimal logits z^* and logits z_θ of the parameterized policy π_θ at each time step. To enforce this alignment, we propose two variants of the LCO objective using standard regression metrics: MSE and log-cosh loss. The first variant applies MSE, yielding the $\mathcal{L}_{\text{LCO-MSE}}$:

$$\mathcal{L}_{\text{LCO-MSE}} = \mathbb{E}_t \left[\frac{1}{|\mathcal{V}|} \sum_{a_t \in \mathcal{V}} \left(z_\theta(s_t, a_t) - z^*(s_t, a_t) \right)^2 \right], \tag{8}$$

where $\mathcal{V}$ is the vocabulary. Moreover, to provide a smoother optimization landscape that is more robust to outliers, we define the log-cosh [Saleh and Saleh, 2022] variant $\mathcal{L}_{\text{LCO-LCH}}$ as follows:

$$\mathcal{L}_{\text{LCO-LCH}} = \mathbb{E}_t \left[\frac{1}{|\mathcal{V}|} \sum_{a_t \in \mathcal{V}} \log\cosh\left(z_\theta(s_t, a_t) - z^*(s_t, a_t) \right) \right]. \tag{9}$$

The $\log\cosh(\cdot)$ loss behaves like MSE for small errors but transitions to a linear penalty for large discrepancies, ensuring gradient stability even with noisy advantage estimates.

Distribution-based LCO Objective While the regression-based losses above directly target logit values, we can also formulate the objective in terms of the optimal policy distribution. We then define $\mathcal{L}_{\text{LCO-KLD}}$ as the minimization of the forward KL divergence between the optimal policy π^* and the parameterized policy π_θ:

$$\mathcal{L}_{\text{LCO-KLD}} = \mathbb{E}_t \left[\sum_{a_t \in \mathcal{V}} \pi^*(a_t|s_t) \log \frac{\pi^*(a_t|s_t)}{\pi_\theta(a_t|s_t)} \right]. \tag{10}$$

Theoretically, since π^* is the unique global maximizer of Equation (2), minimizing any LCO loss (e.g., $\mathcal{L}_{\text{LCO-MSE}}$, $\mathcal{L}_{\text{LCO-LCH}}$, or $\mathcal{L}_{\text{LCO-KLD}}$) is equivalent to solving the original RL problem, provided the parameterized policy π_θ has sufficient capacity to represent π^* or z^*.

4.2 Advantage Estimation

The LCO framework requires advantage signals $A(s_t, a_t)$ to construct the optimal target. Depending on the availability of feedback, we employ three strategies to estimate this advantage, ranging from sparse, sample-based signals to dense, distribution-level signals.

Sampled Action Sparse Estimation In traditional reinforcement learning settings, such as PPO, advantage values are typically only available for the action a_t actually sampled from the behavioral policy. For any other non-sampled action a'_t in the vocabulary $\mathcal{V}$, the advantage signal is unknown. In this case, we adopt a sparse estimation strategy:

$$\hat{A}(s_t, a'_t) = \begin{cases} A(s_t, a_t), & \text{if } a'_t = a_t, \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

Log Probability-Based Dense Estimation A more comprehensive approach provides training signals across the entire vocabulary $\mathcal{V}$. Building on the theoretical framework of Ziebart [2010], the advantage at the distribution level can be derived from the log probability assigned by an LLM ϕ:

$$\log \phi(a_t|s_t) = Q(s_t, a_t) - V(s_t), \tag{12}$$

where $Q(s_t, a_t)$ is the Q-function and $V(s_t)$ is the value function. Using the soft Bellman equation, the advantage estimator is therefore defined as: $A(s_t, a_t) := \log \phi(a_t|s_t)$.

DPO-Based Preference Estimation As provided by Rafailov et al. [2024a], the log ratio between a DPO-tuned LLM ϕ_{DPO} and a reference LLM ϕ_{ref} can serve as a token-level advantage signal for RL training. This provides an implicit representation of the advantage at time step t:

$$\log \frac{\phi_{\text{DPO}}(a_t|s_t)}{\phi_{\text{ref}}(a_t|s_t)} = r(s_t, a_t) + V(s_{t+1}) - V(s_t). \tag{13}$$

The advantage estimator is defined as: $A(s_t, a_t) := \log \frac{\phi_{\text{DPO}}(a_t|s_t)}{\phi_{\text{ref}}(a_t|s_t)}$. Due to its natural alignment with the Bradley-Terry preference model and its widespread use in previous literature [Zhong et al., 2024, Chen et al., 2025b], we adopt this dense estimator as the default for our primary experiments.

4.3 Gradient Analysis of LCO

To characterize the optimization landscapes of different training objectives for LLMs, we first establish the relationship between the parameter-space gradient and the functional change in the logit space. Consider a loss function $\mathcal{L}$ with a finite lower bound $\mathcal{L}^* > -\infty$. Let $\boldsymbol{z}_\theta \in \mathbb{R}^{|\mathcal{V}|}$ denote the logit vector produced by an LLM with parameters θ. We define $\Theta^* = \{\theta^* : \mathcal{L}(\boldsymbol{z}_{\theta^*}) - \mathcal{L}^* < \epsilon\}$ as the set of near-optimal parameters for a given tolerance $\epsilon > 0$.

Assumption 4.1. For any $\theta^* \in \Theta^*$ in the local neighborhood of θ, we assume that the network operates in a regime where the parameter displacement $\Delta\theta = \theta^* - \theta$ is sufficiently small, so that the first-order Taylor expansion:

$$\boldsymbol{z}_{\theta^*} \approx \boldsymbol{z}_\theta + \nabla_\theta \boldsymbol{z}_\theta^\top \Delta\theta \tag{14}$$

holds with negligible higher-order residual terms.

This assumption aligns with Neural Tangent Kernel theory [Jacot et al., 2018] and is supported by other literature: in over-parameterized networks (especially LLMs), individual parameters change negligibly while their collective contribution to the final output is significant [Lee et al., 2019]. We now introduce the *logits convexity*, which serves as an indicator for well-behaved optimization.

Definition 4.2 (Logits Convexity)**.** Let $\mathcal{L}$ be a twice-differentiable loss function that takes logits $\boldsymbol{z}_\theta$ parameterized by θ as input. The $\mathcal{L}$ is called logits convex if its Hessian matrix with respect to all $\boldsymbol{z}_\theta$ is positive semi-definite.

The following proposition establishes why this property is desirable for favorable parameter updates:

Proposition 4.3 (Gradient Directionality)**.** *Under Assumption 4.1, if $\mathcal{L}$ is logits convex, then the parameter-space gradient and logit-space gradient satisfy:*

$$\langle \nabla_\theta \mathcal{L}, \theta - \theta^* \rangle \approx \langle \nabla_{\boldsymbol{z}_\theta} \mathcal{L}, \boldsymbol{z}_\theta - \boldsymbol{z}_{\theta^*} \rangle \geq 0. \tag{15}$$

Derivation can be found in Appendix G. Proposition 4.3 implies that the negative gradient of the loss $\mathcal{L}$ with logits convexity consistently points toward the near-optimal parameter θ^* in a local sense. This provides a theoretical guarantee that gradient descent on those objectives is not misled by spurious stationary points that may arise from non-convexity in the parameter landscape. Below, we analyze the logits convexity of SFT, PPO, and LCO objectives:

Lemma 4.4. *The SFT loss function $\mathcal{L}_{SFT}$, as defined in Equation* (1)*, is logits convex at each time step. In contrast, the PPO loss function $\mathcal{L}_{PPO}$, defined in Equation* (3)*, is not logits convex at any time step. (Proof. See Appendix H.1.)*

These findings elucidate the disparity in gradient behavior observed between SFT and PPO. The non-convexity of the PPO objective in logit space can lead to a violation of gradient directionality, resulting in the erratic gradient spikes and explosions frequently encountered in empirical settings. In contrast, LCO restores this desirable optimization property:

Lemma 4.5. *The LCO loss functions $\mathcal{L}_{LCO\text{-}MSE}$, $\mathcal{L}_{LCO\text{-}LCH}$, and $\mathcal{L}_{LCO\text{-}KLD}$, defined in Equations* (8) *to* (10)*, are all logits convex at each time step. Specifically, $\mathcal{L}_{LCO\text{-}MSE}$ is strongly convex, while $\mathcal{L}_{LCO\text{-}LCH}$ is locally strongly convex on any bounded set of logits. (Proof. See Appendix H.2.)*

While standard PPO attempts to mitigate this instability through heuristic constraints like clipping or trust regions, the LCO objectives offer a more fundamental solution grounded in Proposition 4.3. By ensuring convexity in the logit space, the parameter-space optimization naturally inherits a well-behaved landscape, ensuring that the model updates remain stable throughout the training.

Complementing the guarantee of favorable gradient directionality, we further establish that the gradient norms of LCO objectives are bounded by monotonic functions of the loss. This property ensures gradient magnitude naturally dissipates as the model approaches the optimal target:

Proposition 4.6 (Gradient Norm Upper Bounds)**.** *We consider LCO objectives defined in Equations* (8) *to* (10) *at time step t. Let $\sigma_{\max}$ denote the maximum singular value of $\nabla_\theta \boldsymbol{z}_\theta$. The gradient norm upper bound of $\mathcal{L}_{LCO\text{-}MSE}$ is:*

$$\|\nabla_\theta \mathcal{L}_{LCO\text{-}MSE}\| \leq \frac{2}{|\mathcal{V}|} \sigma_{max} \sqrt{|\mathcal{V}| \mathcal{L}_{LCO\text{-}MSE}}. \tag{16}$$

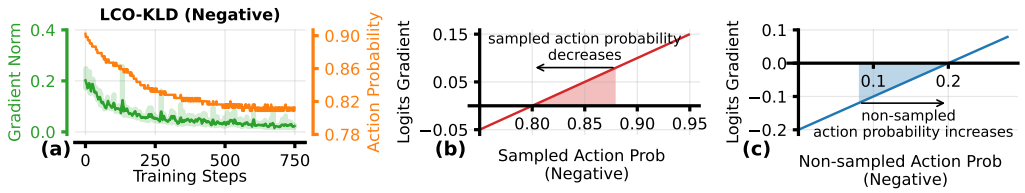


Figure 4: Training dynamics of LCO-KLD for actions with negative advantages. Unlike training dynamics of PPO, LCO-KLD exhibits stable gradient updates that scale with the remaining error and diminish progressively as the model converges.

The gradient norm upper bound of $\mathcal{L}_{\text{LCO-LCH}}$ is:

$$\|\nabla_\theta \mathcal{L}_{\text{LCO-LCH}}\| \leq \frac{1}{|\mathcal{V}|}\sigma_{max}\sqrt{|\mathcal{V}|(1-\exp(-2\mathcal{L}_{\text{LCO-LCH}}))}. \tag{17}$$

The gradient norm upper bound of $\mathcal{L}_{\text{LCO-KLD}}$ is:

$$\|\nabla_\theta \mathcal{L}_{\text{LCO-KLD}}\| \leq \sigma_{max}\sqrt{2\mathcal{L}_{\text{LCO-KLD}}}. \tag{18}$$

Proof. See Appendix I. □

Proposition 4.6 elucidates an inherent self-stabilizing mechanism within LCO: the update magnitude is coupled to the remaining error. Unlike traditional RL objectives that often require heuristic gradient clipping to prevent divergence, LCO's gradient pressure subsides gracefully as the policy aligns with the target. This behavior effectively precludes the gradient spikes that typically destabilize RL training. As visualized in Figures 4 and 8, the optimization landscape remains smooth, with empirical gradient trajectories confirming a controlled decay toward the optimum. Building upon these gradient properties, we formalize the convergence behavior of the LCO framework:

Proposition 4.7 (LCO Convergence Guarantees)**.** *Under Assumption 4.1, for a sufficiently small step size, the LCO objectives $\mathcal{L}_{\text{LCO-MSE}}$, $\mathcal{L}_{\text{LCO-LCH}}$, and $\mathcal{L}_{\text{LCO-KLD}}$ converge at a sublinear rate of at least $\mathcal{O}(1/K)$, where K is the number of iterations. (Proof. See Appendix J.)*

By providing both a bounded gradient norm and a guaranteed convergence rate, LCO offers a robust theoretical foundation for the policy optimization, ensuring efficient and stable training process.

5 Experimental Setup

Evaluation Tasks We evaluate the LCO framework across three primary domains: mathematical reasoning (AMC23 [MAA, 2023], MATH500 [Lightman et al., 2023], and MinervaMath [Lewkowycz et al., 2022]), machine reading comprehension (QA-Feedback [Wu et al., 2023]), and instruction following (AlpacaEval 2.0 [Dubois et al., 2024]). For math reasoning, we report the Pass@1 performance. For QA-Feedback, following Wu et al. [2023], we utilize reward models for relevance, factuality, and completeness, reporting the policy's reward scores on the test set. For AlpacaEval 2.0, we assess response quality relative to GPT-4-1106-Preview using win rate and length-controlled (LC) win rate, with Azure GPT-4o-2024-11-20 serving as judge.

Training Data We first detail the datasets employed for RL training. For mathematical reasoning, we utilize the original training instructions from the MATH dataset, with 500 samples randomly held out as an evaluation set. For QA-Feedback, we adopt its original training instructions. For AlpacaEval 2.0, we utilize instructions from UltraFeedback [Cui et al., 2024]. Regarding RM training, the datasets are curated as follows. For mathematical reasoning, we sample responses from Llama-3 (8B/70B) and Qwen-2.5 (32B/72B) using MATH training instructions. Responses are labeled as chosen or rejected based on their correctness against ground truth, yielding 30,183 pairs after filtering. For QA-Feedback, we use its original preference data, totaling 17,835 pairs. For AlpacaEval 2.0, UltraFeedback serves as the preference dataset. For SFT warm-up, we select 1k chosen responses from each of the above datasets, and train the policy to ensure initial instruction-following capabilities.

Baselines We evaluate our approach against several representative baselines, including REINFORCE [Williams, 1992] and PPO [Schulman et al., 2017], as well as GRPO [Shao et al., 2024] and its stability-enhanced variants, namely DAPO [Yu et al., 2025] and GSPO [Zheng et al., 2025]. To broaden the scope of evaluation, we also include on-policy distillation methods such as MiniLLM [Gu et al., 2023] and GKD [Agarwal et al., 2024]. Furthermore, the RM and π^* defined in Equation (6) are also included. For a fair comparison, all methods are configured following their original reports and the default settings in the TRL library[2].

Training Details Following Zhong et al. [2024], we train an LLM via DPO to serve as the RM. In our main experiments, RMs are built on various backbones, including Qwen-2.5-7B, Qwen-3-8B, Llama-3.1-8B, and Mistral-3-8B. RM training is conducted for 1 epoch with a learning rate of 1e-6 to mitigate overfitting risks. Regarding RL training, we evaluate policy backbones including Qwen-2.5-3B, Qwen-3-4B, Llama-3.2-3B, and Mistral-3-3B. Each policy is consistently paired with an RM from its corresponding model series (e.g., a Qwen-2.5-3B policy with a Qwen-2.5-7B RM). We set the sequence length at 2,048. For GRPO, DAPO, and GSPO, we sample four responses per instruction, while other methods use one. We apply a sampling temperature of 0.6 and top-p of 0.95. For the LCO objectives, we adopt a default $\beta = 1.0$ and a learning rate of 5e-6. Evaluation results are reported as the mean and standard deviation across four random seeds.

6 Results and Analysis

6.1 Main Results

Math Reasoning Table 1 summarizes the performance on mathematical reasoning tasks using Qwen-2.5-3B and Qwen-3-4B as base models. LCO objectives surpass competitive RL baselines across most benchmarks such as MATH500 and MinervaMath. Notably, when built upon Qwen-2.5-3B, LCO-LCH yields the top Pass@1 score of 61.40% on MATH500, while LCO-KLD excels on AMC23 (42.65%) and MinervaMath (16.71%). Similarly, using Qwen-3-4B, LCO-KLD achieves state-of-the-art performance on MATH500 (73.20%) and AMC (55.50%). Analysis of LCO variants reveals that while LCO-MSE is sensitive to advantage noise, LCO-LCH mitigates outlier influence through log-cosh penalty, contributing to its better performance on Qwen-3-4B. Meanwhile, LCO-KLD demonstrates the highest robustness by leveraging probabilistic consistency across the action space. Furthermore, we observe that LCO surpasses the performance of π^*. Unlike a fixed target derived from the initial π_{old} and advantage, LCO facilitates a dynamic optimization process that continuously aligns with evolving targets. Notably, most LCO variants even exceed the performance of the RM ϕ_{DPO} despite its larger parameter count. These results demonstrate the versatility of LCO in improving mathematical reasoning performance across diverse model backbones.

Table 1: Main results of Qwen-2.5-3B and Qwen-3-4B on math reasoning tasks with standard error. Best performances are shown in **bold**, while suboptimal ones are underlined.

Methods	Qwen-2.5-3B MATH500 Pass@1	Qwen-2.5-3B AMC23 Pass@1	Qwen-2.5-3B MinervaMath Pass@1	Qwen-3-4B MATH500 Pass@1	Qwen-3-4B AMC23 Pass@1	Qwen-3-4B MinervaMath Pass@1
ϕ_{DPO}	58.30±1.83	35.70±2.37	16.17±1.56	69.20±2.31	58.20±3.18	16.54±1.34
π^*	51.30±1.53	28.40±2.49	15.18±0.81	61.35±1.67	47.80±3.10	14.39±0.71
SFT	41.60±0.72	27.50±1.44	10.29±0.71	58.80±0.50	45.65±1.81	12.53±0.34
REINFORCE	54.60±0.53	40.15±1.25	12.35±0.93	64.80±0.68	46.65±2.43	19.48±0.41
PPO	55.40±1.02	39.20±2.01	13.97±0.81	67.80±0.83	47.50±1.25	20.95±0.28
GRPO	57.79±0.34	41.75±1.25	14.89±0.87	67.60±1.23	51.95±1.92	21.75±0.19
DAPO	57.05±0.31	40.20±1.46	13.23±0.78	67.80±0.78	53.15±2.45	19.75±0.34
GSPO	56.80±1.38	38.75±2.01	14.87±0.71	66.40±1.64	51.20±2.07	18.38±0.45
LCO-MSE	59.20±1.75	40.20±1.58	15.12±0.43	67.20±1.35	51.30±1.23	21.37±0.36
LCO-LCH	**61.40±2.01**	39.50±2.37	15.38±0.16	69.40±1.98	52.70±2.45	**24.26±0.21**
LCO-KLD	60.80±0.51	**42.65±1.63**	**16.71±0.97**	**73.20±1.34**	**55.50±1.88**	23.95±0.31

Machine Reading Comprehension We report the QA-Feedback results in Table 2, employing Llama-3.2-3B and Mistral-3-3B as backbones. Performance is measured by three oracle RMs: Relevance, Factuality, and Completeness. LCO variants consistently demonstrate superiority over all baselines. Specifically, LCO-KLD achieves the highest average rewards of 0.607 and 0.581 on Llama-3.2-3B and Mistral-3-3B, respectively, signif-

Table 2: Results for QA-Feedback in relevance (Rel.), factuality (Fact.), and completeness (Comp.) with standard error.

Methods	Llama-3.2-3B Rel. $R_1(\uparrow)$	Fact. $R_2(\uparrow)$	Comp. $R_3(\uparrow)$	Avg(↑)	Mistral-3-3B Rel. $R_1(\uparrow)$	Fact. $R_2(\uparrow)$	Comp. $R_3(\uparrow)$	Avg(↑)
ϕ_{DPO}	0.369	0.737	0.549	0.552±0.009	0.412	0.658	0.559	0.556±0.013
π^*	0.321	0.691	0.514	0.509±0.010	0.416	0.597	0.528	0.514±0.003
SFT	0.352	0.597	0.488	0.479±0.011	0.427	0.589	0.513	0.510±0.013
REINFORCE	0.280	0.632	0.506	0.473±0.005	0.416	0.629	0.536	0.527±0.011
PPO	0.374	0.627	0.507	0.503±0.007	0.431	0.614	0.531	0.525±0.002
GRPO	0.342	0.708	0.516	0.522±0.009	0.410	0.622	0.529	0.520±0.005
DAPO	0.337	0.721	0.516	0.525±0.003	0.412	0.625	0.528	0.522±0.003
GSPO	0.332	0.724	0.516	0.524±0.006	0.392	0.639	0.522	0.518±0.007
LCO-MSE	0.378	0.793	0.539	0.570±0.005	0.395	0.669	0.552	0.539±0.003
LCO-LCH	0.383	0.814	0.543	0.580±0.002	0.396	0.639	0.577	0.537±0.004
LCO-KLD	**0.392**	**0.817**	**0.612**	**0.607±0.003**	**0.449**	**0.673**	**0.620**	**0.581±0.006**

[2] https://github.com/huggingface/trl

icantly surpassing PPO (0.503 and 0.525) and RM ϕ_{DPO} (0.552 and 0.556). Compared to SFT and π^*, LCO variants exhibit consistent gains, with LCO-KLD achieving 0.607 on Llama-3.2-3B and 0.581 on Mistral-3-3B. Moreover, all LCO variants surpass the competitive DAPO and GSPO baselines, especially in factuality (up to 0.817) and completeness (up to 0.620). These results highlight the effectiveness of LCO objectives in aligning policies to generate high-quality responses.

Instruction Following Table 3 illustrates that LCO objectives yield substantial gains in the AlpacaEval 2.0 benchmark. Specifically, LCO-KLD with a Qwen-3-4B backbone attains 29.05% WR and 32.93% LC WR, surpassing PPO (26.79%/27.20%), GRPO (25.37%/26.38%), and GSPO (25.67%/26.17%) by a clear margin. This represents an absolute improvement of over 6 percentage points in LC WR compared to the strongest baseline, demonstrating the effectiveness of logit-space convexity in aligning model outputs with human preferences. With Llama-3.2-3B, LCO-LCH and LCO-KLD achieve their highest comparative win rates at 25.37% and 24.38%, respectively, while under Mistral-3-3B, LCO-KLD achieves the highest win rate of 26.81%. Notably, LCO-LCH consistently ranks among the top two methods across all three backbones, suggesting that this logit convexity variant offers a particularly robust trade-off between training stability and alignment performance. These improvements across diverse model families underscore the robustness of LCO objectives in instruction-following tasks, confirming that the proposed framework generalizes well beyond any single model configuration.

Table 3: Results for AlpacaEval 2.0 with standard error.

Methods	Qwen-3-4B		Llama-3.2-3B		Mistral-3-3B	
	WR (%)	LC WR (%)	WR (%)	LC WR (%)	WR (%)	LC WR (%)
ϕ_{DPO}	32.09±1.67	28.56±0.10	19.74±1.33	21.34±0.05	26.78±0.72	27.32±0.17
π^*	25.12±0.91	24.31±0.09	18.31±0.72	21.18±0.16	23.58±0.67	24.17±0.11
SFT	23.99±1.49	25.95±0.01	17.73±1.25	18.04±0.12	19.39±1.31	22.29±0.07
REINFORCE	25.14±1.37	27.18±0.02	21.85±1.32	22.68±0.56	21.57±1.82	23.14±0.04
PPO	26.79±1.45	27.20±0.07	23.74±1.21	25.76±0.71	25.81±1.45	26.41±0.05
GRPO	25.37±1.65	26.38±0.21	23.72±1.32	24.71±0.43	25.70±1.23	**27.81±0.09**
DAPO	24.67±1.37	25.18±0.18	22.07±1.22	24.81±0.34	24.91±1.19	26.71±0.08
GSPO	25.67±1.12	26.17±0.11	24.17±1.98	25.13±0.32	23.85±1.71	27.31±0.13
LCO-MSE	27.01±1.62	29.37±0.03	23.87±1.32	25.86±0.32	25.31±1.01	26.87±0.04
LCO-LCH	27.65±1.43	31.91±0.02	**25.37±1.21**	**26.82±0.09**	26.71±1.32	27.41±0.02
LCO-KLD	**29.05±1.49**	**32.93±0.03**	24.38±1.16	25.55±0.12	**26.81±1.23**	27.10±0.03

6.2 Ablation Study and Analysis

Training Dynamics Analysis To investigate how LCO stabilizes RL training process, we compare training dynamics of LCO-KLD and PPO in Figure 5. (1) *Gradient norms*: the gradient norm dynamics are illustrated in Figure 5(a) and (b). As training progresses, PPO remains relatively stable during the early stages but begins to oscillate after approximately 6K steps. In contrast, the gradient norm of LCO-KLD remains stable throughout the entire training. (2) *Entropy and action probabilities*: we analyze the average entropy of the action distribution and probabilities of sampled actions to evaluate the policy's exploration capability. As shown in Figure 5(a-1) and (b-1), PPO exhibits a sharp drop in sampled action probabilities and a surge in entropy during later stages of training. This phenomenon is strongly linked to oscillations in gradient norms and likely stems from excessive gradients caused by negative advantages. However, LCO-KLD maintains stable entropy and action probabilities, preserving exploration while ensuring effective optimization. (3) *Performance*: as shown in Figure 5(a-2) and (b-2), PPO suffers a performance decline in the later stages due to training collapse, whereas LCO-KLD achieves consistent improvement and ultimately surpasses PPO.

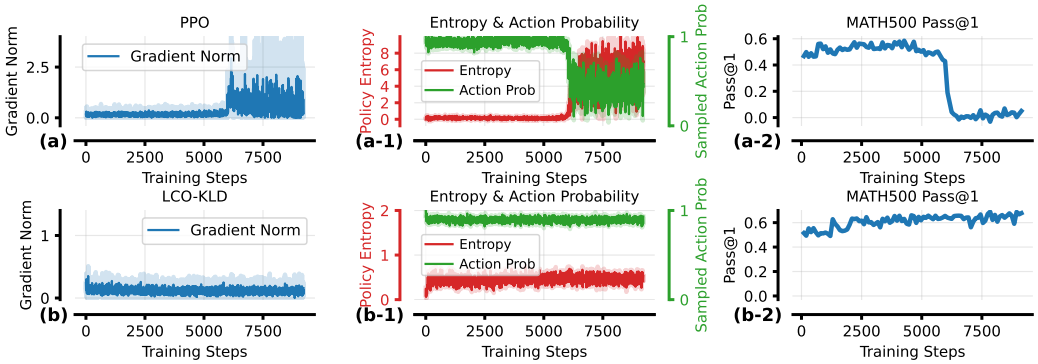


Figure 5: (a) and (b) show the gradient norm dynamics of PPO and LCO-KLD, respectively. (a-1) and (b-1) show the policy entropy and sampled action probability dynamics, while (a-2) and (b-2) present the corresponding Pass@1 performance on the MATH500 dataset.

Training Efficiency To assess the training efficiency of the LCO objectives, we compare the performance of policies based on Qwen-3-4B and Qwen-2.5-3B backbones, trained using LCO-LCH, LCO-KLD, and PPO. As illustrated in Figure 6(a), with the Qwen-2.5-3B backbone, LCO-KLD achieves a Pass@1 comparable to PPO while requiring nearly half the training samples, owing to the logits convexity of LCO-KLD, which accelerates convergence. LCO-LCH further enhances sample efficiency. As depicted in Figure 6(a) and (b), LCO-LCH achieves a Pass@1 similar to LCO-KLD but is nearly three times more sample-efficient. This greater efficiency stems from the stronger convexity of LCO-LCH in the logit space (Lemma 4.5), ensuring a faster convergence rate and speeds up training.

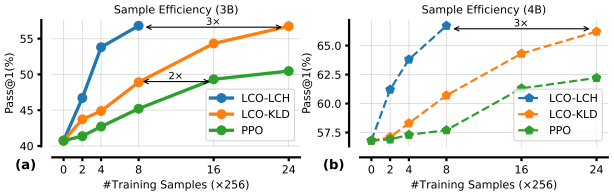


Figure 6: Training efficiency of LCO-LCH/KLD and PPO, evaluated by policy accuracy on MATH500 evaluation set.

Compare to On-Policy Distillation We further compare the LCO-KLD objective against two on-policy distillation methods, GKD and MiniLLM. For a fair comparison, we use the same RMs (ϕ_{DPO}) as teachers in these baselines, consistent with LCO-KLD. As presented in Table 4, LCO-KLD demonstrates competitive performance across three benchmarks. While GKD and MiniLLM primarily focus on mimicking the teacher distribution, LCO-KLD incorporates advantage information derived from original RL objective to guide the policy alignment. This performance gap underscores that LCO-KLD benefits from the dynamic nature of RL, which allow policy to potentially surpass the "performance ceiling" often associated with traditional distillation.

Table 4: Comparison with on-policy distillation methods.

Methods	Qwen-2.5-3B			Qwen-3-4B		
	MATH500	AMC23	MinervaMath	MATH500	AMC23	MinervaMath
MiniLLM	55.20±0.84	40.85±1.24	13.60±0.45	66.30±1.93	53.80±1.23	20.34±0.56
GKD	56.40±1.12	34.95±1.45	15.91±0.89	67.70±1.07	50.35±1.08	21.69±0.54
LCO-KLD	**60.80±0.51**	**42.65±1.63**	**16.71±0.97**	**73.20±1.34**	**55.50±1.88**	**23.95±0.31**

LCO with Limited Advantage Feedback To evaluate the robustness of the LCO framework under sparse feedback, where the advantage signal is restricted to sampled actions, we analyze its performance, presented in Table 5. Specifically, when combined with DPO-RM, all LCO variants surpass PPO across two model backbones. For example, on the Qwen-3-1.7B model, LCO-LCH achieves a Pass@1 score of 64.70% on MATH500, significantly higher than PPO's 55.20%. Similarly, when integrated with a rule-based RM, where a math verifier assigns a score of $+1$ for correct answers and -1 for incorrect ones, LCO demonstrates comparable superiority. In particular, LCO-KLD outperforms GRPO in final accuracy, achieving a MinervaMath score of 10.87% on Qwen-2.5-1.5B, compared to GRPO's 8.65%. These results indicate that the LCO remains effective even with sparse advantage signals, exhibiting a more robust optimization.

Table 5: Results with DPO-RM and rule-based RM, where the LCO advantage signal is limited to sampled action.

Methods	Qwen-2.5-1.5B			Qwen-3-1.7B		
	MATH500	AMC23	MinervaMath	MATH500	AMC23	MinervaMath
w/ DPO-RM						
PPO	41.30±1.32	24.70±2.13	9.23±0.71	55.20±1.32	32.20±2.31	15.31±1.97
LCO-MSE	47.35±1.64	26.80±1.75	9.76±1.44	61.30±1.23	38.10±1.10	16.13±1.38
LCO-LCH	46.80±1.41	26.50±2.01	8.32±1.32	64.70±1.93	37.50±1.03	15.43±0.98
LCO-KLD	48.80±0.94	27.50±2.18	10.32±1.22	63.25±2.01	39.50±1.83	16.98±1.34
w/ Rule-based RM						
GRPO	44.70±1.39	22.70±2.31	8.65±0.83	56.20±1.31	34.50±2.27	14.37±1.02
LCO-MSE	45.85±1.13	24.80±3.01	8.78±0.57	59.40±1.84	35.65±1.31	15.98±1.95
LCO-LCH	46.20±1.54	25.80±2.83	8.72±1.02	58.70±1.42	34.90±2.31	15.43±0.98
LCO-KLD	46.20±2.51	25.80±2.31	10.87±1.36	61.40±1.31	37.10±2.13	16.54±1.34

7 Conclusion

In this work, we analyze the sources of RL instability in LLMs and identified *logits convexity* as essential for well-behaved gradients. We demonstrated the widely used clipped surrogate objective lacks this property, leading to gradient fluctuations and training collapse. Leveraging this insight, we proposed Logits Convex Optimization (LCO) framework, preserving logits convexity and promotes stable training. Empirical results show that LCO delivers consistently stable gradient dynamics and improved performance across various tasks and model families. Our findings provide both a theoretical explanation for RL instability and a practical framework for more reliable optimization.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (No. 62576368).

References

Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Remi Munos, Nicolas Heess, and Martin Riedmiller. Maximum a posteriori policy optimisation. *arXiv preprint arXiv:1806.06920*, 2018.

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The twelfth international conference on learning representations*, 2024.

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

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.

Aili Chen, Aonian Li, Bangwei Gong, Binyang Jiang, Bo Fei, Bo Yang, Boji Shan, Changqing Yu, Chao Wang, Cheng Zhu, et al. Minimax-m1: Scaling test-time compute efficiently with lightning attention. *arXiv preprint arXiv:2506.13585*, 2025a.

Hongzhan Chen, Tao Yang, Shiping Gao, Ruijun Chen, Xiaojun Quan, Hongtao Tian, and Ting Yao. Discriminative policy optimization for token-level reward models. *arXiv preprint arXiv:2505.23363*, 2025b.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. ULTRAFEEDBACK: Boosting language models with scaled AI feedback. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 9722–9744. PMLR, 21–27 Jul 2024. URL `https://proceedings.mlr.press/v235/cui24f.html`.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, et al. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025.

Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. Alpacafarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems*, 36, 2024.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. *arXiv preprint arXiv:2306.08543*, 2023.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Jian Hu, Jason Klein Liu, Haotian Xu, and Wei Shen. Reinforce++: An efficient rlhf algorithm with robustness to both prompt and reward models. *arXiv preprint arXiv:2501.03262*, 2025.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. *Advances in neural information processing systems*, 32, 2019.

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.

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.

MAA. American mathematics competitions, 2023. URL `https://www.maa.org/math-competitions`. Online competition series.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

Rafael Rafailov, Joey Hejna, Ryan Park, and Chelsea Finn. From r to q^*: Your language model is secretly a q-function. In *First Conference on Language Modeling*, 2024a. URL `https://openreview.net/forum?id=kEVcNxtqXk`.

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. *Advances in Neural Information Processing Systems*, 36, 2024b.

Resve A Saleh and AK Saleh. Statistical properties of the log-cosh loss function used in machine learning. *arXiv preprint arXiv:2208.04564*, 2022.

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

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015b.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Ling Team, Bin Hu, Cai Chen, Deng Zhao, Ding Liu, Dingnan Jin, Feng Zhu, Hao Dai, Hongzhi Luan, Jia Guo, et al. Ring-lite: Scalable reasoning via c3po-stabilized reinforcement learning for llms. *arXiv preprint arXiv:2506.14731*, 2025.

Manan Tomar, Lior Shani, Yonathan Efroni, and Mohammad Ghavamzadeh. Mirror descent policy optimization. *arXiv preprint arXiv:2005.09814*, 2020.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256, 1992.

Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A Smith, Mari Ostendorf, and Hannaneh Hajishirzi. Fine-grained human feedback gives better rewards for language model training. *Advances in Neural Information Processing Systems*, 36:59008–59033, 2023.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

Shihui Yang, Chengfeng Dou, Peidong Guo, Kai Lu, Qiang Ju, Fei Deng, and Rihui Xin. Dcpo: Dynamic clipping policy optimization. *arXiv preprint arXiv:2509.02333*, 2025b.

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.

Yufeng Yuan, Yu Yue, Ruofei Zhu, Tiantian Fan, and Lin Yan. What's behind ppo's collapse in long-cot? value optimization holds the secret. *arXiv preprint arXiv:2503.01491*, 2025.

Yi-Fan Zhang, Xingyu Lu, Xiao Hu, Chaoyou Fu, Bin Wen, Tianke Zhang, Changyi Liu, Kaiyu Jiang, Kaibing Chen, Kaiyu Tang, et al. R1-reward: Training multimodal reward model through stable reinforcement learning. *arXiv preprint arXiv:2505.02835*, 2025.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

Han Zhong, Zikang Shan, Guhao Feng, Wei Xiong, Xinle Cheng, Li Zhao, Di He, Jiang Bian, and Liwei Wang. Dpo meets ppo: Reinforced token optimization for rlhf. *arXiv preprint arXiv:2404.18922*, 2024.

Wenqiao Zhu, Ji Liu, Rongjuncheng Zhang, Haipang Wu, and Yulun Zhang. Carft: Boosting llm reasoning via contrastive learning with annotated chain-of-thought-based reinforced fine-tuning. *arXiv preprint arXiv:2508.15868*, 2025.

Brian D Ziebart. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. Carnegie Mellon University, 2010.

A Limitations

Despite the advantages of LCO, this work has several limitations that should be acknowledged.

First, our theoretical framework is primarily built upon the assumption of over-parameterization. This condition is generally satisfied by large-scale models with several billion parameters or more, where the model capacity is sufficient to support the theoretical analysis. Consequently, the theoretical guarantees and the resulting LCO framework may not directly generalize to smaller models with limited parameter counts, where the over-parameterization assumption may not hold.

Second, RL instability is multifaceted and cannot be attributed to a single factor. In addition to the optimization geometry induced by the training objective, factors such as importance sampling variance, off-policy mismatch between the behavior and target policies, and noisy advantage estimation can affect training dynamics. Our work identifies logits convexity as an important, previously underexplored factor that influences gradient stability, rather than claiming it to be the sole cause of RL instability. LCO does not eliminate instability arising from other sources.

B Related Work

Recent research in reinforcement learning has increasingly focused on improving the stability of policy training. These efforts can be broadly categorized into three groups.

The **first category** aims to reduce the variance or bias in advantage estimation. A seminal work in this line is the GAE [Schulman et al., 2015b], which combines Monte Carlo returns and a value model to balance bias and variance. Extending GAE, VC-PPO [Yuan et al., 2025] identifies a failure mode where the value model exhibits bias during training, resulting in large errors in advantage estimation. To address this, they propose a pretraining procedure for the value model, and decouple the λ in GAE for the policy and value model computations. Zhang et al. [2025] identify outliers caused by the imbalance in the advantage distribution. They propose StableReinforce, which applies an advantage filter to retain only those advantages that fall within three standard deviations for stable training. By simplifying the advantage estimation process, RLOO [Ahmadian et al., 2024] employs a leave-one-out baseline across multiple completions to produce advantage estimate for prompt. Similarly, Shao et al. [2024] introduce GRPO, which standardizes sequence-level rewards by subtracting the mean and dividing by the standard deviation, thereby reducing bias and variance. Extending GRPO, Yu et al. [2025] propose DAPO, which re-weights token-level losses to prevent longer responses from being underrepresented in gradient updates.

The **second category** stabilizes training by constraining policy updates through a Kullback-Leibler (KL) divergence penalty relative to a reference model. For example, TRPO [Schulman et al., 2015a] aims to find a policy that increases the probability of advantageous actions while limiting the divergence from the previous policy using a KL constraint, ensuring stable training. Building upon PPO, Ouyang et al. [2022], Hu et al. [2025] add a token-level KL penalty to the reward function, which constrains the policy at each generation step to remain close to the reference SFT model. GRPO [Shao et al., 2024] modifies this approach by applying the KL constraint directly to the policy loss rather than the reward, which allows for more targeted optimization. KL-Cov [Cui et al., 2025] advances this idea by analyzing policy entropy, showing that entropy change is driven by the covariance between action probabilities and advantages, and applying KL penalties selectively to high-covariance tokens to prevent entropy collapse and improve stability.

The **third category** employs clipping mechanisms to stabilize policy updates. PPO and GRPO constrain the importance sampling ratio between current and previous policies within fixed upper and lower bounds to prevent excessively large policy updates. However, such bounds can limit training efficiency and unduly constrain specific updates. To address this, DAPO [Yu et al., 2025] proposes a decoupled clip-higher method that relaxes the upper clipping bound to improve training efficiency while maintaining stability. Building upon the same idea, DCPO [Yang et al., 2025b] addresses the limitation in DAPO, where the same clip range is set for different positions. It further introduces a dynamic clipping method that adaptively adjusts the clipping bounds based on the token probabilities, thereby mitigating the drawbacks of fixed clipping bounds. Chen et al. [2025a] identify a key limitation in PPO/GRPO: clipping can prematurely drop high-advantage tokens from contributing to off-policy gradients. They introduce CISPO, which clips importance sampling weights without clipping token updates to stabilize training. Extending this covariance analysis, Cui et al.

[2025] propose Clip-Cov, which applies clipping selectively to updates on high-covariance tokens to further enhance training stability.

Beyond these approaches to stabilizing policy optimization, mirror descent and policy improvement methods provide a general framework for constructing improved policies and fitting them to parameterized policy. MDPO [Tomar et al., 2020] formulates policy optimization through iterative mirror-descent updates, while AWR [Peng et al., 2019] and MPO [Abdolmaleki et al., 2018] fit improved policies using objectives based on weighted maximum likelihood or KL divergence. In contrast to these methods, LCO reformulates the fitting step directly in logit space by optimizing a unified family of convex logit-matching objectives. Unlike conventional supervised fitting of the improved policy, LCO explicitly enforces desirable convexity properties in the optimization objective, which enables our gradient analysis and the associated optimization guarantees.

C Empirical Validation of the Local Linearization Assumption

Assumption 4.1 relies on the local linearization of the neural network around its initialization. To empirically examine the validity of this assumption in our setting, we measure the relative parameter displacement during PPO training of Qwen3-4B, initialized from a converged SFT checkpoint. Let θ_0 denote the parameters of the initial SFT checkpoint and θ_t denote the parameters after t training steps. We define the relative parameter displacement as $\delta_t = \frac{\|\theta_t - \theta_0\|_2}{\|\theta_0\|_2}$. This quantity measures the magnitude of the parameter change relative to the norm of the initialization. We record the relative parameter displacement at different stages of PPO training. As shown in Table 6, the displacement increases gradually throughout training but remains below 5×10^{-4} at all reported checkpoints.

Table 6: Relative parameter displacement during PPO training of Qwen3-4B.

Training Steps	Relative Parameter Displacement δ_t
1k	9.78×10^{-5}
2k	1.27×10^{-4}
3k	2.47×10^{-4}
4k	4.23×10^{-4}

These results show that PPO post-training induces only a small relative change in the model parameters compared with the initialization. This observation is consistent with the local nature of our theoretical analysis and supports the plausibility of applying a first-order approximation around the SFT checkpoint in our setting. In particular, starting from a converged SFT model and performing post-training with relatively small learning rates (5×10^{-7} to 5×10^{-6}) helps keep the optimization trajectory close to its initialization. We emphasize that a small relative parameter displacement alone does not guarantee that higher-order terms in the network's Taylor expansion are negligible. Nevertheless, the observed parameter stability provides empirical evidence supporting the use of the local linearization assumption in our PPO post-training regime.

D Additional Experiment

Gradient Dynamics of PPO Figure 7 shows the gradient dynamics of PPO for actions with positive advantage. We consider $A(s_t, a_t) > 0$, the gradient norm of the surrogate objective decreases as training progress, as shown in Figure 7(a), while the magnitude of logit gradients decrease before being clipped (Figure 7(b) and (c)).

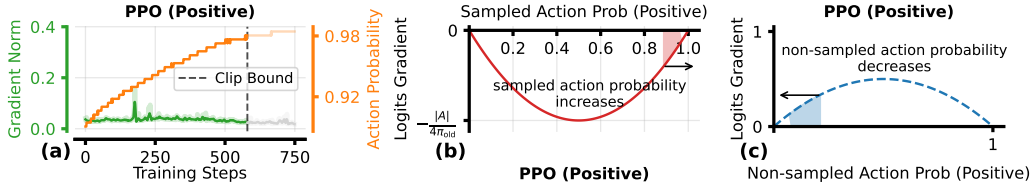


Figure 7: Training dynamics of PPO for actions with positive advantages. **(a)** Gradient norm decreases as training progresses while sampled action probabilities increase. The magnitudes of both sampled **(b)** and non-sampled action logit gradient **(c)** decrease.

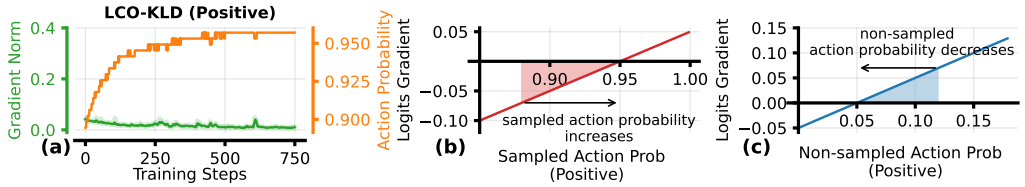


Figure 8: Training dynamics of LCO-KLD for actions with positive advantages. We show the dynamics of gradient norm **(a)**, logit gradient for sampled **(b)** and non-sampled **(c)** actions. LCO-KLD exhibits stable gradient updates as the model converges.

Gradient Dynamics of LCO-KLD Figure 8 shows the gradient dynamics of LCO-KLD for actions with positive advantage. We consider $A(s_t, a_t)$ for the sampled action, the gradient norm of the LCO-KLD objective decreases as training progress, as shown in Figure 8(a), while the magnitude of logit gradients decrease correspondingly (Figure 8(b) and (c)).

Table 7: OOD results. Best performances are shown in **bold**, while suboptimal ones are underlined.

Methods	Qwen-3-4B		Qwen-2.5-3B	
	MMLU	MMLU-Redux	MMLU	MMLU-Redux
PPO	67.87±1.23	73.93±1.53	58.49±1.89	55.59±2.01
GRPO	64.69±1.42	71.35±1.51	58.61±1.42	56.97±1.94
DAPO	65.39±1.21	70.37±1.82	59.94±1.11	52.49±1.34
GSPO	68.71±2.01	72.72±1.71	54.95±1.38	56.44±1.74
LCO-MSE	68.71±1.42	74.21±1.34	58.10±1.67	56.73±2.01
LCO-LCH	69.12±1.31	74.12±1.23	59.23±2.13	**58.71±1.83**
LCO-KLD	**72.11±1.12**	**76.71±1.35**	**60.14±1.35**	57.58±1.67

Out-of-Distribution Performance In order to assess the out-of-distribution (OOD) generalization beyond the mathematical reasoning results in Table 7, we evaluate the models on multi-task language understanding benchmarks, including MMLU and MMLU-Redux. As shown in Table 7, the LCO objectives consistently outperform baseline methods in accuracy. With the Qwen-3-4B backbone, LCO-KLD obtains the highest accuracy on both MMLU (72.11%) and MMLU-Redux (76.71%), surpassing PPO (67.87% and 73.93%, respectively). Using the Qwen-2.5-3B backbone, LCO-KLD delivers the best accuracy on MMLU (60.14%), while LCO-LCH achieves the top accuracy on MMLU-Redux (58.71%). These results highlight the strong OOD generalization and robustness of the LCO objectives across different model families.

E Logit Gradient

E.1 Logit Gradient of SFT

We provide the derivation for Equation (4). At a time step t, let a_t denote the target (ground-truth) token and a'_t be an arbitrary token in the vocabulary $\mathcal{V}$, the derivative of $\mathcal{L}_{\text{SFT}}$ defined in Equation (1) with respect to the logit $z_\theta(s_t, a'_t)$ is:

$$\begin{aligned}\frac{\partial \mathcal{L}_{\text{SFT}}}{\partial z_\theta(s_t, a'_t)} &= -\frac{\partial \log \pi_\theta(a_t|s_t)}{\partial \pi_\theta(a_t|s_t)} \frac{\partial \pi_\theta(a_t|s_t)}{z_\theta(s_t, a'_t)} \\ &= -\frac{1}{\pi_\theta(a_t|s_t)} \pi_\theta(a_t|s_t)(\mathbb{I}_{a_t=a'_t} - \pi_\theta(a'_t|s_t)) \\ &= \pi_\theta(a'_t|s_t) - \mathbb{I}_{a_t=a'_t},\end{aligned} \tag{19}$$

where $\mathbb{I}_{a_t=a'_t}$ is the indicator function, which equals 1 when $a_t = a'_t$, and 0 otherwise.

E.2 Logit Gradient of PPO

We provide the derivation for Equation (5). At a time step t, let a_t be the sampled action and a'_t be an arbitrary action in the $\mathcal{V}$, the derivative of $\mathcal{L}_{\text{PPO}}$ defined in Equation (3) with respect to the logit

$z_\theta(s_t, a'_t)$ is (when the gradient is not clipped):

$$\begin{aligned}\frac{\partial \mathcal{L}_{\text{PPO}}}{\partial z_\theta(s_t, a'_t)} &= -\frac{A(s_t, a_t)}{\pi_{\text{old}}(a_t|s_t)} \frac{\partial \pi_\theta(a_t|s_t)}{\partial z_\theta(s_t, a'_t)} \\ &= \frac{A(s_t, a_t)}{\pi_{\text{old}}(a_t|s_t)} \pi_\theta(a_t|s_t)(\pi_\theta(a'_t|s_t) - \mathbb{I}_{a_t=a'_t}).\end{aligned} \tag{20}$$

F Derivation of Optimal Policy

In this section, we provide the derivation for Equation (6). Under the reinforcement learning objective, we aim to maximize the expected advantage with a KL constraint at each time step t:

$$\max_{\pi_\theta} \mathbb{E}_{a_t \sim \pi_\theta(\cdot|s_t)} [A(s_t, a_t)] - \beta \mathbb{D}_{\text{KL}}(\pi_\theta(\cdot|s_t) \| \pi_{\text{old}}(\cdot|s_t)) \quad s.t. \sum_{a_t \in \mathcal{V}} \pi_\theta(a_t|s_t) = 1. \tag{21}$$

To find the optimal policy π^*, we aim to maximize above objective under the constraint $\sum_{a_t \in \mathcal{V}} \pi_\theta(a_t|s_t) = 1$. To solve the constrained optimization problem, we can use the method of Lagrange multiplier. Let λ be the Lagrange multiplier, the objective can be written as:

$$\begin{aligned}\mathcal{J} &= \mathbb{E}_{a_t \sim \pi_\theta(\cdot|s_t)} [A(s_t, a_t)] - \beta \mathbb{D}_{\text{KL}}(\pi_\theta(\cdot|s_t) \| \pi_{\text{old}}(\cdot|s_t)) + \lambda \left[\sum_{a_t \in \mathcal{V}} \pi_\theta(a_t|s_t) - 1 \right] \\ &= \sum_{a_t \in \mathcal{V}} \pi_\theta(a_t|s_t) A(s_t, a_t) - \beta \left[\sum_{a_t \in \mathcal{V}} \pi_\theta(a_t|s_t) \log \frac{\pi_\theta(a_t|s_t)}{\pi_{\text{old}}(a_t|s_t)} \right] + \lambda \left[\sum_{a_t \in \mathcal{V}} \pi_\theta(a_t|s_t) - 1 \right].\end{aligned} \tag{22}$$

We set the following partial derivative to zero and have:

$$\frac{\partial \mathcal{J}}{\partial \pi_\theta(a_t|s_t)} = A(s_t, a_t) - \beta \left[\log \frac{\pi_\theta(a_t|s_t)}{\pi_{\text{old}}(a_t|s_t)} + 1 \right] + \lambda = 0. \tag{23}$$

Using Equation (23), we have the expression for optimal policy π^* as follows:

$$\pi^*(a_t|s_t) = \pi_{\text{old}}(a_t|s_t) \exp\left[\frac{A(s_t, a_t)}{\beta} \right] \exp\left[\frac{\lambda - \beta}{\beta} \right]. \tag{24}$$

Since the term $\exp\left[\frac{\lambda-\beta}{\beta}\right]$ is constant, the optimal policy π^* is prop to:

$$\pi^*(a_t|s_t) \propto \pi_{\text{old}}(a_t|s_t) \exp\left[\frac{A(s_t, a_t)}{\beta} \right]. \tag{25}$$

By ensuring the property of probability, the optimal policy can be written as:

$$\pi^*(a_t|s_t) = \frac{\pi_{\text{old}}(a_t|s_t) \exp\left[\frac{A(s_t,a_t)}{\beta}\right]}{\sum_{a'_t \in \mathcal{V}} \pi_{\text{old}}(a'_t|s_t) \exp\left[\frac{A(s_t,a'_t)}{\beta}\right]}. \tag{26}$$

By substituting the softmax definition of the behavioral policy $\pi_{\text{old}}(a_t|s_t)$ into Equation (26), the normalization terms $\sum_{a'_t \in \mathcal{V}} \exp z_{\text{old}}(s_t, a'_t)$ in the numerator and denominator cancel out. This allows us to reformulate the optimal policy directly in terms of the behavioral logits $z_{\text{old}}(s_t, a_t)$:

$$\pi^*(a_t|s_t) = \frac{\exp\left[z_{\text{old}}(s_t, a_t) + \frac{A(s_t,a_t)}{\beta}\right]}{\sum_{a'_t \in \mathcal{V}} \exp\left[z_{\text{old}}(s_t, a'_t) + \frac{A(s_t,a'_t)}{\beta}\right]}. \tag{27}$$

By equating the softmax arguments, we can identify a solution for the optimal logits z^* as:

$$z^*(s_t, a_t) = z_{\text{old}}(s_t, a_t) + \frac{A(s_t, a_t)}{\beta}. \tag{28}$$

These optimal logits are obtained via a direct advantage-based adjustment without an additional state-dependent constant shift. For clarity and consistency, we define this specific solution as the unique optimal logits throughout this paper.

G Derivation of Gradient Directionality

In this section, we provide the derivation for Proposition 4.3. Consider logits $\boldsymbol{z}_\theta$ parameterized by θ. Let $\Delta\theta = \theta^* - \theta$ be the parameter displacement, the first-order Taylor expansion around θ is:

$$\boldsymbol{z}_{\theta^*} = \boldsymbol{z}_\theta + \nabla_\theta \boldsymbol{z}_\theta^\top \Delta\theta + o(\|\Delta\theta\|). \tag{29}$$

Under Assumption 4.1, the higher-order term $o(\|\Delta\theta\|)$ can be truncated without significant loss of accuracy:

$$\boldsymbol{z}_{\theta^*} \approx \boldsymbol{z}_\theta + \nabla_\theta \boldsymbol{z}_\theta^\top \Delta\theta. \tag{30}$$

Let $\mathcal{L}$ be a differentiable loss function that take $\boldsymbol{z}_\theta$ as input. By applying the chain rule, we evaluate the inner product between the parameter gradient and the parameter displacement:

$$\langle \nabla_\theta \mathcal{L}, \theta - \theta^* \rangle = \langle \nabla_\theta \boldsymbol{z}_\theta \nabla_{\boldsymbol{z}_\theta} \mathcal{L}, \theta - \theta^* \rangle = \langle \nabla_{\boldsymbol{z}_\theta} \mathcal{L}, \nabla_\theta \boldsymbol{z}_\theta^\top (\theta - \theta^*) \rangle. \tag{31}$$

By rearranging the terms in Equation (30), and substituting into the inner product expression yields:

$$\langle \nabla_{\boldsymbol{z}_\theta} \mathcal{L}, \nabla_\theta \boldsymbol{z}_\theta^\top (\theta - \theta^*) \rangle \approx \langle \nabla_{\boldsymbol{z}_\theta} \mathcal{L}, \boldsymbol{z}_\theta - \boldsymbol{z}_{\theta^*} \rangle. \tag{32}$$

If $\mathcal{L}$ is convex with respect to logits, by the first-order characterization of convex function, we have:

$$\langle \nabla_{\boldsymbol{z}_\theta} \mathcal{L}, \nabla_\theta \boldsymbol{z}_\theta^\top (\theta - \theta^*) \rangle \approx \langle \nabla_{\boldsymbol{z}_\theta} \mathcal{L}, \boldsymbol{z}_\theta - \boldsymbol{z}_{\theta^*} \rangle \geq 0. \tag{33}$$

H Logits Convexity of Different Objectives

H.1 Logits Convexity of SFT and PPO Losses

Logits Convexity of SFT At time step t, the Hessian $\boldsymbol{H}_{\text{SFT}}$ of $\mathcal{L}_{\text{SFT}}$ (Equation (1)) with respect to the logits $\boldsymbol{z}_\theta$ is given by:

$$\boldsymbol{H}_{\text{SFT}} = \text{diag}(\boldsymbol{\pi}_\theta) - \boldsymbol{\pi}_\theta \boldsymbol{\pi}_\theta^\top, \tag{34}$$

where $\boldsymbol{\pi}_\theta \in \mathbb{R}^{|\mathcal{V}|}$ denotes the action probability distribution over the vocabulary $\mathcal{V}$ at time step t, $\text{diag}(\boldsymbol{\pi}_\theta)$ is a diagonal matrix with the elements of $\boldsymbol{\pi}_\theta$ along its main diagonal. For any vector $\boldsymbol{v} \in \mathbb{R}^{|\mathcal{V}|}$, consider the quadratic form:

$$\boldsymbol{v}^\top \boldsymbol{H}_{\text{SFT}} \boldsymbol{v} = \sum_i^{|\mathcal{V}|} \pi_{\theta,i} v_i^2 - \left(\sum_i^{|\mathcal{V}|} \pi_{\theta,i} v_i \right)^2 \geq 0, \tag{35}$$

where the final inequality follows from the Cauchy-Schwarz inequality. This demonstrates that the Hessian matrix $\boldsymbol{H}_{\text{SFT}}$ is positive semi-definite (PSD), which in turn implies that $\mathcal{L}_{\text{SFT}}$ is logits convex at each time step.

Logits Convexity of PPO The gradient of $\mathcal{L}_{\text{PPO}}$, as defined in Equation (3), is non-zero only when the condition ($A(s_t, a_t) > 0$ and $r_t(\theta) < 1 + \epsilon$) or ($A(s_t, a_t) < 0$ and $r_t(\theta) > 1 - \epsilon$) is satisfied. If this condition is not met, the objective enters the clipped region, and the gradient becomes zero. Therefore, we focus our analysis on the active region where the gradient is non-zero. Under this setting, the gradient of $\mathcal{L}_{\text{PPO}}$ with respect to a logit $z_\theta(s_t, a_t')$ at time step t is:

$$\frac{\partial \mathcal{L}_{\text{PPO}}}{\partial z_\theta(s_t, a_t')} = -\frac{A(s_t, a_t)}{\pi_{\text{old}}(a_t|s_t)} \pi_\theta(a_t|s_t)(\mathbb{I}_{a_t = a_t'} - \pi_\theta(a_t'|s_t)). \tag{36}$$

And the second derivative is as follow:

$$\frac{\partial^2 \mathcal{L}_{\text{PPO}}}{\partial z_\theta(s_t, a_t') \partial z_\theta(s_t, a_t'')} =$$
$$-\frac{A(s_t, a_t)}{\pi_{\text{old}}(a_t|s_t)} [\pi_\theta(a_t|s_t)(\mathbb{I}_{a_t = a_t''} - \pi_\theta(a_t''|s_t))(\mathbb{I}_{a_t = a_t'} - \pi_\theta(a_t'|s_t)) - \pi_\theta(a_t|s_t)\pi_\theta(a_t'|s_t)(\mathbb{I}_{a_t' = a_t''} - \pi_\theta(a_t''|s_t))]. \tag{37}$$

According to Equation (37), the Hessian matrix $\boldsymbol{H}_{\text{PPO}}$ of $\mathcal{L}_{\text{PPO}}$ with respect to logits at time step t is given by:

$$\boldsymbol{H}_{\text{PPO}} = \frac{A(s_t, a_t)}{\pi_{\text{old}}(a_t|s_t)} \left[-\boldsymbol{e}^{(k)} \boldsymbol{e}^{(k)\top} + \boldsymbol{\pi}_\theta \boldsymbol{e}^{(k)\top} + \boldsymbol{e}^{(k)} \boldsymbol{\pi}_\theta^\top - 2\boldsymbol{\pi}_\theta \boldsymbol{\pi}_\theta^\top + \text{diag}(\boldsymbol{\pi}_\theta) \right], \tag{38}$$

where k is the index of action a_t in the vocabulary, $\boldsymbol{e}^{(k)}$ represents the standard $|\mathcal{V}|$-dimension basis vector with a 1 at position k, $\boldsymbol{\pi}_\theta \in \mathbb{R}^{|\mathcal{V}|}$ denotes the action probability distribution over the vocabulary $\mathcal{V}$ at time step t, and $\text{diag}(\boldsymbol{\pi}_\theta)$ is a diagonal matrix with the elements of $\boldsymbol{\pi}_\theta$ along its main diagonal. For any vector $\boldsymbol{v} \in \mathbb{R}^{|\mathcal{V}|}$, consider the quadratic form:

$$\boldsymbol{v}^\top \boldsymbol{H}_{\text{PPO}} \boldsymbol{v} = \frac{A(s_t, a_t)}{\pi_{\text{old}}(a_t|s_t)} \left[\underbrace{\sum_i^{|\mathcal{V}|} \pi_{\theta,i} v_i^2 - \left(\sum_i^{|\mathcal{V}|} \pi_{\theta,i} v_i \right)^2}_{\mathbb{D}(\boldsymbol{v})} - \underbrace{\left(v_k^2 - 2v_k \sum_i^{|\mathcal{V}|} \pi_{\theta,i} v_i + \left(\sum_i^{|\mathcal{V}|} \pi_{\theta,i} v_i \right)^2 \right)}_{(v_k - \mathbb{E}(\boldsymbol{v}))^2} \right], \tag{39}$$

where $\mathbb{E}(\boldsymbol{v}) = \sum_i^{|\mathcal{V}|} \pi_{\theta,i} v_i$, and $\mathbb{D}(\boldsymbol{v}) = \sum_i^{|\mathcal{V}|} \pi_{\theta,i} v_i^2 - (\sum_i^{|\mathcal{V}|} \pi_{\theta,i} v_i)^2$. According to Equation (35), we have $\mathbb{D}(\boldsymbol{v}) \geq 0$. Now, consider the case when $A(s_t, a_t) > 0$:

$$\begin{cases} \boldsymbol{v}^\top \boldsymbol{H}_{\text{PPO}} \boldsymbol{v} < 0, & \text{if } v_k > \mathbb{E}(\boldsymbol{v}) + \sqrt{\mathbb{D}(\boldsymbol{v})} \text{ or } v_k < \mathbb{E}(\boldsymbol{v}) - \sqrt{\mathbb{D}(\boldsymbol{v})} \\ \boldsymbol{v}^\top \boldsymbol{H}_{\text{PPO}} \boldsymbol{v} \geq 0, & \text{otherwise.} \end{cases} \tag{40}$$

A symmetric result holds for the case where $A(s_t, a_t) < 0$. This implies Hessian matrix $\boldsymbol{H}_{\text{PPO}}$ is not PSD, which indicates that the PPO loss $\mathcal{L}_{\text{PPO}}$ is not convex with respect to logits.

H.2 Logits Convexity of LCO Losses

Logits Convexity of LCO-MSE Loss At time step t, the Hessian matrix $\boldsymbol{H}_{\text{LCO-MSE}}$ of $\mathcal{L}_{\text{LCO-MSE}}$ (defined in Equation (8)) with respect to the logits $\boldsymbol{z}_\theta$ is given by:

$$\boldsymbol{H}_{\text{LCO-MSE}} = \frac{2}{|\mathcal{V}|} \boldsymbol{I}, \tag{41}$$

where $\boldsymbol{I}$ is the identity matrix. Since the eigenvalues $\frac{2}{|\mathcal{V}|} > 0$, it follows that the Hessian is positive definite. Consequently, $\mathcal{L}_{\text{LCO-MSE}}$ is not only convex but also strongly convex in the logit space, with a modulus of strong convexity equal to $2/|\mathcal{V}|$.

Logits Convexity of LCO-LCH Loss At time step t, the partial derivative of $\mathcal{L}_{\text{LCO-LCH}}$ (defined in Equation (9)) with respect to a logit $z_\theta(s_t, a'_t)$ is given by:

$$\frac{\partial \mathcal{L}_{\text{LCO-LCH}}}{\partial z_\theta(s_t, a'_t)} = \frac{1}{|\mathcal{V}|} \tanh(z_\theta(s_t, a'_t) - z^*(s_t, a'_t)). \tag{42}$$

The corresponding second-order derivative is non-zero only for the diagonal elements ($a'_t = a''_t$):

$$\frac{\partial^2 \mathcal{L}_{\text{LCO-LCH}}}{\partial z_\theta(s_t, a'_t) \partial z_\theta(s_t, a''_t)} = \frac{1}{|\mathcal{V}|} \text{sech}^2(z_\theta(s_t, a'_t) - z^*(s_t, a'_t)). \tag{43}$$

Thus, the Hessian $\boldsymbol{H}_{\text{LCO-LCH}}$ is given by the diagonal matrix:

$$\boldsymbol{H}_{\text{LCO-LCH}} = \text{diag}\left(\frac{1}{|\mathcal{V}|} \text{sech}^2(\boldsymbol{z}_\theta - \boldsymbol{z}^*) \right), \tag{44}$$

where $\boldsymbol{z}_\theta \in \mathbb{R}^{|\mathcal{V}|}$ denotes the logits vector, and $\boldsymbol{z}^* \in \mathbb{R}^{|\mathcal{V}|}$ denotes the optimal logits vector. Since the eigenvalues $\frac{1}{|\mathcal{V}|}\text{sech}^2(x) \in (0, \frac{1}{|\mathcal{V}|}]$ for all $x \in \mathbb{R}$, it follows that the Hessian is positive definite. Furthermore, within any bounded region where $\|\boldsymbol{z}_\theta - \boldsymbol{z}^*\|_\infty \leq R$, the eigenvalues of the Hessian are lower-bounded by $\frac{1}{|\mathcal{V}|}\text{sech}^2(R) > 0$. Consequently, $\mathcal{L}_{\text{LCO-LCH}}$ is not only convex but also locally strongly convex at the logits level.

Logits Convexity of LCO-KLD Loss At time step t, the partial derivative of $\mathcal{L}_{\text{LCO-KLD}}$ (defined in Equation (10)) with respect to a logit $z_\theta(s_t, a'_t)$ is given by:

$$\frac{\partial \mathcal{L}_{\text{LCO-KLD}}}{\partial z_\theta(s_t, a'_t)} = \pi_\theta(a'_t|s_t) - \pi^*(a'_t|s_t). \tag{45}$$

And the second derivative is as follow:

$$\frac{\partial^2 \mathcal{L}_{\text{LCO-KLD}}}{\partial z_\theta(s_t, a_t') \partial z_\theta(s_t, a_t'')} = \pi_\theta(a_t'|s_t)(\mathbb{I}_{a_t'=a_t''} - \pi_\theta(a_t''|s_t)). \tag{46}$$

According to Equation (46), the Hessian matrix $\boldsymbol{H}_{\text{LCO-KLD}}$ of $\mathcal{L}_{\text{LCO-KLD}}$ with respect to logits at time step t is given by:

$$\boldsymbol{H}_{\text{LCO-KLD}} = \text{diag}(\boldsymbol{\pi}_\theta) - \boldsymbol{\pi}_\theta \boldsymbol{\pi}_\theta^\top, \tag{47}$$

From the conclusion in Equation (35), $\boldsymbol{H}_{\text{LCO-KLD}}$ is PSD, which in turn implies $\mathcal{L}_{\text{LCO-KLD}}$ is logits convex at each time step.

I Upper Bound of Gradient Norm for LCO Objectives

I.1 Upper Bound of LCO-MSE

Given the gradient norm expression for $\mathcal{L}_{\text{LCO-MSE}}$ at time step t:

$$\|\nabla_\theta \mathcal{L}_{\text{LCO-MSE}}\| = \frac{2}{|\mathcal{V}|} \|\boldsymbol{J}^\top (\boldsymbol{z}_\theta - \boldsymbol{z}^*)\|, \tag{48}$$

where $\boldsymbol{J}$ is the Jacobian matrix $\nabla_\theta \boldsymbol{z}_\theta^\top$. Under Assumption 4.1, $\boldsymbol{J}$ is approximately constant in the local neighborhood. Let $\sigma_{\max}$ be the maximum singular value of $\boldsymbol{J}^\top$, $\Delta z(s_t, a_t) = z_\theta(s_t, a_t) - z^*(s_t, a_t)$, using property of matrix induced norm:

$$\|\boldsymbol{J}^\top (\boldsymbol{z}_\theta - \boldsymbol{z}^*)\| \leq \sigma_{\max} \sqrt{\sum_{a_t \in \mathcal{V}} \Delta z(s_t, a_t)^2} = \sigma_{\max} \sqrt{|\mathcal{V}| \mathcal{L}_{\text{LCO-MSE}}}. \tag{49}$$

Therefore, the upper bound of the gradient norm is:

$$\|\nabla_\theta \mathcal{L}_{\text{LCO-MSE}}\| \leq \frac{2}{|\mathcal{V}|} \sigma_{\max} \sqrt{|\mathcal{V}| \mathcal{L}_{\text{LCO-MSE}}}. \tag{50}$$

As the loss $\mathcal{L}_{\text{LCO-MSE}}$ decreases, the upper bound of the gradient norm decreases monotonously.

I.2 Upper Bound of LCO-LCH

Given the gradient norm expression for $\mathcal{L}_{\text{LCO-LCH}}$, at time step t:

$$\|\nabla_\theta \mathcal{L}_{\text{LCO-LCH}}\| = \frac{1}{|\mathcal{V}|} \left\| \boldsymbol{J}^\top \tanh(\boldsymbol{z}_\theta - \boldsymbol{z}^*) \right\|, \tag{51}$$

Using the property of the matrix induced norm:

$$\left\| \boldsymbol{J}^\top \tanh(\boldsymbol{z}_\theta - \boldsymbol{z}^*) \right\| \leq \sigma_{\max} \sqrt{\sum_{a_t \in \mathcal{V}} \tanh^2 \Delta z(s_t, a_t)}. \tag{52}$$

To derive the upper bound, we need to express $\tanh^2$ in terms of $\log\cosh$. We use the following hyperbolic identity:

$$\sum_{a_t \in \mathcal{V}} \tanh^2 \Delta z(s_t, a_t) = \sum_{a_t \in \mathcal{V}} \left[1 - \exp\left(-2 \log\cosh \Delta z(s_t, a_t)\right)\right]. \tag{53}$$

By use the Jensen's Inequality for the concave function $1 - e^{-2x}$:

$$\frac{1}{|\mathcal{V}|} \sum_{a_t \in \mathcal{V}} \left[1 - \exp\left(-2 \log\cosh \Delta z(s_t, a_t)\right)\right] \leq 1 - \exp\left(-\frac{2}{|\mathcal{V}|} \sum_{a_t \in \mathcal{V}} \log\cosh \Delta z(s_t, a_t) \right). \tag{54}$$

Therefore, the upper bound of the gradient norm is:

$$\|\nabla_\theta \mathcal{L}_{\text{LCO-LCH}}\| \leq \frac{1}{|\mathcal{V}|} \sigma_{\max} \sqrt{|\mathcal{V}| (1 - \exp(-2\mathcal{L}_{\text{LCO-LCH}}))}. \tag{55}$$

As the loss $\mathcal{L}_{\text{LCO-LCH}}$ decreases, the upper bound of the gradient norm decreases monotonously.

I.3 Upper Bound of LCO-KLD

Given the gradient norm expression for $\mathcal{L}_{\text{LCO-KLD}}$ at time step t:

$$\|\nabla_\theta \mathcal{L}_{\text{LCO-KLD}}\| = \|\boldsymbol{J}^\top(\boldsymbol{\pi}_\theta - \boldsymbol{\pi}^*)\|. \tag{56}$$

Let $\Delta\pi(s_t, a_t) = \pi_\theta(s_t, a_t) - \pi^*(s_t, a_t)$, using the property of the matrix induced norm:

$$\|\boldsymbol{J}^\top(\boldsymbol{\pi}_\theta - \boldsymbol{\pi}^*)\| \leq \sigma_{\max}\sqrt{\sum_{a_t \in \mathcal{V}} \Delta\pi(s_t, a_t)^2}. \tag{57}$$

Since L_2 norm is upper-bounded by the L_1 norm, according to the Pinsker's Inequality:

$$\sqrt{\sum_{a_t \in \mathcal{V}} \Delta\pi(s_t, a_t)^2} \leq \sum_{a_t \in \mathcal{V}} |\Delta\pi(s_t, a_t)| \leq \sqrt{2\sum_{a_t \in \mathcal{V}} \pi^*(a_t|s_t) \log \frac{\pi^*(a_t|s_t)}{\pi_\theta(a_t|s_t)}}. \tag{58}$$

Therefore, the upper bound of the gradient norm is:

$$\|\nabla_\theta \mathcal{L}_{\text{LCO-KLD}}\| \leq \sigma_{\max}\sqrt{2\mathcal{L}_{\text{LCO-KLD}}}. \tag{59}$$

As the loss $\mathcal{L}_{\text{LCO-KLD}}$ decreases, the upper bound of the gradient norm decreases monotonously.

J Convergence Guarantees of LCO

To offer a theoretical guarantee for LCO, we present the following. Let $\mathcal{L}$ be the LCO loss function. Given a step size η, the parameter update rule is: $\theta_{k+1} = \theta_k - \eta J^\top \nabla_z \mathcal{L}$, where $J = \nabla_\theta z_\theta^\top$ denotes the Jacobian matrix. Let z_k be the logits at k-th step, under Assumption 4.1, we have:

$$z_{k+1} - z_k = J(\theta_{k+1} - \theta_k). \tag{60}$$

From the derivation in Appendix H.2, since $\|\boldsymbol{H}_{LCO-MSE}\|_2 \leq \frac{2}{|V|}$, $\|\boldsymbol{H}_{LCO-LCH}\|_2 \leq \frac{1}{|V|}$, and $\|\boldsymbol{H}_{LCO-KLD}\|_2 \leq 1$, using the property of β-smoothness we have:

$$\mathcal{L}(z_{k+1}) \leq \mathcal{L}(z_k) + \nabla_z \mathcal{L}^\top(z_{k+1} - z_k) + \frac{\beta}{2}\|z_{k+1} - z_k\|^2. \tag{61}$$

Combining Equations (60) and (61), we have:

$$\mathcal{L}(z_{k+1}) \leq \mathcal{L}(z_k) - \eta\|J^\top \nabla_z \mathcal{L}\|^2 + \frac{\beta\eta^2}{2}\|JJ^\top \nabla_z \mathcal{L}\|^2. \tag{62}$$

Since $JJ^\top$ is symmetric and PSD, let λ denote its largest eigenvalue. Using spectral decomposition:

$$\|JJ^\top \nabla_z \mathcal{L}\|^2 \leq \lambda\|J^\top \nabla_z \mathcal{L}\|^2. \tag{63}$$

Combining Equations (62) and (63), and setting $\eta = \frac{1}{\beta\lambda}$, we have:

$$\frac{1}{\beta\lambda}\|J^\top \nabla_z \mathcal{L}\|^2 \leq 2(\mathcal{L}(z_k) - \mathcal{L}(z_{k+1})). \tag{64}$$

Now, we consider the distance between the current parameters θ and the optimal parameters θ^*. By expanding the squared norm of the parameter update rule, we have:

$$\begin{aligned} \|\theta_{k+1} - \theta^*\|^2 &= \|\theta_k - \eta J^\top \nabla_z \mathcal{L} - \theta^*\|^2 \\ &= \|\theta_k - \theta^*\|^2 - 2\eta(\theta_k - \theta^*)^\top J^\top \nabla_z \mathcal{L} + \eta^2\|J^\top \nabla_z \mathcal{L}\|^2. \end{aligned} \tag{65}$$

Using the first-order property of convex functions with respect to logits, we have $\mathcal{L}(z^*) \geq \mathcal{L}(z_k) + \nabla_z \mathcal{L}^\top(z^* - z_k)$. By Assumption 4.1, we substitute $z^* - z_k = J(\theta^* - \theta_k)$, which yields:

$$(\theta_k - \theta^*)^\top J^\top \nabla_z \mathcal{L} \geq \mathcal{L}(z_k) - \mathcal{L}(z^*). \tag{66}$$

By setting $\eta = \frac{1}{\beta\lambda}$ and substituting Equation (66) back into the distance expansion, we obtain:

$$\|\theta_{k+1} - \theta^*\|^2 \leq \|\theta_k - \theta^*\|^2 - \frac{2}{\beta\lambda}(\mathcal{L}(z_k) - \mathcal{L}^*) + \frac{1}{\beta^2\lambda^2}\|J^\top \nabla_z \mathcal{L}\|^2. \tag{67}$$

Incorporating the result from Equation (64), the term $\frac{1}{\beta^2\lambda^2}\|J^\top \nabla_z \mathcal{L}\|^2$ is bounded by $\frac{2}{\beta\lambda}(\mathcal{L}(z_k) - \mathcal{L}(z_{k+1}))$. Substituting this bound gives:

$$\begin{aligned}\|\theta_{k+1} - \theta^*\|^2 &\leq \|\theta_k - \theta^*\|^2 - \frac{2}{\beta\lambda}(\mathcal{L}(z_k) - \mathcal{L}^*) + \frac{2}{\beta\lambda}(\mathcal{L}(z_k) - \mathcal{L}(z_{k+1})) \\ &= \|\theta_k - \theta^*\|^2 - \frac{2}{\beta\lambda}(\mathcal{L}(z_{k+1}) - \mathcal{L}^*).\end{aligned} \tag{68}$$

Summing this inequality from step $k = 0$ to $K - 1$, we have:

$$\frac{2}{\beta\lambda}\sum_{k=0}^{K-1}(\mathcal{L}(z_{k+1}) - \mathcal{L}^*) \leq \|\theta_0 - \theta^*\|^2 - \|\theta_K - \theta^*\|^2 \leq \|\theta_0 - \theta^*\|^2. \tag{69}$$

Because Equation (64) implies that the loss is non-increasing ($\mathcal{L}(z_{k+1}) \leq \mathcal{L}(z_k)$), we have:

$$K(\mathcal{L}(z_K) - \mathcal{L}^*) \leq \sum_{k=0}^{K-1}(\mathcal{L}(z_{k+1}) - \mathcal{L}^*). \tag{70}$$

Therefore, combining the above results, we obtain the final convergence rate:

$$\mathcal{L}(z_K) - \mathcal{L}^* \leq \frac{\beta\lambda\|\theta_0 - \theta^*\|^2}{2K}. \tag{71}$$

This provides a general convergence guarantee for LCO, demonstrating that it converges at a sublinear rate of $\mathcal{O}(1/K)$ under the given assumptions.

K Optimal Logits Analysis

While the optimal policy is invariant to a constant shift in logits due to the translation invariance of the softmax function, the choice of the specific target $\boldsymbol{z}^*$ in Equation (7) is not arbitrary. We demonstrate that this specific target minimizes the initial logit-space discrepancy, thereby facilitating a more efficient optimization trajectory.

To justify this choice, consider a general candidate target $\boldsymbol{z}'$ defined as $\boldsymbol{z}' = \boldsymbol{z}_{\text{old}} + \frac{\boldsymbol{A}}{\beta} + C$, where C is an arbitrary constant and is added to each element of advantage vector $\frac{\boldsymbol{A}}{\beta}$. This shift C modifies the initial error vector. For $\mathcal{L}_{\text{LCO-MSE}}$ objective, the initial loss at the start of an update is proportional to:

$$\min_C \left\| \boldsymbol{z}_{\text{old}} - \left(\boldsymbol{z}_{\text{old}} + \frac{1}{\beta}\boldsymbol{A} + C \right) \right\|^2 = \min_C \left\| \frac{1}{\beta}\boldsymbol{A} + C \right\|^2. \tag{72}$$

By setting the derivative with respect to C to zero, we find the optimal shift C^*:

$$\frac{\partial}{\partial C}\sum_{a\in\mathcal{V}}\left(\frac{A(s,a)}{\beta} + C\right)^2 = 0 \implies C^* = -\frac{1}{\beta|\mathcal{V}|}\sum_{a\in\mathcal{V}} A(s,a). \tag{73}$$

In modern reinforcement learning, it is standard practice to employ advantage normalization within each time step, ensuring that the mean advantage $\frac{1}{|\mathcal{V}|}\sum_{a_t\in\mathcal{V}} A(s_t, a_t) = 0$. Under this condition, the optimal shift C^* becomes exactly zero.

This provides a rigorous justification for the definition of $\boldsymbol{z}^*$ in Equation (7): among all possible logit targets that induce the optimal policy, our choice is the unique target that minimizes the L_2 distance from the current behavioral logits. By minimizing this initial discrepancy, we ensure that the parameter updates $\Delta\theta$ are as small as possible to achieve the desired policy shift, which is consistent with the principle of trust region updates. While derived via $\mathcal{L}_{\text{LCO-MSE}}$, this logic extends to $\mathcal{L}_{\text{LCO-LCH}}$ because the Log-Cosh loss is also a symmetric, monotonic function of the residual, thus sharing the same minimum at $C = 0$ under normalized advantages.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly outline the paper's core contributions: the analysis of logits convexity as a factor for stability and the proposal of the LCO framework. The claims are supported by theoretical analysis (Section 4) and empirical experiments across multiple benchmarks.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: The paper provides a "Limitations" section in the appendix.

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The paper provides detailed proofs and assumption in the main text and appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper specifies the model families used, the specific benchmarks, and compares the proposed LCO variants against standard RL baselines.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The paper utilizes standard open-source benchmarks, and code is provided in the supplemental material.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies the training and test details in the experimental setting section.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Evaluation results are reported as the mean and standard deviation across four random seeds.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper provides training efficiency experiment.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research is foundational and focuses on the mathematical and optimization stability of LLMs, adhering to standard responsible AI practices.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: By improving the stability of RL for LLMs, the work contributes to more reliable and controllable AI alignment, which is essential for safe deployment in instruction-following and reasoning tasks.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper focuses on a new optimization framework and theoretical analysis rather than the release of a high-risk model or dataset.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper cites foundational work for models/benchmarks used in the experiments.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [No]

Justification: The paper does not release new assets.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The research does not involve crowdsourcing or human subject experiments

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No human subjects were involved in the research.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The LLM is used only for writing, editing, and formatting purposes in this paper.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.