DYNAMIC MINIMAX REGRET OPTIMIZATION FOR ROBUST LLM POST-TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Modern LLM training increasingly relies on heterogeneous data sources spanning different domains, tasks, preference distributions, and difficulty levels. We study *dynamic minimax regret* for group-distributionally robust LLM post-training under instantaneous mini-batch-only bandit feedback. The framework views the training as a two-player sampler-optimizer process: a sampler adaptively selects among data sources using bandit feedback, while an optimizer updates the model parameters using stochastic gradients from the selected source. We focus on the practically restrictive setting where source losses evolve with model training but historical data are not re-evaluated, requiring the sampler to track instantaneous worst-sources from stale partial feedback. We propose DUCB-OGD, a simple and scalable algorithm that couples a Discounted Upper-Confidence-Bound sampler with an Online Gradient Descent optimizer. The sampler maintains exponential moving average loss estimates and confidence radii based on discounted effective sample sizes, avoiding costly re-evaluation of past data or intrusive changes to standard training pipelines. For K data sources and T training steps, we prove that DUCB-OGD achieves a dynamic minimax regret of $\tilde{O}(K^{1/4}T^{3/4})$, which is optimal up to logarithmic factors for the undiscounted objective under our feedback model. Extensive experiments across supervised fine-tuning, preference optimization, and reinforcement learning show that DUCB-OGD integrates seamlessly into modern LLM training pipelines and improves worst-group robustness with negligible computational overhead compared with standard sampling baselines.

1 INTRODUCTION

Diverse data mixtures are central to modern Large Language Model (LLM) post-training. For example, supervised fine-tuning data may combine instruction-following, coding, mathematical reasoning, safety, and domain-specific examples; preference datasets may reflect different annotator populations or task families; and reinforcement learning objectives may expose the model to prompts whose rewards evolve unevenly during training (Ouyang et al., 2022; Rafailov et al., 2023; Schulman et al., 2017; Shao et al., 2024). As a result, post-training is not merely an average-loss minimization problem. A model can improve substantially on common or easy sources, while remaining weak on smaller, harder, or temporarily under-trained groups. This makes robust performance across data sources a central objective in LLM post-training. In practical deployment, worst-source robustness is a complementary and equally important criterion to the commonly reported mean performance, but often under-emphasized in aggregate evaluations.

Group-distributionally robust optimization (GDRO) provides a principled formulation for this goal by minimizing the worst-case expected loss over a predefined set of data sources (Duchi & Namkoong, 2021; Sagawa* et al., 2020; Oren et al., 2019). Informally, GDRO seeks model parameters that perform well on the most challenging source rather than the average performance across sources. However, the standard GDRO viewpoint is essentially static: robustness is typically evaluated with respect to the worst-performing source at the final or time-averaged solution, or determined in hindsight over the training trajectory (Zhang et al., 2023; Soma et al., 2025). This static benchmark does not fully capture the shift of the worst-performing source as the model evolves in LLM post-training. A robust algorithm should therefore track and respond to the evolving worst source, rather than optimize against a fixed source chosen in hindsight.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Table 1: GDRO formulations of LLM post-training objectives and estimated worst-source (WS) switching. The final column reports the mean number of switches per 1,000 steps, with the range across experiments in brackets. The worst source has the largest exponentially smoothed loss for SFT/DPO or the lowest exponentially smoothed reward for GRPO. Additional optimization dynamics are reported in Sections 4 and D.5.

Objective	$-\ell(\theta, \cdot)$	**WS Switches / 1k Steps**
SFT	$\mathbb{E}_{o\sim\mathcal{D}^k(\cdot\|q)}\left[\log \pi_\theta(o \mid q)\right]$	**41** [3,124]
DPO	$\mathbb{E}_{(o^+,o^-)\sim\mathcal{D}^k(\cdot\|q)}\left[\log p_\theta^{\mathrm{BT}}(o^+ \succ o^- \mid q)\right]$	**31** [13,53]
RL (GRPO)	$\mathbb{E}_{o\sim\pi_\theta(\cdot\|q)}\left[r(q,o)\right]$	**20** [0,32]

Recent work has shown that mixture proportions can strongly affect pretraining and post-training performance, motivating both offline mixture optimization and online adaptive sampling methods (Xie et al., 2023; Chen et al., 2023; Ye et al., 2025; Corrado et al., 2025; Shin et al., 2025). Such mechanisms can be expensive or intrusive in large-scale post-training systems, where throughput, memory efficiency, and compatibility with distributed training libraries are critical. This creates a gap between the theoretical appeal of GDRO and the practical constraints of LLM post-training: we need adaptive algorithms that are robust, theoretically grounded, and lightweight enough to integrate into standard training pipelines relying on well-optimized deep learning libraries. We discuss related work in detail in Appendix A.

In this paper, we study *dynamic minimax regret* for group-distributionally robust LLM post-training under instantaneous mini-batch-only bandit feedback. The training trajectory can be naturally decomposed into a *sampler*, which selects a source using only its observed mini-batch loss, and an *optimizer*, which updates the model using the corresponding stochastic gradient. We propose DUCB-OGD, a simple algorithm that combines a Discounted Upper-Confidence-Bound (DUCB) (Garivier & Moulines, 2011) sampler with an Online Gradient Descent (OGD) (Hazan, 2016) optimizer. In contrast, while Zang et al. (2025) already study dynamic worst-source tracking, they refresh source-loss estimates by re-evaluating accumulated historical samples. Our replay-free formulation instead uses fresh mini-batches as unbiased stochastic estimates of population loss, adds no auxiliary model evaluations, and requires only $O(K)$ scalar state, at the cost of tracking unobserved sources from stale feedback. We show that this simple sampler-optimizer design achieves dynamic minimax regret of $\tilde{O}(K^{1/4}T^{3/4})$ for K data sources over T training steps, which is optimal for the undiscounted objective up to logarithmic factors under our feedback model.

Our main contributions are summarized as follows:

1. We formulate robust LLM post-training under *instantaneous mini-batch-only feedback*, where evolving source losses must be tracked from stale observations without historical re-evaluation or auxiliary model evaluations.
2. We propose DUCB-OGD, an efficient algorithm that combines DUCB sampling with OGD optimization using only mini-batch loss feedback, and prove that it achieves a dynamic minimax regret of $\tilde{O}(K^{1/4}T^{3/4})$, matching the optimal rate up to logarithmic factors for the undiscounted objective under our feedback model.
3. We evaluate the method across supervised fine-tuning, preference optimization, and reinforcement learning, showing improved worst-source robustness with negligible computational overhead.

2 Problem Setup

Let $\mathbb{X}$ be a data space with K *data sources* indexed by $[K] := \{1, \ldots, K\}$. Each source $k \in [K]$ is associated with a distribution $\mathcal{D}^k$ supported on $\mathbb{X}$, and each query to source k returns *i.i.d.* samples from $\mathcal{D}^k$. Let θ be the learnable parameter from a parameter space $\mathbf{\Theta}$. Let $\ell : \mathbf{\Theta} \times \mathbb{X} \to \mathbb{R}$ be a differentiable loss function. The *expected loss* of source k at the model parameter θ is defined as $\mu^k(\theta) := \mathbb{E}_{X\sim\mathcal{D}^k}[\ell(\theta, X)]$. A global minimax solution is any optimizer of the GDRO objective, i.e.,

$$\theta^\star \in \arg\min_{\theta\in\mathbf{\Theta}} \max_{k\in[K]} \mu^k(\theta). \tag{1}$$

This formulation captures a broad class of LLM post-training objectives through appropriate choices of the source distributions and loss functions as described in Tab. 1. Let π_θ be an LLM, and q, o be a pair of prompt and response. Then $\pi_\theta(o|q)$ is the probability that the LLM generates the response o given the prompt q. In supervised fine-tuning (SFT), each source generates prompt-response pairs, and the loss is the negative log-likelihood of the target response (Ouyang et al., 2022). For direct preference optimization (DPO), let (o^+, o^-) denote a preferred and dispreferred response pair. Let $p_\theta^{\mathrm{BT}}(o^+ \succ o^-|q)$ denote the Bradley-Terry probability (Rafailov et al., 2023; Bradley & Terry, 1952) that the model prefers o^+ to o^-, and the loss is the negative Bradley-Terry preference probability. For reinforcement learning (RL), let $r(q, o)$ denote the reward assigned to the response o for the prompt q. Each source generates prompts, the model induces the response distribution, and the loss is the negative reward (Schulman et al., 2017; Shao et al., 2024). For simplicity, Tab. 1 presents these objectives at the sequence level, omitting token-wise factorization, reference-model regularization, and implementation-specific normalization terms.

We consider an iterative training algorithm $\mathcal{A}$, consisting of a *sampler* and an *optimizer*, that runs for T steps. At each step $t \in [T]$, the sampler specifies a distribution $\mathcal{P}_t$ over the data-source indices $[K]$. A source $k_t \sim \mathcal{P}_t$ is then selected, and a mini-batch $\mathcal{B}_t \subseteq \mathbb{X}^{k_t}$ of *i.i.d.* samples is drawn from the corresponding distribution $\mathcal{D}^{k_t}$. The model observes $\mathcal{B}_t$ and calculates the empirical loss $\hat{\mu}^{k_t}(\theta_t; \mathcal{B}_t) := \sum_{X \in \mathcal{B}_t} \ell(\theta_t, X)/|\mathcal{B}_t|$ based on the current model parameter θ_t. Then the optimizer updates the parameter to obtain θ_{t+1} for the next step. The sampler observes only $\hat{\mu}^{k_t}(\theta_t; \mathcal{B}_t)$; it neither observes the current losses of unselected sources nor re-evaluates historical mini-batches. After T steps, the algorithm generates a trajectory of data source indices and model parameters $\mathcal{A}_T := \{(k_t, \theta_t), t = 1, \ldots, T\}$. Our goal is to evaluate this trajectory by how well it controls the *adaptive* sequence of maximum source-wise losses throughout the training. Following the notion of dynamic regret in online learning, the (undiscounted) *dynamic minimax regret* is defined as follows.

Definition 1 (Undiscounted Dynamic Minimax Regret)**.** *Let $\theta^\star$ be the global minimax optimum in Eq.* (1)*. The undiscounted dynamic minimax regret of an algorithm $\mathcal{A}$ after T steps is defined as*

$$R(\mathcal{A}_T) := \sum_{t=1}^{T} \max_{k \in [K]} \mu^k(\theta_t) - T \max_{k \in [K]} \mu^k(\theta^\star). \tag{2}$$

This definition compares the worst-case loss incurred by $\mathcal{A}_T$ with the best achievable minimax loss. Crucially, the maximizing source in $\max_k \mu^k(\theta_t)$ may vary with t, which makes the regret *dynamic* (Zang et al., 2025), *i.e.*, at each step, it reflects the source that is currently under-performing, rather than a single fixed source chosen in hindsight (Zhang et al., 2023; Soma et al., 2025). To analyze the sampler-optimizer structure, we decompose the regret into an optimization component and a bandit component. The optimization component measures how well the optimizer competes with the minimax solution on the sampled sources. The bandit component measures how well the sampler tracks the instantaneous worst-performing source. We also introduce discount factors $\rho, \gamma \in [0, 1]$ to emphasize recent observations, which is natural in post-training because source-wise losses evolve as the model changes.

Definition 2 (Optimization Regret and Dynamic Bandit Regret)**.** *Let $\rho, \gamma \in [0, 1]$. The discounted optimization regret of an algorithm $\mathcal{A}$ after T steps is defined as*

$$R_o^\rho(\mathcal{A}_T) := \sum_{t=1}^{T} \rho^{T-t} \Big(\mu^{k_t}(\theta_t) - \mu^{k_t}(\theta^\star) \Big), \tag{3}$$

and the discounted dynamic bandit regret is defined as

$$R_b^\gamma(\mathcal{A}_T) := \sum_{t=1}^{T} \gamma^{T-t} \Big(\max_{k \in [K]} \mu^k(\theta_t) - \mu^{k_t}(\theta_t) \Big). \tag{4}$$

The optimization regret R_o^ρ captures the loss incurred by following the optimizer trajectory instead of using the minimax solution on the sampled sources. The bandit regret R_b^γ captures the loss incurred by sampling k_t instead of the instantaneous worst-performing source at time t. The following decomposition connects these quantities to the original undiscounted regret. Its proof simply follows from the fact that $\max_k \mu^k(\theta) \geq \mu^{k_t}(\theta)$ for all θ and t and is provided in Appendix C.1.

Proposition 1 (Regret Decomposition). *Any trajectory $\mathcal{A}_T$ satisfies $R(\mathcal{A}_T) \leq R_o^1(\mathcal{A}_T) + R_b^1(\mathcal{A}_T)$.*

Finally, we define the normalized discounted dynamic minimax regret as follows.

Definition 3 (Dynamic Minimax Regret). *For any $\rho, \gamma \in [0, 1]$, the normalized (ρ, γ)-discounted dynamic minimax regret of an algorithm $\mathcal{A}$ after T steps is defined as*

$$R^{\rho,\gamma}(\mathcal{A}_T) := \frac{T}{N_T(\rho)} R_o^\rho(\mathcal{A}_T) + \frac{T}{N_T(\gamma)} R_b^\gamma(\mathcal{A}_T), \tag{5}$$

where $N_T(\beta) := \sum_{t=1}^T \beta^{T-t}$ for any $\beta \in [0, 1]$.

The normalization converts each discounted sum to the scale of a length-T horizon, making regret bounds comparable across choices of ρ and γ. Here, ρ and γ separately discount the optimization and bandit components, whereas β will be selected by the analyses. When $\rho = \gamma = 1$, the normalized regret reduces to the undiscounted sampler-optimizer decomposition, and by Proposition 1, $R(\mathcal{A}_T) \leq R^{1,1}(\mathcal{A}_T)$. For $\rho, \gamma < 1$, the discounted statistics used in our analysis can be viewed as exponential moving-average (EMA) estimates of past observations. This connects our theoretical quantities to EMA-based state tracking, which is widely used in deep learning optimization and training diagnostics Polyak & Juditsky (1992); Kingma & Ba (2014).

We make the following standard assumptions for the theoretical analysis. Specifically, Assumptions 1 and 2 are required for analyzing gradient descent, and Assumption 3 is required to ensure that the empirical loss and gradient estimators are unbiased.

Assumption 1 (Finite Optimization Domain). *There exists a closed convex $\mathbf{\Theta}'$ and $D > 0$ such that, for any θ, θ' generated by the algorithm, there is $\theta, \theta' \in \mathbf{\Theta}'$ and $\|\theta - \theta'\| \leq D$.*

Assumption 2 (Bounded Lipschitz Loss). *There exists $C, G > 0$ such that, for all $\theta \in \mathbf{\Theta}'$ and $X \in \mathbb{X}$, $\ell(\theta, X) \in [0, C]$ and ℓ is G-Lipschitz in θ.*

Assumption 3 (Source-Wise IID Sampling). *All data (i.e., prompts and responses, if applicable) collected by querying $\mathcal{D}^k$ are sampled i.i.d. for all $k \in [K]$.*

We interpret the Lipschitz assumption as a regularity condition along the realized training trajectory rather than as a claim over the entire parameter space. Standard implementation techniques such as gradient clipping, KL regularization, and bounded reward design help control the observed loss variation as is illustrated in Appendix D.5, providing empirical evidence consistent with the assumptions. Furthermore, the low-epoch regime of LLM post-training ensures that most training samples are observed at most once, which is aligned with the source-wise *i.i.d.* assumption.

3 METHOD

3.1 DUCB-OGD: HIGH-LEVEL DESCRIPTION

The training of modern deep neural networks, especially LLMs, relies heavily on well-optimized libraries. A standard pipeline typically involves a *sampler*, which selects a mini-batch of data samples as the model inputs, and an *optimizer*, which updates the model parameters using gradients computed from those mini-batches. Thus, we consider a two-player sampler-optimizer framework for seamless deployment in real-world training systems.

Optimizer. Recall that in each step t, the sampler generates a mini-batch of training samples $\mathcal{B}_t$ from $\mathcal{D}^{k_t}$. The optimizer updates the model parameters according to

$$\theta_{t+1} \leftarrow \theta_t - \eta \nabla \hat{\mu}^{k_t}(\theta_t; \mathcal{B}_t), \tag{6}$$

where η is the learning rate and $\hat{\mu}^{k_t}(\theta_t; \mathcal{B}_t)$ is the average empirical loss. When each data source corresponds to a distinct task or domain (*e.g.*, verbal reasoning *v.s.*, math reasoning), the loss function depends on the data source selected at the current step. Thus the optimizer is an instance of *online gradient descent (OGD)* Hazan (2016). We analyze unprojected OGD under Assumption 1, which explicitly requires the iterates to remain in the optimization domain.

Sampler. At the beginning of each step t, the sampler observes the algorithm trajectory $\mathcal{A}_{t-1}$ and the historical empirical losses $\{\hat{\mu}^{k_\tau}(\theta_\tau; \mathcal{B}_\tau), \tau = 1, \ldots, t-1\}$. Selecting which data source to query

is challenging for three reasons. First, empirical mini-batch losses are stochastic estimates of the corresponding source-wise expected losses. Second, the sampling environment is non-stationary because the source-wise losses change as the optimizer updates the model parameters. Third, the sampler faces an exploration-exploitation tradeoff: it must collect enough observations from all sources to estimate their losses accurately, while also allocating samples to sources that appear to be currently worst-performing.

However, unlike the adversarial bandit settings, where losses could change arbitrarily over time, the variation in source-wise losses between consecutive steps is controlled by the OGD learning rate η in Eq. (6) together with the Lipschitz condition in Assumption 2. Thus, the sampling problem is naturally related to slowly-varying non-stationary bandits (Krishnamurthy & Gopalan, 2025). While the classical upper-confidence-bound (UCB) algorithm suits stationary environments, we use Discounted UCB (or DUCB) (Garivier & Moulines, 2011), a simple UCB variant with a discounting mechanism.

Specifically, let $\mathbb{1}\{\cdot\}$ be the indicator function. For a discount factor $\beta \in [0, 1)$, define the EMA estimate of the empirical loss of source k at step t as

$$\hat{\mu}_t^k := \frac{1}{N_t^k(\beta)} \sum_{\tau=1}^{t} \mathbb{1}\{k_\tau = k\} \cdot \beta^{t-\tau} \hat{\mu}^k(\theta_\tau; \mathcal{B}_\tau), \quad N_t^k(\beta) := \sum_{\tau=1}^{t} \mathbb{1}\{k_\tau = k\} \cdot \beta^{t-\tau}. \tag{7}$$

Here, $N_t^k(\beta)$ reflects the *effective sample size* of source k. Let $H(\beta) := 1/(1-\beta)$ for $\beta \in [0, 1)$ (known as the *effective horizon* of a discounted sequence) and set $H(1) = T$ for undiscounted case.

During the first K steps, DUCB performs initial exploration by selecting each source once, i.e., $k_t = t$ for $t = 1, \ldots, K$. Then for $t \geq K + 1$, it selects the source k_t as

$$k_t \leftarrow \arg\max_k \left(\hat{\mu}_{t-1}^k + r_{t-1}^k\right), \quad r_{t-1}^k := 2C\sqrt{\frac{2\alpha \ln H(\beta)}{N_{t-1}^k(\beta)}}, \tag{8}$$

where $\alpha > 0$ is a constant controlling the aggressiveness of exploration and C bounds the loss as in Assumption 2. Equivalently, this sampling rule can be represented as a categorical distribution $k_t \sim \mathrm{Cat}(p_t)$ over $[K]$, with probabilities $p_{t,k} = \mathbb{1}\{k = k_t\}$ for each $k \in [K]$. We discuss the implementation details of DUCB using practical LLM post-training libraries in Appendix B.

Intuitively, the DUCB index simply couples the normalized EMA estimated empirical loss $\hat{\mu}_t^k$ with an optimistic confidence radius of $O(N_t^k(\beta)^{-1/2})$ for each $k \in [K]$. Due to the recursive identity of $N_t^k(\beta) = \beta N_{t-1}^k(\beta) + \mathbb{1}\{k_t = k\}$, the effective sample size of each data source decays by a factor of β whenever it is not selected at the current step. This naturally reflects a loss of confidence in $\hat{\mu}_t^k$ over time, as the current model parameter θ_t drifts further away from the states in which the historical losses were observed.

3.2 Main Theoretical Results

Now we present the theoretical guarantees of DUCB-OGD. While the milestones of the proof are outlined below, we provide the complete version in Appendix C.

Theorem 2 (Dynamic Minimax Regret of DUCB-OGD)**.** *Suppose μ^k is convex in θ for all k. Then with $\eta = K^{-1/4}T^{-3/4}$ in Eq.* (6) *and $\beta = 1 - K^{-1/2}T^{-1/2}$ in Eq.* (7)*, DUCB-OGD satisfies*

$$\mathbb{E}\big[R^{\rho,\gamma}(\text{DUCB-OGD}_T)\big] \leq \tilde{O}\big(K^{\frac{1}{4}}T^{\frac{3}{4}}\big), \tag{9}$$

for any $\rho, \gamma \in [1 - T^{-1}, 1]$.

In other words, the dynamic minimax regret achieves the rate of $\tilde{O}(K^{\frac{1}{4}}T^{\frac{3}{4}})$ in both undiscounted ($\rho = \gamma = 1$) and discounted scheme ($1 - T^{-1} \leq \rho, \gamma < 1$). To prove this, we first present the OGD optimization regret and the DUCB bandit regret respectively, and then combine them to obtain the overall dynamic minimax regret.

Proposition 3 (Optimization Regret of OGD)**.** *Suppose μ^k is convex in θ for all k. OGD satisfies*

$$\frac{T}{N_T(\rho)}\mathbb{E}\big[R_o^\rho(\text{OGD}_T)\big] \leq O\big(\eta^{-1}\big) + O\big(\eta T\big), \tag{10}$$

for any $\rho \in [1 - T^{-1}, 1]$.

Remark. *In the standard analysis of undiscounted and discounted OGD, the RHS of Eq.* (10) *can be minimized by choosing $\eta = O(T^{-1/2})$, which gives the rate of $O(T^{1/2})$.*

The proof of Proposition 3 follows the standard OGD analysis for convex Lipschitz functions, with modifications to account for the stochasticity in the empirical loss observations. The full version is provided in Appendix C.2. Crucially, while $\eta = O(T^{\xi})$ for any $\xi \in (-1, 0)$ in Eq. (10) yields an optimization regret sublinear in T, we defer the choice of η here since it dictates the rate that the sampling environment changes for DUCB, dramatically affecting its behavior. Instead, we first present the bandit regret with also an η factor, then jointly tune η to optimize both regrets.

Proposition 4 (Bandit Regret of DUCB)**.** *Suppose $\eta = O(T^{\xi})$ for some $\xi \in [-3/2, 0)$. Then with $\beta = 1 - \eta^{2/3} K^{-1/3}$, DUCB satisfies*

$$\frac{T}{N_T(\gamma)} \mathbb{E}\big[R_b^{\gamma}(\mathrm{DUCB}_T)\big] \le \tilde{O}\big(\eta^{\frac{1}{3}} K^{\frac{1}{3}} T\big), \tag{11}$$

for any $\gamma \in [1 - T^{-1}, 1]$ and any K independent of T.

Remark. *Note that when $\gamma = 1$, $R_b^1(\mathrm{DUCB})$ matches the standard regret of slowly-varying non-stationary bandits by Krishnamurthy & Gopalan (2025). Alternatively, let $V := O(\eta T)$ be the total variation budget of the sampling environment over T steps. Then Proposition 4 can be rewritten as $R_b^1(\mathrm{DUCB}) \le \tilde{O}(K^{\frac{1}{3}} T^{\frac{2}{3}} V^{\frac{1}{3}})$, matching the result of Besbes et al. (2014).*

We provide a proof outline of Proposition 4. While standard DUCB analysis focuses on abruptly changing environments Garivier & Moulines (2011), we adapt the proof to our slowly-varying environment based on the following decomposition. Denote $k_t^\star := \arg\max_k \mu^k(\theta_t)$. Let $\tilde{\mu}_t^k := \mathbb{E}[\hat{\mu}_t^k | \mathcal{A}_t]$ be the expected EMA estimated loss of the source k at step t. Then by Definition 2, we have $R_b^{\gamma}(\mathcal{A}_T) = \sum_{t=1}^T \gamma^{T-t}(\mu^{k_t^\star}(\theta_t) - \mu^{k_t}(\theta_t))$, and for any $t \in [T]$,

$$\mu^{k_t^\star}(\theta_t) - \mu^{k_t}(\theta_t) = \underbrace{\big(\tilde{\mu}_{t-1}^{k_t^\star} - \tilde{\mu}_{t-1}^{k_t}\big)}_{\text{Term 1}} - \underbrace{\big(\tilde{\mu}_{t-1}^{k_t^\star} - \mu^{k_t^\star}(\theta_t)\big)}_{\text{Term 2}} + \underbrace{\big(\tilde{\mu}_{t-1}^{k_t} - \mu^{k_t}(\theta_t)\big)}_{\text{Term 3}}. \tag{12}$$

We prove in Proposition 7 in Appendix C.4 that the DUCB update rule guarantees that Term 1 is bounded by $O(r_{t-1}^k)$ with *high probability*, where r_{t-1}^k is the confidence radius defined in Eq. (8). Terms 2 and 3 reflect the bias of the EMA estimated loss w.r.t. the true loss, which are shown to be bounded by $\tilde{O}(\eta G^2 H(\beta))$ in Proposition 8 in Appendix C.5. The complete proof of Proposition 4 is provided in Appendix C.3. Now we can prove Theorem 2.

Proof of Theorem 2. Recall that the optimization regret in Proposition 3 is sublinear only when $\eta = O(T^{\xi})$ for some $\xi \in (-1, 0)$, which satisfies the condition of Proposition 4. Then by Definition 3, and using Proposition 3 and Proposition 4, we have

$$\mathbb{E}\big[R^{\rho,\gamma}(\text{DUCB-OGD}_T)\big] \le O\big(\eta^{-1}\big) + O\big(\eta T\big) + \tilde{O}\big(\eta^{\frac{1}{3}} K^{\frac{1}{3}} T\big). \tag{13}$$

The RHS is minimized (order-wise) by taking $\eta = K^{-1/4} T^{-3/4}$, which concludes the proof. □

Further note that when $\rho = \gamma$, the average dynamic minimax regret $R^{\rho,\gamma}/T$ gives the minimax optimality gap measured by the EMA estimated worst-case loss. Then the following result is a direct consequence of Theorem 2.

Corollary 5. *Under the conditions of Theorem 2, the expected minimax optimality gap at the EMA-weighted iterate of* $\mathrm{DUCB\text{-}OGD}_T$*, for any discount factor $\gamma \in [1 - T^{-1}, 1]$, satisfies*

$$\mathbb{E}\left[\max_{k \in [K]} \mu^k(\tilde{\theta}_T^{\gamma})\right] - \max_{k \in [K]} \mu^k(\theta^\star) \le \tilde{O}\big(K^{\frac{1}{4}} T^{-\frac{1}{4}}\big), \tag{14}$$

where $\tilde{\theta}_T^{\gamma} := \sum_{t=1}^T \gamma^{T-t} \theta_t / N_T(\gamma)$.

Specifically when $\rho = \gamma = 1$, the results in Propositions 3 and 4 are shown to be optimal up to logarithmic factors. This follows from the lower bounds for the optimization regret and the dynamic bandit regret, by Hazan (2016) and Krishnamurthy & Gopalan (2025) respectively, based on constructing a hard instance where the loss landscape is non-smooth, and the worst-performing data source alters periodically over time. Then it immediately follows that, in the undiscounted setting, the bound in Theorem 2 is also optimal.

Theorem 6 (Lower Bound)**.** *For every admissible sampler-optimizer algorithm $\mathcal{A}_T$, there exsits a problem instance for which the undiscounted regret satisfies $\mathbb{E}[R^{1,1}(\mathcal{A}_T)] \ge \tilde{\Omega}(K^{\frac{1}{4}} T^{\frac{3}{4}})$.*

3.3 IMPLICATIONS

We highlight two implications following the theoretical analysis.

Comparison to Pseudo-Regret GDRO. Other GDRO frameworks proposed by Zhang et al. (2023) and Soma et al. (2025) adopt the notion of *pseudo regret*, defined as $\tilde{R}(\mathcal{A}_T) := \max_k \sum_{t=1}^{T} \mu^k(\theta_t) - T \max_k \mu^k(\theta^\star)$, which can be bounded by $\tilde{O}(K^{1/2}T^{1/2})$. The pseudo regret is fundamentally easier to analyze than our dynamic regret in Definition 1, since it always holds that $\tilde{R} \leq R$ (and $R \leq R^{1,1}$ by Proposition 1). In real-world learning problems such as LLM post-training, the worst-case data source may vary as the training progresses, and dynamic regret captures this by tracking the gap w.r.t. the worst-case loss in each step (we provide an example in Section 4.3). Moreover, pseudo regret controls only the uniform time-average of algorithm performance, while our dynamic-regret formulation yields guarantees for exponentially weighted trajectory averages emphasizing on later iterates as shown in Corollary 5. This better matches the practical deep learning workflows, where training dynamics are often monitored through EMA-style statistics and final checkpoints are selected from the later stage of model training.

Scope of the Lower Bound. We further note that Theorem 6 establishes optimality for the undiscounted objective under the instantaneous mini-batch-only feedback oracle, and does not imply optimality for discounted objectives or when richer feedback is available. Indeed, Zang et al. (2025) re-evaluate historically collected samples under the current model and obtain a dynamic-regret rate of $\tilde{O}(K^{1/2}T^{1/2})$. Yet this stronger oracle requires expensive model evaluations over a linearly growing replay-buffer, and a generalization step from replay-based empirical losses to population losses.

4 EXPERIMENTS

4.1 OBJECTIVES AND DATASETS

We evaluate DUCB-OGD across three LLM post-training objectives:

- *SFT:* We train base models for 32k samples on the *Math* and *Orca* sources of Tulu-v2 (Ivison et al., 2023) using TRL (von Werra et al., 2020). We evaluate the minimum of GSM8k accuracy (Cobbe et al., 2021) and IFEval prompt accuracy (Zhou et al., 2023).
- *DPO:* We train instruction-tuned models for 16k samples on the *Code*, *Multilingual*, and *Stem* sources of HelpSteer3 (Wang et al., 2025c;e;d) using TRL. The primary metric is minimum source-wise test preference accuracy.
- *GRPO:* We consider (i) *Reasoning*, comprising *Arc*, *Countdown*, and *Zebra* as three sources (Chollet, 2019; Katz et al., 2025; Lin et al., 2025), with a 10k-sample budget using Verl (Sheng et al., 2024); and (ii) CommonsenseQA (Talmor et al., 2019), partitioned into *Social*, *Physical*, *Causal-Temporal*, and *General* sources, with a 6.4k-sample budget using TRL. We report the minimum source-wise test accuracy.

For each objective, we train models of different sizes from the Qwen2.5 (Yang et al., 2024; Team, 2024), Qwen3 (Team, 2025), Llama3 (AI@Meta, 2024), and GLM4 (GLM et al., 2024) families using different sampling algorithms.

4.2 ROBUSTNESS ACROSS POST-TRAINING OBJECTIVES AND MODEL FAMILIES

Tab. 2 reports minimum source-wise accuracy as the primary metric and unweighted average accuracy as a reference. All metrics are evaluated at the final training checkpoint. Additional models and metrics are reported in Appendix D.

For SFT, we evaluate an adapted Exp3-style proxy based on DynamixSFT (Shin et al., 2025). A faithful implementation requires virtual parameter updates to estimate anticipated source-wise loss decreases and is not directly compatible with our training pipeline, and our proxy instead uses observed source-wise loss decreases. DUCB achieves the highest minimum accuracy for both models, whereas the Exp3-style sampler obtains higher average accuracy, illustrating the robustness-average performance trade-off.

Table 2: Main results of different sampling algorithms on various LLM post-training objectives. The primary metric is *minimum* source-wise accuracy.

Objective	Dataset		Model	Sampler	Accuracy	
	Train	Test			***Min***	**Avg**
SFT	Tulu-v2 (subset)	GSM8k + IFEval	Qwen2.5-7B (Base)	Uniform	27.15	39.36
				Exp3-style	27.73	**40.72**
				DUCB (Ours)	**28.52**	37.89
			Llama3.1-8B (Base)	Uniform	22.07	25.98
				Exp3-style	21.48	**27.93**
				DUCB (Ours)	**24.22**	25.29
DPO	HelpSteer3 (subset)		Qwen2.5-3B (Instruct)	Uniform	76.76	80.53
				AMA-R[1]	78.19	79.45
				AMA-S[1]	**80.25**	**82.35**
				DUCB (Ours)	**79.33**	81.59
			Llama3.2-3B (Instruct)	Uniform	73.79	81.06
				AMA-R[1]	74.22	78.00
				AMA-S[1]	**77.37**	80.48
				DUCB (Ours)	**77.42**	**81.88**
GRPO	Reasoning		Qwen2.5-1.5B (Instruct)	Uniform	18.89	28.98
				Exp3-style	20.91	28.78
				DUMP[2]	19.45	**29.70**
				DUCB (Ours)	**21.81**	28.74
	CommonsenseQA		GLM4-9B (Instruct)	Uniform	77.91	84.08
				Exp3-style	82.56	85.55
				DUMP[2]	81.40	85.55
				DUCB (Ours)	**84.88**	**87.74**

[1] Corrado et al. (2025), requires training K specialist models; [2] Wang et al. (2025b).

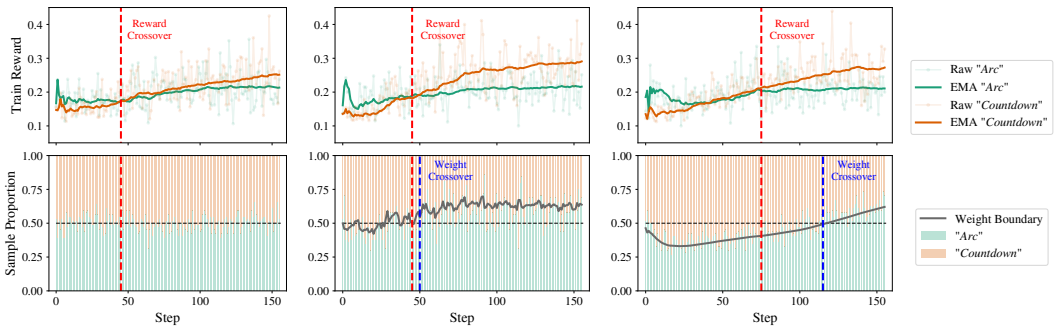
For DPO, we compare against faithful implementations of AutoMixAlign Corrado et al. (2025), including both AMA-R and AMA-S. AutoMixAlign uses source-specific specialist models to compute excess-loss-based sampling signals, which requires training K additional specialists before the generalist model. DUCB achieves competitive or better worst-source performance without this auxiliary training cost. These results further support the robustness of DUCB under a substantially lighter feedback and computational requirement. Specifically, across three Qwen2.5-3B runs, DUCB obtains 79.33 ± 0.62 minimum accuracy versus 76.76 ± 1.65 for Uniform, corresponding to a paired gain of 2.57 ± 1.04 percentage points.

For GRPO, we compare against an Exp3-style sampler-only proxy of MT-GRPO Ramesh et al. (2026) and a faithful implementation of DUMP Wang et al. (2025b) which adopts a switching-window UCB sampler. We use the source-wise training reward with improvement correction and the training advantage as the sampling signal, respectively. We omit additional components of MT-GRPO such as rollout filtering and only implement the sampler in order to maintain uniformity with other samplers and compatibility between the TRL and Verl libraries. The results show clear margins of DUCB over other uniform and adaptive sampling baselines on the *minimum* accuracies across both datasets.

Across the six main settings, DUCB achieves the highest minimum accuracy in five and remains competitive with AMA-S in the remaining DPO setting. Detailed discussions of the results are also provided in Appendix D.

4.3 Computation Efficiency and Dynamic Worst-Source Adaptation

Training overhead. We measure per-step runtime across three Qwen2.5-3B DPO runs, with full results reported in Appendix D.4. DUCB requires 21.94 ± 1.14 seconds per step, compared with 21.90 ± 1.29 for Uniform; the difference is small relative to the observed run-to-run variation. Its



(a) Uniform (b) DUCB (c) Exp3

Figure 1: GRPO training rewards and sampling weights for uniform sampling, DUCB (guided by dynamic regret), and Exp3 (guided by pseudo regret) on two data sources (*Arc* and *Countdown*).

sampler adds no auxiliary model evaluations and maintains only $O(K)$ scalar state. By comparison, AMA additionally trains K source-specific specialists before training the generalist model. Thus, DUCB introduces little measurable online overhead beyond the underlying LLM training step and avoids AMA's auxiliary specialist-training cost.

Adaptation to evolving source difficulty. We further compare DUCB's instantaneous tracking with the cumulative-reward updates of Exp3 on the *Arc* and *Countdown* sources of the Reasoning task (Fig. 1). Arc initially has higher reward than Countdown, but this ordering later reverses, making Arc the new worst-performing source. DUCB shifts its sampling priority toward Arc shortly after the crossover, whereas Exp3 continues to favor Countdown until later observations outweigh its accumulated history. This representative example, together with the switching frequencies in Tab. 1, illustrates DUCB's ability to adapt to changing source difficulty. Because instantaneous tracking is also more sensitive to short-term reward variation, we use DEMA smoothing and forced exploration as practical stabilizers, as detailed in Appendix B.

5 Conclusion

We study dynamic minimax regret for robust multi-source LLM post-training under instantaneous mini-batch-only bandit feedback. We propose DUCB-OGD, which couples discounted UCB sampling with online gradient descent and requires neither historical re-evaluation nor auxiliary model evaluations. Across SFT, DPO, and GRPO, DUCB-OGD achieves the highest minimum source-wise accuracy in the majority of the settings and remains competitive with contemporary baselines but avoids expensive model re-evaluations. Frequent worst-source switching is observed across all three objectives, while DUCB's per-step runtime remains close to Uniform, supporting the relevance and practical efficiency of instantaneous dynamic tracking.

It is shown that DUCB-OGD is most applicable when worst-source robustness is the primary objective, source difficulty changes during training, and only instantaneous mini-batch feedback is available or affordable due to practical constraints. Trade-offs may occur between worst-source and average performances. The guarantees involving optimization also do not cover non-convex loss landscapes. Our comparisons include both faithful baseline implementations and explicitly adapted sampler-only proxies, which avoid objective-specific training modifications. The evaluation is limited to 0.5B-9B models and unimodal datasets with relatively few sources. The performance on larger models and datasets with more complicated distributional characteristics remains to be studied.

AI Use Statement

In this work, we did not use generative AI tools to generate datasets, develop theoretical models or methods, formulate or prove mathematical claims, design experiments, implement methods, process data, or interpret experimental results. We used generative AI tools only to polish writing and improve readability. We reviewed the accuracy and consistency of all AI-assisted work, including revised text,

equations, and descriptions of experimental results. We take responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

Reproducibility statement

We provide the complete DUCB-OGD algorithm and theoretical assumptions in the main text, with full proofs in Appendix C. Appendix B and D document the implementation, practical stabilizers, key training configurations, and hyperparameters. The datasets, pretrained models, sample budgets, evaluation metrics, final-checkpoint protocol, multi-seed results, and hardware used for runtime measurements are described in Section 4 and Appendix D. All datasets and model checkpoints are publicly available and cited. We intend to release the training and evaluation code, full configurations, and scripts upon publication.

References

Alekh Agarwal and Tong Zhang. Minimax Regret Optimization for Robust Machine Learning under Distribution Shift. In Po-Ling Loh and Maxim Raginsky (eds.), *Proceedings of Thirty Fifth Conference on Learning Theory*, volume 178 of *Proceedings of Machine Learning Research*, pp. 2704–2729. PMLR, July 2022.

AI@Meta. The Llama 3 Herd of Models, 2024. URL https://doi.org/10.48550/arXiv.2407.21783.

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time Analysis of the Multiarmed Bandit Problem. *Machine Learning*, 47(2/3):235–256, 2002a. ISSN 08856125. doi: 10.1023/A:1013689704352.

Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The Nonstochastic Multiarmed Bandit Problem. *SIAM Journal on Computing*, 32(1):48–77, January 2002b. ISSN 0097-5397, 1095-7111. doi: 10.1137/S0097539701398375.

Jiamu Bai, Daoyuan Chen, Bingchen Qian, Liuyi Yao, and Yaliang Li. Federated Fine-tuning of Large Language Models under Heterogeneous Tasks and Client Resources. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 14457–14483. Curran Associates, Inc., 2024. doi: 10.52202/079017-0461.

Omar Besbes, Yonatan Gur, and Assaf Zeevi. Stochastic Multi-Armed-Bandit Problem with Non-stationary Rewards. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

Ralph Allan Bradley and Milton E. Terry. Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons. *Biometrika*, 39(3/4):324, December 1952. ISSN 00063444. doi: 10.2307/2334029.

J. S. Butler and Robert Moffitt. A Computationally Efficient Quadrature Procedure for the One-Factor Multinomial Probit Model. *Econometrica*, 50(3):761, May 1982. ISSN 00129682. doi: 10.2307/1912613.

Mayee Chen, Nicholas Roberts, Kush Bhatia, Jue WANG, Ce Zhang, Frederic Sala, and Christopher Ré. Skill-it! A data-driven skills framework for understanding and training language models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 36000–36040. Curran Associates, Inc., 2023.

Xiaoyin Chen, Jiarui Lu, Minsu Kim, Dinghuai Zhang, Jian Tang, Alexandre Piché, Nicolas Gontier, Yoshua Bengio, and Ehsan Kamalloo. Self-Evolving Curriculum for LLM Reasoning, 2025. Version Number: 4.

François Chollet. On the Measure of Intelligence, 2019. Version Number: 2.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training Verifiers to Solve Math Word Problems. *arXiv preprint arXiv:2110.14168*, 2021.

Nicholas E. Corrado, Julian Katz-Samuels, Adithya M Devraj, Hyokun Yun, Chao Zhang, Yi Xu, Yi Pan, Bing Yin, and Trishul Chilimbi. AutoMixAlign: Adaptive Data Mixing for Multi-Task Preference Optimization in LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 20234–20258, Vienna, Austria, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.990.

John C. Duchi and Hongseok Namkoong. Learning models with uniform performance via distributionally robust optimization. *The Annals of Statistics*, 49(3), June 2021. ISSN 0090-5364. doi: 10.1214/20-AOS2004.

Aurélien Garivier and Eric Moulines. On Upper-Confidence Bound Policies for Switching Bandit Problems. In Jyrki Kivinen, Csaba Szepesvári, Esko Ukkonen, and Thomas Zeugmann (eds.), *Algorithmic Learning Theory*, volume 6925, pp. 174–188. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. ISBN 978-3-642-24411-7 978-3-642-24412-4. doi: 10.1007/978-3-642-24412-4_16. Series Title: Lecture Notes in Computer Science.

Soumya Suvra Ghosal and Yixuan Li. Distributionally Robust Optimization with Probabilistic Group. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(10):11809–11817, June 2023. ISSN 2374-3468, 2159-5399. doi: 10.1609/aaai.v37i10.26394.

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Mingdao Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiao Xia, Xiaohan Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yifan An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhen Yang, Zhengxiao Du, Zhenyu Hou, and Zihan Wang. ChatGLM: A Family of Large Language Models from GLM-130B to GLM-4 All Tools, 2024.

Nika Haghtalab, Michael Jordan, and Eric Zhao. On-Demand Sampling: Learning Optimally from Multiple Distributions. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 406–419. Curran Associates, Inc., 2022.

Elad Hazan. Introduction to Online Convex Optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, August 2016. ISSN 2167-3888, 2167-3918. doi: 10.1561/2400000013.

Weihua Hu, Gang Niu, Issei Sato, and Masashi Sugiyama. Does Distributionally Robust Supervised Learning Give Robust Classifiers? In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2029–2037. PMLR, July 2018.

Hamish Ivison, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. Camels in a Changing Climate: Enhancing LM Adaptation with Tulu 2, 2023. Version Number: 2.

Liuyuan Jiang, Quan Xiao, Lisha Chen, and Tianyi Chen. Beyond Value Functions: Single-Loop Bilevel Optimization under Flatness Conditions. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 52003–52040. Curran Associates, Inc., 2025. doi: 10.52202/085713-1733.

Jikai Jin, Bohang Zhang, Haiyang Wang, and Liwei Wang. Non-convex Distributionally Robust Optimization: Non-asymptotic Analysis. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 2771–2782. Curran Associates, Inc., 2021.

Michael Katz, Harsha Kokel, and Sarath Sreedharan. Seemingly Simple Planning Problems are Computationally Challenging: The Countdown Game, 2025. Version Number: 2.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization, 2014. Version Number: 9.

Woosung Koh, Jeyoung Jeon, Youngjin Song, Yujin Cheon, Soowon Oh, Jaehyeong Choi, and Se-Young Yun. mSFT: Addressing Dataset Mixtures Overfitting Heterogeneously in Multi-task SFT, 2026. Version Number: 5.

Ramakrishnan Krishnamurthy and Aditya Gopalan. On Slowly-varying Non-stationary Bandits. In *Reinforcement Learning Conference*, 2025.

Zeman Li, Yuan Deng, Peilin Zhong, Meisam Razaviyayn, and Vahab Mirrokni. PiKE: Adaptive Data Mixing for Large-Scale Multi-Task Learning Under Low Gradient Conflicts. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026.

Bill Yuchen Lin, Ronan Le Bras, Kyle Richardson, Ashish Sabharwal, Radha Poovendran, Peter Clark, and Yejin Choi. ZebraLogic: On the Scaling Limits of LLMs for Logical Reasoning, 2025. Version Number: 2.

Xinyu Lin, Wenjie Wang, Yongqi Li, Shuo Yang, Fuli Feng, Yinwei Wei, and Tat-Seng Chua. Data-efficient Fine-tuning for LLM-based Recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 365–374, Washington DC USA, July 2024. ACM. ISBN 979-8-4007-0431-4. doi: 10.1145/3626772.3657807.

Weijie Lv, Xuan Xia, and Sheng-Jun Huang. Data-efficient LLM Fine-tuning for Code Generation. In *2025 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, Rome, Italy, June 2025. IEEE. ISBN 979-8-3315-1042-8. doi: 10.1109/IJCNN64981.2025.11227906.

Sadhika Malladi, Tianyu Gao, Eshaan Nichani, Alex Damian, Jason D Lee, Danqi Chen, and Sanjeev Arora. Fine-Tuning Language Models with Just Forward Passes. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 53038–53075. Curran Associates, Inc., 2023. doi: 10.52202/075280-2308.

Patrick G. Mulloy. Smoothing Data With Faster Moving Averages. *Technical Analysis of Stocks & Commodities*, 12(1):11–19, January 1994.

Yonatan Oren, Shiori Sagawa, Tatsunori Hashimoto, and Percy Liang. Distributionally Robust Language Modeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4226–4236, Hong Kong, China, 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1432.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744. Curran Associates, Inc., 2022.

Kishan Panaganti, Zhenwen Liang, Wenhao Yu, Haitao Mi, and Dong Yu. Group Distributionally Robust Optimization-Driven Reinforcement Learning for LLM Reasoning, 2026. Version Number: 1.

B. T. Polyak and A. B. Juditsky. Acceleration of Stochastic Approximation by Averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, July 1992. ISSN 0363-0129, 1095-7138. doi: 10.1137/0330046.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 53728–53741. Curran Associates, Inc., 2023.

Shyam Sundhar Ramesh, Xiaotong Ji, Matthieu Zimmer, Sangwoong Yoon, Zhiyong Wang, Haitham Bou Ammar, Aurelien Lucchi, and Ilija Bogunovic. Multi-Task GRPO: Reliable LLM Reasoning Across Tasks, 2026. Version Number: 1.

Noveen Sachdeva, Benjamin Coleman, Wang-Cheng Kang, Jianmo Ni, Lichan Hong, Ed H. Chi, James Caverlee, Julian McAuley, and Derek Zhiyuan Cheng. How to train data-efficient LLMs. In *The Fourteenth International Conference on Learning Representations*, 2026.

Shiori Sagawa*, Pang Wei Koh*, Tatsunori B. Hashimoto, and Percy Liang. Distributionally Robust Neural Networks. In *International Conference on Learning Representations*, 2020.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms, 2017. Version Number: 2.

Pengyang Shao, Naixin Zhai, Lei Chen, Yonghui Yang, Fengbin Zhu, Xun Yang, and Meng Wang. BalDRO: A Distributionally Robust Optimization based Framework for Large Language Model Unlearning. In *Proceedings of the ACM Web Conference 2026*, pp. 8874–8884, Dubai United Arab Emirates, April 2026. ACM. ISBN 979-8-4007-2307-0. doi: 10.1145/3774904.3792975.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models, 2024. Version Number: 3.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. HybridFlow: A Flexible and Efficient RLHF Framework. *arXiv preprint arXiv: 2409.19256*, 2024.

Haebin Shin, Lei Ji, Xiao Liu, Zhiwei Yu, Qi Chen, and Yeyun Gong. DynamixSFT: Dynamic Mixture Optimization of Instruction Tuning Collections, 2025. Version Number: 1.

Tasuku Soma, Khashayar Gatmiry, Sharut Gupta, and Stefanie Jegelka. Near-Optimal Algorithms for Group Distributionally Robust Optimization and Beyond, January 2025. arXiv:2212.13669 [cs].

Yifan Sun, Jingyan Shen, Yibin Wang, Tianyu Chen, Zhendong Wang, Mingyuan Zhou, and Huan Zhang. Improving Data Efficiency for LLM Reinforcement Fine-tuning Through Difficulty-targeted Online Data Selection and Rollout Replay. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4149–4158, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1421.

Qwen Team. Qwen2.5: A Party of Foundation Models, September 2024. URL `https://qwenlm.github.io/blog/qwen2.5/`.

Qwen Team. Qwen3 Technical Report, 2025. URL `https://arxiv.org/abs/2505.09388`.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. TRL: Transformers Reinforcement Learning, 2020. URL `https://github.com/huggingface/trl`.

Weixuan Wang, Minghao Wu, Barry Haddow, and Alexandra Birch. HBO: Hierarchical Balancing Optimization for Fine-Tuning Large Language Models. In *The Fourteenth International Conference on Learning Representations*, 2026.

Yudong Wang, Zixuan Fu, Jie Cai, Peijun Tang, Hongya Lyu, Yewei Fang, Zhi Zheng, Jie Zhou, Guoyang Zeng, Chaojun Xiao, Xu Han, and Zhiyuan Liu. Ultra-FineWeb: Efficient Data Filtering and Verification for High-Quality LLM Training Data. *CoRR*, abs/2505.05427, May 2025a.

Zhenting Wang, Guofeng Cui, Kun Wan, and Wentian Zhao. DUMP: Automated Distribution-Level Curriculum Learning for RL-based LLM Post-training. *CoRR*, abs/2504.09710, April 2025b.

Zhilin Wang, Jiaqi Zeng, Olivier Delalleau, Daniel Egert, Ellie Evans, Hoo-Chang Shin, Felipe Soares, Yi Dong, and Oleksii Kuchaiev. HelpSteer3: Human-Annotated Feedback and Edit Data to Empower Inference-Time Scaling in Open-Ended General-Domain Tasks, 2025c. URL https://arxiv.org/abs/2503.04378.

Zhilin Wang, Jiaqi Zeng, Olivier Delalleau, Ellie Evans, Daniel Egert, Hoo-Chang Shin, Felipe Soares, Yi Dong, and Oleksii Kuchaiev. RLBFF: Binary Flexible Feedback to bridge between Human Feedback & Verifiable Rewards, 2025d. URL https://arxiv.org/abs/2509.21319.

Zhilin Wang, Jiaqi Zeng, Olivier Delalleau, Hoo-Chang Shin, Felipe Soares, Alexander Bukharin, Ellie Evans, Yi Dong, and Oleksii Kuchaiev. HelpSteer3-Preference: Open Human-Annotated Preference Data across Diverse Tasks and Languages, 2025e. URL https://arxiv.org/abs/2505.11475.

Runzhe Wu, Ankur Samanta, Ayush Jain, Scott Fujimoto, Jeongyeol Kwon, Ben Kretzu, Youliang Yu, Kaveh Hassani, Boris Vidolov, and Yonathan Efroni. Imbalanced Gradients in RL Post-Training of Multi-Task LLMs. In *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 3137–3150, Rabat, Morocco, 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.findings-eacl.164.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy S Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. DoReMi: Optimizing Data Mixtures Speeds Up Language Model Pretraining. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 69798–69818. Curran Associates, Inc., 2023.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. Qwen2 Technical Report. *arXiv preprint arXiv:2407.10671*, 2024.

Wenhao Yang, Sifan Yang, and Lijun Zhang. Discounted Online Convex Optimization: Uniform Regret Across a Continuous Interval. In *The Fourteenth International Conference on Learning Representations*, 2026.

Jiasheng Ye, Peiju Liu, Tianxiang Sun, Jun Zhan, Yunhua Zhou, and Xipeng Qiu. Data Mixing Laws: Optimizing Data Mixtures by Predicting Language Modeling Performance. In *The Thirteenth International Conference on Learning Representations*, 2025.

Chengbo Zang, Mehmet Kerem Turkcan, Gil Zussman, Zoran Kostic, and Javad Ghaderi. Adaptive Data Collection for Robust Learning Across Multiple Distributions. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 73974–73994. PMLR, July 2025.

Lijun Zhang, Peng Zhao, Zhen-Hua Zhuang, Tianbao Yang, and Zhi-Hua Zhou. Stochastic Approximation Approaches to Group Distributionally Robust Optimization. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 52490–52522. Curran Associates, Inc., 2023.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-Following Evaluation for Large Language Models, 2023. URL https://arxiv.org/abs/2311.07911.

APPENDIX

A RELATED WORK

Distributionally Robust Optimization. Distributionally Robust Optimization studies learning algorithms that optimize model performance under worst-case shifts or uncertainty in the data distribution Hu et al. (2018); Duchi & Namkoong (2021); Jin et al. (2021); Agarwal & Zhang (2022). Based on this formulation, GDRO explicitly minimizes the worst-case loss across predefined distribution groups Sagawa* et al. (2020); Haghtalab et al. (2022); Ghosal & Li (2023); Zhang et al. (2023). Recent theoretical work formulates GDRO as a stochastic convex-concave saddle-point problem and studies algorithms that combine data sampling with parameter optimization. For example, Zhang et al. (2023) use Exp3-style sampling Auer et al. (2002a) to track the *fixed* worst-performing data source *in hindsight*, and bound the minimax optimality gap for the time-averaged model parameters, but does not account for the evolving nature of the training process. Closest to our setting, Zang et al. (2025) propose to track the *adaptive* sequence of worst-performing data sources using UCB-style sampling Auer et al. (2002a). However, their approach requires repeatedly re-evaluating the current model on previously collected samples to maintain up-to-date source-wise loss estimates. This introduces substantial computational overhead and requires intrusive modifications to the standard training pipeline, limiting its practibility and compatibility with modern LLM training libraries.

Non-stationary Bandits. Multi-armed bandit algorithms provide a natural framework for adaptive data-source selection. Classical stochastic bandits assume fixed reward distributions Auer et al. (2002a), while adversarial bandits allow arbitrary reward sequences Auer et al. (2002b). Non-stationary bandits interpolate between these extremes by imposing structure on how rewards evolve over time Garivier & Moulines (2011); Besbes et al. (2014); Krishnamurthy & Gopalan (2025). Discounted-UCB methods were originally designed for environments with abrupt changes Garivier & Moulines (2011), while variation-budget and slowly-varying models provide alternative ways to quantify non-stationarity Besbes et al. (2014); Krishnamurthy & Gopalan (2025). Our setting differs from standard non-stationary bandits because the non-stationarity is endogenous: source-wise losses change because the optimizer updates the model parameters. Thus, the sampler and optimizer are coupled. We exploit this structure by combining discounted bandit estimation with online gradient descent and analyzing the resulting sampler-optimizer behavior through dynamic minimax regret.

Adaptive Sampling for LLM Training. Adaptive sampling has become increasingly important in LLM training and post-training because data sources vary widely in quality, difficulty, and relevance. Some methods estimate mixture weights offline using data-value models, filtering scores, or proxy evaluations Lin et al. (2024); Lv et al. (2025); Wang et al. (2025a); Sachdeva et al. (2026). Others update sampling probabilities online using training losses, gradient statistics, task difficulty, or model feedback Shin et al. (2025); Koh et al. (2026); Chen et al. (2025); Wang et al. (2025b); Sun et al. (2026); Panaganti et al. (2026). Recent work on multi-task post-training also studies how to balance heterogeneous reasoning, coding, safety, and instruction-following data so that easier or larger sources do not dominate the training Li et al. (2026); Ramesh et al. (2026); Wu et al. (2026); Panaganti et al. (2026). DRO-inspired objectives have also been explored for improving worst-case performance across groups or domains Oren et al. (2019); Corrado et al. (2025); Wang et al. (2026); Shao et al. (2026). However, existing approaches typically optimize static or heuristic objectives and lack a principled dynamic minimax regret formulation that explicitly accounts for the non-stationary nature of LLM training while providing provable theoretical guarantees.

B Implementation Details

Algorithm 1 Adaptive Sampling for Robust Multi-Source LLM Post-Training.

Require: Data sources $\{\mathcal{D}^k\}_{k=1}^K$, initial parameter θ_1, horizon T

1: Set $\beta \leftarrow 1 - K^{-1/2}T^{-1/2}$
2: Initialize $\hat{\mu}_t^k, N_t^k(\beta) \leftarrow 0$ for all $k \in [K]$
3: **for** $t = 1, \ldots, T$ **do**
4: If $t \leq K$, set $k_t \leftarrow t$; otherwise, generate sampling probabilities

$$p_t \leftarrow \textsc{Sampler}(\{\mathcal{D}^k, \hat{\mu}_{t-1}^k, N_{t-1}^k(\beta)\}_{k=1}^K)$$

5: Sample the mini-batch

$$\mathcal{B}_t \sim \left\{X \sim \mathcal{D}^{k_{t,i}} : k_{t,i} \sim \mathrm{Cat}(p_t), i = 1, 2, \ldots\right\}$$

6: Parameter update

$$\theta_{t+1} \leftarrow \textsc{Optimizer}(\theta_t, \mathcal{B}_t)$$

7: Update $\hat{\mu}_t^k$ and $N_t^k(\beta)$ by Eq. (7) for all $k \in [K]$
8: **end for**
9: **return** θ_{T+1}

The training process is outlined in Algorithm 1. We further adopt the following techniques during implementation.

Batch Sampling. Instead of selecting only one data source in each step, we propose a sampling version of DUCB, defined as

$$k_t \leftarrow \arg\max_{k \in [K]} \ \hat{\mu}_{t-1}^k + \phi_{t-1}^k, \quad \phi_{t-1}^k \sim \mathcal{N}\left(2r_{t-1}^k, \frac{C^2}{2N_{t-1}^k(\beta)}\right).$$

Due to the stochasticity of ϕ_{t-1}^k, the selection rule induces a *non-degenerate* categorical distribution over $[K]$ with probabilities p_t s.t. $k_t \sim \mathrm{Cat}(p_t)$, as opposed to the *deterministic* rule in Eq. (8). Therefore, every mini-batch may contain a mixture of samples from multiple data sources shown by Algorithm 1. This is especially important when using a large batch size, since training on a large amount of samples from a single data source increases the risk of catastrophic forgetting or mode collapse. We further prove in Appendix C.6 that the bandit regret of the sampling version of DUCB is only a constant factor worse than that of the standard DUCB in Proposition 4. Since p_t does not admit a closed-form expression, we approximate it using the Gauss-Hermite quadrature (Butler & Moffitt, 1982). At the end of each training step, we update $\hat{\mu}_t^k$ and $N_t^k(\beta)$ using the empirical loss and sample proportion in the mini-batch for each data source.

Double-EMA. Traditional EMA is known to have a delayed response to changing inputs due to the discounting mechanism. Thus we employ *double-EMA* Mulloy (1994) to track the training losses in the implementation of all samplers, defined as

$$\textsc{DoubleEma}(x) \leftarrow 2 \cdot \textsc{Ema}(x) - \textsc{Ema}(\textsc{Ema}(x)).$$

As shown in Fig. 1, the DEMA reward is able to closely track the trend of raw reward while providing sufficient denoising ability.

Enforced Exploration. In the early stage of training, the sampling weights of DUCB can be affected by noisy initial observations and experience fluctuations. Inspired by Exp3 (and its variants), we augment the sampling weights by

$$p_t^k \leftarrow (1 - \beta^t)p_t^k + \frac{\beta^t}{K}, \quad \forall k \in [K].$$

This enforces uniform exploration across all data sources in the early stage of training, and exponentially decays as the training progresses and becomes more stable. It is worth noting that the extra regret incurred by the enforced exploration is *at most* $O(H(\beta))$, which is dominated by the standard DUCB regret given the choice of β in Proposition 4.

C PROOFS

C.1 PROOF OF PROPOSITION 1

Proof of Proposition 1. Recall that $k_t^\star = \max_k \mu^k(\theta)$. By Definition 2,

$$\begin{aligned} R_o^1(\mathcal{A}_T) + R_b^1(\mathcal{A}_T) &= \sum_{t=1}^T \Big(\mu^{k_t}(\theta_t) - \mu^{k_t}(\theta^\star)\Big) + \sum_{t=1}^T \Big(\mu^{k_t^\star}(\theta_t) - \mu^{k_t}(\theta_t)\Big) \\ &= \sum_{t=1}^T \Big(\mu^{k_t^\star}(\theta_t) - \mu^{k_t}(\theta^\star)\Big) \\ &\geq \sum_{t=1}^T \mu^{k_t^\star}(\theta_t) - T \max_{k\in[K]} \mu^k(\theta^\star) \end{aligned},$$

where the last inequality follows from the fact that $\max_k \mu^k(\theta) \geq \mu^{k_t}(\theta)$ for all θ and t. Further recalling Definition 1 concludes the proof. □

C.2 PROOF OF PROPOSITION 3

The proof follows from standard results for undiscounted and discounted OGD. Suppose a player of an online game observes a sequence of losses $\{f_t(\theta_t), t = 1, \ldots, T\}$ for T steps, and updates the parameter using OGD defined in Eq. (6). Then the following lemmas hold. Note that we defer the choice of η in both cases.

Lemma 1 (Undiscounted OGD (Hazan, 2016))**.** *For any $\theta \in \mathbf{\Theta}$, OGD satisfies*

$$\sum_{t=1}^T f_t(\theta_t) - \sum_{t=1}^T f_t(\theta) \leq \frac{D^2}{\eta} + \eta G^2 T.$$

Lemma 2 (Discounted OGD (Yang et al., 2026))**.** *Let $\rho \in [0, 1)$ be a discount factor. For any $\theta \in \mathbf{\Theta}$, OGD satisfies*

$$\sum_{t=1}^T \rho^{T-t} f_t(\theta_t) - \sum_{t=1}^T \rho^{T-t} f_t(\theta) \leq \left(\frac{(1-\rho)D^2}{\eta} + \frac{\eta G^2}{2}\right) N_T(\rho).$$

Now we can prove the optimization regret.

Proof of Proposition 3. Let $f_t(\theta_t) = \hat{\mu}^{k_t}(\theta_t; \mathcal{B}_t)$. Recall that $X \sim \mathcal{D}^{k_t}$ *i.i.d.* for all $X \in \mathcal{B}_t$ according to Assumption 3, *i.e.*,

$$\mathbb{E}\big[\hat{\mu}^{k_t}(\theta_t; \mathcal{B}_t)\big| k_t, \theta_t\big] = \mu^{k_t}(\theta_t).$$

Since $\theta^\star \in \mathbf{\Theta}$ is fixed, we have, by Lemma 1,

$$\begin{aligned} \mathbb{E}\big[R_o^1(\mathcal{A}_T)\big] &= \sum_{t=1}^T \mathbb{E}\big[\mu^{k_t}(\theta_t) - \mu^{k_t}(\theta^\star)\big] \\ &= \sum_{t=1}^T \mathbb{E}\big[\mathbb{E}\big[\hat{\mu}^{k_t}(\theta_t; \mathcal{B}_t) - \hat{\mu}^{k_t}(\theta^\star; \mathcal{B}_t)\big| k_t, \theta_t\big]\big] \\ &= \mathbb{E}\left[\sum_{t=1}^T \hat{\mu}^{k_t}(\theta_t; \mathcal{B}_t) - \sum_{t=1}^T \hat{\mu}^{k_t}(\theta^\star; \mathcal{B}_t)\right] \\ &\leq \frac{D^2}{\eta} + \eta G^2 T \end{aligned},$$

and similarly by Lemma 2,

$$\mathbb{E}\big[R_o^\rho(\mathcal{A}_T)\big] \le \left(\frac{(1-\rho)D^2}{\eta} + \frac{\eta G^2}{2}\right) N_T(\rho), \quad \forall \rho \in [0,1).$$

Further note that $N_T(1) = T$. Combining together, we have

$$\frac{T}{N_T(\rho)}\mathbb{E}\big[R_o^\rho(\mathcal{A}_T)\big] \le O\big(\eta^{-1}(1-\rho)T\big) + O\big(\eta T\big) \le O\big(\eta^{-1}\big) + O\big(\eta T\big),$$

for any $\rho \in [1 - T^{-1}, 1]$, which concludes the proof. □

C.3 Proof of Proposition 4

We first present the following results, proved respectively in Sections C.4 and C.5. As opposed to stationary bandits, the explorations for non-stationary bandits conducted in distant steps gradually expire due to changing environments. Therefore, continuous re-exploration is needed to ensure reliable concentration results, *i.e.*, we need to keep $N_t^k(\beta)$ above some threshold $A > 0$.

Proposition 7 (Bandit Gap of DUCB)**.** *Suppose DUCB picks k_t in step $t \ge K+1$ as in Eq.* (8)*. Then for any $A > 0$,*

$$\Pr\left[\tilde{\mu}_{t-1}^{k_t^\star} - \tilde{\mu}_{t-1}^{k_t} > 2r_t^{k_t}\sqrt{1 + A^{-1}}, N_{t-1}^{k_t}(\beta) \ge A\right] \le 2H(\beta)^{-\alpha}.$$

Proposition 8 (EMA Bias)**.** *Suppose $\{\theta_t, t = 1, \dots, T\}$ is generated by OGD with learning rate η as in Eq.* (6)*. Then for any $t \in [T]$ and $A > 0$ s.t. $N_t^k(\beta) \ge A$,*

$$\left|\tilde{\mu}_t^k - \mu^k(\theta_{t+1})\right| \le \eta G^2 H(\beta) \ln\left(\frac{2Ce}{\eta A G^2}\right),$$

where $\beta \ge 1/2$ and $H(\beta)$ is the effective horizon as in Eq. (8)*.*

We further state following lemmas.

Lemma 3. *For any $\tau \in [T]$ and constant $A > 0$,*

$$\sum_{t=K+1}^{T} \mathbb{1}\left\{k_t = k, N_{t-1}^k(\beta) < A\right\} \cdot \gamma^{T-t} \le \lceil T/\tau \rceil N_{\beta^{-\tau}A}^k(\gamma).$$

Proof. We first write

$$\sum_{t=K+1}^{T} \mathbb{1}\left\{k_t = k, N_{t-1}^k(\beta) < A\right\} \cdot \gamma^{T-t} \le \sum_{m=1}^{\lceil T/\tau \rceil} \sum_{t=(m-1)\tau}^{m\tau} \mathbb{1}\left\{k_t = k, N_{t-1}^k(\beta) < A\right\} \cdot \gamma^{T-t}.$$

By Garivier & Moulines (2011) (Lemma 25 and Corollary 26),

$$\sum_{t=(m-1)\tau}^{m\tau} \mathbb{1}\left\{k_t = k, N_{t-1}^k(\beta) < A\right\} \le \beta^{-\tau} A.$$

Further note that

$$\begin{aligned}
&\sum_{t=(m-1)\tau}^{m\tau} \mathbb{1}\left\{k_t = k, N_{t-1}^k(\beta) < A\right\} \cdot \gamma^{T-t} \\
&\quad = \gamma^{T-m\tau} \sum_{t=(m-1)\tau}^{m\tau} \mathbb{1}\left\{k_t = k, N_{t-1}^k(\beta) < A\right\} \cdot \gamma^{m\tau - t} \\
&\quad \le \gamma^{T-m\tau} N_{\beta^{-\tau}A}^k(\gamma) \\
&\quad \le N_{\beta^{-\tau}A}^k(\gamma)
\end{aligned}.$$

Plugging back to the first equation concludes the proof. □

Lemma 4. *For all $T \geq K+1$,*

$$\sum_{t=K+1}^{T} \frac{\mathbb{1}\{k_t = k\} \cdot \gamma^{T-t}}{\sqrt{N_t^k(\beta)}} \leq 2N_T^k(\beta)^{\frac{1}{2}} + N_T^k(\gamma)H(\beta)^{-\frac{1}{2}}.$$

Proof. Recall that

$$\mathbb{1}\{k_t = k\} = N_t^k(\beta) - \beta N_{t-1}^k(\beta) = N_t^k(\beta) - N_{t-1}^k(\beta) + (1-\beta)N_{t-1}^k(\beta).$$

Thus we can write

$$\begin{aligned}\frac{\mathbb{1}\{k_t = k\}}{\sqrt{N_t^k(\beta)}} &= \frac{\mathbb{1}\{k_t = k\}^2}{\sqrt{N_t^k(\beta)}} \\ &= \mathbb{1}\{k_t = k\} \cdot \frac{N_t^k(\beta) - N_{t-1}^k(\beta)}{\sqrt{N_t^k(\beta)}} + \mathbb{1}\{k_t = k\} \cdot \frac{(1-\beta)N_{t-1}^k(\beta)}{\sqrt{N_t^k(\beta)}}.\end{aligned}$$

For the first term, note that

$$\frac{N_t^k(\beta) - N_{t-1}^k(\beta)}{\sqrt{N_t^k(\beta)}} \leq 2\Big(\sqrt{N_t^k(\beta)} - \sqrt{N_{t-1}^k(\beta)}\Big).$$

Since $\gamma \leq 1$ and $N_t^k(\beta) \geq N_{t-1}^k(\beta)$ whenever $k_t = k$, we have

$$\begin{aligned}\sum_{t=K+1}^{T} \mathbb{1}\{k_t = k\} \cdot \gamma^{T-t} \left(\frac{N_t^k(\beta) - N_{t-1}^k(\beta)}{\sqrt{N_t^k(\beta)}}\right) &\leq \sum_{t=K+1}^{T} \left(\frac{N_t^k(\beta) - N_{t-1}^k(\beta)}{\sqrt{N_t^k(\beta)}}\right) \\ &\leq 2\sum_{t=K+1}^{T} \Big(\sqrt{N_t^k(\beta)} - \sqrt{N_{t-1}^k(\beta)}\Big) \cdot \\ &\leq 2N_T^k(\beta)^{\frac{1}{2}}\end{aligned}$$

For the second term, note that $N_t^k(\beta) \leq H(\beta) = 1/(1-\beta)$ for all t. Thus

$$\begin{aligned}\sum_{t=K+1}^{T} \mathbb{1}\{k_t = k\} \cdot \gamma^{T-t} \left(\frac{(1-\beta)N_{t-1}^k(\beta)}{\sqrt{N_t^k(\beta)}}\right) &\leq (1-\beta)\sum_{t=K+1}^{T} \gamma^{T-t}\sqrt{N_{t-1}^k(\beta)} \\ &\leq (1-\beta)N_T^k(\beta)^{\frac{1}{2}} \sum_{t=K+1}^{T} \gamma^{T-t} \quad \cdot \\ &\leq N_T^k(\gamma)H(\beta)^{-\frac{1}{2}}\end{aligned}$$

Combining together gives the statement of the lemma. □

Now we can prove the dynamic bandit regret. In fact, we prove a more explicit version as follows.

Proposition 9 (Unnormalized Dynamic Bandit Regret of DUCB)**.** *For any $\beta \geq 1/2$, DUCB satisfies*

$$\begin{aligned}\mathbb{E}\big[R_b^\gamma(\mathrm{DUCB}_T)\big] \leq{}& O\big(KTH(\beta)^{-1}\big) + \tilde{O}\big(K^{\frac{1}{2}}H(\beta)^{\frac{1}{2}}\big) \\ &+ \tilde{O}\big(N_T(\gamma)H(\beta)^{-\frac{1}{2}}\big) + \tilde{O}\big(\eta N_T(\gamma)H(\beta)\big).\end{aligned}$$

Proof. Let $A \geq 1/3$ be a constant. We start by writing

$$\begin{aligned} R_b^\gamma(\mathrm{DUCB}_T) &= \sum_{t=1}^{T} \gamma^{T-t}\Big(\mu^{k_t^\star}(\theta_t) - \mu^{k_t}(\theta_t)\Big) \\ &\leq \gamma^{T-K} KC + C \sum_{t=K+1}^{T} \mathbb{1}\left\{N_{t-1}^{k_t}(\beta) < A\right\} \cdot \gamma^{T-t} \\ &\quad + \sum_{t=K+1}^{T} \mathbb{1}\left\{N_{t-1}^{k_t}(\beta) \geq A\right\} \cdot \gamma^{T-t}\Big(\mu^{k_t^\star}(\theta_t) - \mu^{k_t}(\theta_t)\Big) \end{aligned}.$$

Denote $N_T(\cdot) = \sum_{k=1}^K N_t^k(\cdot)$. By Lemma 3, the first sum writes

$$\begin{aligned} C \sum_{t=K+1}^{T} \mathbb{1}\left\{N_{t-1}^{k_t}(\beta) < A\right\} \cdot \gamma^{T-t} &\leq C \sum_{k=1}^{K} \sum_{t=K+1}^{T} \mathbb{1}\left\{k_t = k, N_{t-1}^{k}(\beta) < A\right\} \cdot \gamma^{T-t} \\ &\leq CK\lceil T/\tau \rceil N_{\beta^{-\tau}A}(\gamma) \end{aligned}.$$

Letting $\tau = N_T(\beta)$, we have

$$\beta^{-\tau} = \beta^{-N_T(\beta)} \leq \begin{cases} \beta^{-1/(1-\beta)} \leq 4 & \frac{1}{2} \leq \beta < 1 \\ 1 & \beta = 1 \end{cases}.$$

Thus $N_{\beta^{-\tau}A}(\gamma) \leq \beta^{-\tau} A \leq 4A$ for all $1/2 \leq \beta \leq 1$, and

$$C \sum_{t=K+1}^{T} \mathbb{1}\left\{N_{t-1}^{k_t}(\beta) < A(\beta)\right\} \cdot \gamma^{T-t} \leq 4ACK\big(TN_T(\beta)^{-1} + 1\big).$$

For the second sum, recall by Eq. (12) that

$$\begin{aligned} \sum_{t=1}^{T} \gamma^{T-t}\big(\mu^{k_t^\star}(\theta_t) - \mu^{k_t}(\theta_t)\big) &= \sum_{t=1}^{T} \gamma^{T-t}\big(\tilde{\mu}_{t-1}^{k_t^\star} - \tilde{\mu}_{t-1}^{k_t}\big) \\ &\quad + \sum_{t=1}^{T} \gamma^{T-t}\big(\mu^{k_t^\star}(\theta_t) - \tilde{\mu}_{t-1}^{k_t^\star}\big) - \sum_{t=1}^{T} \gamma^{T-t}\big(\mu^{k_t}(\theta_t) - \tilde{\mu}_{t-1}^{k_t}\big) \end{aligned}.$$

By Proposition 7, the first term on the RHS satisfies

$$\begin{aligned} \sum_{t=K+1}^{T} \mathbb{1}\left\{N_{t-1}^{k_t}(\beta) \geq A\right\} \cdot \gamma^{T-t}\Big(\tilde{\mu}_{t-1}^{k_t^\star} - \tilde{\mu}_{t-1}^{k_t}\Big) &\leq 2\sqrt{1 + A^{-1}} \sum_{t=K+1}^{T} \gamma^{T-t} r_t^{k_t} \\ &\leq 4 \sum_{t=K+1}^{T} \gamma^{T-t} r_t^{k_t} \end{aligned},$$

with probability *at least* $1 - 2H(\beta)^{-\alpha}$, where $A^{-1} \leq 3$ by construction. Thus it follows from a union bound that

$$\begin{aligned} &\Pr\left[\sum_{t=K+1}^{T} \mathbb{1}\left\{N_{t-1}^{k_t}(\beta) \geq A\right\} \cdot \gamma^{T-t}\Big(\tilde{\mu}_{t-1}^{k_t^\star} - \tilde{\mu}_{t-1}^{k_t}\Big) \geq 4 \sum_{t=K+1}^{T} \gamma^{T-t} r_t^{k_t}\right] \\ &\leq \sum_{t=K+1}^{T} \Pr\left[\tilde{\mu}_{t-1}^{k_t^\star} - \tilde{\mu}_{t-1}^{k_t} \geq 4r_t^{k_t}, N_{t-1}^{k_t}(\beta) \geq A\right] \\ &\leq 2TH(\beta)^{-\alpha} \end{aligned}.$$

By the Cauchy-Schwarz inequality, we have $\sum_{k=1}^{K}\sqrt{N_T^k(\beta)} \le \sqrt{KN_T(\beta)}$. Thus

$$\begin{aligned}
4\sum_{t=K+1}^{T}\gamma^{T-t}r_t^{k_t} &= 8C\sqrt{2\alpha\ln H(\beta)}\sum_{t=K+1}^{T}\frac{\gamma^{T-t}}{\sqrt{N_t^k(\beta)}} \\
&= 8C\sqrt{2\alpha\ln H(\beta)}\sum_{k=1}^{K}\sum_{t=K+1}^{T}\frac{\mathbb{1}\{k_t=k\}\cdot\gamma^{T-t}}{\sqrt{N_t^k(\beta)}} \\
&\le 8C\sqrt{2\alpha\ln H(\beta)}\sum_{k=1}^{K}\Big(2N_T^k(\beta)^{\frac{1}{2}} + N_T^k(\gamma)H(\beta)^{-\frac{1}{2}}\Big) \\
&\le 8C\sqrt{2\alpha\ln H(\beta)}\big(2K^{\frac{1}{2}}N_T(\beta)^{\frac{1}{2}} + N_T(\gamma)H(\beta)^{-\frac{1}{2}}\big)
\end{aligned},$$

where the first inequality follows from Lemma 4. Further combining with Proposition 8, we have

$$\begin{aligned}
&\sum_{t=K+1}^{T}\mathbb{1}\Big\{N_{t-1}^{k_t}(\beta)\ge A\Big\}\cdot\gamma^{T-t}\Big(\tilde{\mu}_{t-1}^{k_t^\star}-\tilde{\mu}_{t-1}^{k_t}\Big) \\
&\quad\le 8C\sqrt{2\alpha\ln H(\beta)}\big(2K^{\frac{1}{2}}N_T(\beta)^{\frac{1}{2}} + N_T(\gamma)H(\beta)^{-\frac{1}{2}}\big), \\
&\qquad + 2\eta G^2 N_T(\gamma)H(\beta)\ln\left(\frac{2Ce}{\eta AG^2}\right)
\end{aligned}$$

with probability *at least* $2TH(\beta)^{-\alpha}$. And when the bound fails, the additional regret incurred is *at most* $2CTN_T(\gamma)H(\beta)^{-\alpha}$. Putting together,

$$\begin{aligned}
\mathbb{E}\big[R_b^\gamma(\text{D-UCB}_T)\big] \le{}& \gamma^{T-K}CK + 4ACK\big(TN_T(\beta)^{-1}+1\big) \\
&+ 8C\sqrt{2\alpha\ln H(\beta)}\big(2K^{\frac{1}{2}}N_T(\beta)^{\frac{1}{2}} + N_T(\gamma)H(\beta)^{-\frac{1}{2}}\big) \\
&+ 2\eta G^2 N_T(\gamma)H(\beta)\ln\left(\frac{2Ce}{\eta AG^2}\right) \\
&+ 2CN_T(\gamma)H(\beta)^{1-\alpha}
\end{aligned}$$

Setting a sufficiently large α (*e.g.*, $\alpha = (\ln T + \ln H(\gamma))/\ln H(\beta)$ s.t. $TN_T(\gamma)H(\beta)^{1-\alpha} \le 1$) concludes the proof. □

Proposition 10 (Normalized Dynamic Bandit Regret of DUCB)**.** *The normalized dynamic bandit regret of DUCB satisfies*

$$\frac{T}{N_T(\gamma)}\mathbb{E}\big[R_b^\gamma(\text{DUCB}_T)\big] \le \begin{cases}\tilde{O}\big(K^{\frac{1}{2}}T^{\frac{1}{2}}\big) & \eta \le O(T^{-\frac{2}{3}}) \\ \tilde{O}\big(\eta^{\frac{1}{3}}K^{\frac{1}{3}}T\big) & O(T^{-\frac{3}{2}}) \le \eta \le o(1)\end{cases}$$

for any $\gamma \in [1-T^{-1}, 1]$.

Proof. First write

$$\begin{aligned}
\frac{T}{N_T(\gamma)}\mathbb{E}\big[R_b^\gamma(\text{DUCB}_T)\big] \le{}& O\big(KT^2N_T(\gamma)^{-1}H(\beta)^{-1}\big) + \tilde{O}\big(K^{\frac{1}{2}}TN_T(\gamma)^{-1}H(\beta)^{\frac{1}{2}}\big) \\
&+ \tilde{O}\big(TH(\beta)^{-\frac{1}{2}}\big) + \tilde{O}\big(\eta TH(\beta)\big)
\end{aligned}.$$

Note that when $\gamma \ge 1-T^{-1}$, we have

$$\frac{T}{N_T(\gamma)} \le \frac{1}{1-(1-T^{-1})^T} \le \frac{1}{1-e^{-1}} \le 2,$$

since $(1-T^{-1})^T \nearrow e^{-1}$ as $T \to \infty$. Then it follows that

$$\frac{T}{N_T(\gamma)}\mathbb{E}\big[R_b^\gamma(\mathrm{DUCB}_T)\big] \le O\big(KTH(\beta)^{-1}\big) + \tilde{O}\big(K^{\frac{1}{2}}H(\beta)^{\frac{1}{2}}\big) + \tilde{O}\big(TH(\beta)^{-\frac{1}{2}}\big) + \tilde{O}\big(\eta TH(\beta)\big).$$

Now we discuss the choice of β over different values of η. When $\eta \le o(T^{-3/2})$, *i.e.*, the environment is *almost stationary*, the last term is negligible since $O(\eta TH(\beta)) \le o(T^{1/2})$. Thus we can choose $\beta = 1 - T^{-1}$ s.t. $H(\beta) = T$, and

$$\frac{T}{N_T(\gamma)}\mathbb{E}\big[R_b^\gamma(\mathrm{DUCB}_T)\big] \le O\big(K\big) + \tilde{O}\big(K^{\frac{1}{2}}T^{\frac{1}{2}}\big) + \tilde{O}\big(T^{\frac{1}{2}}\big) + o\big(T^{\frac{1}{2}}\big) = \tilde{O}\big(K^{\frac{1}{2}}T^{\frac{1}{2}}\big).$$

When $O(T^{-3/2}) \le \eta \le o(1)$, *i.e.*, the environment is *slowly changing*, we have $\eta K \le 1$ for any fixed K independent of T. Let $\beta = 1 - \eta^{2/3}K^{-1/3}$ s.t. $H(\beta) = \eta^{-2/3}K^{1/3}$. We have

$$\frac{T}{N_T(\gamma)}\mathbb{E}\big[R_b^\gamma(\mathrm{DUCB}_T)\big] \le O\big(\eta^{\frac{2}{3}}K^{\frac{2}{3}}T\big) + \tilde{O}\big(\eta^{\frac{1}{3}}K^{\frac{1}{3}}\big) + \tilde{O}\big(\eta^{\frac{1}{3}}K^{-\frac{1}{6}}T\big) + \tilde{O}\big(\eta^{\frac{1}{3}}K^{\frac{1}{3}}T\big) = \tilde{O}\big(\eta^{\frac{1}{3}}K^{\frac{1}{3}}T\big).$$

Note that if $\eta = O(1)$, *i.e.*, the environment is *abruptly changing*, no algorithm can achieve a sublinear regret given our setting where θ_t changes in all steps Garivier & Moulines (2011). □

This proves Proposition 4.

C.4 Proof of Proposition 7

First we state a discounted version of Azuma's inequality. Consider the sequence

$$\left\{Z_t^k := \mathbb{1}\left\{k_t = k\right\} \cdot \left(\hat{\mu}^k(\theta_t; \mathcal{B}_t) - \mu^k(\theta_t)\right), t = 1, \ldots, T\right\}, \quad k \in [K].$$

Note that θ_t is $\sigma(\mathcal{A}_{t-1}, \mathcal{X}_{t-1})$-measurable by the design of OGD. Further since

$$\begin{aligned}\mathbb{E}\left[Z_t^k \middle| \mathcal{A}_{t-1}, \mathcal{X}_{t-1}\right] &= \mathbb{E}\left[\mathbb{1}\left\{k_t = k\right\} \cdot \left(\hat{\mu}^k(\theta_t; \mathcal{B}_t) - \mu^k(\theta_t)\right) \middle| \theta_t\right] \\ &= \Pr[k_t = k] \cdot \left(\mathbb{E}\left[\hat{\mu}^k(\theta_t; \mathcal{B}_t) \middle| k_t, \theta_t\right] - \mu^k(\theta_t)\right) + \Pr[k_t \ne k] \cdot 0 \\ &= 0\end{aligned},$$

the sequence $\{Z_t^k, t = 1, \ldots, T\}$ is a *martingale difference sequence* for all fixed $k \in [K]$. We have the following lemma.

Proposition 11 (Discounted Azuma's Inequality)**.** *For any constant $\alpha > 0$ and $\beta \in [0, 1]$,*

$$\Pr\left[\sum_{\tau=1}^{t} \beta^{t-\tau} Z_\tau^k > 2C\sqrt{(\alpha+1)N_t^k(\beta^2)\ln H(\beta)}\right] \le H(\beta)^{-\alpha},$$

where $N_t^k(\cdot)$ is defined in Eq. (7) *and $H(\cdot)$ is defined in Eq.* (8)*.*

Proof. The proof is an adaptation of the standard proof of Azuma's inequality, where we specifically deal with the randomness of $N_t^k(\beta^2)$. Following Garivier & Moulines (2011), we apply a union bound over its possible values using the *peeling technique*. Specifically, let

$$M(\beta, \delta) := \left\lceil \frac{\ln H(\beta)}{\ln(1+\delta)} \right\rceil, \quad \delta > 0.$$

Note that since $1-\beta \le \ln(1/\beta) \le 1-1/\beta$, we have

$$\frac{\beta}{1-\beta} \le H(\beta) \le \frac{1}{1-\beta}.$$

Define the intervals $I_m := (n_{m-1}, n_m]$ for each $m \in [M(\beta,\delta)]$, where we set $n_0 := 0$ and $n_m := (1+\delta)^m$. Then

$$n_M = (1+\delta)^{M(\beta,\delta)} \ge H(\beta) = \frac{1}{1-\beta} \ge \frac{1}{1-\beta^2} \ge N_t^k(\beta^2),$$

and there exists a unique $m \in [M(\beta,\delta)]$ s.t. $N_t^k(\beta^2) \in I_m$. Then for any $\epsilon > 0$,

$$\Pr\left[\sum_{\tau=1}^{t} \beta^{t-\tau} Z_\tau^k > \epsilon\right] = \sum_{m=1}^{M(\beta,\delta)} \Pr\left[\sum_{\tau=1}^{t} \beta^{t-\tau} Z_\tau^k > \epsilon, N_t^k(\beta^2) \in I_m\right].$$

For each event where $N_t^k(\beta) \in I_m$, there is $N_t^k(\beta^2) \le n_m$, and thus

$$\Pr\left[\sum_{\tau=1}^{t} \beta^{t-\tau} Z_\tau^k > \epsilon, N_t^k(\beta^2) \in I_m\right] \le \Pr\left[\sum_{\tau=1}^{t} \beta^{t-\tau} Z_\tau^k - \frac{\lambda C^2}{2} N_t^k(\beta^2) > \epsilon - \frac{\lambda C^2}{2} n_m\right].$$

Then applying Chernoff's bound to the RHS gives

$$\begin{aligned}
&\Pr\left[\sum_{\tau=1}^{t} \gamma^{t-\tau} Z_\tau^k > \epsilon, N_t^k(\beta^2) \in I_m\right] \\
&\le \inf_{\lambda>0} \exp\left(-\lambda\epsilon + \frac{\lambda^2 C^2}{2} n_m\right) \mathbb{E}\left[\exp\left(\lambda \sum_{\tau=1}^{t} \gamma^{t-\tau} Z_\tau^k - \frac{\lambda^2 C^2}{2} N_t^k(\beta^2)\right)\right].
\end{aligned}$$

Recall by Hoeffding's lemma that

$$\mathbb{E}\left[\exp(\lambda \beta^{t-\tau} Z_\tau^k) | \mathcal{A}_\tau, \mathcal{X}_{\tau-1}\right] \le \exp\left(\mathbb{1}\{k_\tau = k\} \cdot \frac{\lambda^2 \beta^{2(t-\tau)} C^2}{2}\right).$$

Then

$$\begin{aligned}
&\mathbb{E}\left[\exp\left(\lambda \sum_{\tau=1}^{t} \beta^{t-\tau} Z_\tau^k - \frac{\lambda^2 C^2}{2} N_t^k(\beta^2)\right)\right] \\
&= \mathbb{E}\left[\exp\left(\lambda \sum_{\tau=1}^{t-1} \beta^{t-\tau} Z_\tau^k - \frac{\lambda^2 C^2}{2} N_t^k(\beta^2)\right) \mathbb{E}\left[\exp(\lambda Z_t^k) | \mathcal{A}_t, \mathcal{X}_{t-1}\right]\right] \\
&\le \mathbb{E}\left[\exp\left(\lambda \sum_{\tau=1}^{t-1} \beta^{t-\tau} Z_\tau^k - \frac{\lambda^2 C^2}{2} N_t^k(\beta^2)\right) \cdot \exp\left(\mathbb{1}\{k_t = k\} \cdot \frac{\lambda^2 C^2}{2}\right)\right]. \\
&\le \cdots \\
&\le \mathbb{E}\left[\exp\left(-\frac{\lambda^2 C^2}{2} N_t^k(\beta^2) + \frac{\lambda^2 C^2}{2} \sum_{\tau=1}^{t} \mathbb{1}\{k_\tau = k\} \cdot \beta^{2(t-\tau)}\right)\right] \\
&= 1
\end{aligned}$$

Therefore,

$$\Pr\left[\sum_{\tau=1}^{t} \beta^{t-\tau} Z_\tau^k > \epsilon, N_t^k(\beta^2) \in I_m\right] \le \inf_{\lambda>0} \exp\left(-\lambda\epsilon + \frac{\lambda^2 C^2}{2} n_m\right) = \exp\left(-\frac{\epsilon^2}{2C^2 n_m}\right),$$

where the RHS of the inequality is minimized by taking $\lambda = \epsilon/(C^2 n_m)$. Since $n_m \le (1+\delta)N_t^k(\beta^2)$ for any $N_t^k(\beta^2) \in I_m$, we have

$$\begin{aligned}
&\Pr\left[\sum_{\tau=1}^{t} \beta^{t-\tau} Z_\tau^k > C\sqrt{2(1+\delta)N_t^k(\beta^2)\ln\frac{M(\beta,\delta)}{H(\beta)^{-\alpha}}}, N_t^k(\beta^2) \in I_m\right] \\
&\le \Pr\left[\sum_{\tau=1}^{t} \beta^{t-\tau} Z_\tau^k > C\sqrt{2n_m \ln\frac{M(\beta,\delta)}{H(\beta)^{-\alpha}}}, N_t^k(\beta^2) \in I_m\right] \\
&\le \frac{H(\beta)^{-\alpha}}{M(\beta,\delta)}
\end{aligned}.$$

Combining together, we have

$$\Pr\left[\sum_{\tau=1}^{t} \beta^{t-\tau} Z_\tau^k > C\sqrt{2(1+\delta)N_t^k(\beta^2)\ln\frac{M(\beta,\delta)}{H(\beta)^{-\alpha}}}\right] \le H(\beta)^{-\alpha}.$$

Finally, we choose $\delta > 0$ s.t. $M(\beta,\delta) = H(\beta)$, *i.e.*, as $\beta \to 1$,

$$1 + \delta = H(\beta)^{1/H(\beta)} \nearrow e^{\frac{1}{e}} < 2.$$

This concludes the proof. □

Then we immediately have the following result.

Lemma 5. *For any $\alpha \ge 1$,*

$$\Pr\left[\hat{\mu}_t^k - \tilde{\mu}_t^k > 2C\sqrt{\frac{2\alpha \ln H(\beta)}{N_t^k(\beta)}}\right] \le H(\beta)^{-\alpha}.$$

Proof. Note that

$$\hat{\mu}_t^k - \tilde{\mu}_t^k = \frac{1}{N_t^k(\gamma)} \sum_{\tau=1}^{t} \gamma^{t-\tau} Z_\tau^k.$$

Since $N_t^k(\gamma) \ge N_t^k(\gamma^2)$ for all $\gamma \in (0,1)$, we have

$$\frac{C}{N_t^k(\gamma)}\sqrt{(\alpha+1)N_t^k(\gamma^2)\ln H(\gamma)} \le C\sqrt{\frac{(\alpha+1)\ln H(\gamma)}{N_t^k(\gamma)}} \le C\sqrt{\frac{2\alpha \ln H(\gamma)}{N_t^k(\gamma)}},$$

for all $\alpha \ge 1$. Then

$$\begin{aligned}
\Pr\left[\hat{\mu}_t^k - \tilde{\mu}_t^k > 2C\sqrt{\frac{2\alpha \ln H(\gamma)}{N_t^k(\gamma)}}\right] &\le \Pr\left[\sum_{\tau=1}^{t} \gamma^{t-\tau} Z_\tau^k > 2C\sqrt{(\alpha+1)N_t^k(\gamma)\ln H(\gamma)}\right] \\
&\le H(\gamma)^{-\alpha}
\end{aligned},$$

where the last inequality follows from Proposition 11. □

Now we can prove the DUCB bandit gap.

Proof of Proposition 7. By the rule of DUCB,

$$\hat{\mu}_{t-1}^k + r_{t-1}^k \le \hat{\mu}_{t-1}^{k_t} + r_{t-1}^{k_t}, \quad \forall k \in [K].$$

Specifically for $k = k_t^\star$, there is

$$\hat{\Delta}_{t-1}^{k_t} := \hat{\mu}_{t-1}^{k_t^\star} - \hat{\mu}_{t-1}^{k_t} \le r_{t-1}^{k_t} - r_{t-1}^{k_t^\star}.$$

When $N_{t-1}^k(\beta) \geq A$, we have

$$\frac{1+A^{-1}}{N_t(\beta)} = \frac{N_{t-1}^k(\beta) + A^{-1}N_{t-1}^k(\beta)}{N_{t-1}^k(\beta)N_t(\beta)} \geq \frac{N_{t-1}^k(\beta)+1}{N_{t-1}^k(\beta)N_t(\beta)} \geq \frac{1}{N_{t-1}^k(\beta)},$$

where the last inequality is due to $N_{t-1}^k(\beta) + 1 \geq N_t^k(\beta)$. Thus

$$r_t^k\sqrt{1+A^{-1}} = 2C\sqrt{\frac{1+A^{-1}}{N_t^k(\beta)} \cdot 2\alpha \ln H(\beta)} \geq 2C\sqrt{\frac{2\alpha \ln H(\beta)}{N_{t-1}^k(\beta)}} = r_{t-1}^k.$$

Then by the standard UCB argument,

$$\begin{aligned}
&\Pr\left[\tilde{\mu}_{t-1}^{k_t^\star} - \tilde{\mu}_{t-1}^{k_t} > 2r_t^{k_t}\sqrt{1+A^{-1}}, N_{t-1}^{k_t}(\beta) \geq A\right] \\
&\leq \Pr\left[\tilde{\mu}_{t-1}^{k_t^\star} - \tilde{\mu}_{t-1}^{k_t} > 2r_{t-1}^{k_t}\right] \\
&= \Pr\left[\tilde{\mu}_{t-1}^{k_t^\star} - \tilde{\mu}_{t-1}^{k_t} - \hat{\Delta}_{t-1}^{k_t} > 2r_{t-1}^{k_t} - \hat{\Delta}_{t-1}^{k_t}\right] \\
&\leq \Pr\left[\left(\tilde{\mu}_{t-1}^{k_t^\star} - \hat{\mu}_{t-1}^{k_t^\star}\right) - \left(\tilde{\mu}_{t-1}^{k_t} - \hat{\mu}_{t-1}^{k_t}\right) > r_{t-1}^{k_t^\star} + r_{t-1}^{k_t}\right] \\
&\leq \Pr\left[\hat{\mu}_{t-1}^{k_t^\star} - \tilde{\mu}_{t-1}^{k_t^\star} < -r_{t-1}^{k_t^\star}\right] + \Pr\left[\hat{\mu}_{t-1}^{k_t} - \tilde{\mu}_{t-1}^{k_t} > r_{t-1}^{k_t}\right]
\end{aligned}.$$

Applying Lemma 5 respectively on each term concludes the proof. □

C.5 Proof of Proposition 8

We prove this by splitting the EMA summands by a horizon $D(\beta)$ at each step t. Note that the early summands for $\tau = 1, \ldots, \lfloor D(\beta) \rfloor$ inherits more bias, since they are calculated based on some model parameter drastically different from the current parameter θ_t. Thus we simply apply the boundedness condition. On the other hand, more recent summands for $\tau = \lceil D(\beta) \rceil, \ldots, t$ are calculated based on the model parameters much closer to the current θ_t. Thus we bound them using the Lipschitz condition in Assumption 2. Finally, we optimize the choice of $D(\beta)$ to obtain Proposition 8.

Proof of Proposition 8. Suppose there exists some $0 \leq D(\beta) \leq t$ for $t \in [T]$. We can write

$$\begin{aligned}
\left|\tilde{\mu}_t^k - \mu^k(\theta_{t+1})\right| &= \frac{1}{N_t^k(\beta)}\left|\sum_{\tau=1}^{t} \mathbb{1}\{k_\tau = k\} \cdot \beta^{t-\tau}\left(\mu^k(\theta_\tau) - \mu^k(\theta_{t+1})\right)\right| \\
&\leq \frac{1}{N_t^k(\beta)}\left|\sum_{\tau=1}^{t-\lfloor D(\beta)\rfloor} \mathbb{1}\{k_\tau = k\} \cdot \beta^{t-\tau}\left(\mu^k(\theta_\tau) - \mu^k(\theta_{t+1})\right)\right| \\
&\quad + \frac{1}{N_t^k(\beta)}\left|\sum_{\tau=t-\lceil D(\beta)\rceil}^{t} \mathbb{1}\{k_\tau = k\} \cdot \beta^{t-\tau}\left(\mu^k(\theta_\tau) - \mu^k(\theta_{t+1})\right)\right|
\end{aligned}.$$

For the first sum, since $|\mu^k(\theta_\tau) - \mu^k(\theta_t)| < C$ for all $\tau, t \in [T]$, we have

$$\begin{aligned}
&\frac{1}{N_t^k(\beta)}\left|\sum_{\tau=1}^{t-\lfloor D(\beta)\rfloor} \mathbb{1}\{k_t = k\} \cdot \beta^{t-\tau}\left(\mu^k(\theta_\tau) - \mu^k(\theta_{t+1})\right)\right| \\
&\leq \frac{C\beta^{\lfloor D(\beta)\rfloor}}{N_t^k(\beta)} \sum_{\tau=1}^{t-\lfloor D(\beta)\rfloor} \mathbb{1}\{k_\tau = k\} \cdot \beta^{t-\lfloor D(\beta)\rfloor-\tau} \\
&= \frac{C\beta^{D(\beta)}N_{t-\lfloor D(\beta)\rfloor}^k(\beta)}{N_t^k(\beta)} \\
&\leq CA^{-1}H(\beta)\beta^{D(\beta)}
\end{aligned}.$$

For the second sum, by the Lipschitz condition and the rule of OGD, we have

$$\max_{\tau\in[t-\lceil D(\beta)\rceil,t]} \left|\mu^k(\theta_\tau)-\mu^k(\theta_{t+1})\right| \le G\left\|\sum_{\tau=t-\lceil D(\beta)\rceil}^{t}\eta\nabla\hat{\mu}^{k_t}(\theta_\tau;\cdot)\right\|. \le \eta G^2 D(\beta)$$

Then

$$\left|\tilde{\mu}_t^k-\mu^k(\theta_{t+1})\right| \le CA^{-1}H(\beta)\beta^{D(\beta)}+\eta G^2 D(\beta).$$

To choose the optimal $D(\beta)$, let

$$\frac{d}{dD(\beta)}\left|\tilde{\mu}_t^k-\mu^k(\theta_{t+1})\right| = CA^{-1}H(\beta)(\ln\beta)\beta^{D(\beta)}+\eta G^2=0,$$

i.e.,

$$D^\star(\beta)=\frac{1}{\ln\beta}\ln\left(\frac{\eta AG^2}{CH(\beta)\ln(1/\beta)}\right).$$

Further note that the LHS is *convex* in $D(\beta)$. Thus the assumption that $0\le D(\beta)\le t$ can be alleviated, and

$$\left|\tilde{\mu}_t^k-\mu^k(\theta_{t+1})\right| \le \frac{\eta G^2}{\ln(1/\beta)}\left(1+\ln\left(\frac{CH(\beta)\ln(1/\beta)}{\eta AG^2}\right)\right).$$

Finally, since $\ln(1/\beta)\ge 1-\beta = H(\beta)^{-1}$, we conclude the proof by realizing that the RHS above is monotonically decreasing w.r.t. $\ln(1/\beta)$, and $H(\beta)\ln(1/\beta)\le 1/\beta\le 2$ for all $\beta\ge 1/2$. □

C.6 Analysis of Sampling DUCB

To show that the bandit regret of the sampling version of DUCB is a constant factor away from the standard DUCB, we only need to show that the bandit gap is a constant factor away from Proposition 7, and the rest naturally follows.

Proposition 12 (Bandit Gap of Sampling DUCB)**.** *Suppose sampling DUCB picks k_t in step $t\ge K+1$ as in Section B. Then for any $A>0$,*

$$\Pr\left[\tilde{\mu}_{t-1}^{k_t^\star}-\tilde{\mu}_{t-1}^{k_t} > 4r_t^{k_t}\sqrt{1+A^{-1}}, N_{t-1}^{k_t}(\beta)\ge A\right] \le 4H(\beta)^{-\alpha}.$$

Proof. By the rule of sampling DUCB, we have

$$\hat{\mu}_{t-1}^k+\phi_{t-1}^k \le \hat{\mu}_{t-1}^{k_t}+\phi_{t-1}^{k_t},\quad \forall k\in[K].$$

And specifically for $k=k_t^\star$ (recall that $k_t^\star=\arg\max_k\mu^k(\theta_t)$),

$$\hat{\Delta}_t^{k_t}=\hat{\mu}_{t-1}^{k_t^\star}-\hat{\mu}_{t-1}^{k_t}\le\phi_{t-1}^{k_t}-\phi_{t-1}^{k_t^\star}.$$

According to the Gaussian tail bound,

$$\Pr\left[\phi_{t-1}^k-2r_{t-1}^k>r_{t-1}^k\right]\le H(\beta)^{-\alpha},\quad \Pr\left[\phi_{t-1}^k-2r_{t-1}^k<-r_{t-1}^k\right]\le H(\beta)^{-\alpha}.$$

Recall by Proposition 7 that $(1+A^{-1})N_t^k(\beta)^{-1}\ge N_{t-1}^k(\beta)^{-1}$. Then

$$\begin{aligned}
&\Pr\left[\tilde{\mu}_{t-1}^{k_t^\star}-\tilde{\mu}_{t-1}^{k_t} > 4r_t^{k_t}\sqrt{1+A^{-1}}, N_{t-1}^{k_t}(\beta)\ge A\right]\\
&\le \Pr\left[\tilde{\mu}_{t-1}^{k_t^\star}-\tilde{\mu}_{t-1}^{k_t} > 4r_{t-1}^{k_t}\right]\\
&= \Pr\left[\tilde{\mu}_{t-1}^{k_t^\star}-\tilde{\mu}_{t-1}^{k_t}-\hat{\Delta}_t^{k_t}+\hat{\Delta}_t^{k_t} > 4r_{t-1}^{k_t}\right]\\
&\le \Pr\left[(\tilde{\mu}_{t-1}^{k_t^\star}-\hat{\mu}_{t-1}^{k_t^\star})-(\tilde{\mu}_{t-1}^{k_t}-\hat{\mu}_{t-1}^{k_t})+\phi_{t-1}^{k_t}-\phi_{t-1}^{k_t^\star} > 4r_{t-1}^{k_t}+r_{t-1}^{k_t^\star}-r_{t-1}^{k_t^\star}\right],\\
&\le \Pr\left[\hat{\mu}_{t-1}^{k_t^\star}-\tilde{\mu}_{t-1}^{k_t^\star}<-r_{t-1}^{k_t^\star}\right]+\Pr\left[\hat{\mu}_{t-1}^{k_t}-\tilde{\mu}_{t-1}^{k_t}(\theta_t)>r_{t-1}^{k_t}\right]\\
&\quad+\Pr\left[\phi_{t-1}^{k_t}>3r_{t-1}^{k_t}\right]+\Pr\left[\phi_{t-1}^{k_t^\star}<r_{t-1}^{k_t^\star}\right]
\end{aligned}$$

where the last inequality follows from a union bound. Applying Lemma 5 and the Gaussian tail bound respectively on each term concludes the proof. □

D Additional Experimental Results

This section provides additional results beyond the compact Tab. 2. The goal is two-fold. First, we evaluate whether the behavior of DUCB transfers across model families and parameter scales. Second, we report auxiliary metrics, including mean token accuracy for SFT and reward margin for DPO, to verify that improvements in the primary robustness metric do not arise from degenerate behavior on secondary measures.

All models are trained either on a single 80G or two 40G NVIDIA A100 GPU(s), using the AdamW optimizer with 8-bit LoRA quantization and bfloat-16 mixed precision. For each set of experiments, we set a uniform *sample budget* and explicitly avoid training for multiple epochs. The goal is to ensure that every training mini-batch is non-repetitive and aligns with Assumption 3. Detailed configurations and scripts will be released with the code upon publication.

D.1 SFT

Table 3: SFT results on the GSM8k + IFEval datasets with a budget of 32k samples.

Model	Sampler	Signal	Accuracy		Mean Token Accuracy	
			Min	**Avg**	**Min**	**Avg**
Qwen2.5-3B (Base)	Uniform	Loss	22.85	**29.79**	81.23	84.97
	Exp3	Loss	23.63	25.68	78.82	82.35
	Exp3	Loss Difference[1]	23.63	27.24	81.46	84.78
	DUCB (Ours)	Loss	**24.02**	26.95	81.35	84.71
Qwen2.5-7B (Base)	Uniform	Loss	27.15	39.36	78.79	82.34
	Exp3	Loss	27.54	37.60	81.42	84.76
	Exp3	Loss Difference[1]	27.73	**40.72**	78.87	82.35
	DUCB (Ours)	Loss	**28.52**	37.89	78.82	82.35
Llama3.1-8B (Base)	Uniform	Loss	22.07	25.98	70.31	73.66
	Exp3	Loss	23.24	26.27	71.27	73.63
	Exp3	Loss Difference[1]	21.48	**27.93**	71.08	73.57
	DUCB (Ours)	Loss	**24.22**	25.29	71.22	73.61

[1]Follows from Shin et al. (2025).

We evaluate the SFT performance on a heterogeneous dataset comprising GSM8k Cobbe et al. (2021) and IFEval Zhou et al. (2023), evaluating the model's ability to balance mathematical reasoning with instruction-following. We compare against Exp3 utilizing standard SFT loss and the DynamixSFT-style *stepwise loss difference* proposed by Shin et al. (2025). We approximate their forward *anticipated* loss probing using virtual parameter update (followed parameter reset) with backward *observed* loss subtraction.

The observations confirm that DUCB consistently fulfills its theoretical promise of robustness, achieving the highest minimum source-wise across all model architectures and sizes. Note that the DynamixSFT-style proxy tends to achieve higher average accuracy, which aligns with their goal of optimizing the *average* performance.

D.2 DPO

We evaluate the DPO performance on the HelpSteer3 Wang et al. (2025c;e) dataset in Tab. 4. We compare against compare against faithful implementations of AutoMixAlign Corrado et al. (2025) including both AMA-R and AMA-S, and Exp3-style samplers based on excess loss. Specifically, we train the model using uniform sampling for 100 steps, and utilize the (smoothed) training loss as a proxy to the loss of a *specialist model* for each data source. Then we use the difference between the instantaneous training losses and the specialist loss proxies as the Exp3 sampling signal. Additionally,

Table 4: DPO results on the Helpsteer3 dataset with a budget of 16k samples.

Model	Sampler	Signal	Accuracy		Margin	
			Min	**Avg**	**Min**	**Avg**
Qwen2.5-3B (Instruct)	Fixed-Weight	-	*75.00*	*80.57*	0.96	1.28
	Uniform	Loss	76.76	80.53	1.08	1.31
	Exp3	Excess Loss	77.78	79.98	1.12	1.40
	AMA-R[1]	Excess Loss	78.19	79.45	1.03	1.30
	AMA-S[1]	Excess Loss	**80.25**	**82.35**	1.07	1.31
	DUCB (Ours)	Loss	**79.33**	81.59	1.10	1.33
Llama3.2-3B (Instruct)	Fixed-Weight	-	*80.24*	*81.75*	1.24	1.33
	Uniform	Loss	73.79	81.06	0.98	1.29
	Exp3	Excess Loss	71.37	79.94	0.93	1.32
	AMA-R[1]	Excess Loss	74.22	78.00	0.77	1.15
	AMA-S[1]	Excess Loss	**77.37**	80.48	0.90	1.20
	DUCB (Ours)	Loss	**77.42**	**81.88**	1.04	1.30

[1]Faithful implementations of Corrado et al. (2025).

we also compare against a *fixed weight* sampler, which assigns a deterministic weight based on the specialist loss proxies for each model, respectively.

The results reveal that DUCB outperforms AMA-R on both models and remains competitive with AMA-S, which achieves higher minimum accuracy on Qwen2.5-3B. It is also worth noting that the simple fixed-weight sampler performs exceptionally well for Llama3.2-3B, while failing to beat the uniform sampling baseline for Qwen2.5-3B. This suggests that a relatively stationary data mixture method is not uniformly applicable for all model families without case-specific tunings. On the other hand, DUCB demonstrates its capacity to automatically detect and adapt to these varying levels of non-stationarity.

Table 5: Three-seed DPO results on HelpSteer3 with Qwen2.5-3B-Instruct. We report final-checkpoint mean accuracy $\pm$ standard deviation. The paired gain compares DUCB and Uniform under matched seeds.

Sampler	Runs	Min Acc.	Avg Acc.
Uniform	3	76.76$\pm$1.65	80.53$\pm$0.70
DUCB (Ours)	3	**79.33$\pm$0.62**	**81.59$\pm$0.40**
Paired gain	3	**+2.57$\pm$1.04**	-

We further evaluate the performance of Qwen2.5-3B-Instruct on the Helpsteer3 dataset over 3 random seeds. Under the original sample budget, hyperparameters, and evaluation protocol, DUCB outperforms Uniform in minimum accuracy in all three matched Qwen2.5-3B DPO runs. As shown in Tab. 5, DUCB improves the final-checkpoint minimum accuracy by 2.57 ± 1.04 percentage points while exhibiting lower variation across seeds. This provides controlled seed-level evidence of improved worst-source robustness without relying on aggregation across different datasets or model families.

D.3 GRPO

We evaluate GRPO under a broader set of model families, scales, and sampling signals inpired by different baselines. Specifically, Ramesh et al. (2026) utilize Exp3 guided by training reward with stepwise reward improvement correction; Wang et al. (2025b) adopt switching-window UCB (SWUCB) guided by training advantages. We focus on the Reasoning Chollet (2019); Katz et al. (2025); Lin et al. (2025) dataset and the CommonsenseQA Talmor et al. (2019) dataset.

Table 6: GRPO results on the Reasoning dataset with a budget of 10k samples.

Model	Sampler	Signal	Accuracy	
			Min	**Avg**
Qwen2.5-0.5B (Instruct)	Uniform	Reward	14.73	23.01
	Exp3	Reward	16.01	22.82
	Exp3	Reward + Difference Ramesh et al. (2026)	16.30	**23.23**
	SWUCB (DUMP)	Advantage Wang et al. (2025b)	13.83	22.61
	DUCB (Ours)	Reward	**16.53**	22.88
Qwen2.5-1.5B (Instruct)	Uniform	Reward	18.89	28.98
	Exp3	Reward	21.70	**29.75**
	Exp3	Reward + Difference Ramesh et al. (2026)	20.91	28.78
	SWUCB (DUMP)	Advantage Wang et al. (2025b)	19.45	**29.70**
	DUCB (Ours)	Reward	**21.81**	28.74
Llama3.2-1B (Instruct)	Uniform	Reward	10.79	21.87
	Exp3	Reward	11.13	22.70
	Exp3	Reward + Difference Ramesh et al. (2026)	11.01	18.19
	SWUCB (DUMP)	Advantage Wang et al. (2025b)	11.58	23.88
	DUCB (Ours)	Reward	**15.40**	**25.07**

Table 7: GRPO results on the CommonsenseQA dataset with a budget of 6.4k samples.

Model	Sampler	Signal	Accuracy	
			Min	**Avg**
Qwen3-8B (Instruct)	Uniform	Reward	**81.40**	83.50
	Exp3	Reward	77.22	85.52
	Exp3	Reward + Difference Ramesh et al. (2026)	79.75	82.81
	SWUCB (DUMP)	Advantage Wang et al. (2025b)	73.26	76.17
	DUCB (Ours)	Reward	**81.40**	**87.20**
Qwen2.5-3B (Instruct)	Uniform	Reward	65.38	72.27
	Exp3	Reward	67.44	**75.78**
	Exp3	Reward + Difference Ramesh et al. (2026)	61.54	66.41
	SWUCB (DUMP)	Advantage Wang et al. (2025b)	65.38	71.88
	DUCB (Ours)	Reward	**69.77**	75.39
Llama3.2-3B (Instruct)	Uniform	Reward	69.23	73.44
	Exp3	Reward	72.09	74.22
	Exp3	Reward + Difference Ramesh et al. (2026)	69.23	72.66
	SWUCB (DUMP)	Advantage Wang et al. (2025b)	69.23	71.88
	DUCB (Ours)	Reward	**74.41**	**75.78**
GLM4-9B (Instruct)	Uniform	Reward	77.91	84.08
	Exp3	Reward	79.07	84.36
	Exp3	Reward + Difference Ramesh et al. (2026)	82.56	85.55
	SWUCB (DUMP)	Advantage Wang et al. (2025b)	81.40	85.55
	DUCB (Ours)	Reward	**84.88**	**87.74**

For Reasoning shown in Tab. 6, DUCB consistently provides the strongest robustness guarantees, achieving the highest minimum source-wise accuracies across all tested models. For CommonsenseQA shown in Tab. 7, DUCB achieves the best or tied-best minimum source-wise accuracy across all four models. This suggests that the robustness of DUCB is not tied to a specific architecture, tokenizer, or pretraining recipe.

The comparison with Exp3-style Ramesh et al. (2026) and DUMP Wang et al. (2025b) baselines further highlights the importance of using a sampling signal aligned with worst-source robustness.

Specifically, samplers that are based on reward difference or advantage does not consistently improve worst-source accuracy, affirming the findings of Ramesh et al. (2026). We also observe that the gains are comparably clearer on the smaller models, indicating that they may be more vulnerable to source-wise imbalance during GRPO. On the other hand, larger models may already possess stronger prior knowledge and thus exhibit more robustness.

D.4 Training Overhead

Table 8: Training overhead on HelpSteer3 with Qwen2.5-3B-Instruct using the same A100 GPU, sample budget, and hyperparameters. Per-step time is reported as mean ± standard deviation. AMA runtimes cover generalist training but exclude the cost of training K specialists. The overhead for Zang et al. (2025) is analytical rather than a wall-clock measurement.

Sampler	Time/Step (s)	Additional Cost
Uniform	21.90 ± 1.29	None
AMA-R†	21.47 ± 1.24	Training K source-specific specialists
AMA-S†	21.72 ± 1.36	Training K source-specific specialists
Zang et al.‡	$21 + 0.36 \times \mathrm{BufSize}_t$	$O(t)$ replay evaluations at step t
DUCB (Ours)	21.94 ± 1.14	$O(K)$ scalar state

† The reported runtime excludes specialist training;

‡ Complexity under a history that grows linearly with the training step.

We measure per-step runtime on the same A100 GPU using the three Qwen2.5-3B DPO runs. DUCB requires 21.94 ± 1.14 seconds per step, compared with 21.90 ± 1.29 for Uniform, with the difference falling within the observed variations. DUCB uses losses already computed during optimization, adds no auxiliary model evaluations, and maintains only $O(K)$ scalar state. The generalist-training runtimes of AMA-R and AMA-S are similarly close to Uniform, but exclude their additional cost of training K source-specific specialists. By contrast, the oracle of Zang et al. (2025) re-evaluates a growing history of collected samples under the current model. Thus, DUCB introduces little measurable online overhead while avoiding both auxiliary specialist training and replay-based model evaluations.

D.5 Optimization Dynamics and Assumption Diagnostics

Lipschitz regularity with respect to model parameters is commonly adopted in theoretical analyses of LLM optimization (Malladi et al., 2023; Bai et al., 2024; Jiang et al., 2025). In our analysis, its primary role is to control changes in source losses between successive iterates visited by the optimization trajectory; we do not characterize a global Lipschitz constant for a transformer over its entire parameter space. We interpret Assumption 2 accordingly as a trajectory-level regularity condition.

Table 9: Optimization dynamics across the SFT, DPO, and GRPO experiments. For gradient norms and per-source signal variation, we report the mean and maximum of the run-level maxima. Worst-source variation is measured using exponentially smoothed training loss for SFT/DPO and reward for GRPO. Estimated worst-source (WS) switches are normalized per 1,000 steps.

Task	Max LR	Max Grad. Norm mean / max	Max Signal Variation mean / max	WS Switches / 1k mean [min, max]
SFT	2e-5	190.24 / 298.79	0.07 / 0.13	41 [3,124]
DPO	1e-5	3.94 / 5.14	0.15 / 0.17	31 [13,53]
GRPO	1e-5	0.22 / 0.24	0.03 / 0.06	20 [0,32]

In practice, small learning rates, gradient clipping, learning-rate schedules, and, where applicable, KL regularization help control parameter updates during LLM post-training. To examine whether the slowly varying loss environment assumed by our analysis is representative of our experiments, we record the learning rate and gradient norm reported by TRL or veRL, together with the maximum

absolute change in each source's exponentially smoothed training signal between consecutive steps. We also report the frequency with which the empirically worst source changes during training.

As shown in Tab. 9, the experiments use maximum learning rates of at most 2×10^{-5}, and the observed source-wise training signals change modestly between consecutive steps. At the same time, the identity of the worst source changes repeatedly across all three objectives, indicating a slowly varying but nonstationary environment. These observations are consistent with the trajectory-level regularity and slow-variation assumptions used in our analysis. They do not prove global Lipschitzness or estimate a global Lipschitz constant for the underlying neural networks.