Lipschitz Dueling Bandits over Continuous Action Spaces

Mudit Sharma
Indian Institute of Technology Ropar

Shweta Jain
Indian Institute of Technology Ropar

Vaneet Aggarwal
Purdue University

Ganesh Ghalme
Indian Institute of Technology Hyderabad

Abstract

We initiate the study of stochastic dueling bandits over continuous action spaces with Lipschitz structure, where the learner observes only noisy pairwise comparisons. Unlike standard Lipschitz bandits, comparative feedback makes estimates across regions over a continuous action space inherently non-comparable, so classical elimination-based methods do not extend directly. Our key idea is to use a round-wise fixed reference arm, which converts pairwise comparisons into a common scalar score and enables safe recursive region elimination. Based on this idea, we propose the first algorithm for this setting, Anchored Recursive Comparisons for dueling Lipschitz bandits (ARC-DUELLI). Under a bi-Lipschitz assumption on the transfer function, we show that ARC-DUELLI achieves regret of $\tilde{O}\left(T^{\frac{d_z+1}{d_z+2}}\right)$, where d_z is the zooming dimension of the near-optimal region, while using only $O(d \log T)$ space. For the linear preference transfer function, we also prove a matching lower bound via reduction from stochastic Lipschitz bandits, establishing optimality up to logarithmic factors. Experiments show convergence to the optimal arm and improved regret over uniform-discretization dueling baselines.

1 Introduction

Many modern learning and optimization problems involve a continuous decision space but provide only *relative* feedback. For example, when tuning a recommendation policy, ranking system, or experimental design, a user may be unable to assign reliable numerical scores to individual decisions, yet can often express a preference between two alternatives. In such settings, two structural features naturally coexist: the action space is continuous, so nearby decisions should behave similarly, and the available observations are purely pairwise comparisons rather than scalar rewards. This motivates the study of learning over continuous spaces under noisy comparative feedback.

We formalize this setting as *stochastic Lipschitz Dueling Bandits*. At each round, the learner selects two actions and observes noisy feedback on which one is preferred. At the same time, the latent utility is assumed to vary smoothly over the action space, so nearby actions induce similar preference behavior. The central question is whether this geometric regularity can be exploited to obtain regret guarantees that depend on the intrinsic complexity of the near-optimal region, rather than on an apriori discretizations of the space.

The main difficulty is that comparative feedback does not produce globally comparable estimates. Existing Lipschitz bandit algorithms (Kleinberg, 2004; Zhu & Huang, 2025) depend on scalar reward observations to form confidence bounds and elimination rules that are directly comparable across regions. Under dueling feedback, however, the outcome of a comparison is meaningful only relative to the opponent arm. As a result, estimates obtained in different regions are not directly comparable

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

unless they are anchored to a common baseline. This destroys the comparability of the underlying zooming-style elimination methods and makes region-level reasoning substantially more delicate. Resolving this lack of cross-region comparability is the key technical challenge in Lipschitz dueling bandits. Prior continuous-action dueling bandit work primarily assumes convexity (Saha et al., 2021; Kumagai, 2017), where pairwise comparisons can be converted into local first-order information and can be used in descent-based optimization frameworks. These methods are fundamentally local and rely on convex geometry; they do not yield the multiscale region-elimination machinery required in general Lipschitz spaces. Hence, neither line of work extends directly to stochastic dueling bandits over continuous Lipschitz action spaces. To our knowledge, this is the first work on noisy stochastic dueling bandits over general Lipschitz continuous action spaces with zooming-dimension-dependent regret guarantees, leaving open the following question:

Can one exploit Lipschitz structure under purely comparative feedback, and if so, can one establish zooming-dimension-dependent regret guarantees?

We answer this question affirmatively and initiate the study of *Lipschitz dueling bandits*. The key idea is to use a *fixed reference arm within each round*, so that all comparisons in that round are anchored to a common baseline. Crucially, the choice of this reference arm is highly non-trivial. If it is chosen poorly, then the resulting region-level estimates can lead to poorly ranked regions and make the elimination step unsafe, potentially discarding the region containing the Condorcet winner. In this paper, we show how to choose and update the reference arm adaptively so that it induces a scalar preference field over the action space whose variation is controlled by the Lipschitz structure and whose values remain suitable for elimination. This viewpoint yields three ingredients that are absent from prior analyses: first, a uniform concentration result for region-level preference estimates relative to the common reference arm; second, a safe elimination argument showing that the region containing the Condorcet winner is never discarded; and third, a shrinking-region property proving that all surviving regions remain geometrically close to the optimum. Together, these ingredients replace the absolute-reward comparisons used in classical Lipschitz bandits by a relative but globally anchored elimination principle. Building on this analysis, we propose Anchored Recursive Comparisons for dueling Lipschitz bandits, ARC-DUELLI, the first algorithm for this setting that achieves regret $\tilde{O}\left(T^{\frac{d_z+1}{d_z+2}}\right)$, where d_z is the zooming dimension of the near-optimal set. ARC-DUELLI is a depth-first elimination strategy which explores the partition tree recursively and stores a constant amount of local information from the active recursive path. This yields an overall memory complexity of $O(d \log T)$. Hence, purely comparative feedback in continuous spaces does not fundamentally increase the space complexity of zooming-style learning.

When pairwise feedback is generated via the linear transfer function, we prove a matching lower bound through a reduction to scalar-feedback Lipschitz bandits and show that every dueling bandit algorithm induces a corresponding scalar-feedback algorithm with exactly twice the regret. Therefore, existing lower bounds for stochastic Lipschitz bandits carry over directly to the dueling setting under the linear transfer model. Consequently, despite the weaker and purely relative nature of the feedback, the achievable regret of ARC-DUELLI matches the best known rate for stochastic Lipschitz bandits up to logarithmic factors (Kleinberg et al., 2019). Finally, we complement our theory with empirical evaluation. The experiments show that ARC-DUELLI converges to the optimal arm and achieves substantially lower regret than dueling-bandit baseline based on uniform discretization, illustrating the advantage of exploiting geometric structure. Overall, this work identifies Lipschitz dueling bandits as a new structured bandit model and shows that zooming-style learning is possible even when feedback is purely comparative, provided one restores cross-region comparability through an adaptive common reference.

2 Related Work

Dueling Bandits and Preference-Based Bandits. The dueling bandits framework proposed by Yue et al. (2012) studies sequential learning from pairwise comparisons between actions, rather than absolute reward feedback, with the goal of identifying or competing with the 'best performing arm' depending upon the notion of best arm that is, Condorcet winner, an action that is preferred over all the other actions with probability greater than half. Existing algorithms (Yue & Joachims, 2011; Urvoy et al., 2013; Zoghi et al., 2014; Komiyama et al., 2015) assume a finite or unstructured action space and focus over arm-level elimination or confidence-based selection.

Over the continuous action space, with concave rewards, Yue & Joachims (2009) proposed the first projected gradient-based algorithm with regret guarantees of $O(T^{3/4})$. Later, Kumagai (2017) improved the regret to $O(\sqrt{T \log T})$ under the assumption of strong convexity and smoothness of the cost function. A more general cost function, α-strong and β-smooth was considered in (Jamieson et al., 2012) which was further generalized to any β-smooth convex functions in (Saha et al., 2021), only for the sign-based relative feedback. A more general preference function was considered in (Saha et al., 2025) with β-smooth convex functions. This paper extends this line of work to the class of more general reward functions, i.e, Lipschitz function. Some works (Xu et al., 2020; Zhang & Li, 2024) do consider non-convex optimization problem; however they establish simple-regret guarantees under RKHS assumptions.

Lipschitz and Continuum-Armed Bandits. A separate line of work studies bandits over continuous action spaces under Lipschitz assumption, where the learner observes noisy scalar rewards (Kleinberg et al., 2019; Zhu & Huang, 2025; Kleinberg, 2004; Feng et al., 2022) or adversarial scalar rewards (Kang et al., 2023). The problem was first introduced by Kleinberg (2004) where first discretization based algorithm was proposed for 1-dimensional function. Later, Kleinberg et al. (2008) presented the first zooming-based algorithm for general d-dimensional Lipschitz functions and also provided the lower bound of $\Omega(T^{\frac{d_z+1}{d_z+2}})$ with d_z being the zooming dimension. Zooming-based algorithms essentially keep zooming in the most promising region until it reaches the near optimal region. In a follow-up work, Bubeck et al. (2011) proposed Hierarchical Optimistic Optimization (HOO) algorithm, which is easier to implement and which exploits the local properties of the reward function at its maxima only. To reduce the space complexity of zooming based algorithms, (Zhu & Huang, 2025) provided a recursive variant of zooming algorithm with space complexity of $O(\log T)$. However, all the above methods crucially rely on access to reward feedback and do not apply to settings with purely comparative observations.

Bayesian Optimization with human feedback Another line of work studies optimization over continuous action spaces using noisy pairwise comparisons through preferential Bayesian optimization and Bayesian optimization from human feedback (BOHF). González et al. (2017) introduced a Gaussian-process-based method for optimization from preferences. Later work used skew Gaussian processes to improve preference prediction and select comparison pairs more efficiently (Benavoli et al., 2021; Takeno et al., 2023). Xu et al. (2024) proposed an optimistic algorithm with likelihood-based confidence sets and regret guarantees. Pásztor et al. (2024) proposed MAXMINLCB, which jointly selects comparison pairs through a Stackelberg game using kernelized logistic confidence sequences. Kayal et al. (2025) obtained tighter information-gain-dependent regret bounds, while Lazzaro et al. (2026) provided a finite-time regret analysis of Thompson sampling. These BOHF guarantees assume that the utility has bounded norm in the RKHS of a specified kernel, with regret bounds depending on kernel-specific quantities such as maximum information gain. In contrast, our method relies on Lipschitz continuity of the utility and a bi-Lipschitz transfer function, without requiring kernel knowledge or RKHS membership.

We study stochastic dueling bandits over continuous action spaces with Lipschitz structure. To our knowledge, this is the first work on noisy stochastic dueling bandits over general Lipschitz continuous action spaces with zooming-dimension-dependent regret guarantees.

3 Problem Setup and Regret Definition

Let $\mathcal{X} \subset \mathbb{R}^d$ be a compact action space endowed with a metric $\|\cdot\|$. Throughout the paper, we work with $\mathcal{X} = [0,1]^d$ under the ℓ_∞ metric, for which metric balls are axis-aligned cubes. This allows a particularly simple description of our recursive elimination procedure via a cube-based multiscale partition. However, the analysis is not tied to the coordinate structure of Euclidean cubes. It relies only on three abstract properties: (i) regions of controlled diameter at each scale, (ii) a hierarchical refinement into smaller regions at finer scales, and (iii) packing or covering bounds for the near-optimal set. Thus, our arguments should extend to compact doubling metric spaces endowed with a suitable hierarchical covering, though we restrict attention to $[0,1]^d$ under the ℓ_∞ metric for clarity. Further, we assume that the unknown reward function $f : \mathcal{X} \to \mathbb{R}$ satisfies the following regularity condition.

Assumption 3.1 (Lipschitz reward)**.** The function f is 1-Lipschitz with respect to $\|\cdot\|$, i.e.,

$$|f(x) - f(y)| \leq \|x - y\|, \qquad \forall x, y \in \mathcal{X}.$$

Unlike the Lipschitz bandit, the learner does not observe the values of f directly in a dueling feedback model. Instead, based on past observations, at each time t, the learner selects two arms denoted by $(x_t, y_t) \in \mathcal{X}^2$ and only observes a noisy binary (duel) outcome $o_t(x_t, y_t)$ of the stochastic duel. The learner operates over a horizon of T duels. The duel outcome is a Bernoulli random variable

$$o_t(x_t, y_t) \sim \mathrm{Ber}(P(x_t \succ y_t)).$$

Here, $x_t \succ y_t$ denote the event that arm x_t is preferred over y_t. We connect the pairwise preference probability $P(x_t \succ y_t)$ through a transfer function $\rho : [-1, 1] \to [0, 1]$ as $P(x_t \succ y_t) = \rho(f(x_t) - f(y_t))$. Note that since f is 1-Lipschitz on $[0, 1]^d$ under ℓ_∞, we have $|f(x) - f(y)| \leq 1$ for all $x, y \in \mathcal{X}$. Hence the transfer function only needs to be specified on $[-1, 1]$. Different examples of transfer functions include : (a) Bradley–Terry transfer (Logistic) (Bengs et al., 2021), (b) Linear transfer function (Ailon et al., 2014), and (c) probit transfer function (Chen & Frazier, 2017). This paper makes the following assumption on the transfer function.

Assumption 3.2. The transfer function $\rho : [-1, 1] \to [0, 1]$ satisfies $\rho(0) = \frac{1}{2}, \rho(-z) = 1 - \rho(z)$, and there exist constants $0 < \lambda \leq \Gamma < \infty$ such that for all $-1 \leq z_2 \leq z_1 \leq 1$,

$$\lambda(z_1 - z_2) \leq \rho(z_1) - \rho(z_2) \leq \Gamma(z_1 - z_2).$$

It is easy to show that Bradley-Terry, Probit, and Linear transfer functions satisfy these assumptions and are bi-Lipschitz with $(\Gamma, \lambda) = \left(\frac{1}{4}, \frac{e}{(1+e)^2}\right)$ (Lemma D.1), $(\Gamma, \lambda) = \left(\frac{1}{\sqrt{2\pi}}, \frac{1}{\sqrt{2\pi}} e^{-1/2}\right)$ (Lemma D.2), and $(\Gamma, \lambda) = \left(\frac{1}{2}, \frac{1}{2}\right)$ (Lemma D.3) respectively.

An arm x^* is called the Condorcet winner if $P(x^* \succ y) \geq \frac{1}{2}$ for all $y \in \mathcal{X}$. Under Assumption 3.1, the function f is continuous. Since $\mathcal{X}$ is compact, there exists $x^* \in \arg\max_{x \in \mathcal{X}} f(x)$. Pairwise preferences are induced via a transfer function $\rho : [-1, 1] \to [0, 1]$ such that $P(x \succ y) = \rho(f(x) - f(y))$, where ρ is non-decreasing and satisfies $\rho(0) = \frac{1}{2}$. Therefore, for any $y \in \mathcal{X}$,

$$f(x^*) \geq f(y) \;\Rightarrow\; P(x^* \succ y) \geq \frac{1}{2}.$$

Thus, a Condorcet winner always exists.

3.1 Regret in Lipschitz Dueling Bandits

For any pair of arms (x, y), define the (pairwise) preference gap $\Delta(x, y) := P(x \succ y) - \frac{1}{2}$. Note that $\Delta(x^\star, x) \geq 0$ for all $x \neq x^\star$ also under Assumption 3.2, $\Delta(x, y) = -\Delta(y, x)$. Further, let $\kappa = \frac{\Gamma}{\lambda}$. We next show that under Assumption 3.2, preference gap Δ function satisfy relaxed stochastic triangular inequality and reverse difference inequality.

Lemma 3.3. *Under Assumption 3.2, the preference gap Δ satisfies:*

1. *if $f(x) \geq f(z) \geq f(y)$, then $\Delta(x, y) \leq \kappa(\Delta(x, z) + \Delta(z, y))$,*
2. *if $f(x) \geq f(y) \geq f(z)$, then $\Delta(x, y) \leq \kappa(\Delta(x, z) - \Delta(y, z))$.*

The proof uses simple inequalities arising from bi-Lipschitz property and is provided in Appendix E.

The instantaneous regret r_t incurred at time t is defined as the sum of preference gaps between the optimal arm $x^\star$ and the two arms played in the duel i.e., $r_t := \Delta(x^\star, x_t) + \Delta(x^\star, y_t)$. The cumulative regret after T duels is

$$R(T) := \sum_{t=1}^{T} r_t = \sum_{t=1}^{T} \Big(\Delta(x^\star, x_t) + \Delta(x^\star, y_t)\Big). \tag{1}$$

This regret notion is standard in stochastic dueling bandits and measures how much worse the learner's comparisons are relative to always comparing the Condorcet winner against itself (Yue et al.,

Algorithm 1 Anchored Recursive Comparison for Dueling Lipschitz Bandits (ARC-DUELLI)

Require: Action space $\mathcal{X} = [0,1]^d$, horizon T, confidence δ, Γ

Choose number of rounds B
Set radius sequence $r_m = 2^{-m+1}$ for $m = 1, \ldots, B+1$
Set sample sizes $n_m = \frac{16 \log(T/\delta)}{r_m^2}$
Initialize: $t \leftarrow 0$, $M \leftarrow 0$, $X^0 \sim \mathrm{Unif}(\mathcal{X})$, $\tilde{\Delta}^0 = 0$
for $m = 1$ to B **do**
  Set $\Delta_{\max}^m = -\infty$, $C_{\max}^m = \emptyset$
  Partition $\mathcal{X}$ into $(1/r_1)^d$ cubes of edge length r_1
  for each cube C **do**
    ROUNDFUNC $(t, m, 1, C, \Delta_{\max}^m, C_{\max}^m, \tilde{\Delta}^{m-1}, X^{m-1})$
  end for
  if $C_{\max}^m \neq \emptyset$ and $t + n_m \leq T$ **then**
    Choose X^m uniformly from the sampled arms in $C_{\max}^m$
    Sample n_m arms uniformly from $C_{\max}^m$ and duel each with X^m
    Compute empirical gap $\widehat{\Delta}_m^m(C_{\max}^m, X^m)$
    $\tilde{\Delta}^m \leftarrow \widehat{\Delta}_m^m(C_{\max}^m, X^m)$, $t \leftarrow t + n_m$
    $M \leftarrow m$
  else
    break
  end if
end for
Cleanup: play duel (X^M, X^M) for all remaining time steps.

Algorithm 2 Recursive cube exploration (ROUNDFUNC)

Require: Time horizon T, current time counter t, round index m, current depth h, cube C, current-round maximum $\Delta_{\max}^m$, best cube $C_{\max}^m$, previous-round estimate $\tilde{\Delta}^{m-1}$, and reference arm X^{m-1}, Γ.

if $t + n_h > T$ **then**
  return
end if
Sample n_h arms uniformly from C and duel each with X^{m-1}
Compute empirical gap $\widehat{\Delta}_h^m(C, X^{m-1})$
$t \leftarrow t + n_h$
if $h = m$ **then**
  if $\widehat{\Delta}_h^m(C, X^{m-1}) > \Delta_{\max}^m$ **then**
    $\Delta_{\max}^m \leftarrow \widehat{\Delta}_h^m(C, X^{m-1})$, $C_{\max}^m \leftarrow C$
  end if
else
  if $\tilde{\Delta}^{m-1} - \widehat{\Delta}_h^m(C, X^{m-1}) \leq 2\, r_h(1+\Gamma)$ **then**
    Partition C into $(r_h/r_{h+1})^d$ subcubes
    for each subcube C' **do**
      ROUNDFUNC$(T, t, m, h+1, C', \Delta_{\max}^m, C_{\max}^m, \tilde{\Delta}^{m-1}, X^{m-1})$
    end for
  else
    Eliminate cube C (terminate recursion)
  end if
end if

2012) . Under the convex loss function, Kumagai (2017) presents a dueling bandit algorithm for achieving regret of $O(\sqrt{T \log T})$. However, under a Lipschitz reward function the regret is often dependent on zooming dimension (Zhu & Huang, 2025; Kleinberg, 2004) which captures the intrinsic dimensionality of the near-optimal region and governs the achievable regret rates.

For $\varepsilon > 0$, define the ε-near-optimal set $\mathcal{S}(\varepsilon) := \{x \in \mathcal{X} : \Delta(x^\star, x) \leq \varepsilon\}$. We assume that $\mathcal{S}(\varepsilon)$ has *zooming dimension* d_z, meaning that there exists $C_z > 0$ such that for all $\varepsilon > 0$, the set $\mathcal{S}(\varepsilon)$ can be covered by at most $C_z \varepsilon^{-d_z}$ balls of radius ε.

Thus, the goal in Lipschitz dueling bandit setting is to design a learning algorithm that minimizes the expected regret $\mathbb{E}[R(T)]$ while only observing noisy dueling feedback, and to characterize how the regret scales with the time horizon T and the geometric complexity of the near-optimal region.

4 ARC-DUELLI: Anchored Recursive Comparison for Dueling Lipschitz

We present ARC-DUELLI, Anchored Recursive Comparison for Dueling Lipschitz in Algorithm 1 (main procedure) and Algorithm 2 (recursive subroutine, ROUNDFUNC). ARC-DUELLI combines round-based exploration with recursive cube elimination to progressively localize the Condorcet winner. For each depth $h \geq 1$, we define a resolution $r_h = 2^{-h+1}$. At this resolution, $\mathcal{X}$ is partitioned into axis-aligned hypercubes ("cubes") of edge length r_h, each representing a region of diameter at

most r_h. A cube at depth $h+1$ is obtained by equally partitioning a cube at depth h into 2^d subcubes, yielding a hierarchical multiscale partition of $\mathcal{X}$.

The algorithm proceeds in rounds $m = 1, \ldots, B$, where $B = \left\lceil \frac{\log_2 T}{d_z+2} \right\rceil + 1$. The parameter B balances exploration depth and the resolution of suboptimality within cubes. Value of B depends on zooming dimension d_z which we assume to be known. When d_z is unknown, since $d_z \leq d$, we can use upper bound of d_z with d being the ambient dimension. The choice of B is justified by the regret guarantee in Theorem 5.4. At the start of round m, the algorithm maintains a *reference arm* X^{m-1} and the best empirical preference gap $\tilde{\Delta}^{m-1}$ from the previous round. All comparisons in round m are made against X^{m-1}, ensuring that empirical gap estimates across cubes are directly comparable.

During round m, the action space is first partitioned into cubes at depth 1 (line 7). Each cube is then explored by invoking the recursive subroutine ROUNDFUNC starting at depth $h = 1$ (lines 8–9). The subroutine processes a cube C at depth h as follows. If sufficient time remains (Algorithm 2, lines 1–2), it samples $n_h = \frac{16 \log(T/\delta)}{r_h^2}$ arms uniformly from C, and each is dueled against common reference arm X^{m-1} (Algorithm 2, lines 4). From these samples, it computes the empirical preference gap $\widehat{\Delta}_h^m(C, X^{m-1})$ (line 5) using equation 2. The sample size n_h guarantees that $\widehat{\Delta}_h^m(C, X^{m-1})$ closely approximates $\Delta(x, X^{m-1})$ uniformly for all $x \in C$ at depth h (Lemma 5.1).

If the maximum depth $h = m$ is reached, the cube is not refined further. Instead, the algorithm sets $\Delta_{\max}^m$ to the maximum of $\widehat{\Delta}_h^m(C, X^{m-1})$ over all terminal cubes in round m, and stores a corresponding maximizer $C_{\max}^m$ (Algorithm 2, lines 8–9). If such a cube exists, the next reference arm is selected from $C_{\max}^m$ (lines 11–12), and $\tilde{\Delta}^m$ is refined by drawing n_m additional samples against the reference arm (lines 13–15). If, at any point during round m, insufficient time remains to complete the next required sampling block, the recursive exploration is terminated and the main procedure returns X^M (lines 21), namely, the reference arm obtained from the most recently completed round M. Thus, an incompletely explored round is never used to update the reference arm. We later prove that even if such a case exists then regret bounds remain the same. The choice of the reference arm is crucial, as all future elimination decisions are made relative to it.

If $h < m$, the algorithm decides whether to eliminate or refine the cube. Specifically, cube C is eliminated if $\tilde{\Delta}^{m-1} - \widehat{\Delta}_h^m(C) > 2\, r_h(1+\Gamma)$ (line 12), which, by concentration and Lipschitz arguments, implies that all arms in C are suboptimal with high probability (Lemma 5.2). Otherwise, C is partitioned into 2^d sub-cubes of edge length r_{h+1} and explored recursively (Algorithm 2, lines 13–15).

The algorithm takes the transfer-function Lipschitz constant Γ as an input. In practice, one may use an estimate or upper bound of Γ which affects the elimination rule. If Γ is underestimated, the elimination threshold becomes too aggressive, and the algorithm may incorrectly eliminate the cube containing the Condorcet winner. On the other hand, if Γ is substantially overestimated, the threshold becomes overly conservative: more suboptimal cubes survive, exploration becomes less focused, and the regret may increase. Thus, the choice of Γ induces a natural safety–efficiency tradeoff.

Although written recursively, the procedure is implemented as an in-place depth-first traversal. At any time, it stores only the current cube, current depth, current reference, current best candidate, and a constant number of counters and empirical estimates. Information is discarded once recursion on a cube terminates, yielding a total memory usage of $O(d \log T)$ due to the maximum depth $B = O(\log T)$. Across rounds, the reference arm is progressively localized near the Condorcet winner with high probability. After $M \leq B$ rounds where M is the last completed round, Lemma 5.3 ensures that all surviving arms lie within $O(r_{M-1})$ preference gap of the Condorcet winner. The algorithm then enters a cleanup phase, repeatedly playing the reference arm X^M from the final completed round M until the horizon T is exhausted (Algorithm 1, line 21).

5 Regret Analysis for ARC-DUELLI

First lemma shows that with n_h number of duels in each round m at depth h, the empirical preference gap for a cube C closely approximate the preference gap for an arbitrary arm in C with any reference arm Y. The proof essentially uses Hoeffding's inequality and Lipschitz property of the transfer function. The detailed proof is provided in Appendix B.1.

Lemma 5.1. *[Uniform concentration for cube-level preference estimates] Fix a round m, depth h, and a cube $C \in \mathcal{A}_h^m$ of diameter at most r_h. Let $Y \in \mathcal{X}$ be the reference arm fixed before that sampling block begins and let $\{x_i\}_{i=1}^{n_h}$ be i.i.d. samples drawn uniformly from C. Define the empirical estimator $\widehat{\Delta}_h^m(C, Y)$ and event $\mathcal{E}$ as:*

$$\widehat{\Delta}_h^m(C, Y) \;:=\; \frac{1}{n_h} \sum_{i=1}^{n_h} o(x_i, Y) - \frac{1}{2}, \tag{2}$$

$$\mathcal{E} := \left\{ \left|\Delta(x, Y) - \widehat{\Delta}_h^m(C, Y)\right| \le \sqrt{\frac{16 \log(T/\delta)}{n_h}} + \Gamma r_h \right\}, \tag{3}$$

for all $1 \le h \le m \le B$, all $C \in \mathcal{A}_h^m$, and all $x \in C$. Then, $\mathbb{P}(\mathcal{E}) \;\ge\; 1 - 2\delta$. Here, $o(x_i, Y) \sim Ber(P(x_i \succ Y))$.

Proof sketch. For a fixed tuple (m, h, C), the error $\left|\Delta(x, Y) - \widehat{\Delta}_h^m(C, Y)\right|$ is decomposed into a stochastic term and a bias term for an arbitrary arm x. Conditioned on the reference arm Y and the cube C, the duel outcomes against Y are i.i.d. Bernoulli, so Hoeffding's inequality gives concentration of $\widehat{\Delta}_h^m(C, Y)$ around its expectation at rate $\sqrt{16 \log(T/\delta)/n_h}$. The remaining bias comes from replacing the gap of the fixed arm x by the average gap over sampled points in C and can be upper bounded Γr_h through Lipschitz continuity of f and ρ. Combining the two bounds yields the stated deviation inequality, and a union bound over all rounds and active cubes gives the event $\mathcal{E}$ with probability at least $1 - 2\delta$.

The next lemma shows that when elimination occurs at line 12 of recursive subroutine, then the eliminated cube does not contain an optimal arm with high probability. The proof essentially uses Lemma 5.1, the optimality of Condorcet arm, and the fact that comparisons are made against the same reference arm. The detailed proof is provided in Appendix B.2

Lemma 5.2. *[Optimal arm is never eliminated] Suppose that the algorithm completes $M \le B$ rounds. Under the event $\mathcal{E}$ defined in Lemma 5.1, an optimal arm $x^\star \in \arg\max_{x \in \mathcal{X}} f(x)$ is not eliminated in any of the completed rounds $m \in [M]$. Consequently, the cube containing $x^\star$ never gets eliminated.*

Proof sketch. Suppose that the algorithm completes $M \le B$ rounds. Fix any completed round $m \in [M]$ and any depth $h < m$, and let $(C_h^m)^\star$ denote the depth-h cube containing the optimal arm $x^\star$. The key point is that, after the benchmark $\tilde{\Delta}^{m-1}$ is redefined in Line 15 using the newly chosen reference arm X^{m-1}, both $\tilde{\Delta}^{m-1}$ and the current-round estimate $\widehat{\Delta}_h^m\big((C_h^m)^\star, X^{m-1}\big)$ are evaluated against the same reference arm. This makes them directly comparable. Under the high-probability event $\mathcal{E}$, Lemma 5.1 controls both quantities up to the usual concentration and Lipschitz error. Since $x^\star$ maximizes f and the transfer function is monotone, its true preference gap against X^{m-1} is at least as large as that of any arm in the cube from which X^{m-1} was selected. Therefore, the difference $\tilde{\Delta}^{m-1} - \widehat{\Delta}_h^m\big((C_h^m)^\star, X^{m-1}\big)$ is at most the elimination threshold $2r_h(1 + \Gamma)$. Hence the cube containing $x^\star$ cannot be eliminated. Since this holds at every depth and every completed round $m \le M$, the optimal arm is never discarded during the M completed rounds. If the algorithm stops during round $M + 1$, all partial updates from that incomplete round are discarded, and the algorithm returns the reference arm from previous completed round i.e. X^M.

The next lemma shows that as we keep shrinking the cube, the gap between optimal arm and any arm in that cube decreases. For this, we use Lemma 3.3 along with the Lemma 5.1. The detailed proof is provided in Appendix B.3.

Lemma 5.3. *[Shrinking-region property for dueling bandits] Suppose that the algorithm completes $M \le B$ rounds. Under the event $\mathcal{E}$, for any $1 \le h \le m \le M$, any cube $C \in \mathcal{A}_h^m$, and any arm $x \in C$, the Condorcet gap satisfies $\Delta(x^\star, x) \le \kappa(2\kappa + 4)(1 + \Gamma) r_{h-1}$, with $\kappa = \frac{\Gamma}{\lambda}$.*

Proof sketch. Fix a completed round $m \le M$, a depth $h \le m$, a surviving cube $C \in \mathcal{A}_h^m$, and an arm $x \in C$. For $h = 1$, the claim is immediate since preference gaps are bounded. For $h \ge 2$, let C_{par} denote the parent of C at depth $h - 1$. The proof compares x to the optimal arm $x^\star$ through the common reference arm X^{m-1} and applies Lemma 3.3 to write $\Delta(x^\star, x) \le \frac{\Gamma}{\lambda}\big(\Delta(x^\star, X^{m-1}) - \Delta(x, X^{m-1})\big)$. The first term is controlled by showing that the newly chosen

reference arm X^{m-1} comes from a cube whose empirical gap against the previous reference X^{m-2} is at least as large as that of the optimal cube; under the concentration event $\mathcal{E}$, this implies that $\Delta(x^\star, X^{m-1})$ is at most $O(r_{m-1}(1+\Gamma))$. The second term is controlled by the fact that the parent cube C_{par} survives elimination in round m, so its empirical estimate against X^{m-1} cannot be much smaller than the benchmark $\tilde{\Delta}^{m-1}$. Using Lemma 5.1 to pass from empirical cube estimates to true preference gaps then gives a lower bound on $\Delta(x, X^{m-1})$. Combining the two bounds yields $\Delta(x^\star, x) \le \kappa(2\kappa+4)(1+\Gamma)r_{h-1}$ showing that every surviving cube in each completed round $m \le M$ lies geometrically close to the optimum.

We now prove our main theorem that bounds the regret for ARC-DUELLI by combining the above lemmas and finding the optimal value of B. The detailed proof is given in Appendix B.4.

Theorem 5.4. *[Regret bound] Assume that the near-optimal set $\mathcal{S}(\varepsilon) := \{x \in \mathcal{X} : \Delta(x^\star, x) \le \varepsilon\}$ has zooming dimension d_z under the metric $\|\cdot\|$. Then, under the event $\mathcal{E}$, the regret of the* ARC-DUELLI *satisfies*

$$R(T) = \tilde{O}\Big(T^{\frac{d_z+1}{d_z+2}}\Big),$$

by setting $B = \left\lceil \frac{\log_2 T}{d_z+2} \right\rceil + 1$. The bound holds even if the algorithm completes only $M < B$ rounds and stops during round $M+1$ due to insufficient remaining time. Here, $\tilde{O}(\cdot)$ hides logarithmic factors in T and $1/\delta$. In particular, for $T \ge 2$, running ARC-DUELLI *with $\delta = 1/T$ yields the expected regret bound $\mathbb{E}[R(T)] = \tilde{O}\Big(T^{\frac{d_z+1}{d_z+2}}\Big)$.*

Proof sketch Let $M \le B$ denote the number of rounds completed by the algorithm. We decompose the regret into the contribution accumulated during the completed rounds, the possible incomplete round $M+1$, and the remaining cleanup phase. For a fixed completed round $m \le M$, every duel compares the round reference arm X^{m-1} with an arm drawn from some surviving cube at depth h. By Lemma 5.3, both the played arm and the reference arm are $O(r_{h-1}(1+\Gamma))$-optimal, so each duel at depth h incurs regret at most $O(r_h(1+\Gamma))$. Since each surviving cube is sampled $n_h = \Theta(\log(T/\delta)/r_h^2)$ times, the regret contribution at depth h is controlled by $|\mathcal{A}_h^m| \cdot \widetilde{O}(1/r_h)$. The zooming-dimension assumption bounds the number of surviving depth-h cubes by $|\mathcal{A}_h^m| \le C_z r_h^{-d_z}$, yielding a round-wise regret of order $\widetilde{O}\big(\sum_{h=1}^{m} 2^{(h-1)(d_z+1)}\big)$. If $M = B$, summing over all rounds gives an exploration term of order $\widetilde{O}(2^{(B-1)(d_z+1)})$. After round B, Lemma 5.3 implies that the final reference arm lies within $O(r_{B-1}(1+\Gamma))$ of the optimum, so the cleanup phase contributes at most $O(2^{-B}T)$. If $M < B$, the algorithm stops during round $M+1$ because insufficient time remains to complete the next sampling block. The regret incurred during the completed blocks of this incomplete round is bounded by the same argument as above. Moreover, if τ is the remaining number of time steps, the stopping condition gives $\tau < n_h \le n_{M+1}$. Since the algorithm plays X^M during these steps, the cleanup regret is at most $O(r_{M-1}n_{M+1}) = \widetilde{O}(2^M) \le \widetilde{O}\big(2^{M(d_z+1)}\big)$. Thus, early stopping does not increase the regret order. Choosing $B = \left\lceil \frac{\log_2 T}{d_z+2} \right\rceil + 1$ balances the exploration and cleanup terms when all rounds are completed and, in both cases, yields $R(T) = \widetilde{O}\Big(T^{\frac{d_z+1}{d_z+2}}\Big)$.

Next, we provide space complexity analysis for ARC-DUELLI which closely follow the existing depth-based elimination algorithm(Zhu & Huang, 2025). Detailed proof is provided in Appendix B.5.

Lemma 5.5. *[Space complexity]* ARC-DUELLI *requires at most $O(d \log T)$ bits of memory at any time.*

6 Lower Bound

In lower bound, we prove that given any dueling algorithm, there exists a scalar-feedback bandit algorithm whose regret is twice of that of dueling algorithm. Since our model class includes the linear transfer function, we provide the lower bound for this subclass. Let $\mathcal{H}$ be any class of stochastic Lipschitz bandit instances on $\mathcal{X}$, where each instance is specified by a mean reward function $\mu : \mathcal{X} \to [0,1]$, and pulling arm x returns an independent Bernoulli reward with mean $\mu(x)$.

For each $\mu \in \mathcal{H}$, define the induced dueling instance $\Phi(\mu)$ on the same action space by

$$P_\mu(x \succ y) = \frac{1}{2}\big(1 + \mu(x) - \mu(y)\big), \qquad x, y \in \mathcal{X}.$$

This is exactly the dueling model with linear transfer function. Then we have the following theorem:

Theorem 6.1. *[Reduction] Let A be any dueling bandit algorithm with linear transfer function operating for T rounds. Then there exists a scalar-feedback bandit algorithm B_A operating for $2T$ rounds such that, for every instance μ, $\mathbb{E}_\mu\big[R^{\mathrm{sc}}_{B_A}(2T)\big] = 2\,\mathbb{E}_\mu\big[R^{\mathrm{duel}}_A(T)\big]$. Here, $R^{\mathrm{sc}}_{B_A}(2T)$ represents regret by algorithm B_A for scalar-feedback bandit algorithm and R^{duel}_A represent regret by algorithm A under duel-feedback bandit algorithm.*

Proof sketch. Given any dueling bandit algorithm A, we construct a scalar-feedback algorithm B_A that simulates A using two scalar pulls per duel. Whenever A queries (x_t, y_t), the algorithm B_A pulls both arms once and converts the two scalar observations into a synthetic duel outcome by declaring the larger reward the winner and breaking ties uniformly at random. Under the linear transfer function, this synthetic comparison has exactly the same distribution as a true duel. Moreover, the scalar regret incurred by the two pulls is exactly twice the dueling regret at that round. Summing over all rounds and taking expectations yields the claim.

Thus, the lower bound on B_A is thus applicable to A. Noting that for every A, there exists a corresponding B_A, we get the desired Corollary:

Corollary 6.2. *If the corresponding stochastic Lipschitz bandit class satisfies the lower bound $\inf_B \sup_{\mu\in\mathcal{H}} \mathbb{E}_\mu[R^{\mathrm{sc}}_B(T)] = \Omega\Big(T^{\frac{d_z+1}{d_z+2}}\Big)$, then the lower bound on dueling bandit class satisfies $\inf_A \sup_{\nu\in\Phi(\mathcal{H})} \mathbb{E}_\nu[R^{\mathrm{duel}}_A(T)] = \Omega\Big(T^{\frac{d_z+1}{d_z+2}}\Big)$.*

7 Experiments

We evaluate the proposed algorithm on a synthetic continuous dueling bandit instance to study four key questions: (i) whether the sequence of reference arms converges toward the optimal action, (ii) whether the preference gap between the optimal arm and the reference arm decreases across rounds as predicted by our theory, (iii) how cumulative regret scales with the time horizon, and (iv) how the proposed adaptive zooming strategy compares with a uniform discretization baseline.

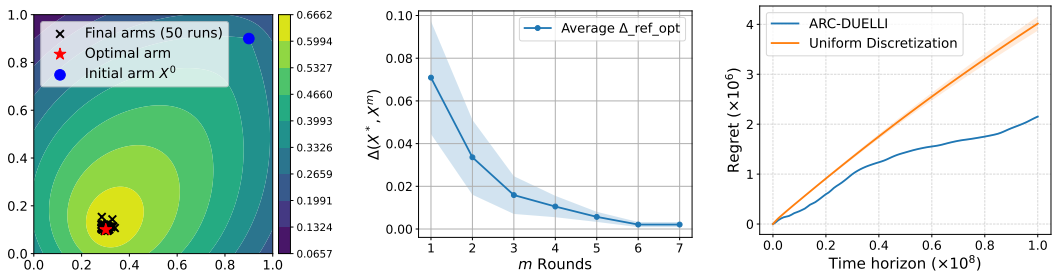
We consider the continuous action space $\mathcal{X} = [0, 1]^2$. The utility of an arm $x \in \mathcal{X}$ is generated according to $\mu(x) = 1 - \frac{2}{3}\|x - x_1\| - \frac{1}{3}\|x - x_2\|$, where $x_1 = (0.3, 0.1), x_2 = (0.9, 0.9)$. This reward landscape induces a unique maximizer at $x^\star = (0.3, 0.1)$. Pairwise comparisons are stochastic and follow the logistic transfer model. Thus, arms with higher utility are more likely to win pairwise duels, with confidence increasing as the utility gap grows.

Unless stated otherwise, we set the zooming dimension to $d_z = 0$. For the convergence experiments, we run the algorithm for horizon $T = 10^8$ over 50 independent trials. To test robustness, we initialize the first reference arm at the deliberately suboptimal point $(0.9, 0.9)$, which lies far from the optimum. All reported results are averaged over repeated runs.

Figure 1a shows that, despite an unfavorable initialization at $(0.9, 0.9)$, the reference arms consistently move toward the optimal action across all runs. To validate our theory, we track $\Delta(x^\star, X^m)$, where X^m denotes the reference arm in round m. As shown in Figure 1b, the gap steadily decreases toward zero, supporting Lemma 5.3. For fixed $T = 10^8$, it eventually stabilizes due to the bounded exploration depth, after which zooming stops.

Figure 1c compares our method with a baseline that uniformly discretizes the action space into a grid with cube radius $r = 2^{-m+1}$, matching the resolution used in ARC-DUELLI for the same depth parameter m. Each cube is treated as an arm, and the Interleaved Filter algorithm (Yue et al., 2012) is applied on this discretized space. Our zooming-based method significantly outperforms this baseline. Moreover, the small standard deviation in our ARC-DUELLI in Figure 1c indicates that our algorithm achieves consistent performance across independent runs.

We conduct additional experiments to evaluate the robustness of ARC-DUELLI across different settings, including random initialization of the reference arm, scaled utility ranges (including the



(a) Reference arm convergence (b) Convergence of $\Delta(x^*, X^m)$ (c) Regret Comparison

Figure 1: Performance of ARC-DUELLI over 50 runs. (a) Final reference arms concentrate near the optimal arm. (b) Decay of the gap $\Delta(x^\star, X^m)$ across rounds, showing convergence of the reference arm. (c) Cumulative regret compared against a uniform discretization baseline (Interleaved Filter).The solid curves denote averages over runs, and the shaded regions represent ± 1 standard deviation.

domain $[0, 10]^2$), utility geometries (Flat, Plateau, Quadratic, Ridge, Sharp, and Two Peaks), transfer functions (sigmoid, arctangent, probit, linear, and hyperbolic tangent), and higher ambient dimensions ($d = 3$ and $d = 4$). Across these settings, ARC-DUELLI consistently achieves decreasing preference gaps and lower regret than the uniform discretization baseline. The results further demonstrate its ability to identify near-optimal points or regions across diverse utility landscapes and transfer functions. In higher dimensions, preference gaps continue to decrease, although the increased number of cubes requires more comparisons per round, reducing the number of completed rounds within the same horizon. All additional experimental results are provided in Appendix F.

Overall, the experiments corroborate the theoretical predictions: the reference arms converge to the optimum, the preference gap decays across rounds, cumulative regret is sublinear, and adaptive zooming significantly outperforms uniform discretization in continuous dueling settings.

8 Discussion and Future Work

We study Lipschitz dueling bandits over continuous action spaces and propose ARC-DUELLI, the first algorithm that provably exploits geometric structure under purely comparative feedback. We show a regret bound of $\tilde{O}\left(T^{\frac{d_z+1}{d_z+2}}\right)$, matching the best known rates for stochastic Lipschitz bandits up to logarithmic factors, while using only logarithmic memory. These results show that purely comparative (dueling) feedback is sufficient to achieve regret guarantees that depend on the zooming dimension, matching those of scalar-reward Lipschitz bandits. This highlights that even without access to absolute rewards, the underlying structure of the problem can still be effectively exploited. As a result, our framework extends the applicability of bandit methods to settings where only relative feedback is available, such as human-in-the-loop optimization and recommender systems.

Several directions remain for future work. First, ARC-DUELLI assumes knowledge of the zooming dimension to select the exploration depth that yields its d_z-dependent regret guarantee. Although the ambient dimension provides a fallback when d_z is unknown, designing a method that automatically adapts to the near-optimal geometry and retains the sharper guarantee remains a direction for our framework. Second, our analysis relies on a bi-Lipschitz transfer condition over the relevant range of utility differences and requires knowledge of an upper Lipschitz bound Γ. Extending the theory to broader preference models, including transfer functions with vanishing slope or weaker identifiability, and achieving comparable guarantees without prior knowledge of Γ, would further clarify the capabilities of comparative feedback.

Our lower-bound analysis currently focuses on the linear transfer model; extending the corresponding argument to fixed nonlinear transfer functions is another direction for investigation. We also conjecture that ARC-DUELLI can be extended to more general compact metric spaces admitting suitable hierarchical coverings, although a rigorous proof remains to be developed. Finally, improving computational scalability in high-dimensional action spaces and adapting to unknown reward smoothness parameters are natural directions for future research.

Acknowledgment

This work is supported by the Science and Engineering Research Board (SERB) under Grant No. CRG/2023/002142. Shweta Jain acknowledges the support from Google India Research award. Ganesh Ghalme acknowledges support from Intel India under the AI Experience Development program.

References

Ailon, N., Karnin, Z., and Joachims, T. Reducing dueling bandits to cardinal bandits. In *International Conference on Machine Learning*, pp. 856–864. PMLR, 2014.

Benavoli, A., Azzimonti, D., and Piga, D. Preferential bayesian optimisation with skew gaussian processes. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pp. 1842–1850, 2021.

Bengs, V., Busa-Fekete, R., El Mesaoudi-Paul, A., and Hüllermeier, E. Preference-based online learning with dueling bandits: A survey. *Journal of Machine Learning Research*, 22(7):1–108, 2021.

Bubeck, S., Munos, R., Stoltz, G., and Szepesvári, C. X-armed bandits. *Journal of Machine Learning Research*, 12(5), 2011.

Chen, B. and Frazier, P. I. Dueling bandits with weak regret. In *International Conference on Machine Learning*, pp. 731–739. PMLR, 2017.

Feng, Y., Huang, Z., and Wang, T. Lipschitz bandits with batched feedback. *Advances in Neural Information Processing Systems*, 35:19836–19848, 2022.

González, J., Dai, Z., Damianou, A., and Lawrence, N. D. Preferential bayesian optimization. In *International Conference on Machine Learning*, pp. 1282–1291. PMLR, 2017.

Jamieson, K. G., Nowak, R., and Recht, B. Query complexity of derivative-free optimization. *Advances in Neural Information Processing Systems*, 25, 2012.

Kang, Y., Hsieh, C.-J., and Lee, T. C. M. Robust lipschitz bandits to adversarial corruptions. *Advances in Neural Information Processing Systems*, 36:10897–10908, 2023.

Kayal, A., Vakili, S., Toni, L., Shiu, D.-s., and Bernacchia, A. Bayesian optimization from human feedback: Near-optimal regret bounds. *arXiv preprint arXiv:2505.23673*, 2025.

Kleinberg, R. Nearly tight bounds for the continuum-armed bandit problem. In *Proceedings of the 18th International Conference on Neural Information Processing Systems*, NIPS'04, pp. 697–704. MIT Press, 2004.

Kleinberg, R., Slivkins, A., and Upfal, E. Multi-armed bandits in metric spaces. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pp. 681–690, 2008.

Kleinberg, R., Slivkins, A., and Upfal, E. Bandits and experts in metric spaces. *Journal of the ACM (JACM)*, 66(4):1–77, 2019.

Komiyama, J., Honda, J., Kashima, H., and Nakagawa, H. Regret lower bound and optimal algorithm in dueling bandit problem. In *Conference on learning theory*, pp. 1141–1154. PMLR, 2015.

Kumagai, W. Regret analysis for continuous dueling bandit. *Advances in Neural Information Processing Systems*, 30, 2017.

Lazzaro, J., Buffelli, D., Shiu, D.-s., and Vakili, S. A finite time analysis of thompson sampling for bayesian optimization with preferential feedback. *arXiv preprint arXiv:2604.25025*, 2026.

Pásztor, B., Kassraie, P., and Krause, A. Bandits with preference feedback: A stackelberg game perspective. *Advances in Neural Information Processing Systems*, 37:11997–12034, 2024.

Saha, A., Koren, T., and Mansour, Y. Dueling convex optimization. In *International Conference on Machine Learning*, pp. 9245–9254. PMLR, 2021.

Saha, A., Koren, T., and Mansour, Y. Dueling convex optimization with general preferences. In *International Conference on Machine Learning*, pp. 52552–52564. PMLR, 2025.

Takeno, S., Nomura, M., and Karasuyama, M. Towards practical preferential bayesian optimization with skew gaussian processes. In *International Conference on Machine Learning*, pp. 33516–33533. PMLR, 2023.

Urvoy, T., Clerot, F., Féraud, R., and Naamane, S. Generic exploration and k-armed voting bandits. In *International conference on machine learning*, pp. 91–99. PMLR, 2013.

Xu, W., Wang, W., Jiang, Y., Svetozarevic, B., and Jones, C. N. Principled preferential bayesian optimization. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Xu, Y., Joshi, A., Singh, A., and Dubrawski, A. Zeroth order non-convex optimization with dueling-choice bandits. In *Conference on Uncertainty in Artificial Intelligence*, pp. 899–908. PMLR, 2020.

Yue, Y. and Joachims, T. Interactively optimizing information retrieval systems as a dueling bandits problem. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 1201–1208, 2009.

Yue, Y. and Joachims, T. Beat the mean bandit. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 241–248, 2011.

Yue, Y., Broder, J., Kleinberg, R., and Joachims, T. The k-armed dueling bandits problem. *Journal of Computer and System Sciences*, 78(5):1538–1556, 2012.

Zhang, C. and Li, T. Comparisons are all you need for optimizing smooth functions. *arXiv preprint arXiv:2405.11454*, 2024.

Zhu, X. and Huang, Z. Lipschitz bandits in optimal space. In *The Thirteenth International Conference on Learning Representations*, 2025.

Zoghi, M., Whiteson, S., Munos, R., and de Rijke, M. Relative upper confidence bound for the k-armed dueling bandit problem. In *International conference on machine learning*, pp. 10–18. PMLR, 2014.

Appendix

A Notation used in the paper

Table 1: Notation used throughout the paper

Symbol	Meaning
	Problem Setup
$\mathcal{X}$	Action space, $\mathcal{X} \subseteq \mathbb{R}^d$
d	Dimension of action space
$\|\cdot\|$	Metric on $\mathcal{X}$ (typically ℓ_∞)
$f(x)$	Unknown reward / utility function
$\rho(\cdot)$	Transfer function
Γ	Lipschitz constant of ρ
$P(x \succ y)$	Probability that arm x is preferred over arm y
$\Delta(x, y)$	Preference gap, $P(x \succ y) - \frac{1}{2}$
x^*	Condorcet winner / optimal arm
T	Time horizon
	Algorithmic Quantities
m	Round index
B	Maximum depth / total rounds
r_h	Region radius (edge length) at level h
$\mathcal{C}_h^m$	Regions at level h in round m
$\mathcal{A}_h^m$	Active regions at level h in round m
d_z	Zooming dimension of near-optimal set
X^m	Reference arm at round m
$\tilde{\Delta}^m$	Estimated best gap in round m
n_h	Samples per region at level h

B Missing Proofs for Regret Upper Bound

B.1 Proof of Lemma 5.1

Lemma 5.1. *[Uniform concentration for cube-level preference estimates] Fix a round m, depth h, and a cube $C \in \mathcal{A}_h^m$ of diameter at most r_h. Let $Y \in \mathcal{X}$ be the reference arm fixed before that sampling block begins and let $\{x_i\}_{i=1}^{n_h}$ be i.i.d. samples drawn uniformly from C. Define the empirical estimator $\widehat{\Delta}_h^m(C, Y)$ and event $\mathcal{E}$ as:*

$$\widehat{\Delta}_h^m(C, Y) \;:=\; \frac{1}{n_h} \sum_{i=1}^{n_h} o(x_i, Y) - \frac{1}{2}, \tag{2}$$

$$\mathcal{E} := \left\{ \left|\Delta(x, Y) - \widehat{\Delta}_h^m(C, Y)\right| \le \sqrt{\frac{16 \log(T/\delta)}{n_h}} + \Gamma r_h \right\}, \tag{3}$$

for all $1 \le h \le m \le B$, all $C \in \mathcal{A}_h^m$, and all $x \in C$. Then, $\mathbb{P}(\mathcal{E}) \;\ge\; 1 - 2\delta$. Here, $o(x_i, Y) \sim Ber(P(x_i \succ Y))$.

Proof. Fix (h, m, C) and an arbitrary $x \in C$. We decompose the estimation error by adding and subtracting the expectation:

$$\begin{aligned}
&\left|\Delta(x, Y) - \widehat{\Delta}_h^m(C, Y)\right| \\
&\leq \left|\widehat{\Delta}_h^m(C, Y) - \mathbb{E}\widehat{\Delta}_h^m(C, Y)\right| \\
&\quad + \left|\mathbb{E}\widehat{\Delta}_h^m(C, Y) - \Delta(x, Y)\right|. \qquad (4)
\end{aligned}$$

Conditional on Y and C, the random variables $o(x_i, Y)$ are i.i.d. Bernoulli and bounded in $[0, 1]$. Therefore, by Hoeffding's inequality,

$$\begin{aligned}
&\mathbb{P}\left(\left|\widehat{\Delta}_h^m(C, Y) - \mathbb{E}\hat{\Delta}_h^m(C, Y)\right| \geq \sqrt{\frac{16\log(T/\delta)}{n_h}}\right) \qquad (5) \\
&\leq \frac{2\delta}{T^8}.
\end{aligned}$$

By definition of the estimator,

$$\mathbb{E}\widehat{\Delta}_h^m(C, Y) = \frac{1}{n_h}\sum_{i=1}^{n_h}\Delta(x_i, Y).$$

Hence,

$$\begin{aligned}
&\left|\Delta(x, Y) - \mathbb{E}\widehat{\Delta}_h^m(C, Y)\right| \\
&\quad = \left|\Delta(x, Y) - \frac{1}{n_h}\sum_{i=1}^{n_h}\Delta(x_i, Y)\right|. \qquad (6)
\end{aligned}$$

Applying definition of pairwise preference gap, triangle inequality and Assumption 3.2,

$$\begin{aligned}
&\leq \frac{1}{n_h}\sum_{i=1}^{n_h}\left|\rho(f(x) - f(Y)) - \rho(f(x_i) - f(Y))\right| \qquad (7) \\
&\leq \frac{\Gamma}{n_h}\sum_{i=1}^{n_h}\left|f(x) - f(x_i)\right|. \qquad (8)
\end{aligned}$$

Since f is 1-Lipschitz and $\|x - x_i\| \leq r_h$ for all $x, x_i \in C$,

$$\left|f(x) - f(x_i)\right| \leq \|x - x_i\| \leq r_h. \qquad (9)$$

Substituting into (7),

$$\left|\Delta(x, Y) - \mathbb{E}\widehat{\Delta}_h^m(C, Y)\right| \leq \Gamma r_h \qquad (10)$$

Combining (5) and (10) with (4), we obtain that for fixed (h, m, C, x),

$$\begin{aligned}
&\mathbb{P}\Bigg(\left|\Delta(x, Y) - \widehat{\Delta}_h^m(C, Y)\right| \\
&\qquad \geq \sqrt{\frac{16\log(T/\delta)}{n_h}} + \Gamma r_h\Bigg) \;\leq\; \frac{2\delta}{T^8}.
\end{aligned}$$

Taking a union bound over all $1 \leq h \leq m \leq B$, all $C \in \mathcal{A}_h^m$, and the finite set of sampled arms in each cube, and noting that the total number of such events is polynomial in T, yields

$$\mathbb{P}(\mathcal{E}) \;\geq\; 1 - 2\delta, \qquad (11)$$

as claimed.

□

B.2 Proof of Lemma 5.2

Lemma 5.2. *[Optimal arm is never eliminated] Suppose that the algorithm completes $M \le B$ rounds. Under the event $\mathcal{E}$ defined in Lemma 5.1, an optimal arm $x^\star \in \arg\max_{x\in\mathcal{X}} f(x)$ is not eliminated in any of the completed rounds $m \in [M]$. Consequently, the cube containing $x^\star$ never gets eliminated.*

Proof. Suppose that the algorithm completes $M \le B$ rounds. The claim is vacuous for $m = 1$, since there is no depth $h \in \{1, \dots, m-1\}$. Fix an arbitrary completed round $m \in \{2, \dots, M\}$. For each depth $h \in \{1, \dots, m-1\}$, let $(C_h^m)^\star \in \mathcal{A}_h^m$ denote a cube at depth h that contains some Condorcet winner $x^\star$. We show that $(C_h^m)^\star$ is not eliminated at depth h.

Let $C \in \mathcal{A}_{m-1}^{m-1}$ be the cube from which reference arm X^{m-1} was selected at depth $m-1$, i.e., C_{max}^{m-1}. Then, under the event $\mathcal{E}$, Lemma 5.1 implies that for any sampled arm $x_C \in C$ and for $x^\star \in (C_h^m)^\star$,

$$\tilde{\Delta}^{m-1} \le \Delta(x_C, X^{m-1}) + \sqrt{\frac{16\log(T/\delta)}{n_{m-1}}} + \Gamma r_{m-1} \tag{12}$$

$$\le \Delta(x_C, X^{m-1}) + \sqrt{\frac{16\log(T/\delta)}{n_h}} + \Gamma r_h \tag{13}$$

$$\widehat{\Delta}_h^m\big((C_h^m)^\star\big) \ge \Delta(x^\star, X^{m-1}) - \sqrt{\frac{16\log(T/\delta)}{n_h}} - \Gamma r_h \tag{14}$$

Subtracting (14) from (13) yields

$$\tilde{\Delta}^{m-1} - \widehat{\Delta}_h^m\big((C_h^m)^\star\big) \le \Delta(x_C, X^{m-1}) - \Delta(x^\star, X^{m-1}) + 2\sqrt{\frac{16\log(T/\delta)}{n_h}} + 2\,\Gamma r_h \tag{15}$$

Since $x^\star$ maximizes f and the preference model $P(x \succ y) = \rho(f(x) - f(y))$ is monotone in $f(x)$, implying

$$\Delta(x_C, X^{m-1}) - \Delta(x^\star, X^{m-1}) \le 0. \tag{16}$$

Combining (15) and (16), we obtain

$$\tilde{\Delta}^{m-1} - \widehat{\Delta}_h^m\big((C_h^m)^\star\big) \le 2\sqrt{\frac{16\log(T/\delta)}{n_h}} + 2\Gamma r_h \tag{17}$$

Choosing

$$n_h = \frac{16\log(T/\delta)}{r_h^2}, \tag{18}$$

we have

$$2\sqrt{\frac{16\log(T/\delta)}{n_h}} + 2\Gamma r_h = 2r_h + 2\Gamma r_h \le 2\,r_h(1+\Gamma) \tag{19}$$

Therefore, the elimination condition is not satisfied for $(C_h^m)^\star$ at depth h, and the cube containing $x^\star$ is not eliminated.

Since the argument holds for all depths $1 \le h \le m-1$ and all completed rounds $m \le M$, the optimal arm $x^\star$ is never eliminated during the M completed rounds. □

B.3 Proof of Lemma 5.3

Lemma 5.3. *[Shrinking-region property for dueling bandits] Suppose that the algorithm completes $M \le B$ rounds. Under the event $\mathcal{E}$, for any $1 \le h \le m \le M$, any cube $C \in \mathcal{A}_h^m$, and any arm $x \in C$, the Condorcet gap satisfies $\Delta(x^\star, x) \le \kappa(2\kappa+4)(1+\Gamma)r_{h-1}$, with $\kappa = \frac{\Gamma}{\lambda}$.*

Proof. Suppose that the algorithm completes $M \le B$ rounds. Fix an arbitrary completed round $m \in [M]$. The case $h = 1$ is trivial, since $\Delta(\cdot,\cdot) \in [-\frac{1}{2}, \frac{1}{2}]$, so the claim holds.

Now fix $h \ge 2$. Let $(C_{m-1}^{m-1})^\star \in \mathcal{A}_{m-1}^{m-1}$ denote the unique cube at depth $m-1$ containing the Condorcet winner $x^\star$. Under event $\mathcal{E}$, this cube exists and is not eliminated by Lemma 5.2.

Let $C \in \mathcal{A}_h^m$ and $x \in C$. Denote by $C_{\mathrm{par}} \in \mathcal{A}_{h-1}^m$ the parent cube of C at depth $h-1$. Clearly, $x \in C_{\mathrm{par}}$.

Introducing the reference arm X^{m-1} by Lemma 3.3, if $f(x^\star) \ge f(X^{m-1}) \ge f(x)$ or if $f(x^\star) \ge f(x) \ge f(X^{m-1})$, then we have:

$$\begin{aligned}\Delta(x^\star, x) &\le \frac{\Gamma}{\lambda}\left(\Delta(x^\star, X^{m-1}) + \Delta(X^{m-1}, x)\right) \\ &\le \frac{\Gamma}{\lambda}\left(\Delta(x^\star, X^{m-1}) - \Delta(x, X^{m-1})\right). \end{aligned} \tag{20}$$

Under event $\mathcal{E}$ and Lemma 5.1, we have the following bounds.

Optimal cube:

$$\begin{aligned}\Delta(x^\star, X^{m-1}) &\le \frac{\Gamma}{\lambda}\left(\Delta(x^\star, X^{m-2}) - \Delta(X^{m-1}, X^{m-2})\right) \\ &\le \frac{\Gamma}{\lambda}\left(\widehat{\Delta}_{m-1}^{m-1}\big((C_{m-1}^{m-1})^\star, X^{m-2}\big) - \widehat{\Delta}_{m-1}^{m-1}(C_{max}^{m-1}, X^{m-2}) + 2\sqrt{\frac{16\log(T/\delta)}{n_{m-1}}} + 2\Gamma r_{m-1}\right) \\ &\le \frac{\Gamma}{\lambda}\left(2\sqrt{\frac{16\log(T/\delta)}{n_{m-1}}} + 2\Gamma r_{m-1}\right) \end{aligned} \tag{21}$$

Parent cube:

$$\begin{aligned}\Delta(x, X^{m-1}) &\ge \widehat{\Delta}_{h-1}^m(C_{\mathrm{par}}, X^{m-1}) - \sqrt{\frac{16\log(T/\delta)}{n_{h-1}}} - \Gamma r_{h-1} \\ &\ge \tilde{\Delta}^{m-1} - 2r_{h-1}(1+\Gamma) - \sqrt{\frac{16\log(T/\delta)}{n_{h-1}}} - \Gamma r_{h-1} \end{aligned} \tag{22}$$

Now,

$$\tilde{\Delta}^{m-1} \ge \Delta(X^{m-1}, X^{m-1}) - \sqrt{\frac{16\log(T/\delta)}{n_{m-1}}} - \Gamma r_{m-1} \tag{23}$$

By the choice $n_h = \frac{16\log(T/\delta)}{r_h^2}$, we have $\sqrt{16\log(T/\delta)/n_h} = r_h$. Moreover, since $h \le m$, the radius sequence is nonincreasing as $r_h = 2^{-h+1}$, so $r_{m-1} \le r_{h-1}$. Along with these facts and substituting (21), (22), and (23) into (20) yields

$$\Delta(x^\star, x) \le \frac{\Gamma}{\lambda} r_{h-1}\left(\frac{2\Gamma}{\lambda}(1+\Gamma) + 4(1+\Gamma)\right) = \kappa(1+\Gamma)(2\kappa+4)r_{h-1}. \tag{24}$$

Here $\kappa = \frac{\Gamma}{\lambda}$

Since the argument holds for every depth $1 \le h \le m$ and every completed round $m \in [M]$, the stated shrinking-region property holds throughout all M completed rounds. If the algorithm stops during round $M+1$, the partial updates from that incomplete round are discarded, and hence the property obtained after round M remains unchanged.

□

B.4 Proof of Theorem 5.4

Theorem 5.4. *[Regret bound] Assume that the near-optimal set $\mathcal{S}(\varepsilon) := \{x \in \mathcal{X} : \Delta(x^\star, x) \le \varepsilon\}$ has zooming dimension d_z under the metric $\|\cdot\|$. Then, under the event $\mathcal{E}$, the regret of the* ARC-DUELLI *satisfies*

$$R(T) = \tilde{O}\Big(T^{\frac{d_z+1}{d_z+2}}\Big),$$

by setting $B = \left\lceil \frac{\log_2 T}{d_z+2} \right\rceil + 1$. *The bound holds even if the algorithm completes only* $M < B$ *rounds and stops during round* $M+1$ *due to insufficient remaining time. Here,* $\tilde{O}(\cdot)$ *hides logarithmic factors in* T *and* $1/\delta$. *In particular, for* $T \geq 2$, *running* ARC-DUELLI *with* $\delta = 1/T$ *yields the expected regret bound* $\mathbb{E}[R(T)] = \tilde{O}\left(T^{\frac{d_z+1}{d_z+2}}\right)$.

Proof. Fix a positive integer B, to be specified later, and let $M \leq B$ denote the number of rounds completed by the algorithm. Let R_m denote the regret incurred during a completed round m. If $M < B$, let R_{M+1}^{part} denote the regret incurred during the incomplete round $M+1$. Finally, let R_{rem} denote the regret incurred after exploration stops and the algorithm repeatedly plays the last completed reference arm. Thus,

$$R(T) = \sum_{m=1}^{M} R_m + \mathbb{1}\{M < B\} R_{M+1}^{\mathrm{part}} + R_{\mathrm{rem}}. \tag{25}$$

We first bound the regret incurred during the completed rounds. Fix a completed round $m \leq M$. During round m, the algorithm considers depths $h = 1, \ldots, m$. For every cube $C \in \mathcal{A}_h^m$, it performs n_h duels between an arm $x_{C,i}^m \in C$ and the reference arm X^{m-1}. In addition, n_m duels are performed to compute the new reference arm. Therefore,

$$R_m = \underbrace{\sum_{h=1}^{m} \sum_{C \in \mathcal{A}_h^m} \sum_{i=1}^{n_h} \left(\Delta(x^\star, x_{C,i}^m) + \Delta(x^\star, X^{m-1}) \right)}_{\text{Regret from all cubes in one round}} + \underbrace{\sum_{j=1}^{n_m} \left(\Delta(x^\star, z_j^m) + \Delta(x^\star, X^m) \right)}_{\text{Regret while selecting } \tilde{\Delta}^m}. \tag{26}$$

where $z_j^m \in C_{\max}^m$ is the arm sampled in the jth additional duel. For convenience, define

$$K := \kappa(2\kappa+4)(1+\Gamma) \qquad \text{and} \qquad L_T := \log(T/\delta).$$

By Lemma 5.3, under the event $\mathcal{E}$, any arm belonging to a surviving cube at depth h satisfies

$$\Delta(x^\star, x_{C,i}^m) \leq K r_{h-1}.$$

The reference arm X^{m-1} was selected at the end of round $m-1$ at height $m-1$, under lemma 5.3 satisfies

$$\Delta(x^\star, X^{m-1}) \leq K r_{m-2}.$$

Since $h \leq m$, we have

$$r_{m-2} \leq 2 r_{h-1},$$

and therefore

$$\Delta(x^\star, X^{m-1}) \leq 2K r_{h-1}.$$

Consequently, each duel performed at depth h incurs regret at most

$$\begin{aligned} \Delta(x^\star, x_{C,i}^m) + \Delta(x^\star, X^{m-1}) &\leq 3K r_{h-1} \\ &= 6K r_h, \end{aligned} \tag{27}$$

where we used $r_{h-1} = 2r_h$.

Since $n_h = \frac{16 L_T}{r_h^2}$, the regret contributed by a single cube at depth h is at most $n_h \cdot 6K r_h = \frac{96 K L_T}{r_h}$. Therefore,

$$R_m \leq \sum_{h=1}^{m} |\mathcal{A}_h^m| \frac{96 K L_T}{r_h} + A_m, \tag{28}$$

where A_m denotes the regret incurred by the additional n_m duels used to compute $\tilde{\Delta}^m$.

We now calculate A_m. By Lemma 5.3,

$$\Delta(x^\star, X^m) \le K r_{m-1}, \quad \Delta(x^\star, z_j^m) \le K r_{m-1}$$

Hence,

$$\begin{aligned} A_m &:= \sum_{j=1}^{n_m} \Big(\Delta(x^\star, z_j^m) + \Delta(x^\star, X^m) \Big) \\ &\le 2 n_m K r_{m-1} \\ &= \frac{32 L_T}{r_m^2} K r_{m-1} \\ &= \frac{64 K L_T}{r_m}, \end{aligned} \tag{29}$$

where the last equality follows from $r_{m-1} = 2 r_m$.

We next bound the number of surviving cubes. By Lemma 5.3, every cube in $\mathcal{A}_h^m$ is contained in

$$\{x \in \mathcal{X} : \Delta(x^\star, x) \le K r_{h-1}\} = \mathcal{S}(2 K r_h).$$

Each cube in $\mathcal{A}_h^m$ contains an ℓ_∞-ball of radius $r_h/2$, and these balls have disjoint interiors. Thus, $\mathcal{A}_h^m$ induces an $(r_h/2)$-packing of $\mathcal{S}(2Kr_h)$.

By the definition of the zooming dimension, the set $\mathcal{S}(2Kr_h)$ can be covered by at most

$$C_z (2 K r_h)^{-d_z}$$

balls of radius $2Kr_h$. Moreover, each such ball intersects at most $(8K+2)^d$ disjoint ℓ_∞-balls of radius $r_h/2$. Therefore, defining

$$\widetilde{C}_z := C_z (2K)^{-d_z} (8K+2)^d,$$

we obtain

$$|\mathcal{A}_h^m| \le \widetilde{C}_z r_h^{-d_z} = \widetilde{C}_z 2^{(h-1) d_z}. \tag{30}$$

By increasing $\widetilde{C}_z$ if necessary, we may assume that $\widetilde{C}_z \ge 1$.

Substituting (30) into (28) gives

$$R_m \le 96 K \widetilde{C}_z L_T \sum_{h=1}^{m} 2^{(h-1)(d_z+1)} + A_m. \tag{31}$$

We now sum over all completed rounds $m \le M$. Exchanging the order of summation yields

$$\begin{aligned} \sum_{m=1}^{M} R_m &\le 96 K \widetilde{C}_z L_T \sum_{m=1}^{M} \sum_{h=1}^{m} 2^{(h-1)(d_z+1)} + \sum_{m=1}^{M} A_m \\ &= 96 K \widetilde{C}_z L_T \sum_{h=1}^{M} (M-h+1) 2^{(h-1)(d_z+1)} + \sum_{m=1}^{M} A_m. \end{aligned} \tag{32}$$

Let

$$q := 2^{d_z+1}.$$

Since $d_z \ge 0$, we have $q \ge 2$. Thus,

$$\begin{aligned} \sum_{h=1}^{M} (M-h+1) q^{h-1} &= q^{M-1} \sum_{j=0}^{M-1} (j+1) q^{-j} \\ &\le q^{M-1} \sum_{j=0}^{\infty} (j+1) q^{-j} \\ &= \frac{q^{M-1}}{(1-q^{-1})^2} \\ &\le 4 q^{M-1}. \end{aligned} \tag{33}$$

Moreover, using $r_m = 2^{-m+1}$ and (29),

$$\begin{aligned}\sum_{m=1}^{M} A_m &\le 64KL_T \sum_{m=1}^{M} \frac{1}{r_m} \\ &= 64KL_T \sum_{m=1}^{M} 2^{m-1} \\ &= 64KL_T(2^M - 1) \\ &\le 128K\widetilde{C}_z L_T q^{M-1}. \end{aligned} \tag{34}$$

Here we used $\widetilde{C}_z \ge 1$ and $2^{M-1} \le q^{M-1}$.

Combining (32), (33), and (34), we obtain

$$\sum_{m=1}^{M} R_m \le C_0 K \widetilde{C}_z L_T 2^{(M-1)(d_z+1)} \tag{35}$$

for some universal constant $C_0 > 0$.

We now consider separately the cases $M = B$ and $M < B$.

Case 1: $M = B$**.** In this case, all B rounds are completed. By Lemma 5.3, the final reference arm satisfies

$$\Delta(x^\star, X^B) \le K r_{B-1}.$$

Therefore, repeatedly playing X^B during the remaining time steps incurs regret at most

$$\begin{aligned} R_{\mathrm{rem}} &\le 2K r_{B-1} T \\ &= 4K 2^{-B+1} T. \end{aligned} \tag{36}$$

Combining (35) and (36), we obtain

$$\begin{aligned} R(T) \le C_0 K \widetilde{C}_z L_T 2^{(B-1)(d_z+1)} \\ + 4K 2^{-B+1} T. \end{aligned} \tag{37}$$

Case 2: $M < B$**.** In this case, the algorithm stops during round $M+1$ because insufficient time remains to complete the next sampling block. Although the partial updates from round $M+1$ are discarded, the regret incurred by the duels already performed in this round must still be counted.

Every sampling block that is started during round $M+1$ is completed before the algorithm proceeds to the next block. Furthermore, the incomplete round visits only a subset of the cubes that could be visited in a complete round $M+1$. The argument of Lemma 5.3 therefore applies to all cubes visited before termination. Consequently, the same argument used in (31) gives

$$R_{M+1}^{\mathrm{part}} \le 96K\widetilde{C}_z L_T \sum_{h=1}^{M+1} 2^{(h-1)(d_z+1)} + A_{M+1}. \tag{38}$$

Since $q = 2^{d_z+1} \ge 2$,

$$\sum_{h=1}^{M+1} q^{h-1} = \frac{q^{M+1} - 1}{q - 1} \le 2q^M.$$

Also,

$$\begin{aligned} A_{M+1} &\le \frac{64KL_T}{r_{M+1}} \\ &= 64KL_T 2^M \\ &\le 64K\widetilde{C}_z L_T q^M. \end{aligned} \tag{39}$$

Thus,

$$R_{M+1}^{\mathrm{part}} \le C_1 K \widetilde{C}_z L_T 2^{M(d_z+1)} \tag{40}$$

for some universal constant $C_1 > 0$.

Combining (35) and (40), we obtain

$$\sum_{m=1}^{M} R_m + R_{M+1}^{\mathrm{part}} \le C_2 K \widetilde{C}_z L_T 2^{M(d_z+1)} \tag{41}$$

for some universal constant $C_2 > 0$.

Let τ denote the number of time steps remaining when the algorithm stops. Suppose that the algorithm stops before starting a sampling block of size n_h during round $M+1$. By the stopping condition,

$$\tau < n_h.$$

Since $h \le M+1$ and n_h is nondecreasing in h,

$$\tau < n_h \le n_{M+1}. \tag{42}$$

By Lemma 5.3, the reference arm X^M obtained from the last completed round satisfies

$$\Delta(x^\star, X^M) \le K r_{M-1}. \tag{43}$$

Therefore, repeatedly playing X^M during the remaining τ steps incurs regret at most

$$\begin{aligned} R_{\mathrm{rem}} &\le 2K r_{M-1} \tau \\ &< 2K r_{M-1} n_{M+1}. \end{aligned} \tag{44}$$

Using

$$n_{M+1} = \frac{16 L_T}{r_{M+1}^2}, \qquad r_{M-1} = 4 r_{M+1}, \qquad r_{M+1} = 2^{-M},$$

we obtain

$$\begin{aligned} R_{\mathrm{rem}} &< 32 K L_T \frac{r_{M-1}}{r_{M+1}^2} \\ &= 128 K L_T \frac{1}{r_{M+1}} \\ &= 128 K L_T 2^M. \end{aligned} \tag{45}$$

Since $d_z \ge 0$,

$$2^M \le 2^{M(d_z+1)}.$$

Combining (41) and (45) gives

$$R(T) \le C_3 K \widetilde{C}_z L_T 2^{M(d_z+1)} \tag{46}$$

for some universal constant $C_3 > 0$.

Finally, choose

$$B^\star = \left\lceil \frac{\log_2 T}{d_z + 2} \right\rceil + 1. \tag{47}$$

When $M = B$, this choice balances the two terms in (37), giving

$$R(T) = \widetilde{O}\left(T^{\frac{d_z+1}{d_z+2}}\right).$$

When $M < B$, we have $M \le B - 1$, and hence

$$\begin{aligned} 2^{M(d_z+1)} &\le 2^{(B-1)(d_z+1)} \\ &= O\left(T^{\frac{d_z+1}{d_z+2}}\right). \end{aligned} \tag{48}$$

Thus, the same regret bound holds even if the algorithm stops before completing all B rounds:

$$R(T) = \widetilde{O}\left(T^{\frac{d_z+1}{d_z+2}}\right).$$

To obtain the expected-regret bound, let $U(T, \delta)$ denote the upper bound on $R(T)$ established above on $\mathcal{E}$. Each duel contributes at most 1 to regret, so $R(T) \leq T$ on every run. By Lemma 5.1, $\Pr(\mathcal{E}^c) \leq 2\delta$. Therefore,

$$\begin{aligned}\mathbb{E}[R(T)] &= \mathbb{E}[R(T)\mathbb{1}_{\mathcal{E}}] + \mathbb{E}[R(T)\mathbb{1}_{\mathcal{E}^c}] \\ &\leq U(T, \delta) + 2\delta T.\end{aligned}$$

For $T \geq 2$, running ARC-DUELLI with $\delta = 1/T$ makes the second term at most 2. Since $\log(T/\delta) = 2\log T$ for this choice, we obtain

$$\mathbb{E}[R(T)] = \widetilde{O}\Big(T^{\frac{d_z+1}{d_z+2}}\Big).$$

This completes the proof.

□

B.5 Proof of Lemma 5.5

Lemma 5.5. *[Space complexity]* ARC-DUELLI *requires at most $O(d \log T)$ bits of memory at any time.*

Proof. We account for all quantities stored by the algorithm at any time step. We measure space by the number of stored numerical coordinates, counters, and empirical statistics, with each occupying one storage unit.

Quantities maintained. Apart from the recursion stack, the algorithm maintains a constant number of global objects:

- The previous-round reference arm and benchmark: $(X^{m-1}, \tilde{\Delta}^{m-1})$.
- The current best cube and its empirical gap: $(C^m_{\max}, \Delta^m_{\max})$.
- A representative sampled arm from the current best cube, used to select the next reference arm X^m.
- The time counter t and round index m.

We next account for the space occupied by these objects and by the recursive exploration.

Representation of cubes. Each cube C is represented by its lower corner in $[0, 1]^d$ and its edge length. The lower corner has d coordinates, so storing one cube requires $O(d)$ space. This representation directly specifies the geometry of the cube.

Representation of arms. Each arm X^m is represented by its d coordinates. Therefore, storing one arm requires $O(d)$ space. Since only a constant number of arms and global cubes are maintained, their total storage is $O(d)$.

Empirical gaps and sampling. For binary duel outcomes $o_i \in \{0, 1\}$, the empirical gap is

$$\hat{\Delta} = \frac{1}{n}\sum_{i=1}^{n} o_i - \frac{1}{2}.$$

It can be computed by storing only the number of comparisons and a running win count. Thus, each empirical gap requires $O(1)$ space, and the individual observations need not be stored.

To select a uniformly random arm from a terminal cube's sampled arms, the algorithm uses reservoir sampling. Specifically, the first sampled arm is retained, and the ith sampled arm replaces the retained arm with probability $1/i$. At the end of the block, the retained arm is uniform among the sampled arms.

Whenever a terminal cube becomes the current empirical maximizer, its retained arm replaces the previous candidate. Therefore, the algorithm stores only a constant number of sampled arms, requiring $O(d)$ space.

Recursion and memory reuse. The recursive procedure explores the partition tree in depth-first order. Subcubes are generated sequentially and processed one at a time.

Each active recursive call stores:

- Its current cube C.
- Its depth h.
- The index of the next child to explore.
- A constant number of local counters and empirical statistics.

The current cube requires $O(d)$ space. The child index can be represented by d binary choices, one for each coordinate, also requiring at most $O(d)$ space. Thus, each active recursive call requires $O(d)$ space.

Since the maximum recursion depth is $B = O(\log T)$, there are at most $O(\log T)$ simultaneously active recursive calls. Hence, the recursion stack requires

$$O(dB) = O(d \log T)$$

space.

When a recursive call returns, its local storage is released. Consequently, the algorithm does not store the full partition tree, all subcubes simultaneously, or the history of comparisons.

Conclusion. The global quantities and sequential sampling require $O(d)$ space, while the recursion stack requires $O(d \log T)$ space. Therefore, the total working space is

$$O(d) + O(d \log T) = O(d \log T).$$

For fixed dimension d, this simplifies to $O(\log T)$. □

C Proof for Regret Lower Bound

C.1 Proof of Theorem 6.1

Theorem 6.1. *[Reduction] Let A be any dueling bandit algorithm with linear transfer function operating for T rounds. Then there exists a scalar-feedback bandit algorithm B_A operating for $2T$ rounds such that, for every instance μ, $\mathbb{E}_\mu\big[R^{\mathrm{sc}}_{B_A}(2T)\big] = 2\,\mathbb{E}_\mu\big[R^{\mathrm{duel}}_A(T)\big]$. Here, $R^{\mathrm{sc}}_{B_A}(2T)$ represents regret by algorithm B_A for scalar-feedback bandit algorithm and R^{duel}_A represent regret by algorithm A under duel-feedback bandit algorithm.*

Proof. The algorithm B_A internally runs A. Whenever A queries a pair (x_t, y_t), the algorithm B_A pulls x_t and y_t once each, obtaining independent samples

$$U_t \sim \mathrm{Ber}(\mu(x_t)), \qquad V_t \sim \mathrm{Ber}(\mu(y_t)).$$

It then generates a synthetic duel outcome

$$O_t = \begin{cases} 1, & U_t > V_t, \\ 0, & U_t < V_t, \\ C_t, & U_t = V_t, \end{cases}$$

where $C_t \sim \mathrm{Ber}(1/2)$ is independent, and feeds O_t back to the internal copy of A.

A direct calculation shows

$$\mathbb{P}(O_t = 1 \mid x_t, y_t) = \frac{1}{2}\big(1 + \mu(x_t) - \mu(y_t)\big),$$

so the synthetic duel has exactly the same distribution as a true comparison in the induced linear-link dueling instance. Hence the internal transcript of A inside B_A has the correct law.

Let $x^\star \in \arg\max_x \mu(x)$. Under the linear link,

$$\Delta(x^\star, x) = \frac{1}{2}\big(\mu(x^\star) - \mu(x)\big).$$

Therefore, at round t,

$$\Delta(x^\star, x_t) + \Delta(x^\star, y_t) = \frac{1}{2}\big(\mu(x^\star) - \mu(x_t)\big) + \frac{1}{2}\big(\mu(x^\star) - \mu(y_t)\big).$$

The two scalar pulls used by B_A at that step incur regret

$$\big(\mu(x^\star) - \mu(x_t)\big) + \big(\mu(x^\star) - \mu(y_t)\big),$$

which is exactly twice the dueling regret. Summing over $t = 1, \dots, T$ gives

$$R^{\mathrm{sc}}_{B_A}(2T) = 2\, R^{\mathrm{duel}}_A(T)$$

□

D Bi-Lipschitz continuity of transfer functions

Lemma D.1 (Assumption 3.2 logistic sigmoid)**.** *Let $\rho : [-1, 1] \to (0, 1)$ be the logistic sigmoid defined by*

$$\rho(z) := \frac{1}{1 + e^{-z}}. \tag{49}$$

Then $\rho(0) = \frac{1}{2}$, $\rho(z) = 1 - \rho(-z)$, and ρ is bi-Lipschitz with $\Gamma = \frac{1}{4}$ and $\lambda = \frac{e}{(1+e)^2}$.

Proof. $\rho(0) = \frac{1}{2}$, $\rho(z) = 1 - \rho(-z)$ are easy to verify. For bi-Lipschitz, we first compute the derivative of ρ. Differentiating yields

$$\rho'(z) = \frac{e^{-z}}{(1 + e^{-z})^2} = \frac{e^z}{(1 + e^z)^2} = \rho(z)\big(1 - \rho(z)\big). \tag{50}$$

Since $\rho(z) \in (0, 1)$ for all $z \in \mathbb{R}$, the function $u \mapsto u(1 - u)$ is maximized over $u \in [0, 1]$ at $u = 1/2$. Therefore,

$$\sup_{z \in \mathbb{R}} |\rho'(z)| = \rho(0)(1 - \rho(0)) = \frac{1}{4}. \tag{51}$$

Applying mean value theorem, we get that the logistic transfer is globally Lipschitz with upper constant

$$\Gamma = \frac{1}{4}.$$

The derivative $\rho'(z)$ is symmetric around zero and decreases as $|z|$ increases. Hence, over $[-1, 1]$, its minimum is attained at $z = \pm 1$. Thus,

$$\inf_{z \in [-1,1]} \rho'(z) = \rho'(1) = \rho'(-1) = \frac{e}{(1 + e)^2}. \tag{52}$$

Thus, we obtain

$$\frac{e}{(1 + e)^2}(a - b) \le \rho(a) - \rho(b) \le \frac{1}{4}(a - b).$$

Therefore, for all $-1 \le b \le a \le 1$, the logistic transfer satisfies

$$\lambda(a - b) \le \rho(a) - \rho(b) \le \Gamma(a - b),$$

with

$$\lambda = \frac{e}{(1 + e)^2}, \qquad \Gamma = \frac{1}{4}.$$

□

Lemma D.2 (Assumption 3.2 for the probit transfer)**.** *Let $\Phi : \mathbb{R} \to (0,1)$ denote the probit function, defined as the cumulative distribution function of a standard normal random variable,*

$$\Phi(z) := \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} e^{-t^2/2}\, dt. \tag{53}$$

Then $\Phi(0) = \frac{1}{2}$, $\Phi(z) = 1 - \Phi(-z)$, and Φ is bi-Lipschitz continuous with $\Gamma = \frac{1}{\sqrt{2\pi}}$ and $\lambda = \frac{1}{\sqrt{2\pi}} e^{-1/2}$.

Proof. The standard normal distribution is symmetric about zero, so $\Phi(0) = \frac{1}{2}$ and $\Phi(z) = 1 - \Phi(-z)$. Since Φ is differentiable on $\mathbb{R}$, its derivative is given by the standard normal density

$$\Phi'(z) = \frac{1}{\sqrt{2\pi}} e^{-z^2/2}, \qquad z \in \mathbb{R}. \tag{54}$$

The exponential term attains its maximum at $z = 0$. Therefore,

$$|\Phi'(z)| \le \frac{1}{\sqrt{2\pi}}, \qquad z \in \mathbb{R}. \tag{55}$$

Consider arbitrary $a, b \in \mathbb{R}$ with $a \neq b$. By the Mean Value Theorem, there exists a point ξ between a and b such that

$$\Phi(a) - \Phi(b) = \Phi'(\xi)(a - b). \tag{56}$$

Combining this identity with (55) gives

$$|\Phi(a) - \Phi(b)| = |\Phi'(\xi)|\, |a - b| \le \frac{1}{\sqrt{2\pi}}\, |a - b|.$$

This upper bound also holds when $a = b$.

We now prove the lower bound on $[-1, 1]$. Since $e^{-z^2/2}$ decreases as $|z|$ increases, the minimum of $\Phi'(z)$ on $[-1, 1]$ is attained at $z = \pm 1$. Hence,

$$\inf_{z \in [-1,1]} \Phi'(z) = \Phi'(1) = \Phi'(-1) = \frac{1}{\sqrt{2\pi}} e^{-1/2}. \tag{57}$$

Now fix $-1 \le b < a \le 1$. The point ξ in (56) lies in $(b, a) \subseteq [-1, 1]$. Since $a - b > 0$, we obtain

$$\Phi(a) - \Phi(b) = \Phi'(\xi)(a - b) \ge \frac{1}{\sqrt{2\pi}} e^{-1/2}(a - b). \tag{58}$$

The lower bound holds with equality when $a = b$. Thus, the probit transfer function is (λ, Γ)-bi-Lipschitz on $[-1, 1]$, with

$$\lambda = \frac{1}{\sqrt{2\pi}} e^{-1/2} \qquad \text{and} \qquad \Gamma = \frac{1}{\sqrt{2\pi}}.$$

□

Lemma D.3 (Assumption 3.2 for linear link)**.** *Let $\rho : \mathbb{R} \to \mathbb{R}$ be defined by*

$$\rho(z) := \tfrac{1}{2}(1 + z). \tag{59}$$

Then $\rho(0) = \frac{1}{2}$, $\rho(z) = 1 - \rho(-z)$, and ρ is bi-Lipschitz on $[-1, 1]$ with $\Gamma = \frac{1}{2}$ and $\lambda = \frac{1}{2}$.

Proof. It is easy to see $\rho(0) = \frac{1}{2}$ and $\rho(z) = 1 - \rho(-z)$. For bi-Lipschitz, fix arbitrary $a, b \in \mathbb{R}$. By direct computation,

$$\rho(a) - \rho(b) = \tfrac{1}{2}(1 + a) - \tfrac{1}{2}(1 + b) = \tfrac{1}{2}(a - b). \tag{60}$$

This gives the value of $\Gamma = \lambda = \frac{1}{2}$. □

E Proof of Lemma 3.3

Lemma 3.3. *Under Assumption 3.2, the preference gap Δ satisfies:*

1. *if $f(x) \geq f(z) \geq f(y)$, then $\Delta(x, y) \leq \kappa(\Delta(x, z) + \Delta(z, y))$,*
2. *if $f(x) \geq f(y) \geq f(z)$, then $\Delta(x, y) \leq \kappa(\Delta(x, z) - \Delta(y, z))$.*

Proof. When $f(x) \geq f(y) \geq f(z)$, we have:

$$\begin{aligned}
\Delta(x, y) &= \rho(f(x) - f(y)) - \rho(0) \geq \lambda(f(x) - f(y)) \\
\Delta(y, z) &= \rho(f(y) - f(z)) - \rho(0) \geq \lambda(f(y) - f(z)) \\
\Delta(x, z) &= \rho(f(x) - f(z)) - \rho(0) \leq \Gamma(f(x) - f(y)) + \Gamma(f(y) - f(z)) \\
&\leq \frac{\Gamma}{\lambda}(\Delta(x, y) + \Delta(y, z))
\end{aligned}$$

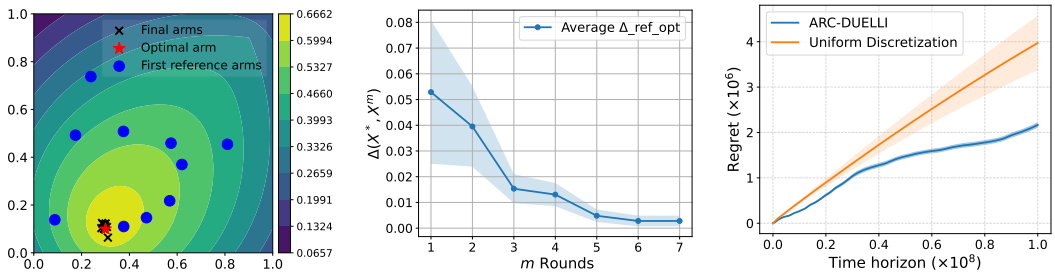
When $f(x) \geq f(z) \geq f(y)$, we have:

$$\begin{aligned}
\Delta(x, z) &= \rho(f(x) - f(z)) - \rho(0) \leq \Gamma(f(x) - f(z)) \\
&\leq \frac{\Gamma}{\lambda}[\rho(f(x) - f(y)) - \rho(f(z) - f(y))] \\
&\leq \frac{\Gamma}{\lambda}(\Delta(x, y) + \Delta(y, z))
\end{aligned}$$

Here, second last inequality follows from $\rho(f(x) - f(y)) - \rho(f(z) - f(y)) \geq \lambda(f(x) - f(z))$. □

F Additional Experimental Results

F.1 Experiments with Random Initial Points



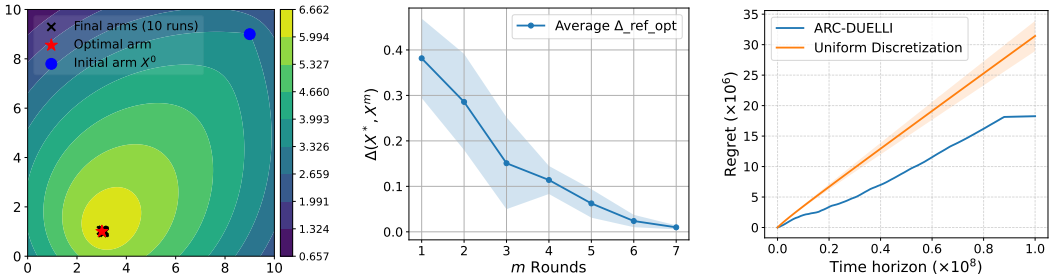
(a) Reference arm convergence (b) Convergence of $\Delta(x^*, X^m)$ (c) Regret comparison

Figure 2: Performance of ARC-DUELLI with randomly initialized reference arms over 10 runs. (a) Reference arms converge toward the optimal arm. (b) The optimality gap $\Delta(x^*, X^m)$ decreases across rounds. (c) Cumulative regret remains comparable to the fixed-initialization setting.

Even when initialized with a randomly chosen reference arm for the same reward function that we use in the main paper, ARC-DUELLI maintains strong performance. As shown in Figure 2a, the reference arms consistently converge toward the optimal arm across runs, indicating robustness to initialization. Figure 2b further shows that the optimality gap $\Delta(x^*, X^m)$ decreases steadily over rounds and approaches zero, demonstrating effective learning. Finally, Figure 2c shows that the cumulative regret remains low and continues to outperform the uniform discretization baseline. Overall, these results highlight that ARC-DUELLI is insensitive to the choice of initial reference arm and reliably converges to near-optimal solutions.

F.2 Experiments on Larger Domain with a Rescaled Utility Landscape.

We further evaluate ARC-DUELLI beyond the unit domain by considering a scaled action space $\mathcal{X} = [0, 10]^2$. The utility function is defined as $\mu(x) = 10 - \frac{2}{3}\|x - x_1\| - \frac{1}{3}\|x - x_2\|$, where



(a) Reference arm convergence (b) Convergence of $\Delta(x^*, X^m)$ (c) Regret comparison

Figure 3: Performance of ARC-DUELLI under scaled rewards in $[0, 10]$. (a) Reference arms converge toward the optimal arm despite the larger reward scale. (b) The gap $\Delta(x^\star, X^m)$ decreases across rounds, indicating consistent improvement in the reference arm. (c) Cumulative regret compared with a uniform discretization baseline, showing superior performance of ARC-DUELLI. The solid curves denote averages over runs, and the shaded regions represent ± 1 standard deviation.

$x_1 = (3.0, 1.0)$ and $x_2 = (9.0, 9.0)$. This rescaled utility landscape has a unique maximizer at $x^\star = (3.0, 1.0)$. The rescaling preserves the unit Lipschitz bound with respect to the norm used in the utility definition. By the reverse triangle inequality, each distance term is 1-Lipschitz, and the two weights sum to one. The larger domain allows utility differences to exceed the interval $[-1, 1]$, so the transfer function's slope bounds are evaluated over the expanded range of utility differences. For the logistic transfer function, the upper slope bound remains $\Gamma = 1/4$, while the lower slope bound λ decreases. The baseline discretization follows the same resolution rule as in the unit-domain experiment, applied to the scaled action space.

As shown in Figure 3a, the final reference arms concentrate near the optimal arm. Figure 3b shows that the optimality gap $\Delta(x^\star, X^m)$ decreases across rounds and becomes small by the final round. Figure 3c shows that ARC-DUELLI achieves lower cumulative regret than uniform discretization in the scaled setting.

Overall, these results show that ARC-DUELLI continues to identify near-optimal reference arms and outperforms the uniform-discretization baseline on the larger action space with a wider range of utility differences.

F.3 Experiments with different reward function

To demonstrate that ARC-DUELLI is robust to a wide range of utility landscapes, we evaluate it on six benchmark reward functions that differ significantly in the geometry around their optimum. These functions range from sharp unimodal objectives to flat plateaus, ring-shaped optima, and multimodal landscapes, thereby testing the algorithm under varying levels of optimization difficulty.

Flat ($d_z = \frac{4}{3}$). The Flat reward function is designed to be particularly challenging by introducing a large nearly-optimal region around the maximizer. Near the optimum at $(0.78, 0.22)$, the utility decreases cubically, producing very small reward differences between nearby actions. Consequently, distinguishing the optimal arm requires substantially more comparisons than in functions with steeper gradients.

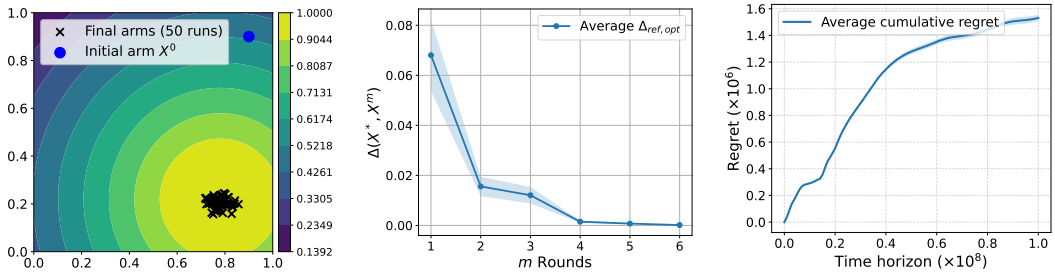
Let $d(x) = \|x - (0.78, 0.22)\|_2$ and $r = 0.22$. The utility is defined as

$$\mu(x) = \mathrm{clip}(1 - g(x), 0, 1),$$

where

$$g(x) = \begin{cases} \dfrac{0.9 d(x)^3}{3r^2}, & d(x) \le r, \\ 0.9\left(d(x) - \dfrac{2r}{3}\right), & \text{otherwise.} \end{cases}$$

The experimental results are shown in Fig. 4.



(a) Reference arm convergence (b) Convergence of $\Delta(x^*, X^m)$ (c) Cumulative Regret

Figure 4: Performance of ARC-DUELLI on the Flat reward function, averaged over 50 independent runs. From left to right: (a) evolution of the reference arm overlaid on the utility landscape, illustrating convergence toward the optimal region, (b) convergence of the optimality gap $\Delta(x^*, X^m)$ with the shaded region denoting one standard deviation, and (c) cumulative regret as a function of the number of comparisons.

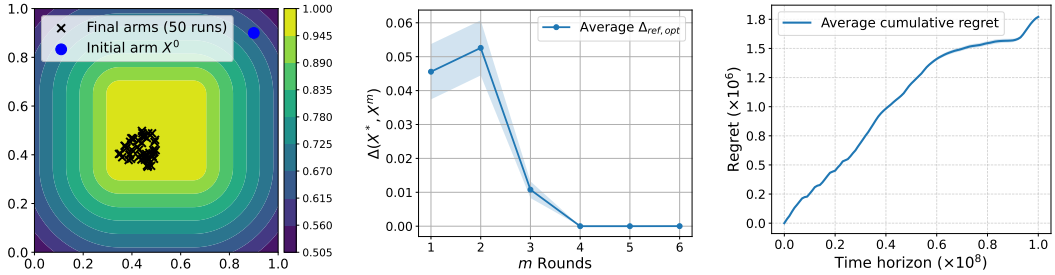
Plateau ($d_z = 2$). Unlike standard optimization benchmarks that possess a unique maximizer, the Plateau reward contains an entire square region of optimal actions. Since every point inside the plateau attains the maximum reward, the algorithm must identify the optimal region rather than converge to a single point. This benchmark evaluates whether the adaptive partitioning strategy can efficiently localize a flat optimum.

The utility function is

$$\mu(x) = \mathrm{clip}\big(1 - \mathrm{dist}(x, [0.35, 0.65]^2), 0, 1\big)\,,$$

where $\mathrm{dist}(x, S)$ denotes the Euclidean distance from x to the set S.

The corresponding experimental results are presented in Fig. 5.



(a) Reference arm convergence (b) Convergence of $\Delta(x^*, X^m)$ (c) Cumulative Regret

Figure 5: Performance of ARC-DUELLI on the Plateau reward function, averaged over 50 independent runs. From left to right: (a) evolution of the reference arm on the utility landscape, (b) convergence of the optimality gap $\Delta(x^*, X^m)$ with one standard deviation, and (c) cumulative regret over time.

Quadratic ($d_z = 1$). The Quadratic reward has a smooth quadratic basin around the optimum located at $(0.5, 0.5)$, followed by linear decay outside a small neighborhood. This function represents a well-conditioned optimization problem and serves as an intermediate benchmark between the Sharp and Flat landscapes.

Let

$$d(x) = \|x - (0.5, 0.5)\|_2, \qquad r = 0.12,$$

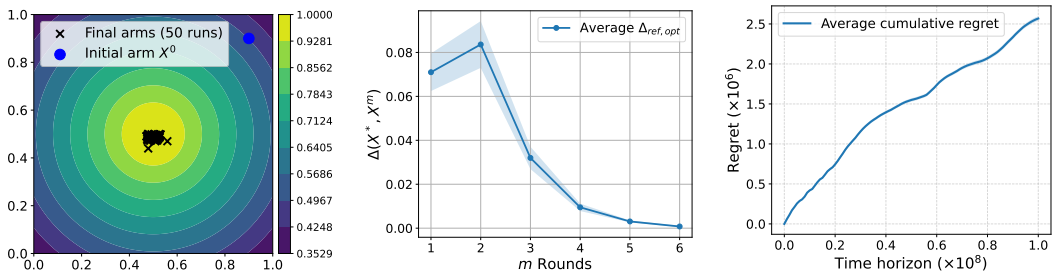
and define

$$g(x) = \begin{cases} \dfrac{d(x)^2}{2r}, & d(x) \le r, \\ d(x) - \dfrac{r}{2}, & \text{otherwise.} \end{cases}$$

The utility is

$$\mu(x) = \text{clip}(1 - g(x), 0, 1).$$

The experimental performance is illustrated in Fig. 6.



(a) Reference arm convergence (b) Convergence of $\Delta(x^*, X^m)$ (c) Cumulative Regret

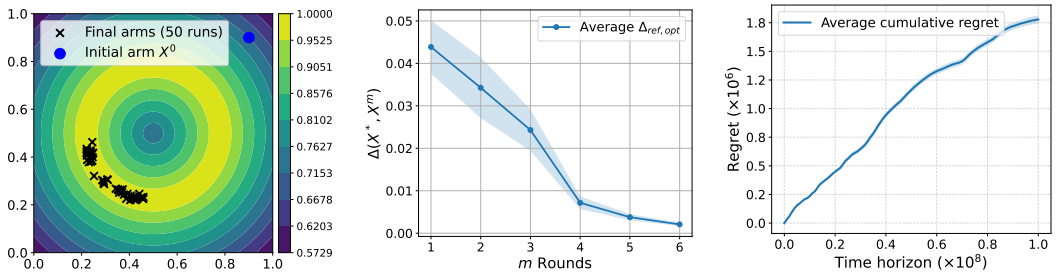
Figure 6: Performance of ARC-DUELLI on the Quadratic reward function, averaged over 50 independent runs. From left to right: (a) convergence of the reference arm on the utility landscape, (b) convergence of the optimality gap $\Delta(x^*, X^m)$ with one standard deviation, and (c) cumulative regret as the learning process progresses.

Ridge ($d_z = 1$)**.** The Ridge reward is fundamentally different from unimodal objectives because its optimum forms a continuous circular ring rather than a single point. Consequently, the learner only needs to identify the optimal manifold instead of a unique maximizer. This benchmark evaluates the algorithm's ability to adapt to non-isolated optima.

The utility function is

$$\mu(x) = \text{clip}(1 - |\|x - (0.5, 0.5)\|_2 - 0.28|, 0, 1).$$

The experimental results are reported in Fig. 7.



(a) Reference arm convergence (b) Convergence of $\Delta(x^*, X^m)$ (c) Cumulative Regret

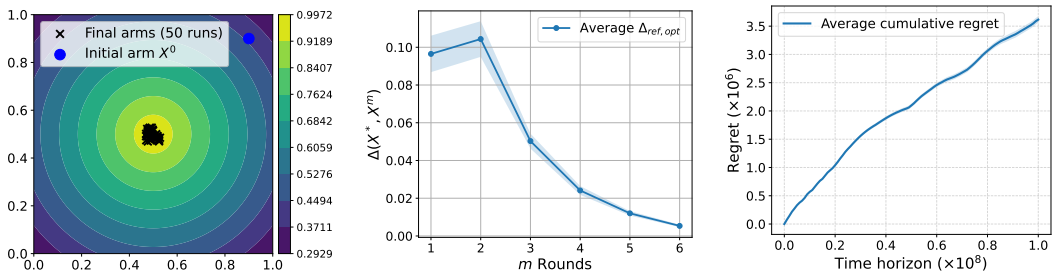
Figure 7: Performance of ARC-DUELLI on the Ridge reward function, averaged over 50 independent runs. From left to right: (a) convergence of the reference arm toward the optimal ridge, (b) convergence of the optimality gap $\Delta(x^*, X^m)$ with one standard deviation, and (c) cumulative regret over time.

Sharp ($d_z = 0$)**.** The Sharp reward is the simplest benchmark considered in our experiments. Its utility decreases linearly from a unique optimum located at $(0.5, 0.5)$, producing large reward gaps between neighboring actions. Owing to its steep landscape, the optimal arm can typically be identified with relatively few comparisons.

The utility function is

$$\mu(x) = \text{clip}(1 - \|x - (0.5, 0.5)\|_2, 0, 1).$$

Figure 8 summarizes the corresponding experimental results.



(a) Reference arm convergence (b) Convergence of $\Delta(x^*, X^m)$ (c) Cumulative Regret

Figure 8: Performance of ARC-DUELLI on the Sharp reward function, averaged over 50 independent runs. From left to right: (a) convergence of the reference arm on the utility landscape, (b) convergence of the optimality gap $\Delta(x^*, X^m)$ with one standard deviation, and (c) cumulative regret as a function of the number of comparisons.

Two Peaks ($d_z = 0$). The Two Peaks reward introduces two competing maxima with slightly different heights. While the global optimum lies at $(0.2, 0.2)$, the second peak at $(0.8, 0.75)$ is nearly optimal, making exploration more difficult since the algorithm must discriminate between two highly competitive regions before identifying the true optimum.

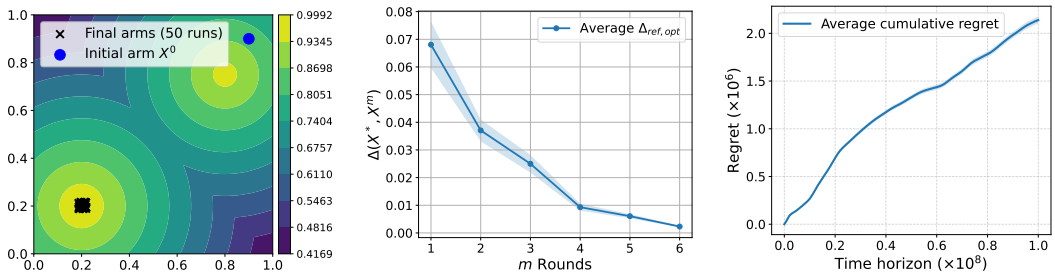
Define

$$g_1(x) = 1 - \frac{\|x - (0.2, 0.2)\|_2}{\sqrt{2}}, \qquad g_2(x) = 0.97 - \frac{\|x - (0.8, 0.75)\|_2}{\sqrt{2}}.$$

The utility is

$$\mu(x) = \mathrm{clip}(\max\{g_1(x), g_2(x)\}, 0, 1)\,.$$

The corresponding experimental results are shown in Fig. 9.



(a) Reference arm convergence (b) Convergence of $\Delta(x^*, X^m)$ (c) Cumulative Regret

Figure 9: Performance of ARC-DUELLI on the Two Peaks reward function, averaged over 50 independent runs. From left to right: (a) convergence of the reference arm toward the global optimum despite the presence of a competing local optimum, (b) convergence of the optimality gap $\Delta(x^*, X^m)$ with one standard deviation, and (c) cumulative regret over time.

F.4 Experiments with different transfer function

To examine the robustness of our algorithm to the choice of the transfer function, we additionally consider the sigmoid, arctangent, probit, linear, and hyperbolic-tangent links. For a utility difference $u = f(x) - f(y)$, these transfer functions are defined as

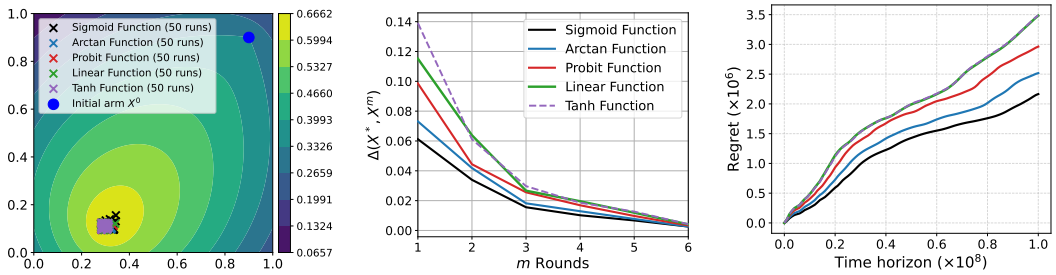
$$\rho_{\mathrm{sig}}(u) = \frac{1}{1 + \exp(-u)}, \qquad \rho_{\mathrm{arc}}(u) = \frac{1}{2} + \frac{1}{\pi}\arctan(u), \qquad \rho_{\mathrm{probit}}(u) = \Phi(u),$$

$$\rho_{\mathrm{lin}}(u) = \frac{1 + u}{2}, \qquad \rho_{\tanh}(u) = \frac{1 + \tanh(u)}{2}.$$

where Φ denotes the cumulative distribution function of the standard normal distribution. All these functions satisfy $\rho(0) = 1/2$ and $\rho(-u) = 1 - \rho(u)$. Their Lipschitz constants are obtained by bounding their derivatives:

$$\Gamma_{\text{sig}} = \frac{1}{4}, \qquad \Gamma_{\text{arc}} = \frac{1}{\pi}, \qquad \Gamma_{\text{probit}} = \frac{1}{\sqrt{2\pi}}, \qquad \Gamma_{\text{lin}} = \frac{1}{2}, \qquad \Gamma_{\tanh} = \frac{1}{2}.$$

Figure 10 shows that the reference arm converges towards the near-optimal region for all five transfer functions, while the preference gap $\Delta(x^\star, X^m)$ decreases consistently across rounds. The cumulative-regret curves exhibit the same overall sublinear behavior, although their numerical values differ because the transfer functions have different slopes and therefore induce different preference gaps for the same utility difference. These results indicate that the empirical behavior of the algorithm is robust to the particular transfer function used to generate the pairwise comparisons.



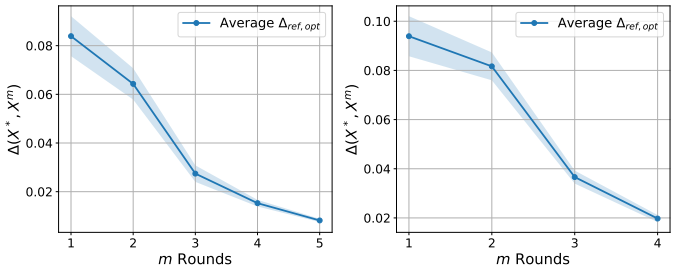
(a) Reference arm convergence (b) Convergence of $\Delta(x^*, X^m)$ (c) Cumulative Regret

Figure 10: Performance of ARC-DUELLI under five transfer functions—sigmoid, arctangent, probit, linear, and hyperbolic tangent—averaged over 50 independent runs. From left to right: (a) convergence of the reference arm, (b) convergence of the preference gap $\Delta(x^\star, X^m)$ with one standard deviation, and (c) cumulative regret over time. .

F.5 Experiments with different Ambient Dimension d

We further evaluate the performance of our algorithm across different ambient dimensions. In particular, we consider the cases of $d = 3$ and $d = 4$, in addition to the two-dimensional setting. For both settings, we use the utility function $\mu(x) = 1 - \frac{2}{3}\|x - x_1\| - \frac{1}{3}\|x - x_2\|$, where the locations of x_1 and x_2 are specified according to the ambient dimension. For $D = 3$, we consider the action space $[0, 1]^3$ with $x_1 = (0.3, 0.1, 0.5), x_2 = (0.9, 0.9, 0.5)$, while for $D = 4$, we consider the action space $[0, 1]^4$ with $x_1 = (0.3, 0.1, 0.5, 0.5), x_2 = (0.9, 0.9, 0.5, 0.5)$. For these higher-dimensional settings, the algorithm terminates in lesser rounds. This behavior is primarily due to the substantially larger number of cubes generated during the partitioning process as the ambient dimension increases. Consequently, each round involves a larger number of pairwise comparisons than in the $D = 2$ setting, resulting in a higher computational cost per round.

Despite this increased computational burden, the results in Figure 11 demonstrate that the reference arm continues to move progressively closer to the optimal arm in both $d = 3$ and $d = 4$. This indicates that the algorithm maintains its ability to identify increasingly promising regions of the action space even in higher dimensions. Although the available time horizon of 10^8 limits the number of completed rounds, the observed convergence trend suggests that, with a sufficiently larger time horizon T, the reference arm would move even closer to the optimal arm and ultimately approach the true optimal solution.



(a) $D = 3$ (b) $D = 4$

Figure 11: Convergence of the preference gap $\Delta(x^\star, X^m)$ for higher-dimensional action spaces under ARC-DUELLI. Panel (a) shows the convergence of $\Delta(x^\star, X^m)$ for an ambient dimension of $D = 3$, while Panel (b) shows the corresponding convergence for $D = 4$. In both settings, the preference gap decreases with the number of rounds, indicating that the reference arm progressively approaches the optimal arm.

NeurIPS Paper Checklist

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

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

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

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

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

IMPORTANT, please:

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

1. **Claims**

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

   Answer: [Yes]

   Justification: We have clearly mention the paper's contribution and scope in the abstract and introduction.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

Justification: We have added last paragraph to bring out the limitation of the work.

Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: All the assumptions are stated clearly, and for all the theoretical results, proof sketch is provided in the main paper with full proofs referred to appendix.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: All the experimental parameters are clearly specified for reproducibility.

   Guidelines:

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

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

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

   Answer: [Yes]

   Justification: We have provided anonymized github repository.

   Guidelines:

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

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The experiment setting is clearly specified in the experiment setting and the appendix corresponding to each experiments run.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: All plots have standard deviation.

   Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We have added the details on computing resources used for the experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

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

   Answer:[Yes]

   Justification: The paper doesn't have any ethical concerns.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: The paper is primarily theoretical and does not cater to any societal impact.

    Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: Paper has no such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [N/A]

    Justification: Paper does not use any existing asset.

    Guidelines:

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

13. **New assets**

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

    Answer: [N/A]

    Justification: No new assests are released.

    Guidelines:

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

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

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

    Answer: [N/A]

Justification: [N/A]

Guidelines:

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

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

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

    Answer: [No]

    Justification: [N/A]

    Guidelines:

    - The answer [No] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: LLMs were used only for minor edits.

    Guidelines:

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