PROVABLY SAMPLE-EFFICIENT ACTIVE PREFERENCE DATA COLLECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Collecting human preference feedback is often expensive, leading recent works to develop algorithms to select them more efficiently. However, these works assume that the underlying reward function is linear, an assumption that does not hold in many real-life applications, e.g., online recommendation. To address this limitation, we propose `Neural-ADB`, an algorithm based on the neural contextual dueling bandit framework that provides a practical method for collecting human preference feedback when the underlying latent reward function is non-linear. We theoretically show that when preference feedback follows the Bradley-Terry-Luce model, the worst sub-optimality gap of the policy learned by `Neural-ADB` decreases at a sub-linear rate as the preference dataset increases. Our experimental results on preference datasets further corroborate the effectiveness of `Neural-ADB`.

1 INTRODUCTION

Collecting human preference feedback is essential in many real-life applications, like online recommendations (Kohli et al., 2013; Wu et al., 2023; Zhang & Wang, 2023; Yang et al., 2024), content moderation (Avadhanula et al., 2022), medical treatment design (Lai & Robbins, 1985; Bengs et al., 2021), prompt optimization (Lin et al., 2024), and aligning large language models (Bai et al., 2022; Menick et al., 2022; Mehta et al., 2023; Chaudhari et al., 2024; Das et al., 2024; Ji et al., 2024), to ensure systems effectively align with user preferences, and exhibit desired behaviors. However, this process is often costly due to the need for skilled evaluators, the complexity of tasks, and the time-intensive nature of producing high-quality, reliable human feedback. To address the challenge of balancing cost and effectiveness in aligning systems, this paper proposes principled and practical algorithms for efficiently collecting human feedback sequentially and adaptively to achieve the desired system behavior. Specifically, we aim to answer the following fundamental question: ***How to achieve desired system behavior while using as minimal human feedback as possible?***

Recent works (Mehta et al., 2023; Das et al., 2024) have modeled the problem of active human feedback collection as an active version of the contextual dueling bandit problem (ADB for brevity) (Saha, 2021; Bengs et al., 2022; Li et al., 2024), where a context-arm pair corresponds to a task for which human preference feedback is collected and then proposed algorithms to select context-arm pairs for human feedback sequentially and adaptively by exploiting the collected preference dataset, i.e., past context-arm pairs with their preference feedback. The preference feedback between two context-arm pairs is commonly assumed to follow the Bradley-Terry-Luce (BTL) model[1] (Hunter, 2004; Bengs et al., 2022; Li et al., 2024; Verma et al., 2025) in which the probability of preferring a context-arm pair over others is proportional to the exponential of its reward. In many real-life applications, the number of context-arm pairs (e.g., user-movie pair in online movie recommendation) can be large or even infinite. Therefore, the reward for each context-arm pair is assumed to be an unknown function of its feature vector, such as a linear function (Mehta et al., 2023; Das et al., 2024).

To better align the system for optimal performance, we consider two key components: *context selection* and *arm selection*. The context selection aims to encourage diversity by exploring the context space, such as selecting prompts as diverse as possible in prompt optimization. Whereas arm selection focuses on identifying the arms that help learn the best arm for each context, such as

[1]For more than two context-arm pairs, preferences are typically modeled using the Plackett-Luce model (Soufiani et al., 2014).

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

selecting the most effective pair of responses to a given prompt that maximizes the system's learning (Lin et al., 2024). Since the goal is to identify the best arm for each context, selecting suboptimal arms provides less useful information than choosing better arms. Existing methods for active contextual dueling bandits (Mehta et al., 2023; Das et al., 2024) do not incorporate an efficient arm selection strategy during data collection, limiting their ability to achieve optimal performance.

An efficient arm selection strategy requires estimating the reward function to guide the arm selection process effectively. Since the reward function may not always be linear in practice, this paper parameterizes the reward function via a *non-linear function*, which needs to be estimated from the available preference dataset by using methods like Gaussian processes (Williams & Rasmussen, 2006; Srinivas et al., 2010) or neural networks (Zhou et al., 2020; Zhang et al., 2021). However, Gaussian processes have limited expressive power and fail to optimize highly complex functions. In contrast, neural networks (NNs) have greater expressive power, making them well-suited for modeling complex functions (Verma et al., 2025; Dai et al., 2023).

In this paper, we propose a neural active contextual dueling bandit algorithm, `Neural-ADB`, which uses an NN to estimate the unknown reward function using the available preference dataset. The context selection in `Neural-ADB` is adapted from Das et al. (2024), while arm selection strategies are based on, respectively, upper confidence bound (UCB) and Thompson sampling (TS), and adapted from Verma et al. (2025). Due to the differences in context selection strategy, arm selection strategies, and the use of a non-linear reward function, our theoretical analysis is completely different than related existing work (Mehta et al., 2023; Das et al., 2024). One of the key theoretical contributions of this paper is *providing an upper bound on the maximum Mahalanobis norm* of a vector from the fixed input space, measured with respect to the inverse of a positive definite Gram matrix that is constructed using finite, adapted samples from that space. Building on this result, we prove that the worst sub-optimality gap (defined in Eq. (1)) of the policy learned by `Neural-ADB` decreases at a sub-linear rate as the preference dataset size increases.

Specifically, our key contributions can be summarized as follows:

- We introduce the setting of active contextual dueling bandits with a non-linear reward function in Sec. 3. In Sec. 4, We propose a neural active contextual dueling bandit algorithm, `Neural-ADB`, which uses a neural network to estimate the unknown reward function from the available preference dataset and integrates this estimate into the arm selection strategy.
- We prove an upper bound on the maximum Mahalanobis norm of a vector from the fixed input space, as measured with respect to the inverse of a positive definite Gram matrix (Lemma 1), where the Gram matrix is constructed using finite, adapted samples from that input space. We show that this upper bound decays at a sub-linear rate as the number of samples used in the Gram matrix increases. This theoretical result is of independent interest, as it offers fundamental insights into the behavior of the bound and has implications beyond this work.
- We prove that the worst sub-optimality gap of the policy learned by `Neural-ADB` with both of our arm selection strategies (Theorem 1 and Theorem 2) decreases at a sub-linear rate with respect to the size of preference dataset, specifically at rate of $\tilde{O}((\tilde{d}/T)^{\frac{1}{2}})$, where $\tilde{O}$ hides the logarithmic factors and constants and $\tilde{d}$ is the effective dimension of context-arm feature vectors. The decay rate of the worst sub-optimality gap for `Neural-ADB` improves by a factor of $\tilde{O}((\log T)^{\frac{1}{2}})$ compared to existing algorithms (Mehta et al., 2023; Das et al., 2024).
- In Sec. 5, our experimental results further validate the different performance aspects of `Neural-ADB`, highlighting its strong sample efficiency in preference data collection across a range of experimental settings.

2 Related Work

This section reviews the most relevant work to our setting: contextual dueling bandits and active contextual dueling bandits. We discuss related topics, such as neural contextual bandits and finite-armed dueling bandits, in Sec. A, where we provide detailed comparisons with existing work.

Contextual Dueling Bandits. Many real-life applications, such as online recommendations, content moderation, medical treatment design, prompt optimization, and aligning large language models, can be effectively modeled using contextual dueling bandits, where a learner observes a context

(additional information before selecting a pair of arms) and then selects the arms based on that context and observes preference feedback for the selected arms. Since the number of context-arm pairs can be potentially large or even infinite, the mean latent reward of each context-arm is assumed to be parameterized by an unknown function of its features. Common assumptions include linear reward models (Saha, 2021; Bengs et al., 2022; Di et al., 2023; Saha & Krishnamurthy, 2022; Li et al., 2024) and non-linear models (Verma et al., 2025). For our setting, we adopt the neural contextual dueling bandit algorithms proposed in (Verma et al., 2025) to construct confidence ellipsoids for the latent non-linear reward function. Note that `Neural-ADB` can incorporate alternative confidence ellipsoids by appropriately modifying Lemma 3. Furthermore, our work addresses an active learning problem and analyzes the convergence rate of the worst sub-optimality gap, whereas (Verma et al., 2025) focus on a regret minimization setting and derive upper bounds on cumulative regret.

Active contextual dueling bandits. The work most closely related to ours is active contextual dueling bandit (Mehta et al., 2023; Das et al., 2024), which takes a principled approach to actively collecting preference datasets. However, two key differences exist between our work and existing research: the non-linear reward function and the arm selection strategy. Existing studies typically assume a linear reward function, which may not be suitable for many real-world applications. Our work addresses this gap by extending the existing framework to incorporate non-linear reward functions in contextual dueling bandits. Additionally, existing approaches use different methods for selecting the pair of arms, leading to distinct arm selection strategies compared to ours. As a result of these differences in both the arm selection strategy and the non-linear reward function (which we estimate using a neural network), our analysis diverges significantly from that of prior work.

3 Problem Setting

We model active human preference feedback collection as an active contextual dueling bandit problem, where a labeler (human or simulator) provides preference feedback for a chosen pair of arms.

Active contextual dueling bandit. We consider an active contextual dueling bandit problem in which the underlying latent reward function can be non-linear. In each iteration of this problem, the learner's goal is to select a triplet containing a context and two arms to collect preference feedback from a labeler/human, such that the resulting preference dataset yields superior performance. Let $\mathcal{C}$ be the set of contexts and $\mathcal{A}$ be the set of all possible arms. In each iteration, the learner selects a context $c_t \in \mathcal{C}$ and then two arms (denoted as $a_{t,1}$ and $a_{t,2}$) from the set of arms $\mathcal{A}$. After selecting the triplet of context and two arms, the learner receives a stochastic preference feedback y_t, where $y_t = 1$ implies the arm $a_{t,1}$ is preferred over arm $a_{t,2}$ for the context c_t and $y_t = 0$ otherwise. We use $\varphi(c_t, a)$ to denote the context-arm feature vector for context c_t and an arm a, where $\varphi : \mathcal{C} \times \mathcal{A} \to \mathbb{R}^d$ is a known feature map, such as one that concatenates the context and arm features.

Preference model. Following the dueling bandits literature (Saha, 2021; Bengs et al., 2022; Li et al., 2024; Verma et al., 2025), we assume the preference feedback follows the Bradley-Terry-Luce (BTL) model[2] (Hunter, 2004; Luce, 2005). Under the BTL preference model, the preference feedback has a Bernoulli distribution, where the probability that the first selected arm $a_{t,1}$ is preferred over the second selected arm $a_{t,2}$ for the given context c_t is given by

$$\mathbb{P}\left\{a_{t,1} \succ a_{t,2}\right\} = \mathbb{P}\left\{y_t = 1 | c_t, a_{t,1}, a_{t,2}\right\} = \mu\left(f(\varphi(c_t, a_{t,1})) - f(\varphi(c_t, a_{t,2}))\right),$$

where $a_{t,1} \succ a_{t,2}$ used for brevity and denotes that $a_{t,1}$ is preferred over $a_{t,2}$ for the given context c_t, $\mu(x) = 1/(1 + \mathrm{e}^{-x})$ is the sigmoid function, $f : \mathbb{R}^d \to \mathbb{R}$ is an unknown non-linear bounded reward function, and $f(\varphi(c, a))$ is the latent reward of the arm a for the context c. We require the following standard assumptions on the function μ (commonly referred to as a *link function* in the contextual bandit literature (Bengs et al., 2022; Li et al., 2017; Verma et al., 2025)):

Assumption 1.
- *Let* $\kappa_\mu \doteq \inf_{c \in \mathcal{C}, a, b \in \mathcal{A}} \dot{\mu}(f(\varphi(c, a)) - f(\varphi(c, b))) > 0$ *for all triplets of context* (c) *and pair of arms* (a, b).
- *The link function* $\mu : \mathbb{R} \to [0, 1]$ *is continuously differentiable and Lipschitz with constant* L_μ. *For the logistic function, we have* $L_\mu \leq 1/4$.

[2]Our results are also applicable to any preference models, such as the Thurstone-Mosteller model and Exponential Noise, as long as stochastic transitivity holds (Bengs et al., 2022).

Performance measure. We denote the collected preference dataset up to T iterations by $\mathcal{D}_T = \{(c_s, a_{s,w}, a_{s,l}, y_s)\}_{s=1}^T$, where $a_{s,w} \succ a_{s,l}$ for the selected context c_s in iteration s. We aim to learn a policy, $\pi : \mathcal{C} \to \mathcal{A}$ from the collected preference dataset $\mathcal{D}_T$ that achieves the worst sub-optimality gap across all contexts in $\mathcal{C}$, which is defined as follows:

$$\Delta^{\pi}_{\mathcal{D}_T} = \max_{c \in \mathcal{C}} \left[\max_{a \in \mathcal{A}} f(\varphi(c, a)) - f\left(\varphi(c, \pi(c))\right) \right], \tag{1}$$

where policy π is a learned policy from the collected preference dataset $\mathcal{D}_T$ up to the iteration T. The policy $\pi_{\mathcal{D}_T}$ competes with the Condorcet winner (Bengs et al., 2021; Das et al., 2024) for a given context, i.e., an arm that is better than all other arms. The suboptimality gap is the worst possible difference in latent rewards over the set of contexts, and the same performance measure is used in prior work (Mehta et al., 2023; Das et al., 2024).

4 Algorithm for Active Human Preference Feedback Collection

In this section, we introduce `Neural-ADB`, a simple yet principled and practical algorithm designed to efficiently select context-arm pairs for collecting preference feedback. `Neural-ADB` consists of two main components: context selection and arm selection. Since the arm selection strategy depends on the estimated reward function, we first explain how an NN can be used to estimate the unknown reward function. We will then give details of the context and arm selection strategies, followed by our theoretical results that validate the effectiveness of `Neural-ADB`.

4.1 Reward function estimation using neural network

For estimating the latent reward function, we use a fully connected neural network (NN) with depth $D \geq 2$, a hidden layer width w, and ReLU activations as done in Zhou et al. (2020), Zhang et al. (2021), and Verma et al. (2025). Let $h(x; \theta)$ be the output of a fully-connected NN with parameters θ for context-arm feature vector $x = \varphi(c, a)$ of context c and arm a, which we define as follows:

$$h(x; \theta) = \boldsymbol{W}_D \text{ReLU}\left(\boldsymbol{W}_{D-1} \text{ReLU}\left(\cdots \text{ReLU}\left(\boldsymbol{W}_1 x\right)\right)\right),$$

where $\text{ReLU}(v) = \max\{v, 0\}$, $\boldsymbol{W}_1 \in \mathbb{R}^{w \times d}$, $\boldsymbol{W}_l \in \mathbb{R}^{w \times w}$ for $2 \leq l < D$, $\boldsymbol{W}_D \in \mathbb{R}^{w \times 1}$. The parameters of the NN are represented by $\theta = (\text{vec}(\boldsymbol{W}_1); \cdots \text{vec}(\boldsymbol{W}_D))$, where $\text{vec}(A)$ transforms an $m \times n$ matrix A into a vector of dimension mn. We use p to represent the total number of NN parameters, which is given by $p = dw + w^2(D-1) + w$, and $g(x; \theta)$ to denote the gradient of NN $h(x; \theta)$ with respect to θ. At the end of each iteration t, the preference dataset $\mathcal{D}_t = \{(c_s, a_{s,w}, a_{s,l}, y_s)\}_{s=1}^t$ is used to estimate the reward function f by training an NN h (parameterized by θ_{t+1}) using gradient descent to minimize the following binary cross entropy loss function:

$$\min_{\theta} \mathcal{L}_t(\theta) = -\frac{1}{w} \sum_{s=1}^{t} \Big[\log \mu\big(h(\varphi(c_s, a_{s,w}); \theta) - h(\varphi(c_s, a_{s,l}); \theta)\big) \Big] + \frac{1}{2} \lambda \left\| \theta - \theta_0 \right\|_2^2, \tag{2}$$

where θ_0 denotes the initial parameter of the NN that is initialized according to the standard practice in neural bandits (Zhou et al., 2020; Zhang et al., 2021) (see Algorithm 1 in Zhang et al. (2021) for details). Minimizing the first term in the above loss function (that involves the summation over the t terms) corresponds to finding the maximum log-likelihood estimate of the parameters θ.

4.2 `Neural-ADB`

We next propose a simple yet principled and practical algorithm, `Neural-ADB`, that consists of two key components: Context selection and arm selection. `Neural-ADB` works as follows: At the beginning of the iteration t, we first select the context as follows:

$$c_t = \underset{c \in \mathcal{C}}{\text{argmax}} \max_{(a,b) \in \mathcal{A} \times \mathcal{A}} \left\| \Phi(c, a) - \Phi(c, b) \right\|_{V_t^{-1}}, \tag{3}$$

where $\Phi : \mathcal{C} \times \mathcal{A} \to \mathcal{Z} \subset \mathbb{R}^p$ is a known feature map and $V_{t-1} = \frac{\lambda}{\kappa_\mu} \mathbb{I}_p + \sum_{s=1}^{t-1} z_s {z_s}^\top \frac{1}{w}$ in which $z_s = \Phi(c_s, a_{s,w}) - \Phi(c_s, a_{s,l}) = g(\varphi(c_s, a_{s,w}); \theta_0) - g(\varphi(c_s, a_{s,l}); \theta_0)$, and $g(\varphi(c_s, a_{s,i}); \theta_0)/\sqrt{w}$ is used as the Random features approximation for the context-arm feature vector $\varphi(c_s, a_{s,i})$. This strategy is adapted from the context selection strategy[3] from Das et al. (2024). After selecting context

[3] Selecting contexts uniformly at random suffers a constant sub-optimality gap (Das et al., 2024, Theorem 3.2).

c_t, `Neural-ADB` uses the trained NN (as an estimate of the unknown reward function) to decide which two arms must be selected. To do so, `Neural-ADB` uses UCB- and TS-based arm selection strategies, which efficiently balance the trade-off between exploration and exploitation (Lattimore & Szepesvári, 2020) due to the bandit nature of preference feedback, as preference feedback is only observed for the selected pair of arms.

`Neural-ADB` Neural Active Dueling Bandit algorithm

Input parameters: $\delta \in (0, 1)$, $\lambda > 0$, and $w > 0$
Initialize: NN parameters θ_1 and $D_0 = \emptyset$
for $t = 1, \ldots, T$ **do**
  Select a context c_t from $\mathcal{C}$ using Eq. (3)
  Select first arm $a_{t,1}$ using Eq. (4)
  Select second arm $a_{t,2}$ using Eq. (5) (UCB-based selection) or Eq. (7) (TS-based selection)
  Observe preference feedback $y_t = \mathbb{1}_{\{a_{t,1} \succ a_{t,2}\}}$
  Update $D_t = D_{t-1} \cup \{(c_t, a_{t,1}, a_{t,2}, y_t)\}$
  Retrain NN parameters θ_{t+1} using D_t by minimizing the loss function defined in Eq. (2)
end for
Return policy $\pi(c) = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\, h(\varphi(c, a); \theta_T),\ \forall c \in \mathcal{C}$

UCB-based arm selection strategy. Algorithms based on Upper confidence bound (UCB) are commonly used to address the exploration-exploitation trade-off in many sequential decision-making problems (Bengs et al., 2022; Zhou et al., 2020; Auer et al., 2002; Abbasi-Yadkori et al., 2011). Our UCB-based arm selection strategy works as follows: In the iteration t, it selects the first arm greedily (i.e., by maximizing the output of the trained NN with parameters θ_t) for the selected context c_t, ensuring the best-performing arm is always selected as follows:

$$a_{t,1} = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\, h(\varphi(c_t, a); \theta_t). \tag{4}$$

The second arm $a_{t,2}$ is selected optimistically by maximizing the UCB value as follows:

$$a_{t,2} = \underset{b \in \mathcal{A} \setminus \{a_{t,1}\}}{\operatorname{argmax}} \left[h(\varphi(c_t, b); \theta_t) + \text{cf}(t, c_t, a_{t,1}, b) \right], \tag{5}$$

where $\text{cf}(t, c_t, a_{t,1}, b) = \nu_T \sigma_{t-1}(c_t, a_{t,1}, b)$, $\nu_T \doteq (\beta_T + B\sqrt{\lambda/\kappa_\mu} + 1)\sqrt{\kappa_\mu/\lambda}$ in which $\beta_T \doteq \frac{1}{\kappa_\mu}\sqrt{\widetilde{d} + 2\log(1/\delta)}$, $\widetilde{d}$ is the *effective dimension* (defined in Eq. (8)), and

$$\sigma^2_{t-1}(c, a, b) \doteq \frac{\lambda}{\kappa_\mu} \left\| \frac{1}{\sqrt{w}} (\Phi(c, a) - \Phi(c, b)) \right\|^2_{V^{-1}_{t-1}}. \tag{6}$$

A larger value of $\sigma^2_{t-1}(c_t, a_{t,1}, b)$ implies that arm b is significantly different from $a_{t,1}$, given the contexts and arm pairs already selected. As a result, the second term in Eq. (5) makes the second arm different from the first arm, which ensures exploration.

TS-based arm selection strategy. Thompson sampling (TS) selects an arm based on its probability of being the best (Thompson, 1933). Several works (Li et al., 2024; Chapelle & Li, 2011; Agrawal & Goyal, 2013; Chowdhury & Gopalan, 2017) have shown that TS empirically outperforms UCB-based bandit algorithms. Therefore, we also propose a TS-based arm selection strategy in which the first arm is also selected using Eq. (4) and the second arm is selected differently. To select the second arm, it first samples a score $s_t(b) \sim \mathcal{N}\big(h(\varphi(c_t, b); \theta_t) - h(\varphi(c_t, a_{t,1}); \theta_t), \nu_T^2 \sigma^2_{t-1}(c_t, a_{t,1}, b)\big)$ for every arm $b \in \mathcal{A} \setminus \{a_{t,1}\}$ and then selects the second arm that maximizes the samples scores as:

$$a_{t,2} = \operatorname{argmax}_{b \in \mathcal{A} \setminus \{a_{t,1}\}}\, s_t(b). \tag{7}$$

After selecting context and arms in iteration t, stochastic preference feedback is observed, denoted by $y_s = \mathbb{1}_{\{a_{t,1} \succ a_{t,2}\}}$, which is equal to 1 if arm $a_{s,1}$ is preferred over arm $a_{s,2}$ for context c_t and 0 otherwise. With the new observation, the preference dataset is updated to $\mathcal{D}_t = \mathcal{D}_{t-1} \cup \{(c_t, a_{t,w}, a_{t,l}, y_t)\}$ and then the NN is retrained using the updated preference dataset $\mathcal{D}_t$. Once the preference data collection process concludes (i.e., end of iteration T, which may not be fixed a priori), `Neural-ADB` returns the following policy: $\forall c \in \mathcal{C} :\ \pi(c) = \operatorname{argmax}_{a \in \mathcal{A}}\, h(\varphi(c, a); \theta_T)$.

4.3 THEORETICAL RESULTS

Let the number of arms in $\mathcal{A}$ be finite, and define $\mathbb{V} = \sum_{s=1}^{T} \sum_{(a,b)\in\mathcal{A}\times\mathcal{A}} z_{a,b}(s) z_{a,b}(s)^\top \frac{1}{w}$, where $z_{a,b}(s) = \Phi(c_s, a) - \Phi(c_s, b)$ and $C_2^{|\mathcal{A}|}$ denotes all pairwise combinations of arms. Then, the *effective dimension* of context-arm feature vectors is defined as follows:

$$\widetilde{d} = \log\det\left(\frac{\kappa_\mu}{\lambda}\mathbb{V} + \mathbb{I}_p\right). \tag{8}$$

In the following, we present a novel theoretical result that gives an upper bound on the maximum Mahalanobis norm of a vector selected from the fixed input space, measured with respect to the inverse of a positive definite Gram matrix constructed from finite, adapted samples of the same space.

Lemma 1. *Let $\{Z_s = z_s z_s^\top\}_{s=1}^T$ be a finite adapted sequence of self-adjoint matrices in $\mathbb{R}^d$ and $\mathcal{F}_s$ be the all information available up to the iteration s. Define $\mathbb{E}\left[z_s z_s^\top | \mathcal{F}_{s-1}\right] = \Sigma_s$, $\overline{\Sigma}_T = \sum_{s=1}^T \Sigma_s / T$, $V_0 = \frac{\lambda}{\kappa_\mu}\mathbb{I}_p$, $V_T = V_0 + \sum_{s=1}^T z_s z_s^\top$. Assume $\|z\|_2 \le L$ for all $z \in \mathcal{Z} \subset \mathbb{R}^p$, and $\lambda_{\min}(A)$ denote the minimum eigenvalue of a matrix A. Then, with a probability at least $1-\delta$, $\max_{z\in\mathcal{Z}} \|z\|_{V_T^{-1}} \le L/G_T$, where $G_T = \sqrt{\max\left\{\frac{\lambda}{\kappa_\mu}, \frac{\lambda}{\kappa_\mu} + T\lambda_{\min}(\overline{\Sigma}_T) - \sqrt{8TL^4\log(p/\delta)}\right\}}$.*

Proof sketch. To derive the upper bound, we use various results related to the positive definite matrix (detailed in Fact 1 of the supplementary material). First, if V_T is a positive definite matrix V_T, then for any $z \in \mathcal{Z}$, $\|z\|_{V_T^{-1}} \le \|z\|_2 \sqrt{\lambda_{\max}(V_T^{-1})} = \|z\|_2 / \sqrt{\lambda_{\min}(V_T)}$. Thus, $\max_{z\in\mathcal{Z}} \|z\|_{V_T^{-1}} \le \|z\|_2 / \sqrt{\lambda_{\min}(V_T)} \le L/\sqrt{\lambda_{\min}(V_T)}$. Since $\{Z_s\}_{s=1}^T$ is a finite adapted sequence of self-adjoint matrices (i.e., Z_s is $\mathcal{F}_s$-measurable for all s, where $\mathcal{F}_s$ represents all information available up to iteration s), we apply the Matrix Azuma inequality (Tropp, 2012) to get a high probability lower bound on $\lambda_{\min}(V_T)$, specifically we have shown that $\lambda_{\min}(V_T) \ge \max\left\{\frac{\lambda}{\kappa_\mu}, \frac{\lambda}{\kappa_\mu} + T\lambda_{\min}(\overline{\Sigma}_T) - \sqrt{8TL^4\log(p/\delta)}\right\}$ holds with probability at least $1-\delta$. Using this bound, we get the desired upper bound L/G_T. The complete proof, along with all other missing proofs, is provided in Sec. B.

This result shows that the upper bound can be expressed in terms of the number of adapted samples used to construct the matrix V_T, and it decays at a sub-linear rate as the number of samples (T) increases. Notably, this result is of independent interest, as it provides valuable insights beyond the specific application of our work. Next, we give an upper bound on the worst sub-optimality gap in terms of the upper bound on the estimation error of the reward difference between any triplet consisting of a context and two arms.

Lemma 2. *Let Assumption 1 hold and $D_T = \{x_s, a_{s,1}, a_{s,2}, y_s\}_{s=1}^T$ be the preference dataset collected up to the iteration T and $\hat{f}_T$ represent the estimate of latent reward function f learned from D_T. $\forall c \in \mathcal{C},\ a, b \in \mathcal{A}:\ \left|[f(\varphi(c,a)) - f(\varphi(c,b))] - \left[\hat{f}_T(\varphi(c,a)) - \hat{f}_T(\varphi(c,b))\right]\right| \le \beta_T(c,a,b)$ holds with probability at least $1-\delta$. If $a^\star = \text{argmax}_{a\in\mathcal{A}} f(\varphi(c,a))$ and $\pi(c)$ is the arm selected by policy for context c, then, with a probability at least $1-\delta$, the worst sub-optimality gap for a policy that greedily selects an arm for a given context is upper bounded by: $\Delta_T^\pi \le \max_{c\in\mathcal{C}} \beta_T(c, a^\star, \pi(c))$.*

The proof follows by starting with the worst sub-optimality gap definition in Eq. (1) and then applying a series of algebraic manipulations to derive the stated result. Our next results give an upper bound on $\beta_T(c,a,b)$ when `Neural-ADB` uses different arm selection strategies.

Lemma 3. *Let Assumption 1 hold and $\nu_T = (\beta_T + B\sqrt{\lambda/\kappa_\mu} + 1)\sqrt{\kappa_\mu/\lambda}$, where $\beta_T = (1/\kappa_\mu)\sqrt{\widetilde{d} + 2\log(1/\delta)}$ and $\delta \in (0,1)$. If $w \ge$ poly$(T, L, K, 1/\kappa_\mu, L_\mu, 1/\lambda_0, 1/\lambda, \log(1/\delta))$, then, with a probability of at least $1-\delta$, for `Neural-ADB` with (i) UCB-based arm selection strategy, for all $c \in \mathcal{C} : \beta_T(c,a,b) = \nu_T\sigma_T(c,a^\star,\pi(c)) + 2\varepsilon'_{w,T}$, (ii) TS-based arm selection strategy, for all $c \in \mathcal{C} : \beta_T(c,a,b) = \nu_T\log\left(KT^2\right)\sigma_T(c,a^\star,\pi(c)) + 2\varepsilon'_{w,T}$,where K denotes the maximum number of arms available in each iteration, and $\varepsilon'_{w,T} = C_2 w^{-1/6}\sqrt{\log w} L^3 (T/\lambda)^{4/3}$ for some absolute constant $C_2 > 0$, is the approximation error that decreases as the width of the NN (w) increases.*

Note that Lemma 3 is the only part of our analysis that relies on the confidence ellipsoid of Verma et al. (2025), and therefore the only part that inherits their assumptions: (i) Assumption 1 on the link function; (ii) the over-parameterization condition w, which controls the NTK approximation error $\varepsilon'_{w,T}$; (iii) the effective-dimension condition $\widetilde{d} = o(T)$; (iv) a finite arm set and bounded gradient features $\|z\|_2 \leq L$; and (v) the norm bound B appearing in ν_T. Since Lemma 2 is independent of the confidence ellipsoid construction, an alternative construction can be substituted by modifying only Lemma 3, while leaving the remainder of the analysis unchanged.

Equipped with Lemma 1, Lemma 2, and Lemma 3, we will now provide an upper bound on the worst sub-optimality gap for a policy learned by `Neural-ADB`.

Theorem 1 (UCB). *Let the conditions in Lemma 1 and Lemma 3 hold. Then, with a probability of at least $1-\delta$, the worst sub-optimality gap of* `Neural-ADB` *when using the UCB-based arm selection is upper bounded by* $\Delta_T^{\pi} \leq \left(\frac{\nu_T L}{G_T}\right)\sqrt{\frac{\lambda}{\kappa_\mu w}} + 2\varepsilon'_{w,T} = \tilde{O}\left(\frac{\nu_T}{\sqrt{T\lambda_{\min}(\widetilde{\Sigma}_T)}}\right) = \tilde{O}\left(\sqrt{\frac{\tilde{d}}{T}}\right)$.

Theorem 2 (TS). *Let the conditions in Lemma 1 and Lemma 3 hold. Then, with a probability of at least $1-\delta$, the worst sub-optimality gap of* `Neural-ADB` *when using the TS-based arm selection is upper bounded by* $\Delta_T^{\pi} \leq \left(\frac{\nu_T L \log\left(KT^2\right)}{G_T}\right)\sqrt{\frac{\lambda}{\kappa_\mu w}} + 2\varepsilon'_{w,T} = \tilde{O}\left(\frac{\nu_T}{\sqrt{T\lambda_{\min}(\widetilde{\Sigma}_T)}}\right) = \tilde{O}\left(\sqrt{\frac{\tilde{d}}{T}}\right)$.

The proof follows by applying Lemma 3, setting $z' = \Phi(c, a^\star) - \Phi(c, \pi(c))$ in Eq. (6), and then using Lemma 1. Note that $\varepsilon'_{w,T} = O(1/T)$ and $\widetilde{d} = \widetilde{o}(\sqrt{T})$ as long as the NN width w is large enough (Zhou et al., 2020; Zhang et al., 2021; Verma et al., 2025). Above Theorem 1 and Theorem 2 show that the worst sub-optimality gap of the policy learned by `Neural-ADB` with UCB- and TS-based arm selection strategies decreases at a sub-linear rate with preference dataset size, specifically $\tilde{O}((\tilde{d}/T)^{\frac{1}{2}})$, where $\tilde{O}$ hides logarithmic factors and constants. Further, the decay rate of the worst sub-optimality gap for `Neural-ADB` improves by a factor of $\tilde{O}((\log T)^{\frac{1}{2}})$ compared to prior algorithms (Mehta et al., 2023; Das et al., 2024), hence bridging the gap between theory and practice.

4.4 Active Dueling Bandits with Regret Minimization

In many real-world applications, such as medical treatment design (Lai & Robbins, 1985; Bengs et al., 2021) and content moderation (Avadhanula et al., 2022), both actively selecting arms and minimizing regret are required. For instance, in personalized medical treatment, active learning is used to identify the most informative treatments to test, while cumulative regret minimization ensures the system adapts to deliver better patient outcomes. Such scenarios also arise in other fields, such as dynamic pricing and personalized education, enabling systems to make smarter decisions, reduce suboptimal choices, and optimize overall performance as they gather more valuable data. We next define the *cumulative regret* ('regret' for brevity). After receiving preference feedback for T pairs of arms, the regret of a sequential arm selection policy is given by: $\mathfrak{R}_T = \sum_{t=1}^{T}\left[f(\varphi(c_t, a_t^\star)) - \left(f(\varphi(c_t, a_{t,1})) + f(\varphi(a_{t,2}))\right)/2\right]$, where $a_t^\star = \mathrm{argmax}_{a\in\mathcal{A}_t} f(\varphi(c_t, a))$ denotes the arm that maximizes the reward function for a given context c_t. Since the arm selection strategies in `Neural-ADB` are directly adapted from UCB- and TS-based algorithms for contextual dueling bandits of Verma et al. (2025), the regret upper bounds for these algorithms also apply to `Neural-ADB`. For completeness, we state the regret upper bounds of `Neural-ADB` as follows.

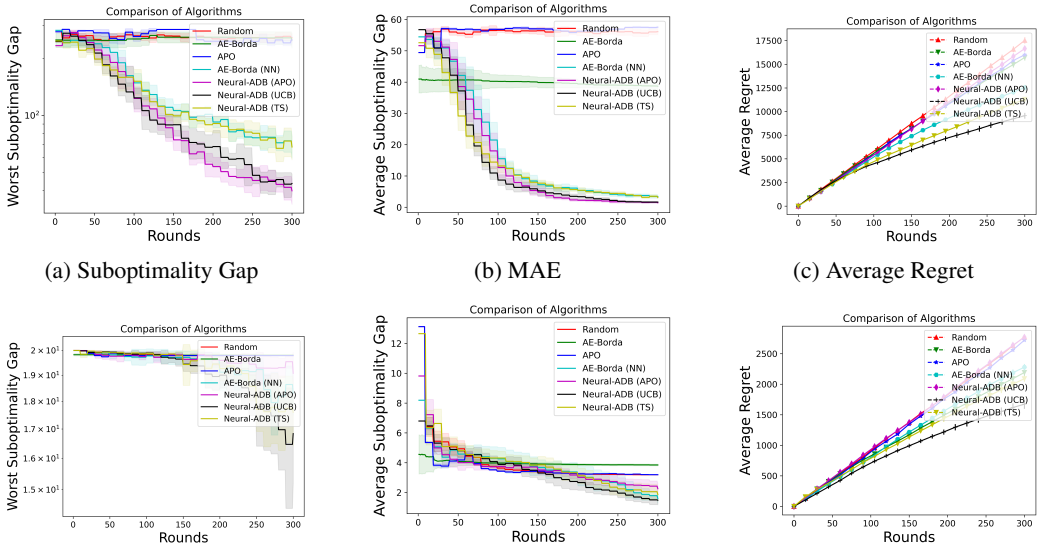
Corollary 1 (Regret Upper Bound). *(Verma et al., 2025, Theorem 2 and Theorem 3) Let $\lambda > \kappa_\mu$ and $w \geq$ poly$(T, L, K, 1/\kappa_\mu, L_\mu, 1/\lambda_0, 1/\lambda, \log(1/\delta))$. Then, with a probability of at least $1-\delta$, the regret of* `Neural-ADB` *when using UCB- or TS-based arm selection strategy is upper bounded by* $\mathfrak{R}_T = \widetilde{O}\left(\left(\frac{\sqrt{\tilde{d}}}{\kappa_\mu} + \sqrt{\frac{\lambda}{\kappa_\mu}}\right)\sqrt{T\tilde{d}}\right)$.

Ignoring logarithmic factors and constants, the asymptotic growth rates of `Neural-ADB` with UCB- and TS-based arm selection strategy are identical and sub-linear.

5 Experiments

Since the contribution of this paper is primarily theoretical, our experiments are designed to empirically validate our theoretical guarantees and compare `Neural-ADB` against existing active

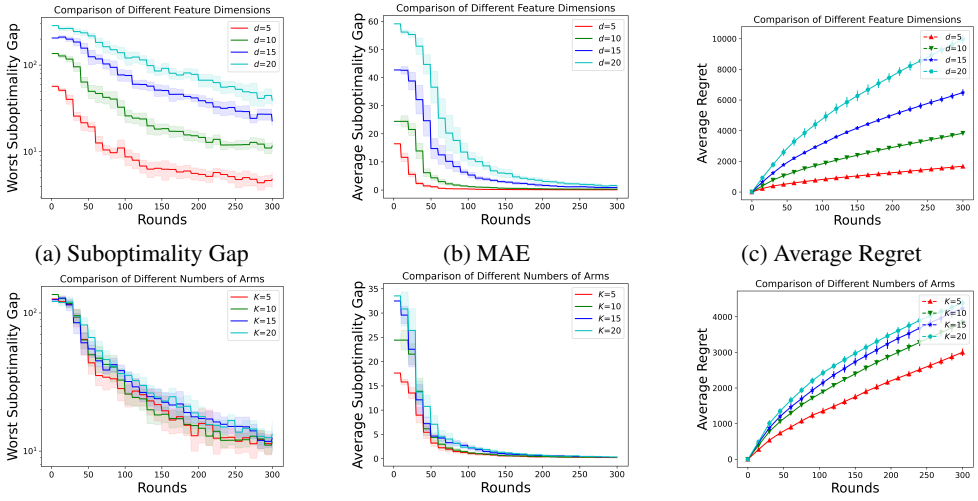
contextual dueling bandit algorithms, rather than to demonstrate state-of-the-art empirical performance across diverse tasks. We therefore use synthetic reward functions ($f(x) = 10(x^\top\theta)^2$ and $f(x) = 2\sin(x^\top\theta)$) that are standard in the neural bandit literature (Verma et al., 2025; Zhou et al., 2020; Zhang et al., 2021; Dai et al., 2023), enabling controlled ablations over problem dimension, number of arms, and network hyperparameters, and additionally evaluate on an LLM alignment task (Sec. C.4). All experiments are repeated 10 times, and we report the average results with 95% confidence intervals (depicted as vertical lines on each curve).



(a) Suboptimality Gap (b) MAE (c) Average Regret

(d) Suboptimality Gap (e) MAE (f) Average Regret

Figure 1: Performance comparison of `Neural-ADB` against different active dueling bandit algorithms on synthetic functions: Square function (top row) and Sine function (bottom row).

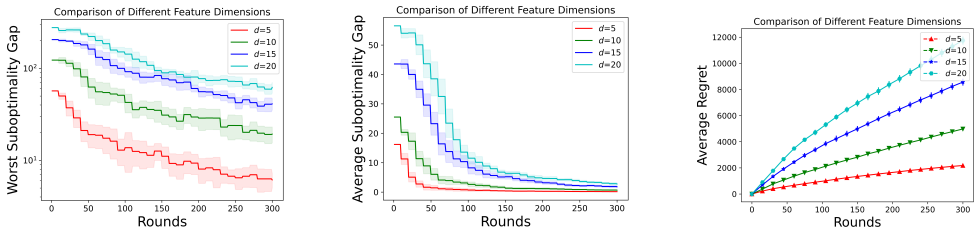


(a) Suboptimality Gap (b) MAE (c) Average Regret

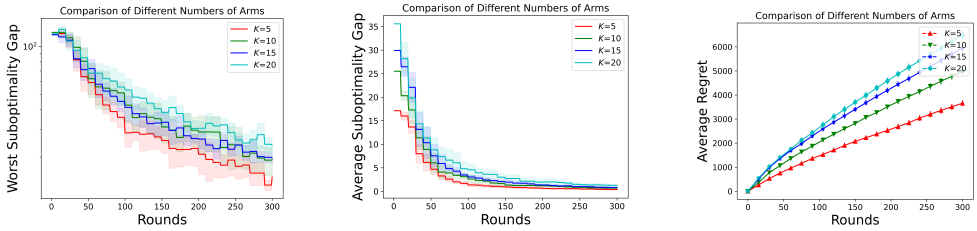
(d) Suboptimality Gap (e) MAE (f) Average Regret

Figure 2: Performance of `Neural-ADB` (UCB) on the Square function, evaluated across varying input dimensions (top row) and numbers of arms (bottom row).

Synthetic dataset. We generate sample features for each context-arm pair in a d-dimensional space. Let $x_{t,a}$ be the context-arm feature vector for context c_t and an arm a. For all $t \geq 1$, $x_{t,a}$ is sampled independently from $(-1, 1)^d$. We keep the number of arms constant across all rounds, denoted by K. In our experiments, the binary preference feedback indicating whether $x_{t,1}$ preferred over $x_{t,2}$ (representing human preference feedback) is sampled from a Bernoulli distribution with parameter $\mu\left(f(x_{t,1}) - f(x_{t,2})\right)$, where f is either a Square or Sine function.



(a) Sub-Optimality Gap (b) MAE (c) Average Regret



(d) Sub-Optimality Gap (e) MAE (f) Average Regret

Figure 3: Performance of `Neural-ADB` (TS) on the Square function, evaluated across varying input dimensions (top row) and numbers of arms (bottom row).

Reward function estimation. We use a neural network with 2 hidden layers with width 50 to estimate the latent reward function, $\lambda = 1.0$, $\delta = 0.05$, $d = 20$, $K = 10$, $T = 1000$, and fixed value of $\nu_T = \nu = 1.0$ in all our experiments (unless we specifically indicate d and K). Note that we did not perform any hyperparameter search for `Neural-ADB`, whose performance can be further improved by doing the hyperparameter search.

Comparison with baselines. We compare the worst suboptimality gap (defined in Eq. (1)), MAE (average suboptimality gap, i.e., $\sum_{t=1}^{T} \left[\max_{a \in \mathcal{A}} f(\varphi(c_t, a)) - f\left(\varphi(c_t, \pi(c))\right)\right]/T$), and average regret (defined in Sec. 4.4) against the different baselines of active contextual dueling bandits to evaluate the performance of UCB- and TS-variant of `Neural-ADB`. We use three baselines: Random, AE-Borda (Mehta et al., 2023), AE-DPO (Mehta et al., 2023), APO (Das et al., 2024), and the neural variants of AE-Borda and APO in which we use a neural network to estimate the latent reward function. They are named AE-Borda (NN) and `Neural-ADB` (APO) respectively. Experimental results in Fig. 1 show that our algorithm, `Neural-ADB` (UCB), outperforms other baselines in almost all synthetic functions (i.e., square function and sine function) in terms of the suboptimality gap. Moreover, both UCB- and TS-variants of `Neural-ADB` also outperform baselines on all synthetic functions in MAE and average regret. We have included more comparisons in different settings (e.g., different d or K) in Sec. C.

Varying dimensions and arms vs. performance. As we increase the dimension of the context-arm feature vectors (d) and the number of arms (K), the problem becomes more challenging. To assess how the changes in K and$d$ affect the performance of our proposed algorithms, we vary $K = \{5, 10, 15, 20\}$ and $d = \{5, 10, 15, 20\}$, while keeping the other parameters fixed. As expected, the performance of our algorithms gets worse with higher values of K and d, as shown in Fig. 2. We have included similar results for `Neural-ADB` (TS) in Fig. 3.

6 Conclusion

This paper studies the problem of active human preference feedback collection by modeling it as an active neural contextual dueling bandit problem. We propose `Neural-ADB`, a principled and practical algorithm designed for efficiently gathering human preference feedback in scenarios where the reward function is non-linear. Exploiting the neural contextual dueling bandit framework, `Neural-ADB` extends its applicability to a broad range of real-world applications, including online recommendation systems and LLM alignment. Our theoretical analysis demonstrates that the worst suboptimality gap of `Neural-ADB` decays at a sub-linear rate as the preference dataset grows. Finally, our experimental results further validate these theoretical findings. An interesting direction for future work is applying `Neural-ADB` to real-life applications such as LLM alignment. From a theoretical perspective, exploring the non-stationary setting is a promising future direction.

AI USE STATEMENT

In this work, we used generative AI tools solely to assist with writing and language refinement, specifically, improving the clarity, grammar, phrasing, and readability of author-drafted text. All technical content, including the problem formulation, methodology, theoretical results and proofs, experimental design, and analysis, was conceived, developed, and written by the authors. AI tools were not used to generate research ideas, derive or verify proofs, produce experimental results, write or debug code, conduct the literature review, or create any figures, tables, or data. Any such tasks requiring disclosure under the policy are therefore not applicable to this work.

We have reviewed all AI-assisted text. Every passage edited or rephrased with AI assistance was checked by the authors for factual accuracy, correct representation of our claims, and consistency with the underlying technical content, and was revised as needed. We take full responsibility for the final content of this work, including all text, claims, and artifacts.

ETHICS STATEMENT

This work is primarily theoretical, focusing on the design and analysis of algorithms. The proposed methods do not directly involve human subjects, personal data, or real-world deployments. While the framework could potentially be applied in systems that interact with users, we emphasize that ethical considerations, such as fairness, privacy, and informed consent, must be addressed in practical deployments. Our primary goal is to advance the theoretical understanding of active preference data collection, and we do not anticipate any immediate negative societal impacts.

REPRODUCIBILITY STATEMENT

This paper primarily presents theoretical results, including formal proofs. All assumptions, definitions, and derivations are stated explicitly in the main text (see Sec. 4) and the Appendix. The details of our experimental setup are provided in Sec. 5 and the Appendix. Additionally, the code used in our experiments has been included in the supplementary material, enabling full reproduction of the results reported in this paper.

REFERENCES

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Proc. NeurIPS*, pp. 2312–2320, 2011.

Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. In *Proc. ICML*, pp. 127–135, 2013.

Nir Ailon, Zohar Karnin, and Thorsten Joachims. Reducing dueling bandits to cardinal bandits. In *Proc. ICML*, pp. 856–864, 2014.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, pp. 235–256, 2002.

Vashist Avadhanula, Omar Abdul Baki, Hamsa Bastani, Osbert Bastani, Caner Gocmen, Daniel Haimovich, Darren Hwang, Dima Karamshuk, Thomas Leeper, Jiayuan Ma, et al. Bandits for online calibration: An application to content moderation on social media platforms. *arXiv:2211.06516*, 2022.

Seoungbin Bae and Dabeen Lee. Neural logistic bandits. *arXiv:2505.02069*, 2025.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv:2204.05862*, 2022.

Yikun Ban, Yuchen Yan, Arindam Banerjee, and Jingrui He. Ee-net: Exploitation-exploration neural networks in contextual bandits. In *Proc. ICLR*, 2022.

Viktor Bengs, Róbert Busa-Fekete, Adil El Mesaoudi-Paul, and Eyke Hüllermeier. Preference-based online learning with dueling bandits: A survey. *Journal of Machine Learning Research*, pp. 1–108, 2021.

Viktor Bengs, Aadirupa Saha, and Eyke Hüllermeier. Stochastic contextual dueling bandits under linear stochastic transitivity models. In *Proc. ICML*, pp. 1764–1786, 2022.

Olivier Chapelle and Lihong Li. An empirical evaluation of thompson sampling. In *Proc. NeurIPS*, pp. 2249–2257, 2011.

Shreyas Chaudhari, Pranjal Aggarwal, Vishvak Murahari, Tanmay Rajpurohit, Ashwin Kalyan, Karthik Narasimhan, Ameet Deshpande, and Bruno Castro da Silva. Rlhf deciphered: A critical analysis of reinforcement learning from human feedback for llms. *arXiv:2404.08555*, 2024.

Sayak Ray Chowdhury and Aditya Gopalan. On kernelized multi-armed bandits. In *Proc. ICML*, pp. 844–853, 2017.

Zhongxiang Dai, Yao Shu, Arun Verma, Flint Xiaofeng Fan, Bryan Kian Hsiang Low, and Patrick Jaillet. Federated neural bandits. In *Proc. ICLR*, 2023.

Nirjhar Das, Souradip Chakraborty, Aldo Pacchiano, and Sayak Ray Chowdhury. Provably sample efficient rlhf via active preference optimization. *arXiv:2402.10500*, 2024.

Qiwei Di, Tao Jin, Yue Wu, Heyang Zhao, Farzad Farnoud, and Quanquan Gu. Variance-aware regret bounds for stochastic contextual dueling bandits. *arXiv:2310.00968*, 2023.

Pratik Gajane, Tanguy Urvoy, and Fabrice Clérot. A relative exponential weighing algorithm for adversarial utility-based dueling bandits. In *Proc. ICML*, pp. 218–227, 2015.

David R Hunter. Mm algorithms for generalized bradley-terry models. *Annals of Statistics*, pp. 384–406, 2004.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Proc. NeurIPS*, pp. 8580–8589, 2018.

Kaixuan Ji, Jiafan He, and Quanquan Gu. Reinforcement learning from human feedback with active queries. *arXiv:2402.09401*, 2024.

Pushmeet Kohli, Mahyar Salek, and Greg Stoddard. A fast bandit algorithm for recommendation to users with heterogenous tastes. In *Proc. AAAI*, pp. 1135–1141, 2013.

Junpei Komiyama, Junya Honda, Hisashi Kashima, and Hiroshi Nakagawa. Regret lower bound and optimal algorithm in dueling bandit problem. In *Proc. COLT*, pp. 1141–1154, 2015.

Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, pp. 4–22, 1985.

Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020.

Lihong Li, Yu Lu, and Dengyong Zhou. Provably optimal algorithms for generalized linear contextual bandits. In *Proc. ICML*, pp. 2071–2080, 2017.

Xuheng Li, Heyang Zhao, and Quanquan Gu. Feel-good thompson sampling for contextual dueling bandits. *arXiv:2404.06013*, 2024.

Xiaoqiang Lin, Zhongxiang Dai, Arun Verma, See-Kiong Ng, Patrick Jaillet, and Bryan Kian Hsiang Low. Prompt optimization with human feedback. *arXiv:2405.17346*, 2024.

R Duncan Luce. *Individual choice behavior: A theoretical analysis*. Courier Corporation, 2005.

Viraj Mehta, Vikramjeet Das, Ojash Neopane, Yijia Dai, Ilija Bogunovic, Jeff Schneider, and Willie Neiswanger. Sample efficient reinforcement learning from human feedback via active exploration. *arXiv:2312.00267*, 2023.

Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, et al. Teaching language models to support answers with verified quotes. *arXiv:2203.11147*, 2022.

OpenAssistant. DeBERTa large summarization reward model. https://huggingface.co/OpenAssistant/reward-model-deberta-v3-large, 2024. Accessed: 2026-07-19.

Carlos Riquelme, George Tucker, and Jasper Snoek. Deep bayesian bandits showdown: An empirical comparison of bayesian deep networks for thompson sampling. In *Proc. ICLR*, 2018.

Aadirupa Saha. Optimal algorithms for stochastic contextual preference bandits. In *Proc. NeurIPS*, pp. 30050–30062, 2021.

Aadirupa Saha and Aditya Gopalan. Battle of bandits. In *Proc. UAI*, pp. 805–814, 2018.

Aadirupa Saha and Aditya Gopalan. Active ranking with subset-wise preferences. In *Proc. AISTATS*, pp. 3312–3321, 2019a.

Aadirupa Saha and Aditya Gopalan. Pac battling bandits in the plackett-luce model. In *Proc. ALT*, pp. 700–737, 2019b.

Aadirupa Saha and Akshay Krishnamurthy. Efficient and optimal algorithms for contextual dueling bandits under realizability. In *Proc. ALT*, pp. 968–994, 2022.

Hossein Azari Soufiani, David Parkes, and Lirong Xia. Computing parametric ranking models via rank-breaking. In *Proc. ICML*, pp. 360–368, 2014.

Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proc. ICML*, pp. 1015–1022, 2010.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. In *Proc. NeurIPS*, pp. 3008–3021, 2020.

William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, pp. 285–294, 1933.

Joel A Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of computational mathematics*, pp. 389–434, 2012.

Arun Verma, Manjesh K Hanawal, Csaba Szepesvári, and Venkatesh Saligrama. Online algorithm for unsupervised sensor selection. In *Proc. AISTATS*, pp. 3168–3176, 2019.

Arun Verma, Manjesh K Hanawal, and Nandyala Hemachandra. Thompson sampling for unsupervised sequential selection. In *Proc. ACML*, pp. 545–560, 2020a.

Arun Verma, Manjesh K Hanawal, Csaba Szepesvári, and Venkatesh Saligrama. Online algorithm for unsupervised sequential selection with contextual information. In *Proc. NeurIPS*, pp. 778–788, 2020b.

Arun Verma, Zhongxiang Dai, Xiaoqiang Lin, Patrick Jaillet, and Bryan Kian Hsiang Low. Neural dueling bandits: Preference-based optimization with human feedback. In *Proc. ICLR*, 2025.

Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. Tl; dr: Mining reddit to learn automatic summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*, pp. 59–63, 2017.

Christopher KI Williams and Carl Edward Rasmussen. *Gaussian Processes for Machine Learning*. MIT press, 2006.

Chuhan Wu, Fangzhao Wu, Yongfeng Huang, and Xing Xie. Personalized news recommendation: Methods and challenges. *ACM Transactions on Information Systems*, pp. 1–50, 2023.

Pan Xu, Zheng Wen, Handong Zhao, and Quanquan Gu. Neural contextual bandits with deep representation and shallow exploration. In *Proc. ICLR*, 2022.

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

Shuhua Yang, Hui Yuan, Xiaoying Zhang, Mengdi Wang, Hong Zhang, and Huazheng Wang. Conversational dueling bandits in generalized linear models. In *Proc. SIGKDD*, pp. 3806–3817, 2024.

Yisong Yue and Thorsten Joachims. Interactively optimizing information retrieval systems as a dueling bandits problem. In *Proc. ICML*, pp. 1201–1208, 2009.

Yisong Yue and Thorsten Joachims. Beat the mean bandit. In *Proc. ICML*, pp. 241–248, 2011.

Yisong Yue, Josef Broder, Robert Kleinberg, and Thorsten Joachims. The k-armed dueling bandits problem. *Journal of Computer and System Sciences*, pp. 1538–1556, 2012.

Weitong Zhang, Dongruo Zhou, Lihong Li, and Quanquan Gu. Neural Thompson sampling. In *Proc. ICLR*, 2021.

Zizhuo Zhang and Bang Wang. Prompt learning for news recommendation. In *Proc. SIGIR*, pp. 227–237, 2023.

Dongruo Zhou, Lihong Li, and Quanquan Gu. Neural contextual bandits with UCB-based exploration. In *Proc. ICML*, pp. 11492–11502, 2020.

Banghua Zhu, Michael Jordan, and Jiantao Jiao. Principled reinforcement learning with human feedback from pairwise or k-wise comparisons. In *Proc. ICML*, pp. 43037–43067, 2023.

Masrour Zoghi, Shimon Whiteson, Remi Munos, and Maarten Rijke. Relative upper confidence bound for the k-armed dueling bandit problem. In *Proc. ICML*, pp. 10–18, 2014a.

Masrour Zoghi, Shimon A Whiteson, Maarten De Rijke, and Remi Munos. Relative confidence sampling for efficient on-line ranker evaluation. In *Proc. WSDM*, pp. 73–82, 2014b.

A Additional Related Work

Neural Contextual Bandits. To model complex and non-linear reward functions, neural contextual bandits (Riquelme et al., 2018; Zhou et al., 2020; Zhang et al., 2021; Xu et al., 2022; Bae & Lee, 2025; Verma et al., 2025) use deep neural networks for reward function estimation. (Riquelme et al., 2018) employ multi-layer neural networks to learn arm embeddings and then use Thompson Sampling at the final layer for exploration. Zhou et al. (2020) propose the first neural contextual bandit algorithm with sub-linear regret guarantees, using a UCB exploration strategy. Building on this, Zhang et al. (2021) propose an algorithm with a TS exploration strategy. Ban et al. (2022) introduces an adaptive exploration strategy incorporating an auxiliary neural network to estimate the potential gain of the exploitation neural network, diverging from traditional UCB and TS exploration strategies. To reduce the computational overhead of using gradient-based features, Xu et al. (2022) only perform UCB-based exploration on the final layer of the neural network. More recent works (Bae & Lee, 2025; Verma et al., 2025) extend these techniques to handle neural contextual bandit settings with binary feedback (i.e., neural logistic bandits).

Finite-Armed Dueling Bandits. Learning from preference feedback has been extensively studied in the bandit literature. In the finite-armed dueling bandits setting, the learner aims to find the best arm while only observing preference feedback for two selected arms (Yue & Joachims, 2009; 2011; Yue et al., 2012). To determine the best arm in dueling bandits, different criteria, such as the Borda winner, Condorcet winner, Copeland winner, or von Neumann winner, have been used while focusing on minimizing regret using only pairwise preference feedback (Ailon et al., 2014; Zoghi et al., 2014b;a; Gajane et al., 2015; Komiyama et al., 2015; Saha & Gopalan, 2018; 2019a;b; Verma et al., 2019; 2020a;b; Zhu et al., 2023). For a comprehensive overview of algorithms for various dueling bandits settings, we refer readers to the survey by Bengs et al. (2021).

A.1 Comparison with Existing Works

Two components of `Neural-ADB` are adapted from prior work at the algorithmic level. The context selection rule in Eq. (3) is adapted from Das et al. (2024), while the UCB- and TS-based arm selection rules in Eqs. (5) and (7), as well as their associated confidence ellipsoid, are adapted from Verma et al. (2025). Our remaining algorithmic and theoretical contributions are novel and do not follow from a direct composition of the analyses in these two works, as we detail below.

The novel contributions of this paper are as follows:

- **Problem setting.** We study active contextual dueling bandits with a *non-linear* latent reward function (Sec. 3), where the learner sequentially selects a context and then a pair of arms.
- **Bound for adaptive Gram matrices (Lemma 1).** Each arm pair is represented by p-dimensional neural tangent kernel (NTK) gradient features evaluated at an adaptively chosen context, so V_T is formed from an *adapted*, non-i.i.d. sequence of rank-one matrices in $\mathbb{R}^{p\times p}$. We bound $\max_{z\in\mathcal{Z}} \|z\|\, V_T^{-1}$ for this adaptive sequence by establishing a high-probability lower bound on $\lambda\min(V_T)$ using the matrix Azuma inequality (Tropp, 2012). This bound may be of independent interest beyond our active contextual dueling bandit setting.
- **Reduction to the worst sub-optimality gap (Lemma 2).** We bound the worst-case sub-optimality gap of the learned policy uniformly over $\mathcal{C}$ (Eq. (1)) solely in terms of the estimation error of the reward difference for an arbitrary context-arm-arm triplet. This reduction is independent of the construction of the confidence ellipsoid, so alternative constructions can be accommodated by modifying only Lemma 3.
- **Convergence rates (Theorem 1 and Theorem 2).** Combining these results yields a $\tilde{O}((\log T)^{\frac{1}{2}})$ improvement in the convergence rate over existing active contextual dueling bandit algorithms (Mehta et al., 2023; Das et al., 2024), while supporting a strictly larger class of reward functions.

The key differences between this paper and prior work.

- **Linear active contextual dueling bandits** (Mehta et al., 2023; Das et al., 2024). These works assume a linear reward function, with their Gram matrices constructed from *fixed* d-dimensional features. In our setting, the reward difference $f(\varphi(c,a)) - f(\varphi(c,b))$ is non-linear in the network parameters. We therefore construct the confidence set via an NTK linearization, which introduces

an approximation error $\varepsilon'_{w,T}$ that vanishes only as the network width w grows (Lemma 3); this error term is absent in the linear reward setting.

- **Neural contextual dueling bandits** (Verma et al., 2025). This is the closest prior work to ours in terms of the reward model, and we build on its confidence ellipsoid (Lemma 3). However, their learner selects a pair of arms for an *exogenously given* context and is evaluated using cumulative regret, corresponding to a regret minimization objective. In contrast, ours is an active learning problem: the learner *selects the context* and then the pair of arms, and we bound the worst-case sub-optimality gap of the policy learned after data collection. A cumulative regret guarantee does not directly yield a uniform sub-optimality guarantee over the context space.

B LEFTOVER PROOFS

To simplify the presentation, we use a common error probability of δ for all probabilistic statements. Our final results naturally follow by applying a union bound over all individual δ. Next, we will describe the key properties of positive definite matrices crucial for the subsequent proofs. These properties form the basis for several key parts of our analysis.

Fact 1 (Properties of a positive definite matrix)**.** *Let $V_0 = \lambda \mathbb{I}_d$, $V_T = V_0 + \sum_{s=1}^{T} z_s z_s^\top$ be a positive definite matrix, where $\lambda > 0$, $z_s \in \mathbb{R}^p$, and $\{Z_s = z_s z_s^\top\}_{s=1}^{T}$ is a finite adapted sequence of self-adjoint matrices, i.e., V_s and Z_s are $\mathcal{F}_s$-measurable for all s, where $\mathcal{F}_s$ represents all information available up to s. We use $\lambda_{\max}(V_T)$ and $\lambda_{\min}(V_T)$ to denote the maximum and minimum eigenvalue of matrix V_T. Then, the following properties hold for V_T:*

1. *Let $\delta \in (0,1)$, $\forall s \leq T:\ \|V_s - V_{s-1}\|^2 = \|z_s\|_2^4 = C_s$, where $\|A\|$ denotes the operator norm. Then, using Theorem 7.1 and Corollary 7.2 of Tropp (2012), with probability at least $1-\delta$,*
$$\mathbb{P}\left\{\lambda_{\max}\left(V_T - \mathbb{E}\left[V_T|\mathcal{F}_{T-1}\right]\right) \geq \sqrt{8\sum_{s=1}^{T} C_s \log\left(\frac{p}{\delta}\right)}\right\} \leq \delta.$$
2. *$\lambda_{\max}(V_T) = -\lambda_{\min}(-V_T)$.*
3. *Let $\lambda_i(V)$ be the i-th eigenvalue of matrix V. If W is any Hermitian matrix, then, from Weyl's inequality:*
   1. *$\lambda_i(V_T) + \lambda_{\min}(W) \leq \lambda_i(V_T + W) \leq \lambda_i(V_T) + \lambda_{\max}(W)$ and*
   2. *$\lambda_i(V_T) - \lambda_{\max}(W) \leq \lambda_i(V_T - W) \leq \lambda_i(V_T) - \lambda_{\min}(W)$.*
4. *For $a > 0:\ \|az\|_{V_T} = a\|z\|_{V_T}$ and $\lambda_i(aV_T) = a\lambda_i(V_T)$.*
5. *Let $\forall z \in \mathbb{R}^d:\ \|z\|_2 \leq L$. Then, $\max_{z\in\mathbb{R}^d} \|z\|_{V_T^{-1}} \leq \|z\|_2 \sqrt{\lambda_{\max}(V_T^{-1})} \leq \frac{L}{\sqrt{\lambda_{\min}(V_T)}}$.*

B.1 PROOF OF LEMMA 1

We now prove the upper bound on the maximum Mahalanobis norm of a vector from the fixed input space, measured with respect to the inverse of a positive definite Gram matrix defined by finite, adapted samples from the same input space.

Lemma 1. *Let $\{Z_s = z_s z_s^\top\}_{s=1}^{T}$ be a finite adapted sequence of self-adjoint matrices in $\mathbb{R}^d$ and $\mathcal{F}_s$ be the all information available up to the iteration s. Define $\mathbb{E}\left[z_s z_s^\top | \mathcal{F}_{s-1}\right] = \Sigma_s$, $\overline{\Sigma}_T = \sum_{s=1}^{T} \Sigma_s / T$, $V_0 = \frac{\lambda}{\kappa_\mu}\mathbb{I}_p$, $V_T = V_0 + \sum_{s=1}^{T} z_s z_s^\top$. Assume $\|z\|_2 \leq L$ for all $z \in \mathcal{Z} \subset \mathbb{R}^p$, and $\lambda_{\min}(A)$ denote the minimum eigenvalue of a matrix A. Then, with a probability at least $1-\delta$, $\max_{z\in\mathcal{Z}} \|z\|_{V_T^{-1}} \leq L/G_T$, where $G_T = \sqrt{\max\left\{\frac{\lambda}{\kappa_\mu}, \frac{\lambda}{\kappa_\mu} + T\lambda_{\min}(\overline{\Sigma}_T) - \sqrt{8TL^4 \log\left(p/\delta\right)}\right\}}$.*

Proof. Using Property 1 in Fact 1 with $Y_T - \mathbb{E}\left[Y_T|\mathcal{F}_{T-1}\right] = \mathbb{E}\left[V_T|\mathcal{F}_{T-1}\right] - V_T$, we have

$$\mathbb{P}\left\{\lambda_{\max}\left(\mathbb{E}\left[V_T|\mathcal{F}_{T-1}\right] - V_T\right) \geq \tau\right\} \leq p \exp\left(\frac{-\tau^2}{8\sum_{s=1}^{T} C_s}\right)$$

$$\implies \mathbb{P}\left\{-\lambda_{\min}\left(-(\mathbb{E}\left[V_T|\mathcal{F}_{T-1}\right]-V_T)\right)\geq\tau\right\}\leq p\exp\left(\frac{-\tau^2}{8\sum_{s=1}^{T}C_s}\right) \qquad \text{(Property 2 in Fact 1)}$$

$$\implies \mathbb{P}\left\{\lambda_{\min}\left(V_T-\mathbb{E}\left[V_T|\mathcal{F}_{T-1}\right]\right)\leq-\tau\right\}\leq p\exp\left(\frac{-\tau^2}{8\sum_{s=1}^{T}C_s}\right).$$

Using upper bound on $\lambda_{\min}\left(V_T-\mathbb{E}\left[V_T|\mathcal{F}_{T-1}\right]\right)$ from Property 3 in Fact 1, we get

$$\implies \mathbb{P}\left\{\lambda_{\min}(V_T)-\lambda_{\min}(\mathbb{E}\left[V_T|\mathcal{F}_{T-1}\right])\leq-\tau\right\}\leq p\exp\left(\frac{-\tau^2}{8\sum_{s=1}^{T}C_s}\right)$$

$$\implies \mathbb{P}\left\{\lambda_{\min}(V_T)\leq\lambda_{\min}(\mathbb{E}\left[V_T|\mathcal{F}_{T-1}\right])-\tau\right\}\leq p\exp\left(\frac{-\tau^2}{8\sum_{s=1}^{T}C_s}\right).$$

Note that $\mathbb{E}\left[V_T|\mathcal{F}_{T-1}\right]=\mathbb{E}\left[V_0+\sum_{t=1}^{T}z_sz_s^\top|\mathcal{F}_{T-1}\right]=\frac{\lambda}{\kappa_\mu}\mathbb{I}_p+\sum_{t=1}^{T}\mathbb{E}\left[z_sz_s^\top|\mathcal{F}_{T-1}\right]=\frac{\lambda}{\kappa_\mu}\mathbb{I}_p+\sum_{t=1}^{T}\Sigma_s=\frac{\lambda}{\kappa_\mu}\mathbb{I}_p+T\overline{\Sigma}_T$, where $\overline{\Sigma}_T=\sum_{t=1}^{T}\Sigma_s/T$. Using Property 4 in Fact 1, $\lambda_{\min}(\frac{\lambda}{\kappa_\mu}\mathbb{I}_p+T\overline{\Sigma}_T)=\frac{\lambda}{\kappa_\mu}+T\lambda_{\min}(\overline{\Sigma}_T)$. Using $\|z\|\leq L\implies C_s\leq L^4$, we get

$$\mathbb{P}\left\{\lambda_{\min}(V_T)\leq\max\left\{\frac{\lambda}{\kappa_\mu},\frac{\lambda}{\kappa_\mu}+T\lambda_{\min}(\overline{\Sigma}_T)-\sqrt{8TL^4\log\left(\frac{p}{\delta}\right)}\right\}\right\}\leq\delta.$$

Hence, $\lambda_{\min}(V_T)\geq\max\left\{\frac{\lambda}{\kappa_\mu},\frac{\lambda}{\kappa_\mu}+T\lambda_{\min}(\overline{\Sigma}_T)-\sqrt{8TL^4\log\left(p/\delta\right)}\right\}$ holds with probability at least $1-\delta$. Note that, by definition, the minimum eigenvalue of V_T satisfies $\lambda_{\min}(V_T)\geq\frac{\lambda}{\kappa_\mu}$. Using these facts with Property 5 in Fact 1, we now use to prove our key result as follows:

$$\begin{aligned}\max_{z\in\mathcal{Z}}\|z\|_{V_T^{-1}}&\leq L/\sqrt{\lambda_{\min}(V_T)}\\&\leq L/\sqrt{\max\left\{\frac{\lambda}{\kappa_\mu},\frac{\lambda}{\kappa_\mu}+T\lambda_{\min}(\overline{\Sigma}_T)-\sqrt{8TL^4\log\left(\frac{p}{\delta}\right)}\right\}}\\&=L/G_T\\\implies\max_{z\in\mathcal{Z}}\|z\|_{V_T^{-1}}&\leq L/G_T.\end{aligned}$$

□

We emphasize that adaptivity is the key difficulty in our setting: the samples $z_s{}_{s=1}^{T}$ forming V_T are chosen by the algorithm and are therefore generally dependent and non-i.i.d. This is why our analysis relies on the matrix Azuma inequality (Tropp, 2012), rather than standard matrix concentration results for i.i.d. samples. Existing analyses of active contextual dueling bandits (Mehta et al., 2023; Das et al., 2024) construct the Gram matrix from fixed d-dimensional features under a linear reward model, while neural dueling bandits (Verma et al., 2025) assume exogenously given contexts and therefore do not require a bound that is uniform over $\mathcal{Z}$. Neither setting establishes the uniform bound in Lemma 1. It is also useful to characterize when the bound is informative. The second term in G_T provides the nontrivial improvement over the trivial bound $\sqrt{\lambda/\kappa_\mu}$ once $T\lambda_{\min}(\overline{\Sigma}_T)\gtrsim\sqrt{8TL^4\log(p/\delta)}$. Therefore, the bound becomes informative after sufficiently many adaptively collected samples, relative to the feature dimension p.

B.2 Proof of Lemma 2 and Lemma 3

Our next result gives an upper bound of worst sub-optimality gap in terms of the upper bound of estimation error in the reward difference between any triple of context and two arms.

Lemma 2. *Let Assumption 1 hold and $D_T=\{x_s,a_{s,1},a_{s,2},y_s\}_{s=1}^{T}$ be the preference dataset collected up to the iteration T and $\hat{f}_T$ represent the estimate of latent reward function f learned from D_T. $\forall c\in\mathcal{C},\ a,b\in\mathcal{A}:\ \left|[f(\varphi(c,a))-f\left(\varphi(c,b)\right)]-\left[\hat{f}_T(\varphi(c,a))-\hat{f}_T(\varphi(c,b))\right]\right|\leq\beta_T(c,a,b)$ holds with probability at least $1-\delta$. If $a^\star=\operatorname{argmax}_{a\in\mathcal{A}}f(\varphi(c,a))$ and $\pi(c)$ is the arm selected by policy for context c, then, with a probability at least $1-\delta$, the worst sub-optimality gap for a policy that greedily selects an arm for a given context is upper bounded by: $\Delta_T^\pi\leq\max_{c\in\mathcal{C}}\beta_T(c,a^\star,\pi(c))$.*

Proof. Define $a^\star = \operatorname{argmax}_{a\in\mathcal{A}} f(\varphi(c,a))$. Here, a ranges over the entire arm set $\mathcal{A}$ and is not tied to iteration t. We use $\varphi(c,a)$ for the generic feature map of an arbitrary context-arm pair, and reserve the subscript t for quantities selected by the algorithm at iteration t, such as c_t, $a_{t,1}$, and $a_{t,2}$. Recall the definition of worst suboptimality across all contexts, which is :

$$\begin{aligned}
\Delta^{\pi}_{\mathcal{D}_T} &= \max_{c\in\mathcal{C}} \left[\max_{a\in\mathcal{A}} f(\varphi(c,a)) - f\left(\varphi(c,\pi(c))\right)\right] \\
&= \max_{c\in\mathcal{C}} \left[f(\varphi(c,a^\star)) - f\left(\varphi(c,\pi(c))\right)\right] \\
&= \max_{c\in\mathcal{C}} \left[f(\varphi(c,a^\star)) - f\left(\varphi(c,\pi(c))\right) + \hat{f}_T(\varphi(c,a^\star)) - \hat{f}_T(\varphi(c,a^\star))\right] \\
&\le \max_{c\in\mathcal{C}} \left|[f(\varphi(c,a^\star)) - f\left(\varphi(c,\pi(c))\right)] + \left[\hat{f}_T(\varphi(c,\pi(c))) - \hat{f}_T(\varphi(c,a^\star))\right]\right| \\
&= \max_{c\in\mathcal{C}} \left|[f(\varphi(c,a^\star)) - f\left(\varphi(c,\pi(c))\right)] - \left[\hat{f}_T(\varphi(c,a^\star)) - \hat{f}_T(\varphi(c,\pi(c)))\right]\right| \\
\implies \Delta^{\pi}_{\mathcal{D}_T} &\le \max_{c\in\mathcal{C}} \beta_T(c,a^\star,\pi(c)).
\end{aligned}$$

The inequality follows from the fact we have greedy policy, i.e., $\pi(c) = \operatorname{argmax}_{a\in\mathcal{A}} \hat{f}_T(\varphi(c,a))$ for any context c. Therefore, if $\pi(c) \neq a^\star$, then $\hat{f}_T(\varphi(c,\pi(c))) \ge \hat{f}_T(\varphi(c,a^\star))$ must hold. □

Lemma 3. *Let Assumption 1 hold and* $\nu_T = (\beta_T + B\sqrt{\lambda/\kappa_\mu} + 1)\sqrt{\kappa_\mu/\lambda}$, *where* $\beta_T = (1/\kappa_\mu)\sqrt{\widetilde{d} + 2\log(1/\delta)}$ *and* $\delta \in (0,1)$. *If* $w \ge$ *poly*$(T, L, K, 1/\kappa_\mu, L_\mu, 1/\lambda_0, 1/\lambda, \log(1/\delta))$, *then, with a probability of at least* $1-\delta$, *for* `Neural-ADB` *with (i) UCB-based arm selection strategy, for all* $c \in \mathcal{C} : \beta_T(c,a,b) = \nu_T \sigma_T(c,a^\star,\pi(c)) + 2\varepsilon'_{w,T}$, *(ii) TS-based arm selection strategy, for all* $c \in \mathcal{C} : \beta_T(c,a,b) = \nu_T \log\left(KT^2\right)\sigma_T(c,a^\star,\pi(c)) + 2\varepsilon'_{w,T}$, *where* K *denotes the maximum number of arms available in each iteration, and* $\varepsilon'_{w,T} = C_2 w^{-1/6}\sqrt{\log w} L^3 \left(T/\lambda\right)^{4/3}$ *for some absolute constant* $C_2 > 0$, *is the approximation error that decreases as the width of the NN* (w) *increases.*

Proof. Recall that we are using the arm-selection strategies proposed in (Verma et al., 2025). Since their confidence bounds hold for any adapted sequence of contexts, the proof of the first part follows directly from Theorem 1 in (Verma et al., 2025), while the second part follows from Lemma 10 together with Eq. (27) of (Verma et al., 2025). □

The constant $B > 0$ in Lemma 3 appearing in ν_T arises from the confidence ellipsoid construction of Verma et al. (2025) and bounds the RKHS norm of the latent reward function induced by the NTK. Specifically, B is such that there exists a parameter $\theta^\star$ satisfying $f(\varphi(c,a)) = \langle g(\varphi(c,a);\theta_0)/\sqrt{w}, , \theta^\star - \theta_0\rangle$ for all $c \in \mathcal{C}$ and $a \in \mathcal{A}$, with $\sqrt{w}\left\|\theta^\star - \theta_0\right\|_2 \le B$. Thus, B plays the same role as the norm bound on the unknown parameter in linear contextual and dueling bandits (Abbasi-Yadkori et al., 2011; Das et al., 2024).

We next discuss the sensitivity of our guarantees to these assumptions in Remark 1.

Remark 1 (Sensitivity to the assumptions)**.** *We now discuss how sensitive* `Neural-ADB` *and its guarantees are to each of the assumptions we make.*

- ***Preference model.*** *Our analysis does not require the BTL model specifically. As noted in Sec. 3, our results apply to any preference model satisfying the stochastic transitivity assumption, including the Thurstone–Mosteller and Exponential Noise models (Bengs et al., 2022). Assumption Assumption 1 further requires the link function to be non-flat, with* $\kappa_\mu > 0$. *The quantity* κ_μ *characterizes the degree of flatness and enters our guarantees only through* β_T *and* ν_T *(Lemma 3). Consequently, the guarantees degrade continuously as* $\kappa_\mu \to 0$, *rather than becoming invalid abruptly.*

- ***Finite arm set.*** *The number of arms* K *enters our analysis in two ways: implicitly through the effective dimension* $\widetilde{d}$ *in Eq. (8), which aggregates the contributions of all pairwise arm combinations, and explicitly through the* $\log(KT^2)$ *factor in the TS branch of Lemma 3. Thus, the explicit dependence on* K *is only logarithmic, allowing moderate growth in the number of arms. Extending the analysis to a continuous arm space would require an additional covering argument, which is beyond the scope of this work. Empirically, Figs. 2 and 3 (bottom rows) and Fig. 5 illustrate the effect of increasing* K *on performance.*

- ***NTK and effective dimension.*** *We adopt the arm selection strategies from existing neural dueling bandit algorithms (Verma et al., 2025), which assume $\tilde{d} = o(T)$. In some settings, $\tilde{d} = \Omega(T)$ (Ban et al., 2022), in which case the resulting convergence rate may remain constant. Our focus is on leveraging neural networks to estimate non-linear reward functions in active contextual dueling bandits. Since neural dueling bandit methods primarily determine the arm selection strategy, their variants can be incorporated into our framework by appropriately modifying Lemma 3. Two additional conditions are specific to our analysis. First, Lemma 1 is informative only when $T\lambda_{\min}(\overline{\Sigma}T)$ dominates $\sqrt{8TL^4\log(p/\delta)}$; otherwise, G_T coincides with the trivial bound $\sqrt{\lambda/\kappa\mu}$. Second, interpreting Theorem 1 and Theorem 2 as $\tilde{O}(T^{-1/2})$ implicitly assumes that $\lambda_{\min}(\overline{\Sigma}_T)$ is bounded away from zero.*

B.3 Proof of Theorem 1 and Theorem 2

Equipped with Lemma 1, Lemma 2, and Lemma 3, we will next prove the upper bound on the worst sub-optimality gap for a policy learned by `Neural-ADB` while using UCB- and TS-based arm selection strategy for a given context.

Theorem 1 (UCB)**.** *Let the conditions in Lemma 1 and Lemma 3 hold. Then, with a probability of at least $1-\delta$, the worst sub-optimality gap of* `Neural-ADB` *when using the UCB-based arm selection is upper bounded by* $\Delta_T^\pi \leq \left(\frac{\nu_T L}{G_T}\right)\sqrt{\frac{\lambda}{\kappa_\mu w}} + 2\varepsilon'_{w,T} = \tilde{O}\left(\frac{\nu_T}{\sqrt{T\lambda_{\min}(\overline{\Sigma}_T)}}\right) = \tilde{O}\left(\sqrt{\frac{\tilde{d}}{T}}\right)$.

Proof. Using Lemma 3 and setting value of $\beta_T(c, a^\star, \pi(c))$ using Lemma 3 and Eq. (6), we have

$$\begin{aligned}
\Delta_T^\pi &\leq \max_{c\in\mathcal{C}} \beta_T(c, a^\star, \pi(c)) \quad \text{(from Lemma 2)} \\
&\leq \max_{c\in\mathcal{C}} \left(\nu_T \sigma_T(c, a^\star, \pi(c)) + 2\varepsilon'_{w,T}\right). \quad \text{(from Lemma 3)}
\end{aligned}$$

As ν_T and $\varepsilon'_{w,T}$ independent of context c, we get

$$\begin{aligned}
\Delta_T^\pi &\leq \nu_T \max_{c\in\mathcal{C}} \left(\sigma_T(c, a^\star, \pi(c))\right) + 2\varepsilon'_{w,T} \\
&= \nu_T \max_{c\in\mathcal{C}} \left(\sqrt{\frac{\lambda}{\kappa_\mu}} \left\| \frac{\Phi(c, a^\star) - \Phi(c, \pi(c))}{\sqrt{w}} \right\|_{V_T^{-1}}\right) + 2\varepsilon'_{w,T} \quad \text{(using Eq. (6))} \\
&= \nu_T \max_{c\in\mathcal{C}} \left(\sqrt{\frac{\lambda}{\kappa_\mu w}} \left\| \Phi(c, a^\star) - \Phi(c, \pi(c)) \right\|_{V_T^{-1}}\right) + 2\varepsilon'_{w,T} \quad \text{(Property 4 in Fact 1)} \\
&= \nu_T \sqrt{\frac{\lambda}{\kappa_\mu w}} \max_{c\in\mathcal{C}} \left(\left\| \Phi(c, a^\star) - \Phi(c, \pi(c)) \right\|_{V_T^{-1}}\right) + 2\varepsilon'_{w,T} \\
&\leq \nu_T \sqrt{\frac{\lambda}{\kappa_\mu w}} \left(\frac{L}{\sqrt{\max\left\{\frac{\lambda}{\kappa_\mu}, \frac{\lambda}{\kappa_\mu} + T\lambda_{\min}(\overline{\Sigma}_T) - \sqrt{8TL^4\log\left(\frac{p}{\delta}\right)}\right\}}}\right) + 2\varepsilon'_{w,T} \quad \text{(Lemma 1)} \\
&\leq \tilde{O}\left(\frac{\nu_T}{\sqrt{T\lambda_{\min}(\overline{\Sigma}_T)}}\right) = \tilde{O}\left(\sqrt{\frac{\tilde{d}}{T}}\right).
\end{aligned}$$

□

Theorem 2 (TS)**.** *Let the conditions in Lemma 1 and Lemma 3 hold. Then, with a probability of at least $1-\delta$, the worst sub-optimality gap of* `Neural-ADB` *when using the TS-based arm selection is upper bounded by* $\Delta_T^\pi \leq \left(\frac{\nu_T L \log\left(KT^2\right)}{G_T}\right)\sqrt{\frac{\lambda}{\kappa_\mu w}} + 2\varepsilon'_{w,T} = \tilde{O}\left(\frac{\nu_T}{\sqrt{T\lambda_{\min}(\overline{\Sigma}_T)}}\right) = \tilde{O}\left(\sqrt{\frac{\tilde{d}}{T}}\right)$.

Proof. Using Lemma 3 and setting value of $\beta_T(c, a^\star, \pi(c))$ using Lemma 3 and Eq. (6), we have

$$\Delta_T^\pi \leq \max_{c\in\mathcal{C}} \beta_T(c, a^\star, \pi(c)) \qquad \text{(from Lemma 2)}$$

$$\leq \max_{c\in\mathcal{C}} \left(\nu_T \log\left(KT^2\right) \sigma_T(c, a^\star, \pi(c)) + 2\varepsilon'_{w,T}\right). \qquad \text{(from Lemma 3)}$$

The value of ν_T and $\varepsilon'_{w,T}$ independent of context c. By following similar steps to those in the proof of Theorem 1, we have

$$\begin{aligned}
\Delta_T^\pi &\leq \nu_T \log\left(KT^2\right) \max_{c\in\mathcal{C}} \left(\sigma_T(c, a^\star, \pi(c))\right) + 2\varepsilon'_{w,T} \\
&= \nu_T \log\left(KT^2\right) \max_{c\in\mathcal{C}} \left(\sqrt{\frac{\lambda}{\kappa_\mu}} \left\| \frac{\Phi(c, a^\star) - \Phi(c, \pi(c))}{\sqrt{w}} \right\|_{V_T^{-1}}\right) + 2\varepsilon'_{w,T} \\
&= \nu_T \log\left(KT^2\right) \max_{c\in\mathcal{C}} \left(\sqrt{\frac{\lambda}{\kappa_\mu w}} \left\| \Phi(c, a^\star) - \Phi(c, \pi(c)) \right\|_{V_T^{-1}}\right) + 2\varepsilon'_{w,T} \\
&= \nu_T \log\left(KT^2\right) \sqrt{\frac{\lambda}{\kappa_\mu w}} \max_{c\in\mathcal{C}} \left(\left\| \Phi(c, a^\star) - \Phi(c, \pi(c)) \right\|_{V_T^{-1}}\right) + 2\varepsilon'_{w,T} \\
&\leq \nu_T \log\left(KT^2\right) \sqrt{\frac{\lambda}{\kappa_\mu w}} \left(\frac{L}{\sqrt{\max\left\{\frac{\lambda}{\kappa_\mu}, \frac{\lambda}{\kappa_\mu} + T\lambda_{\min}(\overline{\Sigma}_T) - \sqrt{8TL^4 \log\left(\frac{p}{\delta}\right)}\right\}}}\right) + 2\varepsilon'_{w,T} \\
&\leq \tilde{O}\left(\frac{\nu_T}{\sqrt{T\lambda_{\min}(\overline{\Sigma}_T)}}\right) = \tilde{O}\left(\sqrt{\frac{\tilde{d}}{T}}\right). \qquad \square
\end{aligned}$$

This improvement is a direct consequence of Lemma 1 and applies to a strictly larger class of reward functions, since (Mehta et al., 2023; Das et al., 2024) assume a linear latent reward. We also note that interpreting the bounds as $\tilde{O}((\tilde{d}/T)^{1/2})$ implicitly treats $\lambda_{\min}(\overline{\Sigma}_T)$ as bounded away from zero. This holds when the adaptively selected context-arm differences continue to excite all directions of the feature space, which is precisely the exploration encouraged by the context selection rule in Eq. (3).

C Additional Experimental Details and Results

We first want to highlight that we do *not* collect human preference feedback online. Following prior work in this setting (Mehta et al., 2023; Das et al., 2024; Verma et al., 2025), a predefined reward function in the synthetic experiments or a pretrained reward model trained on real human preferences in the LLM alignment experiment serves as a simulated labeler. Crucially, the underlying reward function or reward model is not accessible to the algorithms: `Neural-ADB` and all baselines observe only the one-bit preference feedback y_t for the selected context and arm pair, exactly as they would from a human labeler. Thus, the simulator replaces the human labeler only for the purpose of providing preference feedback at each query, while preserving the interaction protocol and feedback available to the algorithms. We discuss this evaluation choice and other limitations in Sec. 6.

C.1 Experimental Details

Computational resources used for experiments. All non-LLM experiments were conducted on a server equipped with an AMD EPYC 7543 32-Core Processor, 256GB of RAM, and 8 NVIDIA GeForce RTX 3080 GPUs. The LLM alignment experiment was conducted on NVIDIA L40 GPUs.

Baselines. Since the baselines differ along three independent axes, we specify the exact configuration of each in Tab. 1, allowing Fig. 1 to also serve as a component-wise ablation. Reading Fig. 1 together with Tab. 1 isolates the contribution of each component: (i) *the neural reward model*, by comparing APO with `Neural-ADB` (APO) and AE-Borda with AE-Borda (NN), where each pair shares the same context and arm selection rules and differs only in the reward model; (ii) *the arm selection strategy*, by comparing `Neural-ADB` (APO) with `Neural-ADB` (UCB)/(TS), which share the neural reward model and the context selection rule in Eq. (3) and differ only in arm selection; and (iii) *the context selection rule*, by comparing AE-Borda (NN) with `Neural-ADB` (UCB)/(TS) and the Random baseline, which samples the context and both arms uniformly. Thus, the gap between

APO and `Neural-ADB` (APO) reflects the contribution of non-linear reward estimation, while the additional gap between `Neural-ADB` (APO) and `Neural-ADB` (UCB)/(TS) isolates the effect of the arm selection strategy. These comparisons show that reducing the worst-case sub-optimality gap requires informative selection of both the context and the arm pair.

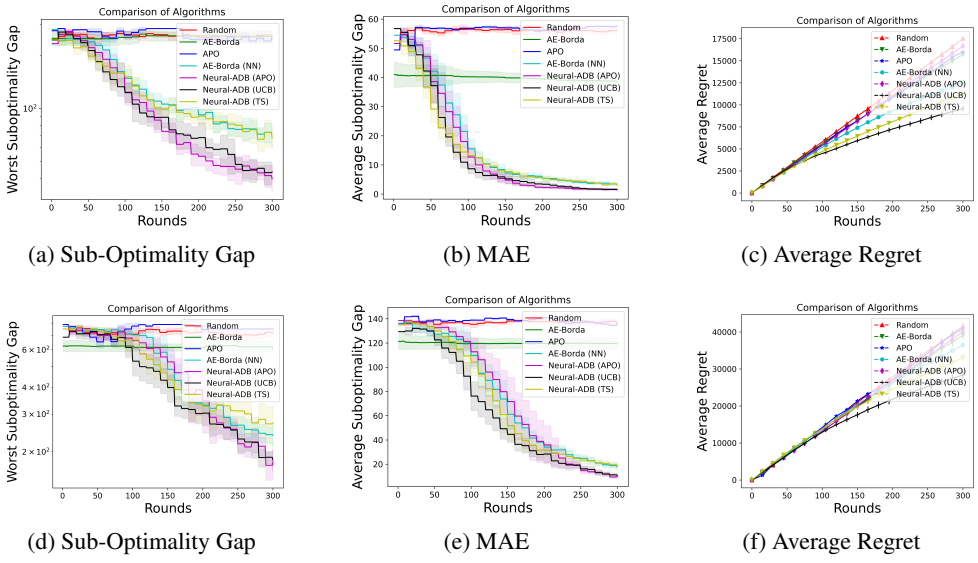
Table 1: Exact configuration of every method compared in Fig. 1, along the three axes that distinguish them: how the context is selected, how the latent reward function is estimated, and how the pair of arms is selected. Rows marked "as in" keep the corresponding rule of the original work unchanged, so that each adjacent pair of rows differs in exactly one axis, which is what makes the comparison a component-wise ablation.

Method	**Context selection**	**Reward model**	**Arm selection**
Random	Uniform	—	Uniform pair
AE-DPO (Mehta et al., 2023)	As in (Mehta et al., 2023)	Linear	As in (Mehta et al., 2023)
AE-Borda (Mehta et al., 2023)	As in (Mehta et al., 2023)	Linear	As in (Mehta et al., 2023)
AE-Borda (NN)	As in (Mehta et al., 2023)	Neural network	As in (Mehta et al., 2023)
APO (Das et al., 2024)	Eq. (3)	Linear	Pair attaining the max in Eq. (3)
`Neural-ADB` (APO)	Eq. (3)	Neural network	Pair attaining the max in Eq. (3)
`Neural-ADB` (UCB)	Eq. (3)	Neural network	Eq. (4) and Eq. (5)
`Neural-ADB` (TS)	Eq. (3)	Neural network	Eq. (4) and Eq. (7)

Practical considerations. Based on the neural tangent kernel (NTK) theory (Jacot et al., 2018), the initial gradient $g(x;\theta_0)$ can be used as the original feature vector x as $g(x;\theta_0)$ effectively represents the random Fourier features of the NTK. To make our algorithm more practical, we use common practices in neural bandits (Zhou et al., 2020; Zhang et al., 2021; Verma et al., 2025). Specifically, we replaced the theoretical regularization parameter $\frac{1}{2}w\lambda \left\|\theta-\theta_0\right\|_2^2$ (where w is the NN's width) with the simpler $\lambda \left\|\theta\right\|_2^2$ in the loss function (defined in Eq. (2)) that is used to train our NN. We retrain the neural network after every 20 rounds for 50 gradient steps across all experiments.

C.2 Additional Experimental Results

Next, we present the additional experiment results comparing the performance of `Neural-ADB` varying input dimension d (Fig. 4) and different numbers of arms K (Fig. 5).

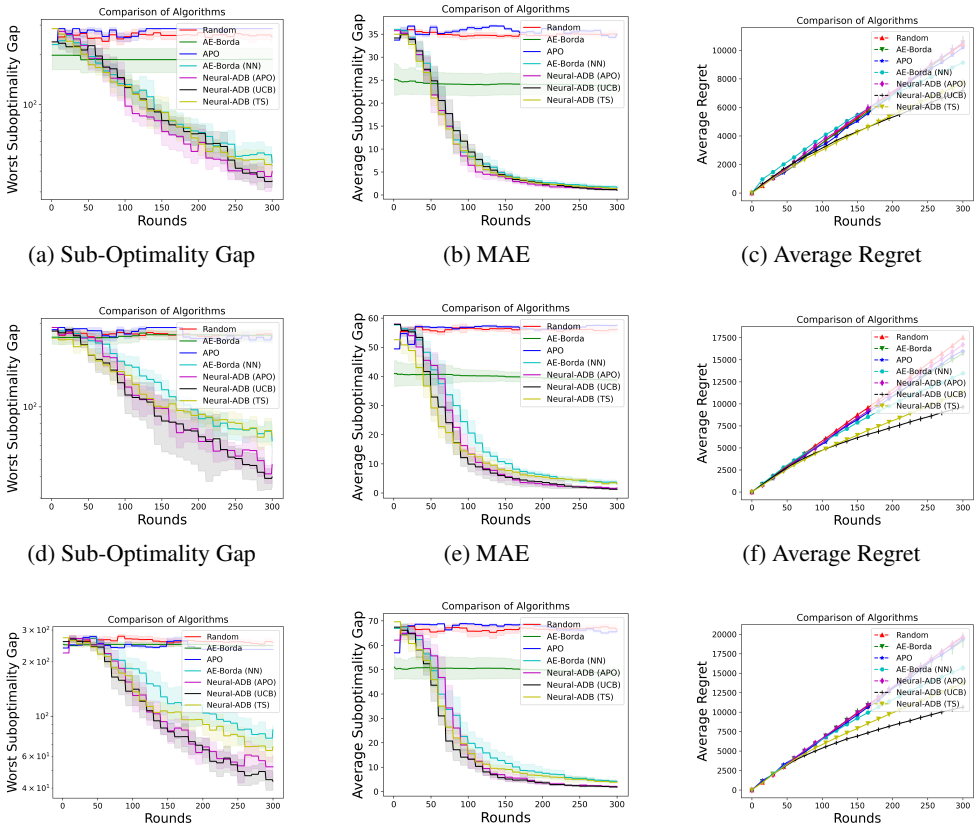


(a) Sub-Optimality Gap (b) MAE (c) Average Regret

(d) Sub-Optimality Gap (e) MAE (f) Average Regret

Figure 4: Performance comparison across different input dimensions d: $d = 20$ (first row) and $d = 40$ (second row). We set the number of arms to 10 and use the Square function for all experiments.

Performance vs. neural network size. To investigate how performance varies with different neural network (NN) sizes, we used the Square and Cosine functions defined in the paper. We varied either



(a) Sub-Optimality Gap (b) MAE (c) Average Regret

(d) Sub-Optimality Gap (e) MAE (f) Average Regret

(g) Sub-Optimality Gap (h) MAE (i) Average Regret

Figure 5: Performance comparison across different numbers of arms K: $K = 5$ (top row), $K = 10$ (middle row), and $K = 15$ (bottom row). We set the input dimension to 20 and use the Square function for all experiments.

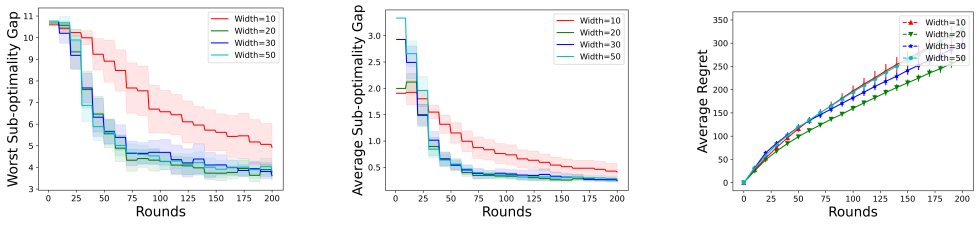
the number of layers (with width = 32) or the width of the NN (with 2 layers), while keeping all other variables consistent with those in the paper. As shown in Fig. 6, we observed that selecting the appropriate size of NN is crucial for the given problem. Using a large NN for a simple problem leads to poor performance due to high bias in the estimation, while a smaller NN may not accurately be able to estimate the complex non-linear function.

C.3 Computational Efficiency.

To discuss the computational efficiency of `Neural-ADB`, we follow the approach of (Verma et al., 2025) and consider the following two key aspects: size of the neural network and then the number of contexts and arms.

Size of the neural network. The primary computational cost in `Neural-ADB` arises from the neural network (NN) used to approximate the latent non-linear reward function. Given a context-arm feature vector of dimension d, an NN with D hidden layers and w neurons per layer incurs an inference cost of $O(dw + Dw^2 + w)$ per context-arm pair. The total number of parameters in the NN is $p = dw + Dw^2 + w$, and the training time per iteration is $O(\mathcal{E}\mathcal{P}Dw^2)$, where $\mathcal{E}$ is the number of training epochs and $\mathcal{P}$ is the number of observed context-arm pairs. Choosing an appropriate NN size is critical, as NNs that are too small may fail to accurately approximate the underlying non-linear reward function, while excessively large NNs can result in substantial training and inference overhead.

Number of contexts and arms. Let K denote the number of arms and p the total number of NN parameters. Since `Neural-ADB` uses NN gradients as context-arm features, the cost of computing



(a) Sub-Optimality Gap (b) MAE (c) Average Regret

(d) Sub-Optimality Gap (e) MAE (f) Average Regret

Figure 6: We compare performance across different neural network widths (first row) and numbers of hidden layers (second row), using the Square function in all experiments. All other parameters are kept fixed, except that the width is set to 32 when varying the number of layers.

gradients for all arms per context is $O(K^2dp)$, where d is the dimension of the context-arm feature vector. The cost of computing reward estimates and confidence terms for all context-arm pairs is $O(K^2p)$ and $O(K^2p^2)$, respectively. For arm selection, the first selection step requires $O(Kp + K)$, consisting of reward estimation for all arms ($O(Kp)$) and then identifying the arm with the highest estimated reward ($O(K)$). The second arm selection incurs a cost of $O(Kp + (K-1)p^2)$, including reward estimation $O(Kp)$ and confidence term computation $O((K-1)p^2)$ relative to the first selected arm. Thus, the total computational cost for selecting a pair of arms per context is $O(K^2dp + K^2p^2)$. Since each context-arm pair is independent, gradients, reward estimates, and optimistic terms can be computed in parallel, reducing the overall cost to $O(dp + p^2)$ for each iteration.

On the p^2 term. The p^2 term originates from the confidence term σ^2_{t-1} in Eq. (6), which involves a Mahalanobis norm with respect to $V_{t-1}^{-1} \in \mathbb{R}^{p\times p}$. In our analysis, p denotes the number of NN parameters and therefore the dimension of the gradient features. These quantities need not coincide in practice. In the LLM alignment experiment, for example, we project the LoRA gradients into an 8,192-dimensional space (Sec. C.4), so the effective p in the theoretical bound is the projection dimension, not the number of parameters in the policy model. Moreover, `Neural-ADB` is modular with respect to the confidence ellipsoid (Remark 1): alternative approaches can be incorporated by modifying only Lemma 3. This includes shallow-exploration (Xu et al., 2022), where exploration is restricted to the final network layer and the effective dimension p is the width of that layer.

Empirical scalability in the feature dimension and the number of arms. Our synthetic problem instances use up to $K = 20$ arms (bottom rows of Figs. 2 and 3 and Fig. 5) and up to 40-dimensional context-arm features (Fig. 4), while the Mahalanobis and confidence computations in the LLM alignment experiment operate on gradient features projected to $p = 8{,}192$ dimensions. Tables 2 and 3 reports the resulting analytical costs. Three points are worth highlighting. First, *the memory used by the confidence computation is independent of K, T, and the policy model size*: V_t^{-1} is a single $p \times p$ matrix determined by the projection dimension, while gradient features are streamed rather than stored (approximately 98 MB for the 3,000 comparisons in an iteration). Projecting the LoRA gradients to 8,192 dimensions therefore decouples the dimension p from the number of policy-model parameters, making the confidence computation tractable for a 4B-parameter policy. Second, *the additional arithmetic is modest relative to generation*: one forward token of a 4B-parameter model requires roughly 8 GFLOP, so the 403 GFLOP required per iteration corresponds to approximately 50 generated tokens. The 3,000 confidence terms are computed in a single batched matrix operation using the same V_t^{-1}, yielding the parallel implementation described above. Third, *the dominant computational cost is obtaining the gradient features*, which requires a backward pass for each candidate response. Thus, `Neural-ADB` adds a confidence-computation overhead on top of the

existing generation-and-gradient pipeline, rather than introducing a separate asymptotic dependence on the policy-model parameter count.

Table 2: Cost of the confidence computations as a function of the gradient-feature dimension p, for the 3,000 pairwise comparisons formed in one iteration of the LLM alignment experiment.

p	V_t^{-1} memory (fp32)	Confidence terms (3,000 comparisons)
4096	67 MB	101 GFLOP
8192 (used)	268 MB	403 GFLOP
16384	1.07 GB	1.61 TFLOP

Table 3: Cost of the confidence computations as a function of the number of arms K, with $p = 8192$ and 1,000 contexts per iteration. The last column converts the cost into forward tokens of the 4B policy model at approximately 8 GFLOP per token.

K	Comparisons	Confidence terms per iteration	Equivalent generated tokens
3 (used)	3,000	0.40 TFLOP	≈ 50
5	10,000	1.34 TFLOP	≈ 170
10	45,000	6.04 TFLOP	≈ 760
20	190,000	25.5 TFLOP	$\approx 3{,}200$

C.4 Preliminary LLM Alignment Experiment

Sample efficient LLM alignment. We use the TLDR summarization dataset (Völske et al., 2017; Stiennon et al., 2020), which contains posts from Reddit and the corresponding summarization written by humans. This dataset also contains human preference feedback from human annotators and will be used later as an oracle to obtain real human preference feedback. We perform experiments using Qwen3-4B (Yang et al., 2025). We evaluate algorithmic performance using win-rate, a practical metric that measures how frequently the aligned system outperforms competing alternatives. As shown in the Fig. 7, `Neural-ADB` (UCB) achieves the highest win-rate among the key baselines.

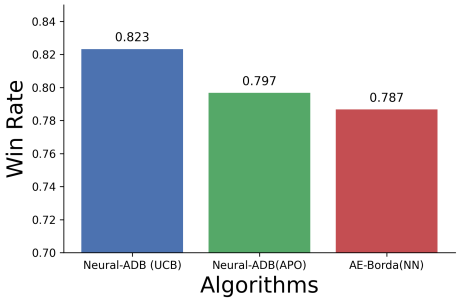


Figure 7: **Comparison of algorithms' win rates for active LLM alignment.** `Neural-ADB` (UCB) achieves the highest win rate among the evaluated methods.

Collecting human preference feedback. Responses generated by the updated model at each iteration are not included in the original preference dataset and therefore lack direct human feedback. To make iterative training feasible, we rely on a pretrained reward model as a proxy for human preference. This reward model is trained on the original human preference data and is used as an oracle to provide preference signals for newly generated responses at each iteration.[4]

We want to highlight that this reward model is not exposed to any algorithm: `Neural-ADB` and all baselines in Fig. 7 observe only the binary preference feedback $y_t \in \{0, 1\}$ for the selected triplet. No method has access to the reward model parameters, their scalar scores, or feedback for unselected triplets. Thus, each method interacts with the reward model only through the same one-bit feedback channel that would be available from a human labeler. Moreover, all methods are evaluated against the same simulator, so the comparison is conducted under a common feedback source and isolates differences in their selection strategies under that feedback model. This does not eliminate potential simulator-specific biases, which may interact with the triplets selected by different methods. The proxy therefore stands in for the *cost* of obtaining a preference label rather than for the existence of a human annotator. A remaining limitation is that both the in-loop labels and the win-rate evaluation are generated by the same reward model (Sec. C.4), which may introduce evaluator-model coupling. We discuss this limitation and the use of an independent judge in Sec. 6.

[4]We use publicly available pretrained reward models from HuggingFace: the model from (OpenAssistant, 2024) for the TLDR dataset.

Evaluation. For each task, the trained LLM generates multiple responses for 100 prompts randomly sampled from the dataset. We then compute the reward for each prompt–response pair and report the average reward across all samples. Higher average reward indicates better alignment with human preferences, as the reward model is trained on real human feedback.

Hyperparameters. We initialize the LLM using supervised fine-tuning on the TLDR SFT dataset for one epoch with a learning rate of 2×10^{-5}. In each iteration, we randomly sample 1000 prompts and generate three responses per prompt, yielding $\binom{3}{2} = 3$ pairwise comparisons per prompt and a total of $3{,}000$ training examples in dataset D_t. From D_t, we select 50 data points per iteration using different selection strategies. The model is then trained for 4 epochs using the DPO objective with a learning rate of 10^{-4}. For LoRA, we use a rank of 128 with a scaling factor of $\alpha = 512$. All LoRA gradients are projected into an $8{,}192$-dimensional space, providing a favorable trade-off between performance and computational cost, as demonstrated in later experiments.

Win-rate. We evaluate alignment performance using *win-rate*, a pairwise preference-based metric. Given a set of evaluation prompts, we generate one response per prompt from the evaluated model and one response from a fixed reference model under the same prompt. The oracle reward model provides a binary judgment indicating which of the two responses is preferred. The win-rate is defined as the fraction of prompts for which the response generated by the evaluated model is preferred over the reference model's response. Intuitively, win-rate measures how often the evaluated model wins in direct pairwise comparisons against the reference model; values above 0.5 indicate improved alignment with human preferences relative to the reference.

D Limitations and Future Work.

We close by stating the limitations of this work explicitly. First, our experiments do not collect human preference feedback online. The synthetic reward functions and the pretrained reward model in the LLM alignment experiment serve as simulated labelers that every method observes only through the same one-bit preference channel (Sec. 5). We therefore approximate the cost of obtaining human feedback, rather than human feedback itself; validating `Neural-ADB` with humans in the loop remains future work. Second, our empirical evaluation is limited to synthetic reward functions and a single LLM alignment task. Evaluation on larger and more realistic preference datasets and across a broader range of alignment tasks is an important next step. Third, in the LLM alignment experiment, both the training labels and the win-rate evaluation are generated by the same reward model (Sec. C.4). Re-evaluating the resulting checkpoints with an independent judge, such as a separate reward model or an LLM-based judge, would provide a stronger assessment of alignment quality. Fourth, our theoretical guarantees assume a finite arm set and the effective-dimension condition $\widetilde{d} = o(T)$ (Remark 1). Extending it to continuous arm spaces would require an additional covering argument. Finally, studying `Neural-ADB` under model misspecification, including violations of Assumption 1 or the NTK approximation assumptions, remains an open direction.