Federated Linear Dueling Bandits

Xuhan Huang[1], Yan Hu[2, 3], Zhiyan Li[2], Zhiyong Wang[4], Zhongxiang Dai[2*]

[1] School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen
[2] School of Data Science, The Chinese University of Hong Kong, Shenzhen
[3]National Health Data Institute, Shenzhen
[4]The University of Edinburgh

Abstract

Contextual linear dueling bandits have recently garnered significant attention due to their widespread applications in important domains such as recommender systems and large language models. Classical dueling bandit algorithms are typically only applicable to a single agent. However, many applications of dueling bandits involve multiple agents who wish to collaborate for improved performance yet are unwilling to share their data. This motivates us to draw inspirations from *federated learning*, which involves multiple agents aiming to collaboratively train their neural networks via gradient descent (GD) without sharing their raw data. Previous works have developed federated linear bandit algorithms which rely on closed-form updates of the bandit parameters (e.g., the linear function parameters) to achieve collaboration. However, in linear dueling bandits, the linear function parameters *lack a closed-form expression* and their estimation requires minimizing a loss function. This renders these previous methods inapplicable. In this work, we overcome this challenge through an innovative and principled combination of online gradient descent (OGD, for minimizing the loss function to estimate the linear function parameters) and federated learning, hence introducing our *federated linear dueling bandit with OGD* (`FLDB-OGD`) algorithm. Through rigorous theoretical analysis, we prove that `FLDB-OGD` enjoys a sub-linear upper bound on its cumulative regret and demonstrate a theoretical trade-off between regret and communication complexity. We conduct empirical experiments to demonstrate the effectiveness of `FLDB-OGD` and reveal valuable insights, such as the benefit of a larger number of agents, the regret-communication trade-off, among others.

1 Introduction

Contextual dueling bandits (Saha 2021; Saha and Krishnamurthy 2022; Bengs, Saha, and Hüllermeier 2022; Li, Zhao, and Gu 2024) have recently attracted significant attention due to their use in real-world systems such as recommender systems (Yue et al. 2012) and large language models (LLMs) (Lin et al. 2024; Ji, He, and Gu 2024). In each iteration, an agent receives a d-dimensional *context* vector and K *arms*, selects a pair of arms, and observes a binary preference indicating which arm is preferred (Bengs, Saha, and Hüllermeier 2022). For instance, when optimizing LLM responses, the context is a prompt, the K arms are generated responses, a pair is selected, and the user indicates the preferred response (Lin et al. 2024). To guide pair selection, a surrogate function models the *latent reward function* (Sec. 2), typically assumed to be linear: $f(x) = \theta^\top \phi(x)$ for unknown $\theta \in \mathbb{R}^d$ and known feature map $\phi(\cdot) \in \mathbb{R}^d$. This setting is known as the contextual *linear dueling bandit* problem.

Classical contextual dueling bandit algorithms address only single-agent settings, whereas many practical applications involve multiple agents and thus creates opportunities to enhance performance via *collaboration*. However, such agents often face privacy concerns and are *unwilling to share their data*, including selected arms and preference observations. For example, users optimizing LLM responses may wish to collaborate while keeping their chosen responses and preference feedback private. These constraints naturally align with *federated learning* (FL), which enables collaborative training without sharing raw data (McMahan et al. 2017). In each FL round, agents compute local gradients or parameters, send them to a central server for aggregation (e.g., averaging), and receive the aggregated result for further local updates (McMahan et al. 2017).

To extend contextual linear dueling bandits to the federated setting, a natural starting point is the literature on federated contextual bandits (Shi and Shen 2021). In particular, Wang et al. (2019) proposed a federated contextual linear bandit algorithm in which agents are only required to share some sufficient statistics, including a d-dimensional vector and a $(d \times d)$ matrix. Crucially, both components of the linear upper confidence bound (Lin-UCB) policy (Abbasi-Yadkori, Pál, and Szepesvári 2011)—(a) the estimate of θ and (b) the exploration term—*admit closed-form expressions* based on the sum of these statistics. Thus, the Central Server aggregates information simply via summation, enabling each agent to exploit observations from others without sharing raw data. This paradigm has since been extended to other bandit settings, including federated neural bandits (Dai et al. 2022).

However, in contextual linear dueling bandits, *the estimated linear function parameters θ lack a closed-form expression* (Bengs, Saha, and Hüllermeier 2022). Instead, we often need to estimate θ by minimizing a loss function via *gradient descent* (GD). This renders the federated bandit paradigm from Wang et al. (2019) inapplicable to our problem of contextual linear dueling bandits. To resolve this challenge, we

*Corresponding author. Email: daizhongxiang@cuhk.edu.cn
Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

propose to allow the agents to *collaboratively use GD to estimate the linear function parameters*. Specifically, the joint loss function (to be minimized for parameter estimation) for all agents can be expressed in terms of a summation among all agents (more details in Sec. 3). Therefore, the gradient of the joint loss function can be *decoupled into the contributions from individual agents*. As a result, we let every agent calculate the local gradient of its own loss function and send it to the Central Server. The Central Server then aggregates all local gradients (via summation) to attain the gradient of the joint loss function, which is then used for gradient descent. Interestingly, in contrast to previous works on federated bandits (Shi and Shen 2021; Dai et al. 2022), this novel approach bears a closer resemblance to the original federated learning (FL) paradigm which also involves the exchange of local gradients (McMahan et al. 2017).

In this work, we firstly propose **a vanilla algorithm** named *Federated Linear Dueling Bandit with Gradient Descent* (`FLDB-GD`), which adopts federated GD to estimate the linear function parameters θ in every iteration. A significant drawback of `FLDB-GD` is that it requires multiple rounds of gradient exchange between the Central Server and the agents in every iteration, making it impractical due to excessive communication costs. Therefore, we also develop **a practical algorithm** named *FLDB with Online GD* (`FLDB-OGD`), which only requires one round of gradient exchange after every $\tau \geq 1$ iterations.

We perform rigorous theoretical analysis of the regret of our `FLDB-GD` and `FLDB-OGD` algorithms and show that they both enjoy sub-linear upper bounds on the cumulative regret (Sec. 4). Our theoretical results show that the vanilla `FLDB-GD` algorithm, which requires an excessive number of communication rounds, enjoys a tighter regret upper bound. Meanwhile, the practical `FLDB-OGD` algorithm has a worse regret upper bound yet is substantially more communication-efficient and hence more practical. We also demonstrate a *theoretical trade-off between the regret and communication complexity* of our `FLDB-OGD` algorithm. Through extensive empirical experiments using synthetic and real-world functions, we show that both our algorithms consistently outperform single-agent linear dueling bandits (Sec. 5). In addition, we also empirically demonstrate the benefit of having a larger number of agents, the robustness of our algorithms to the heterogeneity of the reward functions of different agents, and the practical trade-off between regret and communication for `FLDB-OGD`. We defer proofs and additional experimental details to the full version of our paper (Huang et al. 2025).

2 Background and Problem Setting

At iteration t of contextual dueling bandits, the environment generates a set of K arms, denoted as $\mathcal{X}_t \subset \mathcal{X} \subset \mathbb{R}^d$, where $\mathcal{X}$ represents the domain of all possible arms. The agent then selects a pair of arms $x_{t,1}, x_{t,2} \in \mathcal{X}_t$ and receives feedback $y_t = \mathbb{1}(x_{t,1} \succ x_{t,2})$, which is equal to 1 if $x_{t,1}$ is preferred over $x_{t,2}$ and 0 otherwise.

Preference Model. Following the common practice in dueling bandits (Saha 2021; Bengs, Saha, and Hüllermeier 2022; Li, Zhao, and Gu 2024), we assume that the preference feedback follows the Bradley-Terry-Luce (BTL) model (Hunter 2004; Luce 2005). Specifically, the utility of the arms are represented by a *latent reward function* $f : \mathbb{R}^d \to \mathbb{R}$, which maps any arm x to its corresponding reward value $f(x)$. Here we assume that f is a linear function $f(x) = \theta^\top \phi(x), \forall x$, in which $\theta \in \mathbb{R}^d$ are unknown parameters and $\phi(\cdot) \in \mathbb{R}^d$ denotes a known feature mapping. This reduces to standard linear dueling bandits when $\phi(\cdot)$ is the identity mapping. Then, the probability that the first selected arm $x_{t,1}$ is preferred over the second selected arm $x_{t,2}$ is given by $\mathbb{P}\{x_{t,1} \succ x_{t,2}\} = \mathbb{P}\{y_t = 1 | x_{t,1}, x_{t,2}\} = \mu(f(x_{t,1}) - f(x_{t,2}))$ Here $\mu(x) = 1/(1 + \mathrm{e}^{-x})$ is the *link function* for which we adopt the logistic function.

We list below the assumptions needed for our theoretical analysis, all of which are standard assumptions commonly adopted by previous works on dueling bandits (Bengs, Saha, and Hüllermeier 2022; Li, Lu, and Zhou 2017a).

Assumption 2.1. • $\kappa_\mu \triangleq \inf_{x,x' \in \mathcal{X}} \dot{\mu}(f(x) - f(x')) > 0$ *for all pairs of arms* [1].

- *The link function* $\mu : \mathbb{R} \to [0, 1]$ *is continuously differentiable and Lipschitz with constant* L_μ. *For logistic function,* $L_\mu \leq 1/4$.
- *The difference between feature maps is bounded* $\left\|\phi(x_{t,1}) - \phi(x_{t,2})\right\|_2 \leq 1$ *for all arms.*

Performance Measure. The goal of an agent in contextual dueling bandits is to minimize its *regret*. After selecting a pair of arms, denoted by $x_{t,1}$ and $x_{t,2}$, in iteration t, the learner incurs an instantaneous regret. In our theoretical analysis, we aim to analyze the following regret: $r_t = 2f(x_t^*) - f(x_{t,1}) - f(x_{t,2})$, in which $x_t^* = \arg\max_{x \in \mathcal{X}_t} f(x)$ denotes the optimal arm in iteration t. After observing preference feedback for T pairs of arms, the cumulative regret (or regret, in short) of a sequential policy is given by $R_T = \sum_{t=1}^T r_t = \sum_{t=1}^T \big(2f(x_t^*) - f(x_{t,1}) - f(x_{t,2})\big)$, in which $x_t^* = \arg\max_{x \in \mathcal{X}_t} f(x)$ denotes the optimal arm in iteration t. A good policy should have sub-linear regret, i.e., $\lim_{T \to \infty} R_T / T = 0$.

Federated Contextual Dueling Bandits. In federated contextual dueling bandits involving N agents, we assume that all agents share the same latent reward function $f(x) = \theta^\top \phi(x)$. This is consistent with the previous works on federated bandits (Shi and Shen 2021; Dai et al. 2022). In iteration t, every agent i receives a separate set of arms $\mathcal{X}_{t,i} \subset \mathcal{X}$ and chooses from them a pair of arms denoted as $x_{t,1,i}$ and $x_{t,2,i}$. In this case, we analyze the total cumulative regret from all N agents: $R_{T,N} = \sum_{t=1}^T \sum_{i=1}^N \Big(2f(x_{t,i}^*) - f(x_{t,1,i}) - f(x_{t,2,i})\Big)$, in which $x_{t,i}^* = \arg\max_{x \in \mathcal{X}_{t,i}} f(x)$.

Following the common practice in contextual bandits, we assume that the set of arms $\mathcal{X}_{t,i}$ are independently and identically sampled from the environment, and we make the following assumption on the distribution of the arm features:

Assumption 2.2. *Let* $\delta \in (0, 1)$, *define* $\beta_t = \sqrt{2\log(1/\delta) + d\log\big(1 + tN\kappa_\mu/(d\lambda)\big)}$. *For a fixed* $\theta \in$

[1] Here $\dot{\mu}(\cdot)$ refers to the gradient of the link function.

$\mathbb{S}^d$, *positive definite matrix* $W \succeq \frac{\lambda}{\kappa_\mu} I_{d\times d}$, *let* $\widetilde{x}_{t,i,1} = \arg\max_{x\in\mathcal{X}_{t,i}} \theta^\top \phi(x)$, $\widetilde{x}_{t,i,2} = \arg\max_{x\in\mathcal{X}_{t,i}} \theta^\top \phi(x) + \frac{\beta_t}{\kappa_\mu}\|\phi(x) - \phi(x_1)\|_{W^{-1}}$. *Denote* $\widetilde{\phi}_{t,i} = \phi(\widetilde{x}_{t,i,1}) - \phi(\widetilde{x}_{t,i,2})$, $\Sigma = \mathbb{E}\left[\widetilde{\phi}_{t,i}\widetilde{\phi}_{t,i}^T\right]$ *and* $\lambda_f = \inf_{\theta,W,\beta} \lambda_{\min}(\Sigma)$. *We assume* $\lambda_f > 0$ *is a positive constant.*

Our Assumption 2.2 can be seen as a generalization of Assumption 3 of Ding, Hsieh, and Sharpnack (2021) from generalized linear bandits to linear dueling bandits. Such diversity assumptions are commonly adopted in the analysis of generalized linear bandits and dueling bandits (Li, Lu, and Zhou 2017b; Wu, Yang, and Shen 2020; Ding, Hsieh, and Sharpnack 2021). Intuitively, Assumption 2.2 implies that given the distribution from which the arm features are sampled and our arm selection strategy, the feature differences between the selected pairs of arms explore every direction of $\mathbb{R}^d$ with sufficient probability in every iteration to prevent blind spots in learning. Specifically, this assumption guarantees that, on average, the feature differences supply adequate information in each iteration to facilitate sufficient exploration.

3 Federated Linear Dueling Bandits

In this section, we firstly introduce the vanilla algorithm: `FLDB-GD` (Sec. 3.1), which results from an extension of linear dueling bandits to the federated setting and is impractical due to excessive communication costs. Next, we introduce our practical algorithm: `FLDB-OGD` (Sec. 3.2), which incurs significantly less communication costs while maintaining competitive performance.

Local Arm Pair Selection. The local arm pair selection process is shared by both algorithms. In iteration t, each agent $i \in [N]$ receives a set of K contexts, denoted as $\mathcal{X}_{t,i}$ (line 3 of Algo. 1), and selects two arms using the synchronized global parameters θ_{sync} and W_{sync} (which contain information from all agents) following the classical linear dueling bandit (`LDB`) algorithm (Bengs, Saha, and Hüllermeier 2022). Specifically, the first arm $x_{t,1,i}$ is selected greedily based on θ_{sync} (line 4 of Algo. 1), which is the current estimated linear function parameters *based on the data from all agents*. Next, the second arm $x_{t,2,i}$ is chosen following an upper confidence bound strategy to balance exploration and exploitation (line 5 of Algo. 1). In this way, we encourage $x_{t,2,i}$ to both achieve high predicted reward and be different from $x_{t,1,i}$ as well as *the previously selected arms from all agents*. Then, the binary preference feedback between $x_{t,1,i}$ and $x_{t,2,i}$, denoted as $y_{t,i}$, is observed (line 6 of Algo. 1).

Next, we discuss how `FLDB-GD` and `FLDB-OGD` facilitate the communication between the agents and the Central Server to obtain the synchronized global parameters θ_{sync} and W_{sync}.

3.1 The Vanilla Algorithm: `FLDB-GD`

To estimate the global parameters θ_{sync} in iteration t, the Central Server aims to find $\theta_{\text{sync}} = \arg\min_{\theta'} \mathcal{L}_t^{\text{fed}}(\theta')$, in which the loss function is given by:

$$\mathcal{L}_t^{\text{fed}}(\theta') = \sum_{i=1}^{N} \mathcal{L}_t^i(\theta') + \frac{1}{2}\lambda\|\theta'\|_2^2, \tag{1}$$

$$\mathcal{L}_t^i(\theta') = -\sum_{s=1}^{t-1}\Bigg(y_s^i \log\mu\left(\theta'^\top\left[\phi(x_{s,1}^i) - \phi(x_{s,2}^i)\right]\right) + (1-y_s^i)\log\mu\left(\theta'^\top\left[\phi(x_{s,2}^i) - \phi(x_{s,1}^i)\right]\right)\Bigg). \tag{2}$$

Equivalently, $\theta_{\text{sync}} = \arg\min_{\theta'} \mathcal{L}_t^{\text{fed}}(\theta')$ is the maximum likelihood estimate (MLE) of the unknown parameters θ *given the data from all* N *agents up to iteration* $t-1$ (Bengs, Saha, and Hüllermeier 2022).

Since the loss function (1) is convex for any t, it is natural to apply gradient descent (GD) to optimize (1). Interestingly, the loss function (1) is naturally decoupled across different agents, which allows the Central Server to *estimate the gradient of* (1) *using the gradients of the individual local loss functions* (2). As a result, the agents only need to send their local gradients to the Central Server and are hence not required to share their local data $\{x_{t,1,i}, x_{t,2,i}, y_{t,i}\}$. In addition, the Central Server updates the estimated W_{sync} by aggregating all individual information matrices: $W_{\text{sync}} \leftarrow W_{\text{sync}} + \sum_{i=1}^{N} W_{\text{new},i}$. For the reader's reference, we present this vanilla algorithm (named `FLDB-GD`) in Algos. 3 and 4 in App. A of Huang et al. (2025).

However, although the objective function (1) is convex, finding the optimal solution of the loss function at time t requires multiple rounds of gradient descent. This **necessitates multiple rounds of gradient exchanges between the Central Server and the agents in every iteration**. This makes the vanilla algorithm `FLDB-GD` communication-inefficient and hence **impractical**. To this end, in Sec. 3.2, we introduce our more practical algorithm `FLDB-OGD`, which uses online GD to estimate θ_{sync} and hence only needs a single round of communication after every $\tau \geq 1$ iterations.

3.2 The Practical Algorithm: `FLDB-OGD`

Algorithm 1: `FLDB-OGD` (Agent i)

1: **Initialization:** $W_{sync} = \frac{\lambda}{\kappa_\mu} I_{d\times d}, \theta_{sync} = 0_{d\times 1}$
2: **for** $t = 2, \ldots, T$ **do**
3: Receive contexts $\mathcal{X}_{t,i}$
4: Choose the first arm $x_{t,1,i} = \arg\max_{x\in\mathcal{X}_{t,i}} \theta_{\text{sync}}^\top \phi(x)$
5: Choose the second arm $x_{t,2,i} = \arg\max_{x\in\mathcal{X}_{t,i}} \theta_{\text{sync}}^\top\left(\phi(x) - \phi(x_{t,1,i})\right) + \frac{\beta_t}{\kappa_\mu}\|\phi(x) - \phi(x_{t,1,i})\|_{W_{\text{sync}}^{-1}}$
6: Observe the preference feedback $y_{t,i} = \mathbb{1}(x_{t,1,i} \succ x_{t,2,i})$
7: Calculate local gradient $\nabla l_t^i(\widehat{\theta})$ (5)
8: Update $\nabla l_{\text{new}}^i \leftarrow \nabla l_{\text{new}}^i + \nabla l_t^i(\widehat{\theta})$
9: Update $W_{\text{new},i} \leftarrow W_{\text{new},i} + \left(\phi(x_{t,1,i}) - \phi(x_{t,2,i})\right)\left(\phi(x_{t,1,i}) - \phi(x_{t,2,i})\right)^\top$
10: **if** $t \bmod \tau = 0$ (communication round starts) **then**
11: Send $\{\nabla l_{\text{new}}^i, W_{\text{new},i}\}$ to Central Server
12: Receive $\{\theta_{\text{sync}}, W_{\text{sync}}, \widehat{\theta}\}$ from the Central Server
13: $W_{\text{new},i} = \mathbf{0}$, $\nabla l_{\text{new}}^i = \mathbf{0}$

Algorithm 2: FLDB−OGD (Central Server)

1: $t_c = t/\tau$
2: Receive $\{\nabla l^i_{\text{new}}, W_{\text{new},i}\}_{i=1,\dots,N}$ from all N agents
3: Aggregate local gradients: $\nabla f^{\text{fed}}_{t_c}(\widehat{\theta}^{(t_c)}) = \sum_{i=1}^{N} \nabla l^i_{\text{new}}$
4: Update parameters: $\eta_{t_c} = \frac{1}{\alpha t_c}$, $\widehat{\theta}^{(t_c+1)} = \pi_S\left(\widehat{\theta}^{(t_c)} - \eta_{t_c} \nabla f^{\text{fed}}_{t_c}(\widehat{\theta}^{(t_c)})\right)$, let $\widehat{\theta} = \widehat{\theta}^{(t_c+1)}$
5: Update parameters: $\widetilde{\theta}^{(t_c+1)} = \frac{1}{t_c+1} \sum_{j=1}^{t_c+1} \widehat{\theta}^{(j)}$, let $\theta_{\text{sync}} = \widetilde{\theta}^{(t_c+1)}$
6: Update $W_{\text{sync}} \leftarrow W_{\text{sync}} + \sum_{i=1}^{N} W_{\text{new},i}$
7: Broadcast $\{\theta_{\text{sync}}, W_{\text{sync}}, \widehat{\theta}\}$ to all agents

FLDB−OGD updates its estimate of θ_{sync} after every $\tau \geq 1$ iterations (i.e., when $t \bmod \tau = 0$). To estimate θ_{sync}, the loss function our practical FLDB−OGD algorithm aims to minimize is the same as that of FLDB−GD (1), but reformulated into the following form:

$$\mathcal{L}^{\text{fed}}_t(\theta') = \sum_{s=1}^{t} f^{\text{fed}}_s(\theta'), \tag{3}$$

where

$$f^{\text{fed}}_s(\theta') = \begin{cases} \sum_{i=1}^{N} l^i_s(\theta'), & \text{if } s \neq 1, \\ \sum_{i=1}^{N} l^i_1(\theta') + \frac{\lambda}{2}\|\theta'\|_2^2, & \text{if } s = 1, \end{cases} \tag{4}$$

$$\begin{aligned} l^i_s(\theta') = -\Big(& y^i_s \log \mu\big(\theta'^{\top}\left[\phi(x^i_{s,1}) - \phi(x^i_{s,2})\right]\big) \\ & + (1 - y^i_s) \log \mu\big(\theta'^{\top}\left[\phi(x^i_{s,2}) - \phi(x_{s,1})\right]\big)\Big) \end{aligned} \tag{5}$$

Intuitively, every individual loss function f^{fed}_s (4) corresponds to the total loss of *the data collected in iteration s* from all N agents. Inspired by the work of Ding, Hsieh, and Sharpnack (2021) which used OGD to estimate the parameters in generalized linear bandits, in a communication round $t_c = t/\tau$, we only use the gradients of f^{fed}_s (4) from the newly collected data since the last communication round $t_c - 1$ to update our estimation of θ_{sync}. That is, instead of GD, we use OGD to estimate θ_{sync}.

At the beginning of our FLDB−OGD ($t = 1$), the Central Server finds the minimizer of $\mathcal{L}^{\text{fed}}_t(\theta')$ (3) at iteration $t = 1$ (which is also the minimizer of $f^{\text{fed}}_1(\theta')$) denoted as $\widehat{\theta}^{(1)}$, and let $\widetilde{\theta}^{(1)} = \widehat{\theta}^{(1)}$. This ensures that the subsequent estimated θ_{sync} always lies in a bounded ball centered at the groundtruth parameter θ, which is needed in our theoretical analysis (Sec. 4). In the subsequent communication rounds $t_c = t/\tau$, the server also maintains a ball centered at $\widehat{\theta}^{(t_c)}$ as the projection set during OGD: $\mathcal{S} = \left\{\theta : \left\|\theta - \widehat{\theta}^{(t_c)}\right\| \leq 2r\right\}$, in which $r \triangleq \sqrt{\left(2\log(1/\delta) + d\log\left(1 + TN\kappa_\mu/(d\lambda)\right)\right)/(\lambda\kappa_\mu)}$.

In every iteration $t > 1$, each agent i keeps track of two parameters: ∇l^i_{new} (lines 7-8 of Algo. 1) and $W_{\text{new},i}$ (line 9 of Algo. 1), which represent the accumulated local gradients at $\widehat{\theta}$ and accumulated information matrix, respectively. After every τ iterations, a communication round starts (line 10 of Algo. 1), and every agent i sends its local parameters ∇l^i_{new} and $W_{\text{new},i}$ to the Central Server (line 11 of Algo. 1). In communication round $t_c = t/\tau$, after the Central Server receives $\{\nabla l^i_{\text{new}}, W_{\text{new},i}\}_{i=1,\dots,N}$ from all N agents, it aggregates all individual gradients ∇l^i_{new} via summation to obtain $\nabla f^{\text{fed}}_{t_c}(\widehat{\theta}^{(t_c)})$ following (4) (line 3 of Algo. 2). Then, the Central Server performs one step of projected gradient descent with step size $\eta = \frac{1}{\alpha t_c}$ to obtain $\widehat{\theta}^{(t_c+1)}$ (line 4 of Algo. 2). Next, to leverage historical information, the server aggregates (i.e., averages) the past estimates $\widehat{\theta}^{(s)}$ and uses the resulting $\widetilde{\theta}^{(t_c+1)}$ as the updated estimate θ_{sync} in round t (lines 5 of Algo. 2). In addition, the Central Server also updates the global information matrix W_{sync} using the summation of the individual information matrices $W_{\text{new},i}$ (lines 6 of Algo. 2). Finally, in line 7 of Algo. 2, the Central Server broadcasts θ_{sync}, W_{sync} and $\widehat{\theta} = \widehat{\theta}^{(t_c+1)}$ to all agents. In the subsequent τ iterations, each agent i uses θ_{sync} and W_{sync} for arm pair selection (lines 4-5 of Algo. 1), and uses $\widehat{\theta} = \widehat{\theta}^{(t_c+1)}$ to calculate the local gradients ∇l^i_{new} (line 8 of Algo. 1).

Communication Efficiency. Of note, our FLDB−OGD algorithm *only needs one communication round after every $\tau \geq 1$ iterations*. In contrast, the vanilla FLDB−GD algorithm (Sec. 3.1) *requires $M \geq 1$ communication rounds in every iteration*. Therefore, FLDB−OGD is substantially more communication-efficient and hence more practical. The parameter τ (i.e., the number of local update iterations) incurs a principled trade-off between the performance and communication efficiency of FLDB−OGD, which we will demonstrate both theoretically (Sec. 4) and empirically (Sec. 5).

4 Theoretical Analysis

4.1 Analysis of the Vanilla FLDB−GD Algorithm

We make the following assumption in our analysis of FLDB−GD.

Assumption 4.1. *We assume that in every iteration of FLDB−GD, after M rounds of gradient exchange (lines 8-11 of Algo. 3), the resulting $\theta_{sync} = \theta^{(M)}$ can exactly minimize* (1)*, i.e., $\nabla \mathcal{L}^{fed}_t(\theta^{(M)}) = 0$.*

This assumption is commonly adopted by previous works on (non-federated) generalized linear bandits and dueling bandits (Li, Lu, and Zhou 2017b; Saha 2021; Bengs, Saha, and Hüllermeier 2022). That is, it is a common practice in the literature to assume that the maximum likelihood estimation of the parameter θ is obtained exactly. However, in the vanilla FLDB−GD algorithm, satisfying this assumption may necessitate a significantly large number of communication rounds between the agents and the Central Server. Of note, the analysis of the practical FLDB−OGD algorithm (Sec. 4.2) does not require this assumption.

Denote $\theta_t \triangleq \arg\min_{\theta'} \mathcal{L}^{\text{fed}}_t(\theta')$, i.e., θ_t is the minimizer of the loss function $\mathcal{L}^{\text{fed}}_t$. Note that according to Assumption

4.1, we have that $\theta_{\text{sync}} = \theta_t$. In our proof of the regret upper bound for FLDB-GD, a crucial step is to derive the following concentration bound.

Lemma 4.2. *Let* $\delta \in (0,1)$, $\beta_t \triangleq \sqrt{2\log(1/\delta) + d\log\big(1 + tN\kappa_\mu/(d\lambda)\big)}$. *Then with probability of at least* $1-\delta$, *for all* $t = 1, \ldots, T$, *we have that* $\|\theta - \theta_t\|_{V_t} \leq \frac{\beta_t}{\kappa_\mu}$.

Lemma 4.2 suggests that in every iteration t of FLDB-GD, after running M rounds of gradient exchange (lines 8-11 of Algo. 3), the resulting $\theta_{\text{sync}} = \theta_t$ is an accurate approximation of the groundtruth parameters θ. The regret of FLDB-GD can then be upper-bounded by the following theorem.

Theorem 4.3 (FLDB-GD)**.** *With probability at least* $1-\delta$, *the overall regrets of all agents in all iterations satisfy:*

$$R_{T,N} \leq 2Nd\log\big(1 + TN\kappa_\mu/(d\lambda)\big) + 3\sqrt{2}\frac{\beta_T}{\kappa_\mu}\sqrt{TN2d\log\big(1 + TN\kappa_\mu/(d\lambda)\big)} \quad (6)$$

Ignoring all log factors, we have that $R_{T,N} = \widetilde{O}\left(Nd + \frac{d}{\kappa_\mu}\sqrt{TN}\right)$.

The regret upper bound for FLDB-GD (Theorem 4.3) is sub-linear in T. Theorem 4.3 suggests that the average regret of N agents in our FLDB-GD algorithm is upper-bounded by $\widetilde{O}\left(d + \frac{d}{\sqrt{N}\kappa_\mu}\sqrt{T}\right)$, which becomes smaller with a larger number of agents N. This demonstrates *the benefit of collaboration*, because it shows that if a larger number of agents N join the federation of our algorithm, they are guaranteed (on average) to achieve a smaller regret compared with running their contextual dueling bandit algorithms in isolation (i.e., $N = 1$). When there is a single agent, the regret upper bound from Theorem 4.3 reduces to $\widetilde{O}(d + d\sqrt{T}/\kappa_\mu) = \widetilde{O}(d\sqrt{T}/\kappa_\mu)$, which is of the same order as the classical contextual linear dueling bandit algorithm (Bengs, Saha, and Hüllermeier 2022).

As discussed in Sec. 3.1, the vanilla FLDB-GD algorithm incurs excessive communication costs and is hence not practical. In the next section, we analyze the regret of our practical FLDB-OGD algorithm.

4.2 Analysis of the Practical FLDB-OGD Algorithm

For simplicity, we firstly analyze FLDB-OGD for $\tau = 1$, and then generalize it to the case of $\tau > 1$.

In the analysis of FLDB-OGD, we do not require Assumption 4.1. That is, the $\theta_{\text{sync}} = \widetilde{\theta}^{(t)}$ returned by Algo. 2 is no longer the minimizer of $\mathcal{L}_t^{\text{fed}}$ in (3). However, here we prove that as long as the number of agents N is sufficiently large, the difference between $\theta_{\text{sync}} = \widetilde{\theta}^{(t)}$ and θ_t is still bounded. To begin with, the following proposition proves that as long as the number of agents N is large enough, the loss function $f_s^{\text{fed}}(\theta')$ (4) in every iteration s is strongly convex.

Proposition 4.4 (α-strongly convex)**.** *Denote* $\mathbb{B}_\eta := \{\theta' : \|\theta' - \theta\| \leq \eta\}$. *For a constant* $\alpha > 0$, *let*

$$N \geq \big((C_1\sqrt{d} + C_2\sqrt{\log 2/\delta})/\lambda_f\big)^2 + 2\alpha/(\kappa_\mu\lambda_f),$$

where C_1 *and* C_2 *are two universal constants. Then* $f_s^{fed}(\theta')$ *is an* α*-strongly convex function in* $\mathbb{B}_{3r}$, *with probability at least* $1-\delta$.

Recall that every $f_s^{\text{fed}}(\theta') = \sum_{i=1}^N l_s^i(\theta')$ corresponds to an individual loss function in our overall loss function $\mathcal{L}_t^{\text{fed}}(\theta')$ (3) during the FLDB-OGD algorithm. Therefore, we can make use of Proposition 4.4 and the convergence result of OGD for strongly convex functions (Hazan et al. 2016) to derive the following lemma.

Lemma 4.5. *Under the same condition on* N *as Proposition 4.4, with probability at least* $1-\delta$, *the following holds for all* $t \geq 1$: $\left\|\widetilde{\theta}^{(t)} - \theta_t\right\|_{V_t} \leq \frac{N\sqrt{N + \frac{\lambda}{\kappa_\mu}}}{\alpha}\sqrt{1 + \log t}$.

Lemma 4.5 shows that the difference between $\theta_{\text{sync}} = \widetilde{\theta}^{(t)}$ (returned by Algo. 2) and $\theta_t \triangleq \arg\min_{\theta'} \mathcal{L}_t^{\text{fed}}(\theta')$ is bounded. As a result, Lemma 4.5, combined with Lemma 4.2 which has provided an upper bound on the difference between θ_t and θ, allows us to bound the difference between $\widetilde{\theta}^{(t)}$ and θ. That is, the $\theta_{\text{sync}} = \widetilde{\theta}^{(t)}$ returned by Algo. 2 is an accurate estimation of the groundtruth linear function parameter θ.

With these supporting lemmas, we can derive an upper bound on the cumulative regret of FLDB-OGD when $\tau = 1$ (i.e., if a communication round occurs after every iteration).

Theorem 4.6 (FLDB-OGD with $\tau = 1$)**.** *Ignoring all log factors, with probability of at least* $1-\delta$, *the overall regret of* FLDB-OGD *(when* $\tau = 1$*) can be bounded by*

$$R_{T,N} = \widetilde{O}\big(Nd + \frac{d}{\kappa_\mu}\sqrt{TN} + \frac{N^2\sqrt{d}}{\alpha}\sqrt{T}\big) \quad (7)$$

The regret upper bound of our FLDB-OGD algorithm is also sub-linear in T. Compared with FLDB-GD (Theorem 4.3), the regret upper bound for FLDB-OGD has an additional term of $\widetilde{O}(\frac{N^2\sqrt{d}}{\alpha}\sqrt{T})$. This is the loss we suffer for not ensuring that the $\theta_{\text{sync}} = \widetilde{\theta}^{(t)}$ returned by Algo. 2 could achieve the minimum of $\mathcal{L}_t^{\text{fed}}$ (3). On the other hand, by paying this cost in terms of regrets, FLDB-OGD gains considerably smaller communication costs than FLDB-GD. This is because FLDB-OGD only needs a single round of communication after every τ iterations, whereas FLDB-GD requires a large number of communication rounds in every iteration to find the minimum of $\mathcal{L}_t^{\text{fed}}(\theta')$ (1).

Unlike FLDB-GD (Theorem 4.3), the average regret upper bound of N agents in Theorem 4.6 is not improved as the number N of agents is increased. This is because in the proof of Lemma 4.5 (App. E in Huang et al. (2025)), we have made use of the convergence of OGD for strongly convex functions (Hazan et al. 2016), which requires an upper bound on the expected norm of the gradient: $G^2 \geq E\|\nabla f_s\|^2$. We have shown that the worst-case choice of G is $G = N$ (see (32) in App. E of Huang et al. (2025)), which contributed a dependency of N to the last term in the regret upper bound in

Theorem 4.6. Therefore, if we additionally assume that there exists an upper bound G on expected gradient norm which is independent of N, then the last term in Theorem 4.6 can be replaced by $\frac{GN\sqrt{d}}{\alpha}\sqrt{T}$. This would then make our average regret upper bound in Theorem 4.6 (averaged over N agents) become tighter with a larger number of agents N.

Now we analyze the regret of our practical FLDB−OGD algorithm when $\tau > 1$, i.e., when a communication round occurs after multiple iterations.

Proposition 4.7 (FLDB−OGD with $\tau > 1$)**.** *Ignoring all log factors, with probability of at least $1-\delta$, the overall regret of FLDB−OGD (when $\tau > 1$) can be bounded by*

$$R_{T,N} = \widetilde{O}\big(\tau N d + \frac{d}{\kappa_\mu}\sqrt{TN} + \tau^{\frac{3}{2}}\frac{N^2\sqrt{d}}{\alpha}\sqrt{T}\big) \quad (8)$$

Trade-off between Regrets and Communication. As long as the number of agents N is large enough (i.e., ensured by Prop. 4.4), the second term in the regret upper bound from Prop. 4.7 is dominated by the other two terms. This implies that a larger number of local updates τ (which indicates less frequent communication between the agents and the Central Server) results in a worse regret upper bound for FLDB−OGD. On the other hand, a larger τ reduces the total number of communication rounds by a factor of $1/\tau$. This induces a *theoretical trade-off between regrets and communication complexity*, which is also validated by our empirical experiments (Sec. 5).

5 Experiments

Here we evaluate the performance of our algorithms using both synthetic and real-world experiments. We let $\tau = 1$ unless specified otherwise.

Experimental Settings. In the synthetic experiments, we generate the groundtruth linear function parameters θ in each experiment by randomly sampling from the standard Gaussian distribution. In each round, every agent receives K arms (i.e., contexts) randomly generated from the standard Gaussian distribution $\mathcal{N}(0, I)$ with dimension d. For FLDB−GD, we adopt the LBFGS method from PyTorch (Paszke et al. 2019) to efficiently find the minimizer of the loss function (1) in every iteration. All figures in this section plot the *average cumulative regret* of all N agents up to a given iteration, which allows us to inspect the benefit brought by the federated setting to an agent on average. To verify the benefit of collaboration in the federated setting, we compare our algorithms with the single-agent baseline of linear dueling bandits (LDB) (Bengs, Saha, and Hüllermeier 2022), in which every agent simply performs a local linear dueling bandit algorithm in isolation. For real-world evaluations, we adopt the MovieLens dataset (Harper and Konstan 2015), following the experimental setup from Wang et al. (2023). More details on the experimental settings are deferred to App. B in Huang et al. (2025).

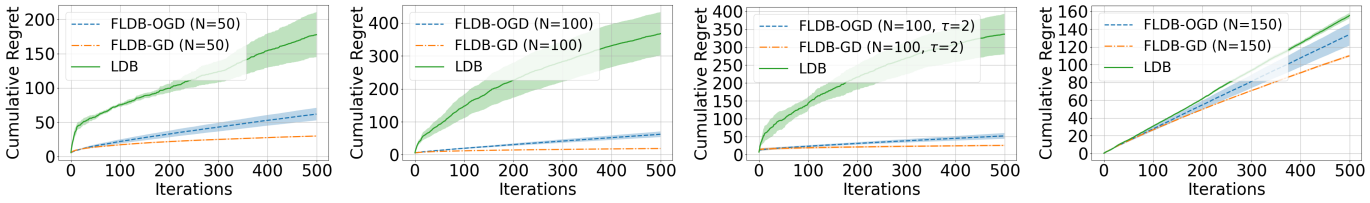
Results. In the synthetic experiments, we evaluate our FLDB−OGD algorithm, as well as the vanilla FLDB−GD algorithm and the LDB baseline, under the same settings of the number of agents N, the number of arms K and the input dimension d. To begin with, we fix $K = 10$ and $d = 5$, and evaluate the impact of different number of agents N and different update period τ. In the MovieLens experiments, we set $K = 5$ and $d = 10$. The results, presented in Figs. 1a–1d, show that both FLDB−GD and FLDB−OGD consistently outperform the single-agent baseline LDB across all settings. Moreover, FLDB−GD achieves lower cumulative regret than FLDB−OGD, which is consistent with our theoretical analysis in Section 4.2, as FLDB−GD enjoys a tighter regret bound compared to FLDB−OGD. However, as we have pointed out in Sec. 3.2 and Sec. 4.2, this performance advantage of FLDB−GD in terms of regret comes at the expense of *significantly increased communication costs*.

In addition, we conduct an ablation study to further examine the behavior of our FLDB−OGD algorithm under different experimental conditions. Specifically, we analyze the effects of (i) the number of agents (N), (ii) reward heterogeneity, and (iii) the number of communication rounds.

Impact of the Number of Agents N. Here we examine how the number of agents N affects the performance of the practical FLDB−OGD algorithm, while fixing the values of K and d. As shown in Fig. 2a and b, the cumulative regret of FLDB−OGD decreases as the number of agents N becomes larger, indicating that the collaboration among more agents indeed leads to improved performance. These results validate the effectiveness of our FLDB−OGD algorithm in leveraging collaborative information from multiple agents, and offer practical motivation for encouraging more agents to participate in the federated learning process. The observed empirical improvement of FLDB−OGD with larger N does not follow directly from the theoretical guarantee provided in Theorem 4.6. This discrepancy suggests that the theoretical bound in Theorem 4.6 may be overly conservative. Meanwhile, it provides justifications for our additional assumption at the end of Section 4.2, which posits that the gradient norm bound G is independent of N. Under this assumption, the regret bound for FLDB−OGD indeed improves with larger N, leading to better agreement between theory and practice.

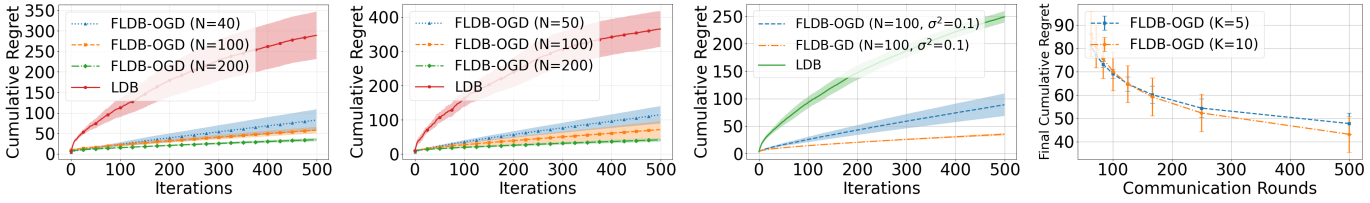
Heterogeneous Agents. To evaluate the performance of our algorithms when the reward functions of different agents are heterogeneous, we introduce variability into the reward functions of different agents by perturbing the global parameters θ^*. Specifically, for each agent i, we define a personalized parameter $\theta_i^* = \theta^* + \epsilon$, where the perturbation ϵ is sampled from a multivariate Gaussian distribution $\mathcal{N}(0, \sigma^2 I)$. We vary the standard deviation σ from 0.1 to 0.5 to control the degree of heterogeneity. Fig. 2c shows the results for $\sigma^2 = 0.1$ (for$\sigma^2 = 0.25$ and a comparison, see App. B in Huang et al. 2025). For each fixed value of σ, we observe that the trends are consistent with those in the homogeneous synthetic setting (Fig. 1). Both FLDB−OGD and FLDB−GD incur higher cumulative regrets as the level of degree of heterogeneity increases. Of note, FLDB−OGD maintains favorable performance in the presence of moderate heterogeneity, demonstrating its robustness under heterogeneous reward functions of different agents.

Regret-Communication Trade-off. To investigate the effect of communication frequency on the performance of FLDB−OGD, we vary the number of local updates τ in each iteration and report the resulting final regret after $T = 500$



(a) $N = 50$ $(K = 10, d = 5)$ (b) $N = 100$ $(K = 10, d = 5)$ (c) $N = 100, \tau = 2$ $(K = 10, d = 5)$ (d) MovieLens $(K = 5, d = 10)$

Figure 1: Cumulative regret for different methods with varying numbers of agents: (a) $N = 50$, (b) $N = 100$, (c) $N = 100, \tau = 2$ under the number of arms $K = 10$ and dimension $d = 5$, and (d) MovieLens Dataset with $K = 5$ and $d = 10$.



(a) FLDB-OGD $(K = 10, d = 5)$ (b) FLDB-OGD $(K = 50, d = 5)$ (c) Heterogeneous $(K = 5, d = 5)$ (d) Communication vs. Regrets $(N = 100, d = 5)$

Figure 2: Cumulative regret of FLDB-OGD under different settings. (a)–(b): Impact of the number of agents with $K = 10, 50$ and $d = 5$. (c): Performance under heterogeneous rewards with $K = 5$, $d = 5$. (d): Final regret versus number of communication rounds with $N = 100$, $d = 5$.

iterations. Given the values of T and τ, the total number of communication rounds is given by T/τ. Therefore, for a fixed T, a smaller τ corresponds to more frequent communication, while a larger τ reduces the number of communication rounds. Fig. 2d illustrates the relationship between communication frequency and the final regret. These results corroborate our theoretical analysis in Prop. 4.7: a larger number of communication rounds (i.e., smaller τ) leads to smaller regrets.

6 Related Work

Federated Bandits. Recent studies have extended the classical K-armed bandit problem to the federated setting. Li and Song (2022); Li et al. (2020) introduced privacy-preserving federated K-armed bandits in centralized and decentralized settings, respectively. Shi and Shen (2021) formulated a global bandit model where arm rewards are averaged across agents, which was later extended to incorporate personalization (Shi, Shen, and Yang 2021). For federated linear contextual bandits, Wang et al. (2019) proposed a distributed algorithm using sufficient statistics to compute the Linear UCB policy, which was later extended to incorporate differential privacy (Dubey and Pentland 2020), agent-specific contexts (Huang et al. 2021), and asynchronous communication (Li and Wang 2022a). Federated kernelized and neural bandits have been developed for hyperparameter tuning (Dai, Low, and Jaillet 2020, 2021; Dai et al. 2022). In addition, many recent works have extended federated bandits to various settings and applied them to solve different real-world problems (Li et al. 2022; Li and Wang 2022b; Zhu et al. 2021; Ciucanu et al. 2022; Blaser, Li, and Wang 2024; Fan et al. 2024; Yang et al. 2024; Solanki, Jain, and Gujar 2024; Fourati, Alouini, and Aggarwal 2024; Wang et al. 2024; Li et al. 2024; Wei et al. 2024; Li, Liu, and Lui 2024).

Dueling Bandits. Dueling bandits have received considerable attention in recent years (Yue and Joachims 2009, 2011; Yue et al. 2012; Zoghi et al. 2014b; Ailon, Karnin, and Joachims 2014; Zoghi et al. 2014a; Komiyama et al. 2015; Gajane, Urvoy, and Clérot 2015; Saha and Gopalan 2018, 2019a,b; Saha and Ghoshal 2022; Zhu, Jordan, and Jiao 2023). To account for complicated real-world scenarios, a number of contextual dueling bandit algorithms have been developed which model the reward function using either a linear function (Saha 2021; Bengs, Saha, and Hüllermeier 2022; Li, Zhao, and Gu 2024; Saha and Krishnamurthy 2022; Di et al. 2023) or a neural network (Verma et al. 2024).

7 Conclusion and Future Direction

We introduce FLDB-OGD, the first federated linear dueling bandit algorithm for privacy-preserving multi-agent collaboration. Integrating online gradient descent with federated learning, our approach efficiently estimates the linear function parameters. We establish a sub-linear regret bound and empirically demonstrate the benefits of collaboration and its trade-off with communication. A potential limitation is that our methods are not applicable to dueling bandit problems with non-linear reward functions (when a known non-linear feature mapping is unavailable). We plan to address this challenge in future work.

Acknowledgments

This work was in part supported by National Natural Science Foundation of China (Grant No. 62506319), Shenzhen Medical Research Fund (Grant No. C10120250085), and Shenzhen Science and Technology Program (Grant No. JCYJ20250604141031003).

References

Abbasi-Yadkori, Y.; Pál, D.; and Szepesvári, C. 2011. Improved Algorithms for Linear Stochastic Bandits. In *Proc. NeurIPS*, 2312–2320.

Ailon, N.; Karnin, Z.; and Joachims, T. 2014. Reducing dueling bandits to cardinal bandits. In *Proc. ICML*, 856–864.

Bengs, V.; Saha, A.; and Hüllermeier, E. 2022. Stochastic Contextual Dueling Bandits under Linear Stochastic Transitivity Models. In *Proc. ICML*, 1764–1786.

Blaser, E.; Li, C.; and Wang, H. 2024. Federated Linear Contextual Bandits with Heterogeneous Clients. In *International Conference on Artificial Intelligence and Statistics*, 631–639. PMLR.

Ciucanu, R.; Lafourcade, P.; Marcadet, G.; and Soare, M. 2022. SAMBA: A Generic Framework for Secure Federated Multi-Armed Bandits. *JAIR*, 73: 737–765.

Dai, Z.; Low, B. K. H.; and Jaillet, P. 2020. Federated Bayesian optimization via Thompson sampling. *Advances in Neural Information Processing Systems*, 33: 9687–9699.

Dai, Z.; Low, B. K. H.; and Jaillet, P. 2021. Differentially private federated Bayesian optimization with distributed exploration. In *Proc. NeurIPS*.

Dai, Z.; Shu, Y.; Verma, A.; Fan, F. X.; Low, B. K. H.; and Jaillet, P. 2022. Federated neural bandits. *arXiv preprint arXiv:2205.14309*.

Di, Q.; Jin, T.; Wu, Y.; Zhao, H.; Farnoud, F.; and Gu, Q. 2023. Variance-Aware Regret Bounds for Stochastic Contextual Dueling Bandits. *arXiv:2310.00968*.

Ding, Q.; Hsieh, C.-J.; and Sharpnack, J. 2021. An Efficient Algorithm For Generalized Linear Bandit: Online Stochastic Gradient Descent and Thompson Sampling. arXiv:2006.04012.

Dubey, A.; and Pentland, A. 2020. Differentially-private federated linear bandits. In *Proc. NeurIPS*, 6003–6014.

Fan, L.; Zhou, R.; Tian, C.; and Shen, C. 2024. Federated linear bandits with finite adversarial actions. *Advances in Neural Information Processing Systems*, 36.

Fourati, F.; Alouini, M.-S.; and Aggarwal, V. 2024. Federated Combinatorial Multi-Agent Multi-Armed Bandits. *arXiv preprint arXiv:2405.05950*.

Gajane, P.; Urvoy, T.; and Clérot, F. 2015. A relative exponential weighing algorithm for adversarial utility-based dueling bandits. In *Proc. ICML*, 218–227.

Harper, F. M.; and Konstan, J. A. 2015. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.*, 5(4).

Hazan, E.; et al. 2016. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4): 157–325.

Huang, R.; Wu, W.; Yang, J.; and Shen, C. 2021. Federated Linear Contextual Bandits. In *Proc. NeurIPS*.

Huang, X.; Hu, Y.; Li, Z.; Wang, Z.; Wang, B.; and Dai, Z. 2025. Federated Linear Dueling Bandits. arXiv:2502.01085.

Hunter, D. R. 2004. MM Algorithms for Generalized Bradley-Terry Models. *Annals of Statistics*, 384–406.

Ji, K.; He, J.; and Gu, Q. 2024. Reinforcement learning from human feedback with active queries. *arXiv preprint arXiv:2402.09401*.

Komiyama, J.; Honda, J.; Kashima, H.; and Nakagawa, H. 2015. Regret lower bound and optimal algorithm in dueling bandit problem. In *Proc. COLT*, 1141–1154.

Li, C.; and Wang, H. 2022a. Asynchronous Upper Confidence Bound Algorithms for Federated Linear Bandits. In *Proc. AISTATS*.

Li, C.; and Wang, H. 2022b. Communication efficient federated learning for generalized linear bandits. *Advances in Neural Information Processing Systems*, 35: 38411–38423.

Li, C.; Wang, H.; Wang, M.; and Wang, H. 2022. Communication efficient distributed learning for kernelized contextual bandits. *Advances in Neural Information Processing Systems*, 35: 19773–19785.

Li, L.; Lu, Y.; and Zhou, D. 2017a. Provably Optimal Algorithms for Generalized Linear Contextual Bandits. In *Proc. ICML*, 2071–2080.

Li, L.; Lu, Y.; and Zhou, D. 2017b. Provably Optimal Algorithms for Generalized Linear Contextual Bandits. arXiv:1703.00048.

Li, T.; Sahu, A. K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; and Smith, V. 2020. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2: 429–450.

Li, T.; and Song, L. 2022. Privacy-Preserving Communication-Efficient Federated Multi-Armed Bandits. *IEEE Journal on Selected Areas in Communications*.

Li, W.; Song, Q.; Honorio, J.; and Lin, G. 2024. Federated x-armed bandit. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 13628–13636.

Li, X.; Zhao, H.; and Gu, Q. 2024. Feel-Good Thompson Sampling for Contextual Dueling Bandits. *arXiv:2404.06013*.

Li, Z.; Liu, M.; and Lui, J. 2024. FedConPE: Efficient Federated Conversational Bandits with Heterogeneous Clients. *arXiv preprint arXiv:2405.02881*.

Lin, X.; Dai, Z.; Verma, A.; Ng, S.-K.; Jaillet, P.; and Low, B. K. H. 2024. Prompt Optimization with Human Feedback. *arXiv preprint arXiv:2405.17346*.

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

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, 1273–1282. PMLR.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Köpf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. arXiv:1912.01703.

Saha, A. 2021. Optimal Algorithms for Stochastic Contextual Preference Bandits. In *Proc. NeurIPS*, 30050–30062.

Saha, A.; and Ghoshal, S. 2022. Exploiting correlation to achieve faster learning rates in low-rank preference bandits. In *Proc. AISTATS*, 456–482.

Saha, A.; and Gopalan, A. 2018. Battle of Bandits. In *Proc. UAI*, 805–814.

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

Saha, A.; and Gopalan, A. 2019b. PAC battling bandits in the plackett-luce model. In *Proc. ALT*, 700–737.

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

Shi, C.; and Shen, C. 2021. Federated multi-armed bandits. In *Proc. AAAI*.

Shi, C.; Shen, C.; and Yang, J. 2021. Federated multi-armed bandits with personalization. In *Proc. AISTATS*, 2917–2925.

Solanki, S.; Jain, S.; and Gujar, S. 2024. Fairness and Privacy Guarantees in Federated Contextual Bandits. *arXiv preprint arXiv:2402.03531*.

Verma, A.; Dai, Z.; Lin, X.; Jaillet, P.; and Low, B. K. H. 2024. Neural dueling bandits. *arXiv preprint arXiv:2407.17112*.

Wang, Y.; Hu, J.; Chen, X.; and Wang, L. 2019. Distributed bandit learning: Near-optimal regret with efficient communication. *arXiv preprint arXiv:1904.06309*.

Wang, Z.; Xie, J.; Liu, X.; Li, S.; and Lui, J. C. S. 2023. Online Clustering of Bandits with Misspecified User Models. arXiv:2310.02717.

Wang, Z.; Zhu, Y.; Wang, D.; and Han, Z. 2024. Towards Fair and Scalable Trial Assignment in Federated Bandits: A Shapley Value Approach. *IEEE Transactions on Big Data*.

Wei, Z.; Li, C.; Ren, T.; Xu, H.; and Wang, H. 2024. Incentivized Truthful Communication for Federated Bandits. *arXiv preprint arXiv:2402.04485*.

Wu, W.; Yang, J.; and Shen, C. 2020. Stochastic Linear Contextual Bandits with Diverse Contexts. arXiv:2003.02681.

Yang, H.; Liu, X.; Wang, Z.; Xie, H.; Lui, J. C.; Lian, D.; and Chen, E. 2024. Federated Contextual Cascading Bandits with Asynchronous Communication and Heterogeneous Users. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 20596–20603.

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

Yue, Y.; and Joachims, T. 2009. Interactively optimizing information retrieval systems as a dueling bandits problem. In *Proc. ICML*, 1201–1208.

Yue, Y.; and Joachims, T. 2011. Beat the mean bandit. In *Proc. ICML*, 241–248.

Zhu, B.; Jordan, M.; and Jiao, J. 2023. Principled reinforcement learning with human feedback from pairwise or k-wise comparisons. In *Proc. ICML*, 43037–43067.

Zhu, Z.; Zhu, J.; Liu, J.; and Liu, Y. 2021. Federated bandit: A gossiping approach. *Proc. ACM Meas. Anal. Comput. Syst.*, 5(1): 1–29.

Zoghi, M.; Whiteson, S.; Munos, R.; and Rijke, M. 2014a. Relative upper confidence bound for the k-armed dueling bandit problem. In *Proc. ICML*, 10–18.

Zoghi, M.; Whiteson, S. A.; De Rijke, M.; and Munos, R. 2014b. Relative confidence sampling for efficient on-line ranker evaluation. In *Proc. WSDM*, 73–82.