Privacy Matters: Vertical Federated Linear Contextual Bandits for Privacy-Protected Recommendation

Zeyu Cao*
University of Cambridge
Cambridge, United Kingdom
zc317@cam.ac.uk

Zhipeng Liang*
Hong Kong University of Science and Technology
Hong Kong, China
zliangao@connect.ust.hk

Bingzhe Wu†
Tencent AI Lab
Shenzhen, China
wubingzheagent@gmail.com

Shu Zhang
Tencent AI Lab
Shenzhen, China
bookzhang@tencent.com

Hangyu Li
Tencent AI Lab
Shenzhen, China
masonhyli@tencent.com

Ouyang Wen
Tencent AI Lab
Shenzhen, China
gdpouyang@tencent.com

Yu Rong
Tencent AI Lab
Shenzhen, China
yu.rong@hotmail.com

Peilin Zhao
Tencent AI Lab
Shenzhen, China
masonzhao@tencent.com

ABSTRACT

Recent awareness of privacy protection and compliance requirement resulted in a controversial view of recommendation system due to personal data usage. Therefore, privacy-protected recommendation emerges as a novel research direction. In this paper, we first formulate this problem as a vertical federated learning problem, i.e., features are vertically distributed over different departments. We study a contextual bandit learning problem for recommendation in the vertical federated setting. To this end, we carefully design a customized encryption scheme named orthogonal matrix-based mask mechanism (O3M). O3M mechanism, a tailored component for contextual bandits by carefully exploiting their shared structure, can ensure privacy protection while avoiding expensive conventional cryptographic techniques. We further apply the mechanism to two commonly-used bandit algorithms, LinUCB and LinTS, and instantiate two practical protocols for online recommendation. The proposed protocols can perfectly recover the service quality of centralized bandit algorithms while achieving a satisfactory runtime efficiency, which is theoretically proved and analysed in this paper. By conducting extensive experiments on both synthetic and real-world datasets, we show the superiority of the proposed method in terms of privacy protection and recommendation performance.

*Both authors contributed equally to this research. This work is done when Zeyu Cao and Zhipeng Liang worked as interns in Tencent AI Lab.
†Corresponding Author

KDD '23, August 6–10, 2023, Long Beach, CA, USA

ACM ISBN 979-8-4007-0103-0/23/08…$15.00
https://doi.org/10.1145/3580305.3599475

CCS CONCEPTS

• **Security and privacy** → **Privacy-preserving protocols**; • **Computing methodologies** → *Online learning settings.*

KEYWORDS

Vertical Federated Learning, Linear Contextual Bandits, Privacy-Preserving Protocols

ACM Reference Format:
Zeyu Cao, Zhipeng Liang, Bingzhe Wu, Shu Zhang, Hangyu Li, Ouyang Wen, Yu Rong, and Peilin Zhao. 2023. Privacy Matters: Vertical Federated Linear Contextual Bandits for Privacy-Protected Recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3580305.3599475

1 INTRODUCTION

Personalized recommendation system is one of the fundamental building blocks of numerous web service platforms such as online retail and digital marketing [15, 27]. From a technical perspective, most of these recommendation tasks can be modelled as an online sequential decision process wherein the system needs to recommend items to the current user based on sequential dynamics (i.e., historical user-item interactions and user/item contexts). Theoretically, seminal contextual bandit algorithms, including LinUCB [1] and LinTS [3] are proved to achieve the optimal balance between exploration and exploitation, whereas empirically, these approaches have shown remarkable performance and have been deployed into various industrial scenarios for recommendation [19, 21, 35].

For such an online process, the cold-start problem is also ubiquitous since new users and items continuously join the online service and these new users typically come with incomplete contextual information and rarely interact with items [25, 31]. The key task of cold-start problems is to collect user-item interactions when one or both are new efficiently. This task is challenging since there are two competing goals in such a collection process, i.e., exploration and exploitation. The contextual bandit-based approach is also seen as a principled way to solve this problem, in which a learning algorithm sequentially recommends an item to users based on contextual information of the user and item, while simultaneously adapting its recommendation strategy based on historical user-click feedback to maximize total user clicks in the long run.

However, in practice, any service provider is hopeless to have direct access to comprehensive user contextual information. For example, as shown in Figure 1, an online shopping platform can only keep track of a user's purchase history, while her social interaction and financial background which can also imply her preference are kept by other companies or departments. These companies may attempt to collaborate on a promotion campaign for a shared user base. Another example could be different departments in a multinational company attempting online recommendations with the same user. Clearly, the recommendation system, or specifically, the bandit algorithm, could benefit from additional data to make a better recommendation for one of the departments or company. However, the emerging awareness of privacy protection and data sovereignty regulations impose a sharp burden prohibiting directly sharing information across different companies or even departments within the same company.

These challenges faced by traditional recommendation systems required research in new privacy-protected recommendation systems [4]. This recommendation system aims to conduct recommendations while preventing private user information from leaking. Vertical federated learning [5] is proposed to solve the joint training under such stringent privacy requirements with mild performance and efficiency loss. A question is naturally raised: *How to design the contextual bandits for the privacy-protected recommendation system under the vertical federated setting?*

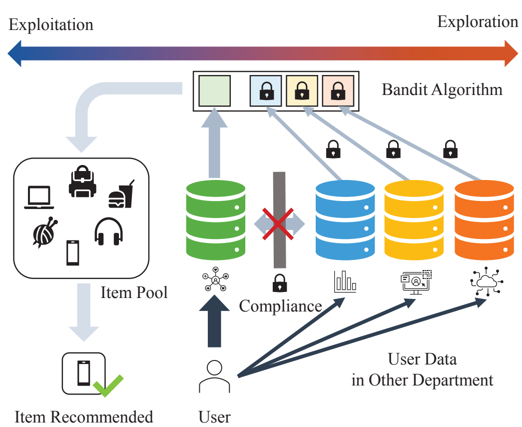


Figure 1: Overview on using bandit algorithm for recommendation problem in the vertical federated setting. Department A wants to make the recommendation for the cold-start user i. Since the user is new to the system, instead of only exploiting user data in department A, one good strategy to improve the recommendation results is exploring the user data from other departments. These departments' data are illustrated in different colours. However, due to compliance requirements, the recommendation system cannot directly use the data and must conduct privacy-protected recommendations. This is illustrated by the lock on the data when the contextual bandit is attempted to use the data to conduct recommendations.

To solve this problem, a natural idea is to apply modern privacy enhancing techniques such as secure multi-party computation (MPC) [11] and differential privacy (DP) [9, 10] to current contextual bandit approaches. However, this manner typically leads to either high computation/communication costs or the degradation of recommendation performance: For example, in the multi-party computation, the Multiplication operator would lead to 380 times slower and the comparison operator would even lead to 34000 times slower [13]. Thus directly applying these cryptographic techniques to the contextual bandit is intractable since there involve large amounts of multiplication/comparison operations.[1] For differential privacy, injecting sufficient noise into the contextual information leads to a significant loss of the statistics utility and worse recommendation performance [12, 40].

All these cryptographic techniques are originally designed for general-purpose computations. In this paper, we note that the core computation operators in the contextual bandits come with unique properties, thus providing the opportunity to design customized encryption schemes with better runtime efficiency and statistical utility. In general, the exploration mechanism is at the core of previous bandit algorithms, which typically use the empirical sample covariance matrix to construct the quantity needed by the exploration mechanism. Based on the property of the sample covariance matrix, we propose a technique named **o**rthogonal **m**atrix based **m**ask **m**echanism (O3M) for computing this quantity in a private and lossless manner, which is motivated by prior works on federated PCA [5].

O3M is used by the data provider for masking the local contextual information (i.e., user features) before sending them to the service provider. Specifically, O3M encodes the raw contextual information by multiplying with an orthogonal matrix (i.e., mask) then the data provider will send the masked data to the service provider. Since the service provider cannot access the mask, it cannot infer the original information even though it knows the masked data. However, due to the mathematical property of the orthogonal matrix, its effect can be removed when constructing the key statistic in the bandit algorithms. Thus we can obtain "lossless" performance in a privacy-preserving manner (see proof details in Section 4). We further apply the O3M technique to two commonly-used bandit algorithms, LinUCB and LinTS, and implement two practical protocols, VFUCB and VFTS, for real-world recommendation tasks. For simplicity, in the following discussions, we refer to these two different protocols as VFCB (Vertical Federated Contextual Bandit) without contextual ambiguity.

Besides, to help the reader better understand our protocols, we conduct comprehensive complexity analysis and provide formal utility and security proofs in Section 4. By performing extensive empirical studies on synthetic and real-world datasets, we show that the proposed protocols can achieve similar runtime efficiency and the same recommendation performance as the centralized bandit with a theoretical privacy-protection guarantee. We also point out that incorporating more user features improves the performance of

[1] A further comparison between these approaches and ours can be found at Appendix C.3.

the contextual bandit in the vertical setting, providing insight to improve the recommendation quality in an online and federated manner.

2 RELATED WORK

2.1 Federated Contextual Bandits

Recently a few works have explored the concept of federated bandits [2, 23, 33, 33, 41]. Among the federated bandits with linear reward structures, Huang et al. consider the case where individual clients face different K-armed stochastic bandits coupled through common global parameters [17]. They propose a collaborative algorithm to cope with the heterogeneity across clients without exchanging local feature vectors or raw data. Dubey and Pentland design differentially private mechanisms for tackling centralized and decentralized (peer-to-peer) federated learning [8]. Tewari and Murphy consider a fully-decentralized federated bandit learning setting with gossiping, where an individual agent only has access to biased rewards, and agents can only exchange their observed rewards with their neighbor agents [36]. Li and Wang study the federated linear contextual bandits problem with heterogeneous clients respectively, i.e., the clients have various response times and even occasional unavailability in reality. They propose an asynchronous event-triggered communication framework to improve our method's robustness against possible delays and the temporary unavailability of clients [20]. Shi et al. consider a similar setting to Tewari and Murphy's setting but the final payoff is a mixture of the local and global model [34]. All the above-mentioned paper belong to the horizontal federated bandits, i.e., each client has heterogeneous users and all the clients jointly train a central bandit model without exchanging its own raw users' features. Since each user has her features all stored by one client, this horizontal federated setting has a sharp contrast with the VFL setting. As far as we know, our paper is the first to consider the contextual bandit problem in the VFL setting.

2.2 Vertical Federated Learning

Vertical Federated Learning (VFL), aiming at promoting collaborations among non-competing organizations/entities with vertically partitioned data, has seen its huge potential in applications [38] but still remains relatively less explored. Most of the existing work focuses on supervised learning and unsupervised learning settings. For unsupervised learning problems, Chai et al. propose the first masking-based VFL singular vector decomposition (SVD) method. Their method recovers to the standard SVD algorithm without sacrificing any computation time or communication overhead [5]. Cheung et al. propose the federated principal component analysis and advanced kernel principal component analysis for vertically partitioned datasets[7]. For supervised learning problems, Chen et al. targets solving vertical FL in an asynchronous fashion, and their solution allows each client to run stochastic gradient algorithms without coordination with other clients[6]. Hardy et al. propose a VFL variant of logistic regression using homomorphic encryption [14]. Wu et al. propose a VF decision tree without dependence on any trusted third party [39]. Hu et al. designed an asynchronous stochastic gradient descent algorithm to collaboratively train a central model in a VFL manner [16]. Romanini et al. introduce a

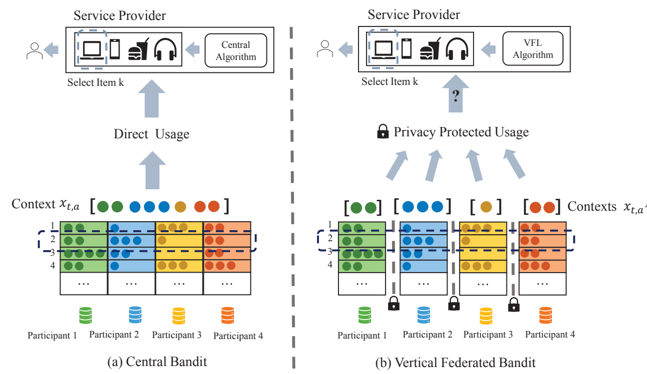


Figure 2: Comparison between centralized(a) and vertical federated(b) setting.

framework to train neural network in the VFL setting using split neural networks [30].

3 PRELIMINARY

3.1 Problem Description

We first introduce the definition of Linear Contextual Bandits.

Definition 3.1 (Linear Contextual Bandits). Consider a sequential decision process, where at each time step t, the environment would release a set of contexts $\{x_{t,a} \in \mathbf{R}^d\}_{a \in \mathcal{A}}$ for some fix $\mathcal{A}$ as the action set. If context x_{t,a_t} is selected, then the environment would release a reward $r_t = x_{t,a_t}^\top \theta^* + \epsilon_t$ for some reward generating parameter $\theta^* \in \mathbf{R}^d$ and i.i.d. noise ϵ_t. A bandit algorithm A is defined as a sequence of maps $\{\pi_t\}_{t=1}^T$. Each map π_t is from the historical information, $\{(x_{s,a_s}, r_s)\}_{s=1}^{t-1}$ and the current context $\{x_{t,a}\}_{a \in \mathcal{A}}$, to the distribution over the action set, $\Delta(\mathcal{A})$. The goal of the bandits algorithm is to minimize long-time regret, where regret is the gap between the best possible achievable rewards and the rewards obtained by the algorithm. Regret is formally defined as:

$$R(T) = \mathbb{E}[\sum_{t=1}^{T} r_{t,a_t^*}] - \mathbb{E}[\sum_{t=1}^{T} r_{t,a_t}],$$

where $a_t^* = \arg\max_{a \in \mathcal{A}_t} x_{t,a_t}^\top \theta^*$ denotes the arm with the optimal reward.

The task of personalized recommendation can be naturally modelled as a contextual bandit problem. At a high level, the contextual bandit aims to sequentially select arms (i.e., recommend items) to serve users when arrived in an online stream. The core of the bandit algorithm is an arm-selection strategy which is based on contextual information of the user (e.g., age and hobbies) and can be updated using user-item historical feedback (click or not). The goal of the algorithm is to maximize the total reward (i.e., total user clicks) in the long run. This paper considers the most widely studied linear contextual bandits, which are formally defined as follows.

Figure 2 shows a high-level comparison between centralized and vertical federated settings. In summary, user-item contextual information is collected by different participants and the figure uses different colours to distinguish the source of the information (e.g., information marked with green colour is from participant 1).

The number of circles denotes the specific feature value. Each row of the table contains the contextual information of one given user (denoted as $x_{t,a}$). The top box denotes the service provider, which is responsible for building contextual bandits for recommending items to users [2]. For the centralized setting (Figure 2a), the service provider has access to all contextual information even though they belong to different participants. In contrast, in the vertical federated setting (Figure 2b), the service provider cannot access the information directly from other participants due to privacy issues (there are locks between different columns in Figure 2b). Therefore, this information needs to be used in a privacy-preserving way which is the main goal of this paper. Specifically, we present an effective yet lossless way to use this information while protecting the users' privacy. We will present our solution in Section 4.

3.2 Contextual Bandit in the Centralized Setting

We first introduce the most commonly-used contextual bandit algorithms, LinUCB and LinTS in the centralized setting, i.e., all users' contextual information can be fully accessed by the service provider.

In this setting, the recommendation service provider runs the linear contextual algorithm A which proceeds in discrete trials indexed by time $t = 1, 2, 3, \cdots, T$. In trial, t, the learning procedure can be formulated as the following steps:

(1) A observes an user u_t and current arm set $\mathcal{A}_t = [K]$ consisting of K different arms a (e.g., items for recommendation). The *context* $x_{t,a}$ can be seen as a feature vector describing the contextual information of both user u_t and the given arm a.

(2) A chooses arm based on the estimated value $v_{t,a}$ for each arm

$$a_t = \arg\max_{a \in \mathcal{A}_t} v_{t,a}. \tag{1}$$

For LinUCB, detailed in the left part of Figure 2, $v_{t,a}$ is called UCB value with definition $v_{t,a} = \hat{r}_{t,a} + \hat{B}_{t,a}$, where $\hat{r}_{t,a} := x_{t,a}^\top \hat{\theta}_t$ is the estimated mean for item a and $\hat{B}_{t,a} := \beta_t \sqrt{x_{t,a}^\top \Lambda_t^{-1} x_{t,a}}$ is the optimistic term to encourage exploring new items. In particular, Abbasi-Yadkori et al. prove that the true reward of arm a is less than the $\hat{r}_{t,a} + \hat{B}_{t,a}$ with high probability, thus $\hat{r}_{t,a} + \hat{B}_{t,a}$ is an optimistic estimation for the true reward of arm a for some appropriate β_t [1]. Here $\Lambda_t := \sum_{s=1}^{t-1} x_{s,a_s} x_{s,a_s}^\top + \lambda \cdot I$ for some $\lambda > 0$. As for LinTS, $v_{t,a} = x_{t,a}^\top \hat{\mu}_t$ where $\hat{\mu}_t \sim \mathcal{N}(\hat{\theta}_t, v^2 \Lambda_t^{-1})$ with v^2 is the variance of the reward [3]. The sampling randomness in the $\tilde{\Lambda}_t$ encourages the algorithm to explore different arms.

(3) Once A recommends a_t for user u_t, then it receives a random reward $r_t = x_{t,a_t}^\top \theta^* + \epsilon_t$ where $\{\epsilon_t\}_{t=1}^T$ are i.i.d. sub-Gaussian mean zero noise [37]. This is demonstrated in the right part of Fig. 2. Here, $\theta^* \in \mathbf{R}^d$ is an unknown parameter used for generating rewards. $\hat{\theta}_t$ is a known parameter for predicting reward and the goal of A is to approximate θ^* with $\hat{\theta}_t$.

(4) Then A uses the newly observed reward r_t and the new sample covariance matrix Λ_{t+1} to update the estimator $\hat{\theta}_t$ via the Ordinary Least Square (OLS) algorithm, i.e., $\hat{\theta}_{t+1} = \Lambda_{t+1}^{-1} u_{t+1}$ where $\Lambda_{t+1} = \Lambda_t + x_{t,a_t} x_{t,a_t}^\top$ and $u_{t+1} = u_t + x_{t,a_t} r_t$.

[2]In Figure 2, the service provider is distinct from the data providers for simplicity. In practice, the service provider can also provide partial contextual information.

3.3 Vertical Federated Contextual Bandits

In this subsection, we formally present the VFCB problem, upon which we will design our privacy-protected algorithms.

Participant Role. In our VFCB setting, there is a single *active* participant (AP) and multiple *passive* participants (PP). Specifically, active participant directly interacts with users and holds historic user feedback (e.g., clicking an ad or not in the recommendation system). This participant is always the recommendation service provider. In contrast, passive participants can be seen as user feature providers who hold complementary contexts with the respective user. They do not directly interact with users. Without loss of generality, we assume the total number of participants is M, and we index the participants as $1, 2, \cdots, M$. Here the participant 1 is the AP and the remaining ones are passive. Additionally, a privacy mask generator(PMG) is required to perform our O3M protocols. This can either be a secure third party or be randomly selected from PP with some mechanism agreed upon.

Vertical Federated Setting. In contrast to the centralized setting, the vertical federated setting refers to that the user contexts ($x_{t,a}, a \in \mathcal{A}$) are distributively stored in different departments and cannot be shared since privacy concerns or data regulations. Formally speaking, a specific context $x_{t,a}$ is divided into different parts $x_{t,a} = [x_{t,a}^1, x_{t,a}^2, \cdots, x_{t,a}^M]$ and the j-th local context $x_{t,a}^j \in \mathbf{R}^{d \times d_j}$ can only be accessed by the j-th participant. In this setting, the core task is how to jointly leverage this local contextual information to approximate or even recover the reward mean and the upper-confidence bound in the centralized setting, i.e., presented in Eq 1. As introduced in Section 4.2, this paper presents a simple yet effective way to achieve this task, which is illustrated in Figure 3.

Threat Models. We consider all parties involved in VFCB semi-honest in our paper. This is in line with previous work on vertical federated learning [29]. The semi-honest model is not as strong as other threat models but nonetheless, it is still powerful for our inter-departmental vertical federated learning setting. In a semi-honest setting, each participant adheres to the designed protocol but it may attempt to infer information about other participant's input (i.e., local contextual information). Additionally, our threat model assumes that our PMG cannot collude with AP under any circumstances. This ensures the privacy constraint can be achieved via following privacy analysis.

4 ALGORITHM

In this section, we present our O3M privacy technique and then demonstrate how to apply it in the linear contextual bandits model to encrypt the user's information.

4.1 Challenges

As shown by the previous literature, LinUCB and LinTS algorithms can both achieve rate-optimal regret for the linear contextual bandits problem. In this section, we design a privacy-protected variant of them to adapt to the vertical federated setting while perfectly recovering to the centralized one. In fact, developing vertical federated variants for them is nontrivial.

As illustrated in Figure 2, due to the vertical federated setting, each participant only has access to the private version of the contexts from other participants. Typically each participant can apply the general widely-used privacy-awareness technique, e.g., differential privacy (DP) or multi-party computation (MPC), to protect their context before sending it out to others. However, DP has been documented to significantly degrade the performance of the contextual bandits [32] while MPC would add a large amount of extra computational cost.

To design a VFCB framework without the above drawbacks, we have to step into the key quantities of the LinUCB/LinTS algorithm. For LinUCB, firstly, both the estimated reward and the optimistic terms $\hat{r}_{t,a}$ and $\hat{B}_{t,a}$ require the Λ_t in their computation. Using private contexts $\tilde{x}_{t,a_t}$ to construct the private version of Λ_t, denoted as $\tilde{\Lambda}_t$, might introduce extra bias. For $\hat{r}_{t,a}$, $\tilde{\Lambda}_t^{-1}$ then would be multiplied with the $\sum_{s=1}^{t} \tilde{x}_{s,a_s}\tilde{r}_s$, where $\tilde{x}_{s,a_s}$ and $\tilde{r}_s$ are both (possibly) private and different from the non-private one. Once the $\tilde{\theta}_t$ is obtained via the private $\tilde{\Lambda}_t^{-1}$ and private historical rewards, it then is utilized to multiply with the private context $\tilde{x}_{t,a}$ to construct the estimated reward for arm a at time t. For $\hat{B}_{t,a}$, $\tilde{\Lambda}_h^{-1}$ is utilized to multiply with the private context to derive the optimistic term $\tilde{x}_{t,a}^{\top}\tilde{\Lambda}_h^{-1}\tilde{x}_{t,a}$. For LinTS, it shares most of the computation steps as LinUCB, especially the sample covariance matrix $\tilde{\Lambda}_t$.

In short, to maintain a small gap between the vertical federated bandits and the centralized ones, we need to design a sound privacy-protection mechanism for the contexts and the rewards by exploiting the computation details described above. The target of the privacy-protection mechanism is to keep the estimator $\tilde{\theta}_t$ and $\tilde{\Lambda}_t$ derived from the private, contextual information, to stay close to the non-private one, without incurring too much extra computational cost.

4.2 O3M

Motivated by [5] on masking data for vertical federated single value decomposition, we propose O3M on contextual bandits.

From a high-level perspective, O3M allocates each participant a part of a pre-defined orthogonal matrix to mask their contexts and send the masked contexts to the AP. The AP sums the masked local contexts to derive the masked contexts and directly runs the LinUCB/LinTS upon them. We illustrate the O3M step by step with Figure 3. To help the presentation, we use d_j to denote the dimension for the local contextual information $x_{t,a}^j$ stored in participant j. The total dimension for the global contextual information $x_{t,a}$ is $d = \sum_{j=1}^{M} d_j$. [3]

In the beginning, all the participants agree with either a third party or a PP randomly selected by some mechanism as the Private Mask Generator (PMG). Since PMG is critical to the system's security, finding a trusted third party to serve as PMG is important. Our approach aligns with the semi-honest settings. One could employ a cryptographically secure method to generate the mask during initialization. Alignment of the user between participants is done via Private Set Intersection (PSI) [24, 28], a common technique under VFL settings across different participants.

Then the PMG will generate an orthogonal matrix $Q \in \mathbf{R}^{d\times d}$. This must not be the identity matrix. The orthogonal mask Q is partitioned by columns into M parts to match the dimension d_j of each participant's context $x_{t,a}^j$, i.e., $Q = [Q^1, Q^2, \cdots, Q^M]$ such that $Q^j \in \mathbf{R}^{d\times d_j}$. Each participant's mask is distinguished by colored solid squares (e.g. solid green mask Q^1 is dedicated for Participant 1). The mask Q^j is then securely sent to participant j (the lock on each arrow indicates the secure peer-to-peer transfer process).

After receiving the masks, each participant would utilize them to mask their local contexts. The details of masking can be seen in the lower section of Figure 3 and we use different colours to distinguish contexts for different participants (following the same colour scheme with the partitioned mask Q^j). The data provider j obtains the masked context $\tilde{x}_{t,a}^j$ by multiplying its mask Q^j (solid square) with its local context $x_{t,a}^j$ (horizontal hollow bars) $\tilde{x}_{t,a}^j$ (vertical hollow bars). Note that after the masking, the masked context $\tilde{x}_{t,a}^j$ shares the same shape as the global context $\tilde{x}_{t,a}$. The data provider then sends her masked contexts to the AP and the AP obtains the masked context $\tilde{x}_{t,a}$ by directly adding the masked local contexts together, following $\tilde{x}_{t,a} = \sum_{j=1}^{M} \tilde{x}_{t,a}^j$. If a user's data is only available to some participants, the remaining participants should send a dummy value to the service provider to preserve privacy. Then the AP runs the LinUCB/LinTS protocol on the private contexts $\tilde{x}_{t,a}$ for $a \in \mathcal{A}_t$. We summarize the above procedure into the VFUCB in algo. 1 and the corresponding VFTS is deferred in Appendix 2.

Note that our O3M is not a direct application of [5]. We successfully identify the key components in the seminal bandits frameworks and transform them into the VFL setting by uncovering the linkage between LinUCB/TS and PCA. As PCA and bandits problem are inherently different, how to utilise the mask to develop VF bandits algorithms and what the performance and computational/storage cost guarantee are all not present with [5]. Our method can be naturally extended to other bandits algorithms, for example, generalized linear bandits [22] and Phased elimination with G-optimal exploration [19], using the same structure for exploration as in LinUCB (line328). The structure inherently contains the uncertainty for contexts and thus is shared by most contextual bandits algorithms.

4.3 Lossless Recommendation

In this subsection, we will prove that our VFUCB/VFTS algorithm running on private data can behave perfectly the same as the centralized ones used on non-private data. [4]

We provide the theoretical justification to show that, although the masked contexts are biased from the original one, all the bias will be cancelled out in the estimators constructed from the masked contexts and the estimators can perfectly recover to the non-private ones. The key observations of our analysis are two-fold: The first is that for any matrix $Q \in \mathbf{R}^{d\times d}$, any vector $x \in \mathbf{R}^{d\times 1}$, with any column partition $Q = [Q^1, Q^2, \cdots, Q^M]$ where $Q^j \in \mathbf{R}^{d\times d_j}$ and $x = [x^1, x^2, \cdots, x^M]$ where $x^j \in \mathbf{R}^{d_j\times 1}$, we have $Qx = \sum_{j=1}^{M} Q^j x^j$. Thus, the summation of all the masked contexts $\sum_{j=1}^{M} Q^j x_{t,a}^j$, is exactly the transformed context with the orthogonal matrix Q as the transformer, i.e., $Qx_{t,a}$.

[3] $Q \in R^{d\times d}$ should be a square as $\sum d_j = d$. However, mask Q and subsequent Q^js are drawn not to scale in Figure 3 for illustration purposes.

[4] For demonstration purposes, we focus on VFUCB for lossless recommendation analysis. The analysis for VFTS can be found in Appendix A.1

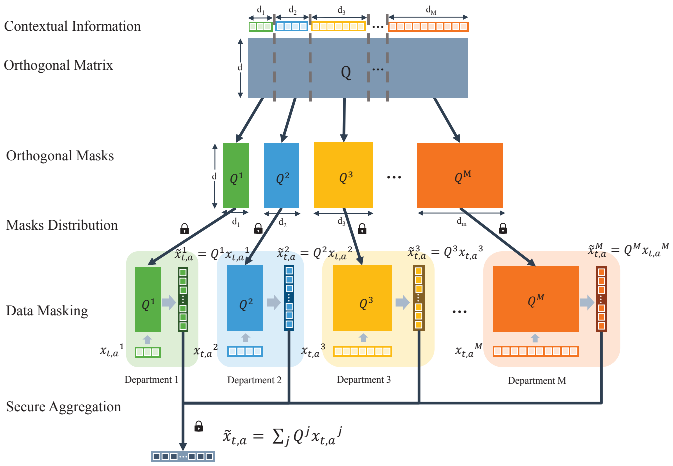


Figure 3: Overview for O3M process in VFCB. In PMG, the orthogonal mask $Q \in R^{d\times d}$ is partitioned with context columns d_j into $Q^j \in R^{d\times d_j}$. The mask Q_j is then distributed to each participant j securely. After that, each participant conducts masking process, producing $\tilde{x}^j_{t,a} = Q_j x_{t,j}$. Finally, the data is securely aggregate as $\tilde{x}_{t,a} = \sum_j Q_j x_{t,j}$.

Algorithm 1: VFUCB

Init: $\tilde{\Lambda}_1 = \mathbf{I}_d$, $\tilde{u}_1 = \mathbf{0}$

Any PP or a secure third party randomly generates an orthogonal Q and conducts a column partition $Q = [Q^1, Q^2, \cdots, Q^M]$ where $Q^i \in \mathbf{R}^{d\times d_i}$ and send the i-th submatrix to the client i

for $t = 1$ **to** T **do**

  An action set $X_t = \{x_{t,i}\}_{i=1}^K$ arrives to the system while the client j can only observe $x^j_{t,i} \in \mathbf{R}^{d_j\times 1}$ and $x_{t,i} = [x^1_{t,i}; x^2_{t,i}; \cdots; x^M_{t,i}]$

  For each $j \in [2, \cdots, M]$, Client j send $\{Q^j x^j_{t,i}\}_{i\in[K]}$ to the AP to derive $\tilde{x}_{t,i} = \sum_{j=1}^M Q^j x^j_{t,i}$

  AP recommend the $a_t \in [K]$ item via

$$a_t = \arg\max_{a\in\mathcal{A}} \tilde{r}_{t,a} + \tilde{B}_{t,a}$$

  where

$$\tilde{r}_{t,a} := \tilde{x}^\top_{t,i}\tilde{\theta}_t, \quad \tilde{B}_{t,a} := \beta_t\sqrt{\tilde{x}^\top_{t,i}\tilde{\Lambda}^{-1}_t\tilde{x}_{t,i}}. \tag{2}$$

  AP receives r_t from the user

  AP update Λ_{t+1} via $\tilde{\Lambda}_{t+1} = \tilde{\Lambda}_t + \tilde{x}_{t,i}\tilde{x}^\top_{t,i}$

  Update $\tilde{u}_{t+1}$ via $\tilde{u}_{t+1} = \tilde{u}_t + r_t\tilde{x}_{t,i}$

  AP update $\tilde{\theta}$ via $\tilde{\theta}_{t+1} = \tilde{\Lambda}^{-1}_{t+1}\tilde{u}_{t+1}$.

end for

Thus, the AP is in fact running a centralized linear contextual bandits algorithms on the private contexts $\tilde{x}_{t,a} = Qx_{t,a}$. We will prove the high privacy protection level using the O3M, i.e., the masked contexts can hardly reveal the information about the original contexts. While the gap between masked contexts and original ones is inevitable in safeguarding privacy, we need to ensure the statistics utility is not severely damaged so that our VFCB/VFTS can still achieve the rate-optimal regret bound. Here comes our second observation: by choosing the mask matrix Q as any orthogonal matrix in our O3M, our VFUCB/VFTS can losslessly recover to the non-private LinUCB/LinTS algorithms.

To be more concrete, taking LinUCB as an example, the decision is made by choosing the item with the largest UCB values, summation of private predicted reward $\tilde{r}_{t,a}$ and the optimistic term $\tilde{r}_{t,a}$. Thus we will prove that $\tilde{r}_{t,a}$ and $\tilde{B}_{t,a}$ are exactly the same as the estimated rewards $\hat{r}_{t,a}$ and optimistic terms $\tilde{B}_{t,a}$ constructed in the centralized LinUCB bandits. We formalized our observations in the following theorem.

Theorem 4.1. *Given the fixed sequence $\{x_{t,a}\}_{t\in[T],a\in A}$ and corresponding return sequence $\{r_{t,a}\}_{t\in[T],a\in A}$, for any time $t \in [T]$, we have:*

- *The estimated value obtained by LinUCB (1) $\hat{r}_{t,a}$, is the same as the one obtained by the VFUCB (Algorithm 1) $\tilde{r}_{t,a}$.*
- *The optimistic value obtained by LinUCB (1) $\hat{B}_{t,a}$, is the same as the one obtained by the VFUCB (Algorithm 1) $\tilde{B}_{t,a}$.*

Proof. Our proof is completed using the mathematical induction:

(1) For $t = 0$, we have $\hat{\theta}_0 = \tilde{\theta}_0 = \mathbf{0}$ and $\hat{\Lambda}_0 = \tilde{\Lambda}_0 = \lambda I$ following the LinUCB and VFUCB. For any $a \in \mathcal{A}_t$,

$$\begin{aligned}\tilde{B}_{0,a} &= \beta_0\sqrt{\tilde{x}^\top_{t,a}\tilde{\Lambda}^{-1}_0\tilde{x}_{t,a}} = \beta_0\sqrt{\lambda^{-1}x^\top_{t,a}Q^\top QQ^\top Qx_{t,a}}\\ &= \beta_0\sqrt{\lambda^{-1}x^\top_{t,a}x_{t,a}} = \beta_0\sqrt{x^\top_{t,a}\Lambda^{-1}_0 x_{t,a}} = \hat{B}_{0,a}.\end{aligned} \tag{3}$$

(2) Assume $\hat{\theta}_s = \tilde{\theta}_s$ and $\hat{B}_{s,a} = \tilde{B}_{s,a}$ for $s \in [t-1]$ and all $a \in \mathcal{A}_s$. Since the decision at any time s is determined by the estimated values and optimistic terms for each arm, the historically selected actions by the LinUCB are the same as those of the VFUCB. Now we consider the case t.

We first prove the relationship between $\tilde{\Lambda}_t$ and Λ_t.

$$\begin{aligned}\tilde{\Lambda}^{-1}_t &= \left(\lambda I + \sum_{s=1}^{t-1}\left(\sum_{j\in[M]} Q^j x^j_{t,a_t}\right)\left(\sum_{j\in[M]} Q^j x^j_{t,a_t}\right)^\top\right)^{-1}\\ &= \left(\lambda I + \sum_{s=1}^{t-1} Qx_{t,a_t}(Qx_{t,a_t})^\top\right)^{-1}\\ &= \left(\lambda I + \sum_{s=1}^{t-1} Qx_{t,a_t}x^\top_{t,a_t}Q^\top\right)^{-1} = \left(Q\Lambda_t Q^\top\right)^{-1} = Q\Lambda^{-1}_t Q^\top.\end{aligned} \tag{4}$$

Thus for any $t \in [T]$ and $a \in \mathcal{A}_t$, we first prove that $\hat{B}_{t,a} = \tilde{B}_{t,a}$, $\hat{B}_{t,a} = x^\top_{t,a}\Lambda^{-1}_t x_{t,a} = x^\top_{t,a}Q^\top Q\Lambda^{-1}_t Q^\top Qx_{t,a} = \tilde{x}^\top_{t,a}\tilde{\Lambda}^{-1}_t\tilde{x}_{t,a} = \tilde{B}_{t,a}$.

Next, we verify that $\hat{r}_{t,a} = \tilde{r}_{t,a}$, we decompose into the following three steps to finish. As we know, $\hat{r}_{t,a} = \hat{\theta}^\top_t x_{t,a}$ and thus we first investigate the relationship between $\hat{\theta}_t$ and $\tilde{\theta}_t$. Then, $\tilde{\theta}_t = \tilde{\Lambda}^{-1}_t\tilde{u}_t = Q\Lambda^{-1}_t Q^\top Q(\sum_{s=1}^t r_s x_{s,a_s}) = Q\Lambda^{-1}_t(\sum_{s=1}^t r_s x_{s,a_s}) = Q\hat{\theta}_t$. Finally, $\tilde{r}_{t,a} = \sum_{j=1}^M (Q^j x^j_{t,a})^\top\tilde{\theta}_t = (Qx_{t,i})^\top\tilde{\theta}_t = x^\top_{t,i}Q^\top Q\hat{\theta}_t = x^\top_{t,i}\hat{\theta}_t = \hat{r}_{t,a}$.

Now we prove that $\tilde{r}_{t,a} = \hat{r}_{t,a}$ and $\tilde{B}_{t,a} = \hat{B}_{t,a}$ for all $a \in \mathcal{A}_t$ and by the mathematical induction, these conclusions hold for all $t \in [T]$.

□

4.4 Complexity Analysis

We present the complexity analysis for VFCB in terms of computation and communication cost in Table 1. We follow the notation introduced in the previous section. The detailed derivations are presented in Appendix C.

Table 1: Complexity Analysis for VFCB

Model	computation cost	communication cost
VFUCB	$O(T \times (K \times M \times d + K \times d^2 + d^3))$	$O(T \times K \times M \times d)$
LinUCB	$O(T \times (K \times d^2 + d^3))$	N/A
VFTS	$O(T \times (K \times M \times d + K \times d^2 + d^3))$	$O(T \times K \times M \times d)$
LinTS	$O(T \times (K \times d + d^3))$	N/A

The table shows that O3M introduces a common cost of $O(K \times M \times d + K \times d^2)$ for computing the mask. This is the major change in computational cost in our VFUCB protocol. On the other hand, for VFTS, as sampling only takes once per step t, the lower cost for calculating a_t results in a complexity of only $(K \times d)$. Hence the O3M process contributes more and results in $O(K \times d^2)$ in VFTS. However, as we will discuss in Section 5.2, in practice, the complexity of VFTS will be manageable under realistic settings of variables. As for communication cost, our communication complexity is linear towards the respective variables T, K, d, M.

4.5 Privacy Analysis

We first give an overview of O3M followed by a detailed privacy analysis. Specifically, our approach aims to protect the user's feedback at time y, which is only accessible to the active participant. The statistics that contain r_t, namely $\tilde{u_t}$, are also retained by the active participant and not shared with other participants. This aligns with our semi-honest setting. However, under a stricter threat model, d_j may also be considered sensitive, which our approach cannot protect.

As illustrated in Section 4.2, a selected PMG generates an orthogonal matrix Q as the mask. In practice, this mask can be generated by generating a random matrix and then applying Gram-Schmitz orthogonalization. This mask is then used throughout the O3M process. Each participant then follows their VFCB protocols such as Algorithm 1 to carry O3M on respective bandit algorithms.

To show that the O3M is indeed secure in our semi-honest threat model, we need to prove that the masked data $\tilde{x}^j_{t,a}$ cannot be recovered without knowing the mask Q^j. Note that the mask must not be an identity matrix in order to be an effective mask. This is achieved via the following theorem[5]:

Theorem 4.2. *Given a masked data $D = Q_1X_1$, there are infinite number of raw data X_2 that can be masked into D, i.e. $Q_2X_2 = Q_1X_1 = D$*

This means that even AP knows all the masked data, given that it does not know any mask (other than the one associated with itself), since there are infinite number of mask and data combinations, it's not possible to directly infer PP's original user context data. Our VFCB protocols can't leak data from the statistical property, since the bandits algorithms don't assume the existence of the data distribution. After applying the rotation to the original context X_1, D would not have the same distribution as X_1. Hence the user context data privacy is preserved within each PP.

[5]The proof of the theorem can be find in Appendix B

5 EXPERIMENT

In this section, we design experiments based on both simulation and real-world datasets to answer the following questions:

(1) What are the performance gap between VFCB with privacy protection and the centralized bandits without privacy protection?
(2) What is the extra computation and communication cost for the VFCB compared with their centralized variant?
(3) Whether more data introduced from other participants improves the performance of VFCB?

In summary, Q1 and Q2 address the cost of safeguarding privacy in vertical federated settings while Q3 depicts the improvements from joint training using different data sources. In the following subsections, we first describe the settings for both synthetic and real-world datasets. Then, we use the experiment results to answer the three questions proposed. For consistency, we adopt the same notation from previous sections. Details on the experiment environment can be found at Appendix D.1.

5.1 Experiment Settings

Synthetic Experiment. In our synthetic dataset, $\forall t \in [T]$ and $\forall a \in [K]$, the context $x_{t,a}$ is generated from a multivariate normal distribution $x_{t,i} \sim \mathcal{N}(0, \sigma^2 I)$ with $\sigma^2 = 0.05$. It is then normalized with l2 norm. The reward generating parameter θ is generated from a multivariate normal distribution $\theta \sim \mathcal{N}(0, \sigma^2 I)$ with $\sigma^2 = 0.05$. It is also normalized with l2 norm. The reward r_t is then generated as $r_t = x^T_{t,a}\theta + \epsilon$ where $\epsilon \sim \mathcal{N}(0, 0.05)$. The regret is then measured with $R(T) = \sum_{t=1}^{T} x^T_{t,a^*}\theta - x^T_{t,a_t}\theta$. The details for the experiment runs can be found in Appendix D.2.

Criteo dataset. Criteo dataset [18] is an ad stream log dataset provided by Criteo AI Lab on the Kaggle Ad display challenge. Since the dataset is anonymized and the label meanings are unknown, we follow the previous approach and assumptions from [26] to construct the experiment dataset. The details for the experiment runs can be found in Appendix D.3.

5.2 Experiment Discussion

We demonstrate our experimental results in Figure 4, Figure 5 and Figure 6. We then use these results to answer in details the 3 questions we asked previously.

The performance gap between VFCB with privacy protection and that of without privacy protection. Although we already establish the theoretical guarantee that our VFCB is lossless in Section 4.3, we want to use experimental results to demonstrate that VFCB is indeed overall lossless. Furthermore, we want to show that VFCB protocols can make the same decision as their centralized variant on experimental data. The result of our synthetic analysis is demonstrated in Figure 4.

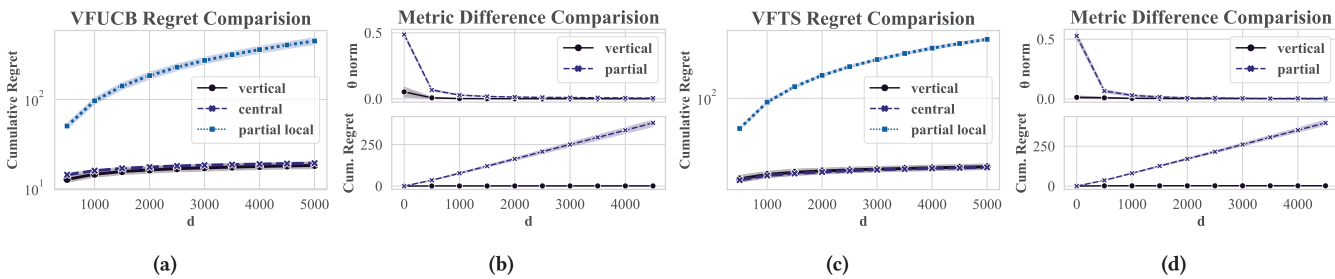


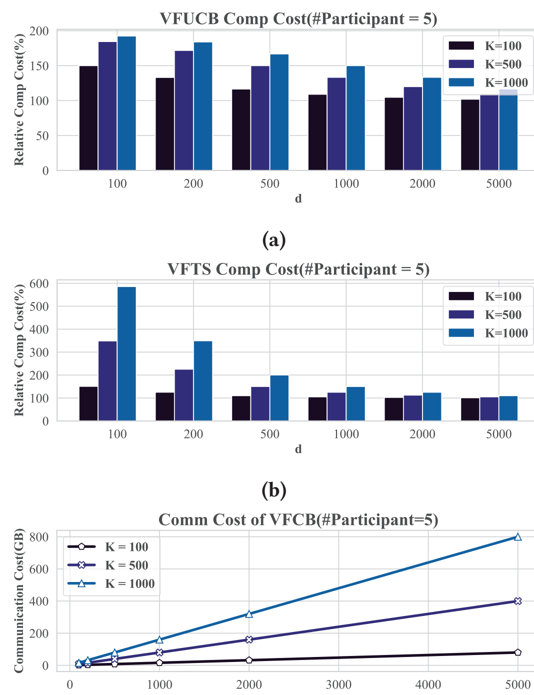
Figure 4: Experimental results for (a) Comparison between cumulative regret of central LinUCB, partial-local LinUCB and VFUCB on a synthetic dataset. (b) Comparison in metrics difference for VFUCB and partial-local LinUCB with Central LinUCB. (c) Comparison between cumulative regret of central LinTS, partial-local LinTS and VFTS on synthetic dataset. (d) Comparison in metrics difference for VFTS and partial-local LinTS with Central LinTS. For ease of comparison, the y scale for (a) and (c) is changed to log scale to differentiate between VFCB and partial bandits. The shaded area represents the confidence interval with 5 repeats. Both *partial local* labels on (a) and (c) represent the bandit algorithm that is only utilizing only the first partition of the data. The *partial* labels on (b) and (d) refer to the metric difference between partial local bandits and central bandits. Noted that on (a) partial local regret is slightly lower than central. This is due to the numerical accuracy issue in the implementation and does not implicate the theoretical lossless guarantee for VFUCB.

As shown in Figure (4a), our VFUCB protocol completely achieves the same regret of central LinUCB. However, partial LinUCB cannot perform as well as VFCB. The regret of it grows significantly compared with VFUCB and full LinUCB, resulting in more than 10 times larger cumulative regret. Similarly, for Figure (4c), our VFTS protocol achieves similar cumulative regret as central LinTS while the cumulative regret for partial LinTS grows to more than 10 times larger.

We further investigate the reconstruction error of the VFCB and partial bandits in Figure (4b) and Figure (4d). In particular, we plot $\|\widehat{\theta}_t^{\text{VFL}}\|_2 - \|\widehat{\theta}_t^{\text{Central}}\|_2$ and $\|\widehat{\theta}_t^{\text{Partial}}\|_2 - \|\widehat{\theta}_t^{\text{Central}}\|_2$ to check the reconstruction error in the estimator for each bandit model. Our VFCB metrics quickly recover to the centralized bandits at almost 0 difference, while the partial recovery is much slower. Additionally, the cumulative regret difference between partial bandits and full bandits grows continuously to more than 250 while our VFCB protocols difference remains at almost 0.

Hence, we can show that the performance gap between VFCB and their central variants is small as our VFUCB and VFTS protocols can indeed achieve the same cumulative regret target as LinUCB and LinTS.

Extra computation and communication cost for the VFCB. We first perform a synthetic calculation for VFCB protocols regarding computational cost. For the computational cost, we assume that a single arithmetic operation, comparison for one element of the matrix has one operation cost. For other known costs outlined in the prior analysis, we follow them directly in the computation. [6] We present the computational cost difference in Figure (5a) and Figure (5b) under realistic settings: Steps $T = 5000$, item size $K = 100, 500, 1000$ and number of participant $M = 5$. For ease of comparison, We use the relative computational cost[7] in our figure. As shown in Figure (5a), VFUCB only increases the computational



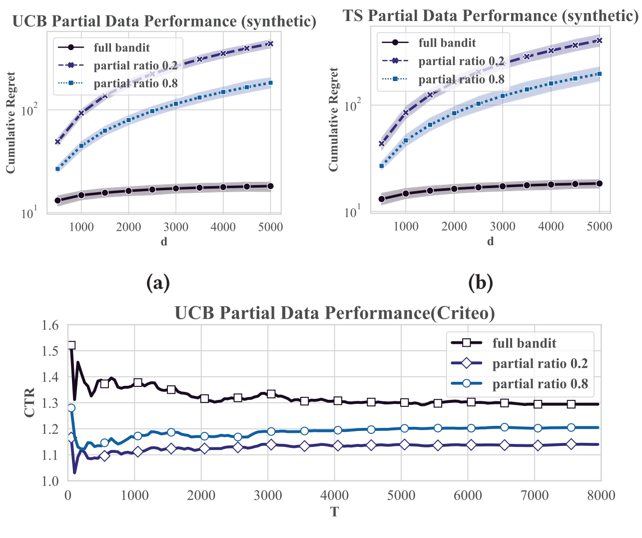
(c)

Figure 5: Complexity costs for (a) Relative Computational Cost for VFUCB. (b) Relative Computational Cost for VFTS. (c) Communication Cost in GB for VFCB active participants.

cost by a maximum of 2 fold on a typical participant setting. Further, this increase is marginal after either item size K, or contextual dimension d grows significant. This shows that compared with LinUCB, VFUCB does not introduce a significant computational complexity burden. As of VFTS, due to the O3M complexity introduced as $O(K \times d^2)$, the relative cost is initially higher at above 5x, but decreases to almost identical with LinTS as context dimension d grows larger, resulting in higher $O(d^3)$ term. Additionally, due to

[6]The details for computational complexity analysis and assumptions can be found in Appendix C.

[7]The relative cost can be calculated via $cost_{relative} = \frac{cost_{VFCB}}{cost_{central}}$.



(a) (b)

(c)

Figure 6: Experimental results for (a) Regret performance with 20% and 80% partial data for LinUCB on synthetic data. (b) Regret performance with 20% and 80% partial data for LinTS on synthetic data. (c) Comparison between relative CTR for VFUCB and Partial LinUCB. For ease of comparison, the y scale for (a) and (b) is changed to log scale to differentiate between VFCB and partial bandits.The partial ratio is $\#data_{used}/\#data_{total}$.

the expensive cost of sampling a multivariate normal distribution (which is also $O(n^3)$), the effect should be much more marginal practically. After all, the overall computational cost for our VFCB is still much smaller compared with the high complexity introduced by homomorphic encryption.

We then analyse communication cost in Figure (5c). We assume that each array element is a double floating point number of 8 byte and each array is transferred without any compression. This means that our communication cost calculated here should be an upper bound. Since the communication process is contributed by O3M only, both VFCB protocols should have the same communication cost. We present the cost with $T = 5000$ under the unit in GB at item size $k = 100, 500$ and 1000. The result shows that VFCB's communication cost grows linearly regarding item size K and context dimension d. Nonetheless, even under extreme settings of $d = 5000$ and $K = 1000$, the data transfer is still relatively small, at 6.25GB per step. Realistic settings would have a much smaller contextual size, resulting in smaller communication per step. For example, $d = 1000, K = 1000$ will only have 0.04GB per step. Additionally, our prior analysis shows that the communication complexity grows linearly with the number of participants M. However, since the number of participants in a vertical federated setting is much smaller compared with a horizontal federated setting, the cost increase is much smaller compared with the other two variables. As this is an upper bound estimated without compression, incorporating other data compression techniques is comparable with the high communication cost introduced in MPC.

In total, we show that the extra computational and communication cost introduced by O3M to our VFCB protocols is manageable under realistic settings. The value of T, k, d, M in our experiments are chosen as illustrative examples to demonstrate the manageability of the extra cost introduced by our VFCB protocols. These results empirically confirm our theoretical analysis of computational complexity and communication complexity in Section 4.4.

The effect of incorporating more data from other participants. We demonstrate that from both of our synthetic dataset in Figure (6a) and Figure (6b), our contextual bandits' performance increase when we incorporate more data from other participants. The synthetic cumulative regret bound is 10x smaller when using the full data than partial data. Additionally, compared with partial ratio=0.2, contextual bandit at partial ratio=0.8 has smaller cumulative regret. To ensure the effect of incorporating all data is applicable to the real-world datasets, we perform experiments on real-world datasets shown in Figure (6c). There is an increase of 0.1 in relative CTR metric from partial ratio=0.2 to partial ratio=0.8 and another 0.1 increase to full LinUCB. This shows that the positive effect of incorporating additional data from different departments is also true on real-world datasets. Therefore, using more context data from another department indeed helps improve the performance of the contextual bandit algorithm, which provides insight to improve the recommendation service quality in a federated manner.

Summary. In summary, our experiments show that our VFCB protocols are indeed lossless compared with the centralized variant. We then conduct synthetic analysis on the extra computational cost and communication cost introduced by O3M on VFCB, and show that the effect is still manageable. Finally, the experiment on both synthetic and real-world dataset suggest that our VFCB protocols can indeed benefit from additional data under the vertical federated settings. Therefore, these 3 answers combined to show the overall value of our VFCB protocols. It shows that our VFCB protocols can provide lossless recommendations with the benefit of additional inter-department features, while still maintaining privacy protection utility at a manageable cost.

6 CONCLUSION

This paper studies an important problem of building online bandit algorithms in the vertical federated setting for a privacy-protected recommendation. We emphasize the significance of this problem in real-world scenarios, where data privacy and compliance are crucial. We point out, a simple yet effective privacy-enhancing technique named O3M can be used to avoid heavy cryptographic operations while maintaining lossless performance for this problem. Since we are the first to formally study this problem to our best knowledge, there are several interesting directions to explore. For example, our algorithmic design is dedicated to exploiting the linear structure in contextual bandits, generalizing to nonlinear nature may be non-trivial. In addition, errors may arise in data provided by other participants, significantly affecting the recommendation performance. One could consider extending this to a variant of robust linear bandits. The research aims to provide insights into efficient bandit algorithms with privacy constraints and encourages greater attention from the community in this direction.

REFERENCES

[1] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. 2011. Improved Algorithms for Linear Stochastic Bandits. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain*, John Shawe-Taylor, Richard S. Zemel, Peter L. Bartlett, Fernando C. N. Pereira, and Kilian Q. Weinberger (Eds.). 2312–2320. https://proceedings.neurips.cc/paper/2011/hash/e1d5be1c7f2f456670de3d53c7b54f4a-Abstract.html

[2] Alekh Agarwal, John Langford, and Chen-Yu Wei. 2020. Federated residual learning. *arXiv preprint arXiv:2003.12880* (2020).

[3] Shipra Agrawal and Navin Goyal. 2013. Thompson sampling for contextual bandits with linear payoffs. In *International conference on machine learning*. PMLR, 127–135.

[4] Ghazaleh Beigi, Ahmadreza Mosallanezhad, Ruocheng Guo, Hamidreza Alvari, Alexander Nou, and Huan Liu. 2020. Privacy-Aware Recommendation with Private-Attribute Protection using Adversarial Learning. In *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020*, James Caverlee, Xia (Ben) Hu, Mounia Lalmas, and Wei Wang (Eds.). ACM, 34–42. https://doi.org/10.1145/3336191.3371832

[5] Di Chai, Leye Wang, Junxue Zhang, Liu Yang, Shuowei Cai, Kai Chen, and Qiang Yang. 2022. Practical Lossless Federated Singular Vector Decomposition over Billion-Scale Data. In *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, Aidong Zhang and Huzefa Rangwala (Eds.). ACM, 46–55. https://doi.org/10.1145/3534678.3539402

[6] Tianyi Chen, Xiao Jin, Yuejiao Sun, and Wotao Yin. 2020. Vafl: a method of vertical asynchronous federated learning. *arXiv preprint arXiv:2007.06081* (2020).

[7] Yiu-ming Cheung, Juyong Jiang, Feng Yu, and Jian Lou. 2022. Vertical Federated Principal Component Analysis and Its Kernel Extension on Feature-wise Distributed Data. *CoRR* abs/2203.01752 (2022). https://doi.org/10.48550/arXiv.2203.01752 arXiv:2203.01752

[8] Abhimanyu Dubey and Alex 'Sandy' Pentland. 2020. Differentially-Private Federated Linear Bandits. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). https://proceedings.neurips.cc/paper/2020/hash/4311359ed4969e8401880e3c1836fbe1-Abstract.html

[9] Cynthia Dwork. 2008. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*. Springer, 1–19.

[10] Cynthia Dwork, Aaron Roth, et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science* 9, 3–4 (2014), 211–407.

[11] Oded Goldreich. 1998. Secure multi-party computation. *Manuscript. Preliminary version* 78 (1998), 110.

[12] Yuxuan Han, Zhipeng Liang, Yang Wang, and Jiheng Zhang. 2021. Generalized Linear Bandits with Local Differential Privacy. In *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (Eds.), Vol. 34. Curran Associates, Inc., 26511–26522. https://proceedings.neurips.cc/paper/2021/file/df0e09d6f25a15a815563df9827f48fa-Paper.pdf

[13] Awni Y. Hannun, Brian Knott, Shubho Sengupta, and Laurens van der Maaten. 2019. Privacy-Preserving Contextual Bandits. *CoRR* abs/1910.05299 (2019). arXiv:1910.05299 http://arxiv.org/abs/1910.05299

[14] Stephen Hardy, Wilko Henecka, Hamish Ivey-Law, Richard Nock, Giorgio Patrini, Guillaume Smith, and Brian Thorne. 2017. Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption. *arXiv preprint arXiv:1711.10677* (2017).

[15] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2017. Fast Matrix Factorization for Online Recommendation with Implicit Feedback. *CoRR* abs/1708.05024 (2017). arXiv:1708.05024 http://arxiv.org/abs/1708.05024

[16] Yaochen Hu, Di Niu, Jianming Yang, and Shengping Zhou. 2019. FDML: A Collaborative Machine Learning Framework for Distributed Features. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, Ankur Teredesai, Vipin Kumar, Ying Li, Rómer Rosales, Evimaria Terzi, and George Karypis (Eds.). ACM, 2232–2240. https://doi.org/10.1145/3292500.3330765

[17] Ruiquan Huang, Weiqiang Wu, Jing Yang, and Cong Shen. 2021. Federated Linear Contextual Bandits. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (Eds.). 27057–27068. https://proceedings.neurips.cc/paper/2021/hash/e347c51419ffb23ca3fd5050202f9c3d-Abstract.html

[18] Criteo AI Labs. 2014. https://labs.criteo.com/2014/02/kaggle-display-advertising-challenge-dataset/

[19] Tor Lattimore and Csaba Szepesvári. 2020. *Bandit algorithms*. Cambridge University Press.

[20] Chuanhao Li and Hongning Wang. 2022. Asynchronous Upper Confidence Bound Algorithms for Federated Linear Bandits. In *International Conference on Artificial Intelligence and Statistics, AISTATS 2022, 28-30 March 2022, Virtual Event (Proceedings of Machine Learning Research, Vol. 151)*, Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera (Eds.). PMLR, 6529–6553. https://proceedings.mlr.press/v151/li22e.html

[21] Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010*, Michael Rappa, Paul Jones, Juliana Freire, and Soumen Chakrabarti (Eds.). ACM, 661–670. https://doi.org/10.1145/1772690.1772758

[22] Lihong Li, Yu Lu, and Dengyong Zhou. 2017. Provably optimal algorithms for generalized linear contextual bandits. In *International Conference on Machine Learning*. PMLR, 2071–2080.

[23] Tan Li, Linqi Song, and Christina Fragouli. 2020. Federated recommendation system via differential privacy. In *2020 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2592–2597.

[24] Gang Liang and Sudarshan S. Chawathe. 2004. Privacy-Preserving Inter-database Operations. In *Intelligence and Security Informatics*.

[25] Blerina Lika, Kostas Kolomvatsos, and Stathes Hadjiefthymiades. 2014. Facing the cold start problem in recommender systems. *Expert Syst. Appl.* 41, 4 (2014), 2065–2073. https://doi.org/10.1016/j.eswa.2013.09.005

[26] Mohammad Malekzadeh, Dimitrios Athanasakis, Hamed Haddadi, and Benjamin Livshits. 2020. Privacy-Preserving Bandits. In *Proceedings of Machine Learning and Systems 2020, MLSys 2020, Austin, TX, USA, March 2-4, 2020*, Inderjit S. Dhillon, Dimitris S. Papailiopoulos, and Vivienne Sze (Eds.). mlsys.org. https://proceedings.mlsys.org/book/310.pdf

[27] Erxue Min, Yu Rong, Tingyang Xu, Yatao Bian, Da Luo, Kangyi Lin, Junzhou Huang, Sophia Ananiadou, and Peilin Zhao. 2022. Neighbour interaction based click-through rate prediction via graph-masked transformer. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 353–362.

[28] Benny Pinkas, T. Schneider, Gil Segev, and Michael Zohner. 2015. Phasing: Private Set Intersection Using Permutation-based Hashing. In *USENIX Security Symposium*.

[29] Zhenghang Ren, Liu Yang, and Kai Chen. 2022. Improving Availability of Vertical Federated Learning: Relaxing Inference on Non-overlapping Data. *ACM Trans. Intell. Syst. Technol.* 13, 4 (2022), 58:1–58:20. https://doi.org/10.1145/3501817

[30] Daniele Romanini, Adam James Hall, Pavlos Papadopoulos, Tom Titcombe, Abbas Ismail, Tudor Cebere, Robert Sandmann, Robin Roehm, and Michael A. Hoeh. 2021. PyVertical: A Vertical Federated Learning Framework for Multi-headed SplitNN. *CoRR* abs/2104.00489 (2021). arXiv:2104.00489 https://arxiv.org/abs/2104.00489

[31] Yu Rong, Xiao Wen, and Hong Cheng. 2014. A Monte Carlo algorithm for cold start recommendation. In *Proceedings of the 23rd international conference on World wide web*. 327–336.

[32] Roshan Shariff and Or Sheffet. 2018. Differentially Private Contextual Linear Bandits. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (Eds.). 4301–4311. https://proceedings.neurips.cc/paper/2018/hash/a1d7311f2a312426d710e1c617fcbc8c-Abstract.html

[33] Chengshuai Shi, Cong Shen, and Jing Yang. 2021. Federated multi-armed bandits with personalization. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 2917–2925.

[34] Chengshuai Shi, Cong Shen, and Jing Yang. 2021. Federated Multi-armed Bandits with Personalization. In *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event (Proceedings of Machine Learning Research, Vol. 130)*, Arindam Banerjee and Kenji Fukumizu (Eds.). PMLR, 2917–2925. http://proceedings.mlr.press/v130/shi21c.html

[35] Aleksandrs Slivkins et al. 2019. Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning* 12, 1-2 (2019), 1–286.

[36] Ambuj Tewari and Susan A. Murphy. 2017. From Ads to Interventions: Contextual Bandits in Mobile Health. In *Mobile Health - Sensors, Analytic Methods, and Applications*, James M. Rehg, Susan A. Murphy, and Santosh Kumar (Eds.). Springer, 495–517. https://doi.org/10.1007/978-3-319-51394-2_25

[37] Martin J Wainwright. 2019. *High-dimensional statistics: A non-asymptotic viewpoint*. Vol. 48. Cambridge University Press.

[38] Kang Wei, Jun Li, Chuan Ma, Ming Ding, Sha Wei, Fan Wu, Guihai Chen, and Thilina Ranbaduge. 2022. Vertical Federated Learning: Challenges, Methodologies and Experiments. *CoRR* abs/2202.04309 (2022). arXiv:2202.04309 https://arxiv.org/abs/2202.04309

[39] Yuncheng Wu, Shaofeng Cai, Xiaokui Xiao, Gang Chen, and Beng Chin Ooi. 2020. Privacy preserving vertical federated learning for tree-based models. *arXiv preprint arXiv:2008.06170* (2020).

[40] Kai Zheng, Tianle Cai, Weiran Huang, Zhenguo Li, and Liwei Wang. 2020. Locally Differentially Private (Contextual) Bandits Learning. In *Advances in*

Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). https://proceedings.neurips.cc/paper/2020/hash/908c9a564a86426585b29f5335b619bc-Abstract.html

[41] Zhaowei Zhu, Jingxuan Zhu, Ji Liu, and Yang Liu. 2021. Federated bandit: A gossiping approach. In *Abstract Proceedings of the 2021 ACM SIGMETRICS/International Conference on Measurement and Modeling of Computer Systems*. 3–4.

A VFTS

A.1 VFTS Algorithm

Algorithm 2: VFTS

Input: prior variance v.
Initial $\tilde{\Lambda}_0 = \mathbf{I}_d$, $\tilde{u}_0 = \mathbf{0}$, and $\hat{\theta}_1 = \mathbf{0}$
Any PP or a secure third party randomly generates a orthogonal Q and conduct a column partition $Q = [Q^1, Q^2, \cdots, Q^M]$ where $Q^i \in \mathbf{R}^{d \times d_i}$ and send the i-th submatrix to the client i.
for $t = 1$ **to** T **do**
  An action set $X_t = \{x_{t,i}\}_{i=1}^K$ arrives to the system while the client j can only observe $x_{t,i}^j \in R^{d_j \times 1}$ and $x_{t,i} = [x_{t,i}^1, x_{t,i}^2, \cdots, x_{t,i}^M]$
  AP collects $\tilde{x}_{t,i} = \sum_{j=1}^M Q^j x_{t,i}^j$ from client 1 to M (includes itself)
  AP samples $\tilde{\mu}_t \sim \mathcal{N}(\tilde{\theta}_t, v^2 \tilde{\Lambda}_t^{-1})$ and recommend the item $a_t = \arg\max_{i \in [K]} \tilde{x}_{t,i}^\top \tilde{\mu}_t$
  AP receives r_t from the user
  AP updates its local estimator by
$$\tilde{\Lambda}_{t+1} = \tilde{\Lambda}_t + \tilde{x}_{t,a_t} \tilde{x}_{t,a_t}^\top, \quad \tilde{u}_{t+1} = \tilde{u}_t + r_t \tilde{x}_{t,a_t}$$
  and
$$\tilde{\theta}_{t+1} = \tilde{\Lambda}_{t+1}^{-1} \tilde{u}_{t+1}$$
end for

A.2 VFTS Loseless Proof

Following the established framework, we prove the following lemma to show VFTS is loseless.

Theorem A.1. *Given the fixed sequence $\{x_{t,a}\}_{t \in [T], a \in A}$ and corresponding return sequence $\{r_{t,a}\}_{t \in [T], a \in A}$, for any time $t \in [T]$, we have*

- *The estimated reward in the VFTS follows the same distribution as the one in the LinTS, i.e., $\tilde{x}_{t,a}^\top \tilde{\mu}_t$ is the same distribution as $x_{t,a}^\top \hat{\mu}_t$.*

Proof. From the design of the Algorithm 2, we have
$$\tilde{\theta}_t \sim \mathcal{N}(Q\tilde{\mu}_t, v^2 Q \hat{\Lambda}_t^{-1} Q^\top).$$
Recall that in the LinTS, $x_{t,a}^\top \hat{\mu}_t \sim \mathcal{N}(x_{t,a}^\top \hat{\theta}_t, v^2 x_{t,a}^\top \hat{\Lambda}_t^{-1} x_{t,a})$ Thus we know $\tilde{x}_{t,a}^\top \tilde{\mu}_t \sim \mathcal{N}(x_{t,a}^\top \hat{\theta}_t, v^2 x_{t,a}^\top \hat{\Lambda}_t^{-1} x_{t,a})$ follows the same distribution as $x_{t,a}^\top \hat{\mu}_t$ for all $a \in \mathcal{A}_t$. □

B PRIVACY ANALYSIS

Here we proof Theorem 4.2.

Proof. Consider an arbitrary rotational matrix R with its inverse R^{-1}. Masked data D is the result of multiplying the orthogonal mask Q_1 and the user context data X_1, i.e., $D = Q_1 X_1$. Let $Q_2 = Q_1 R$ and $X_2 = R^{-1} X_1$. Since an orthogonal matrix is still orthogonal after rotation, we have $D = Q_1 X_1 = Q_2 R^{-1} R X_2 = Q_2 X_2$. As R is arbitrary, we have infinite number of Q_2 and X_2. □

C COMPLEXITY ANALYSIS

C.1 Computational Complexity

We conduct computational analysis on both VFCB protocols. We assumed that for computational cost calculation, a single addition, multiplication, comparison or random generation operation for one element of the matrix has 1 operation cost. We then derive the standard addition, multiplication, dot for matrix operation base on this assumption to calculate the synthetic computational cost. These operation cost are unoptimized naive implementation cost to give an upper bound. For known Big-O cost such as n-dimensional matrix inverse ($O(n^3)$), we directly use them in our calculation.

As illustrated in Algorithm 1 and Algorithm 2, our VFCB protocols can be divided into the following stages:

(1) Mask initialization
(2) Selection process ($\forall t \in T$)
(3) Update process ($\forall t \in T$)

Hence, we perform complexity calculation for each stages of our VFCB protocols and combine them to provide an overall computational complexity. Note that the Mask initialization process is shared between both VFUCB and VFTS.

C.1.1 VFUCB.

Stage (1). In stage (1), our approach to generate orthogonal matrix Q is to generate a random matrix $M \in R^{d \times d}$, which cost $o(d^2)$, and conduct Gram-Schmitz orthogonalization ($O(d^3)$) on it to ensure that the resulted mask $Q \in R^{d \times d}$ matrix is orthogonal. The overall computation cost for this stage is $O(d^3)$.

Stage (2). In stage (2), $\forall t \in T$, we need to consider VFUCB selection process. The masking and adding process for $\tilde{x}_{t,i}$ for every item K is $O(K \times d^2) + O(K \times M \times d)$. The process of calculating $\tilde{r}_{t,i}$ for every item K is $O(K \times d)$. The process of calculating Λ_t^{-1} is $O(d^3)$, which is shared by all item K. The process of calculating $\tilde{B}_{t,i}$ for every item K is $O(K \times d^2)$. The process of calculating for every item K arg max a_t is $O(K)$. The overall cost of this stage is $O(T \times (K \times M \times d + K \times d^2 + d^3))$.

Stage (3). In stage (3), $\forall t \in T$, we need to consider VFUCB update process. The process of calculating Λ_{t+1} is $O(d^2)$. The process of calculating $\bar{u}_{t+1}$ is $O(d)$. The process of calculating $\bar{\theta}_{t+1}$ is $O(d^2)$. The overall cost of this stage is $O(T \times d^2)$.

Overall Cost. By combining the previous three stages, the overall cost for VFUCB is $O(T \times (K \times M \times d + K \times d^2 + d^3))$. Note that the cost is dynamically determined by the size of each variable.

C.1.2 LinUCB.

For LinUCB, stage (1) does not exist. Additionally, in stage (2), the cost for obtaining $\tilde{x}_{t,i}$ from O3M is not necessary. Hence the overall cost is the direct result of removing these two costs, which is $O(T \times (K \times d^2 + d^3)$.

C.1.3 VFTS.

Stage (1). In stage (1), our approach on O3M is identical with VFUCB. Hence the overall computation cost for this stage is $O(d^3)$.

Stage (2). In stage (2), $\forall t \in T$, we need to consider VFTS selection process. The masking and adding process for $\tilde{x}_{t,i}$ for every item K is $O(K \times d^2) + O(K \times M \times d)$. The process of calculating covariance matrix $v^2\Lambda_t^{-1}$ is $O(d^2)$. Note that the Λ_t^{-1} is computed in the stage (3) update process. The process of sampling of $\tilde{\mu}_t$ from multivariate normal distribution is $O(d^3)$ as Chelosky decomposition is $O(d^3)$. The process of calculating $\tilde{x}_{t,i}^{\top}\tilde{\mu}_t$ for every item $K$is $O(K \times d)$. The process of calculating $\arg\max a_t$ for every item K is $O(K)$. The overall cost of this stage is $O(T \times (K \times M \times d + K \times d^2 + d^3))$.

Stage (3). In stage (3), $\forall t \in T$, we need to consider VFTS update process. The process of calculating Λ_{t+1} is $O(d)$. The process of calculating Λ_{t+1}^{-1} is $O(d^3)$. The process of calculating u_{t+1} is $O(d)$. The process of calculating θ_{t+1} is $O(d^2)$. The overall cost of this stage is $O(T \times d^3)$.

Overall Cost. By combining the previous three stages, the overall cost for VFTS is $O(T \times (K \times M \times d + K \times d^2 + d^3))$. Noted that the cost is also dynamically determined by the size of each variable.

C.1.4 LinTS. For LinTS, similarly, stage(1) does not exist. Furthermore, the cost of O3M process does not exist as well. Hence the overall stage (2) cost is reduced to $O(T \times (K \times d + d^3))$. Note that, unlike LinUCB which also has $K \times d^2$ term during UCB value calculation, LinTS only have $K \times d$ term for calculation after sampling, therefore its overall cost reduces to $O(T \times (K \times d + d^3))$.

C.2 Communication Cost

We assume that the communication cost for each array element is 1. For convenience, we give an upper bound of this process, which means we also include the masked communication cost for the acting mask host PP in stage (1) and AP in stage (2). As shown in Algorithm 1 and Alogrithm 2, two stages of communication happened with VFCB.

(1) Mask Delivery (during initialization)
(2) Masked context delivery ($\forall t \in T$)

In stage (1), the mask delivery requires the Mask Generator PMG to send mask Q^i to client i. As $Q = [Q^1, Q^2, \cdots Q^j]$, the total mask delivery cost for $Q \in d \times d$ is d^2.

In stage (2), the masked sums are $Q^j x_{t,i}^{(j)} \in R^{d \times 1}$ for $i \in [K]$, $j \in [M]$ Hence the masked sum delivery cost at each t is $\sum_i \sum_j d = K \times M \times d$, and the total mask delivery sum is $T \times K \times M \times d$.

Therefore, the overall communication cost is $d^2 + T \times K \times M \times d$, which can be simplified to $O(T \times K \times M \times d)$, given that in practice, $T \times K \times M > d$.

C.3 Comparison with Other Privacy-Preserving Techniques

We compared the computation time between O3M and homomorphic encryption (HE), secure multiparty computation (MPC) and differential privacy (DP). We timed each cryptographic method by encrypting local contexts and using it to compute the statistic utilities required for the LinUCB/LinTS algorithms.

Settings	O3M(Ours)	HE	MPC	DP
Small	0.006	1190	2.282	0.005
Large	7.28	-	103	6.48

The table summarises the time used for each statistic utility calculation in seconds. The small setting uses $d = 100, K = 20, N = 5$, whereas the large setting uses $d = 500, K = 10, N = 500$. Due to the high computational requirements of HE, we limited our experiments to the small setting only. Note that, unlike O3M, DP is not lossless for its recommendation performance.

D EXPERIMENTS

D.1 Experiments Environment

Experiments on synthetic datasets and trail runs for real datasets are conducted on a workstation with AMD Epyc 7H12, 256G memory. The complete experiments for real-world dataset and hyperparameters search are conducted on cluster servers each with 23 cores of Intel 8252C, 110G RAM. All experiment environments are built from the docker image.

D.2 Synthetic Data Experiment Setting

To simulate the vertical federated settings, $\forall a \in [K]$, $x_{t,a}$ is partitioned into M parts which are held by the M th participant respectively. The number of contexts for each participant is derived from a partition vector $[1, \cdots, M]$, where M represents the number of columns partitioned for participant M. Hence the context partition is $x_{t,a} = [x_{t,a}^1, \cdots, x_{t,a}^M]$.

The central bandit's experiments are performed on $d = 100, K = 10, T = 5000$. The VFCB experiments are performed with the same parameters and a partition of $[20, 20, 20, 20, 20]$, which represents an equal dimension number in 5 participants. Each experiment is repeated five times to provide a confidence interval. $\beta_t = 0.5$ is used for LinUCB and $v = 0.01$ is used for LinTS.

D.3 Criteo Dataset Experiment Setting

Specifically, we assume that the user features are all numerical values and the item features are all categorical in [26]. Following their feature engineering method, we first use feature hashing to construct three hash-encoded values from 26 category values. After that, a pairing function pairs the three hash-encoded values to a single item label. Then, we filter the 40 most common labels as item labels in experimental data. However, instead of just using the item label, we use the three encoded values as item features for that item label. This helps to capture the user-item interaction in our single-parameter bandit setting. Finally, we build each log entry by contacting the respective user features(dimension d_u), item features(dimension d_i), item label and the respective response value r. We further perform feature engineering by min-max scaling the feature columns and multiplying an individual scaling factor for each user feature. The context features $x_{t,a} \in R^{d_u \times d_i}$ are obtained via outer addition between user features and item features for each log entry.

Following the approach of [21], we use the unbiased offline evaluation policy for our bandits. Similarly, we report the relative CTR ($CTR_{relative} = CTR_{actual}/CTR_{random}$) as the CTR metric. The $CTR_{randoom}$ is obtained by averaging the random policy among five random states. After a grid search of hyperparameter, $\beta_t = 0.6$ is used for LinUCB and all subsequent partial experiments for the dataset. We report each partial LinUCB experiment with 5 different random partitions.