PARTIAL INFORMATION AS FULL: REWARD IMPUTATION WITH SKETCHING IN BANDITS

Anonymous authors
Paper under double-blind review

ABSTRACT

We focus on the setting of contextual batched bandit (CBB), where a batch of rewards is observed from the environment in each episode. But these rewards are partial-information feedbacks where the rewards of the non-executed actions are unobserved. Existing approaches for CBB usually ignore the potential rewards of the non-executed actions, resulting in feedback information being underutilized. In this paper, we propose an efficient reward imputation approach using sketching in CBB, which completes the unobserved rewards with the imputed rewards approximating the full-information feedbacks. Specifically, we formulate the reward imputation as a problem of imputation regularized ridge regression, which captures the feedback mechanisms of both the non-executed and executed actions. To reduce the time complexity of reward imputation on a large batch of data, we use randomized sketching for solving the regression problem of imputation. We prove that the proposed reward imputation approach obtains a relative-error bound for sketching approximation, achieves an instantaneous regret with an exponentially-decaying bias and a smaller variance than that without reward imputation, and enjoys a sublinear regret bound against the optimal policy. Moreover, we present two extensions of our approach, including the rate-scheduled version and the version for nonlinear rewards, which makes our approach more feasible. Experimental results demonstrated that our approach can outperform the state-of-the-art baselines on a synthetic dataset, the Criteo dataset, and a dataset from a commercial app.

1 INTRODUCTION

Contextual bandits have been widely used in real-world sequential decision-making problems (Li et al., 2010; Lan & Baraniuk, 2016; Yom-Tov et al., 2017; Yang et al., 2021), where the agent updates the decision-making policy fully online (i.e., at each step) according to the context and corresponding reward feedback so as to maximize the cumulative reward. In this paper, we consider a more complex setting—*contextual batched bandits* (CBB), where the decision process is partitioned into N episodes, the agent interacts with the environment for B steps in one episode, collects the reward feedbacks and contexts at the end of the episode, and then updates the policy using the collected data for the next episode. CBB is more practical in some real-world applications, since updating the policy once receiving the reward feedback is rather unrealistic due to its high computational cost and decision instability.

In bandit settings, it is inevitable that the environment only reveals the rewards of the executed actions to the agent as the feedbacks, while hiding the rewards of non-executed actions. We refer to this category of limited feedback as the *partial-information feedback* (also called "bandit feedback"). Existing batched bandit approaches in the CBB setting discard the information contained in the potential rewards of the non-executed actions, address the problem of partial-information feedback using an exploitation-exploration tradeoff on the context space and reward space (Han et al., 2020; Zhang et al., 2020).But in the CBB setting, the agent usually estimates and maintains reward models for the action-selection policy, and the potential rewards of the non-executed actions have been somehow captured by the policy. This additional reward structure information is estimated and available in each episode, however, are not utilized by existing batched bandit approaches.

In contextual bandit settings where the policy is updated fully online, several bias-correction approaches have been introduced to address the partial-information feedback. Dimakopoulou et al.

(2019) presented linear contextual bandits integrating the balancing approach from causal inference, which reweight the contexts and rewards by the inverse propensity scores. Chou et al. (2015) designed pseudo-reward algorithms for contextual bandits, which use a direct method to estimate the unobserved rewards for the upper confidence bound (UCB) strategy. Kim & Paik (2019) focused on the feedback bias-correction for LASSO bandit with high-dimensional contexts, and applied the doubly-robust approach to the reward modification using average contexts. Although these approaches have been demonstrated to be effective in contextual bandit settings, little efforts have been spent to address the under-utilization of partial-information feedback in the CBB setting.

Theoretical and experimental analyses in Section 2 indicate that better performance of CBB is achievable if the rewards of the non-executed actions can be received. Motivated by these observations, we propose a novel reward imputation approach for the non-executed actions, which mimics the reward generation mechanisms of environments. We conclude our contributions as follows.

- To fully utilize feedback information in CBB, we formulate the reward imputation as a problem of imputation regularized ridge regression, where the policy can be updated efficiently using sketching.
- We prove that our reward imputation approach obtains a relative-error bound for sketching approximation, achieves an instantaneous regret with an exponentially-decaying bias and a smaller variance than that without reward imputation, has a lower bound of the sketch size independently of the overall number of steps, enjoys a sublinear regret bound against the optimal policy, and reduces the time complexity from $O(Bd^2)$ to $O(cd^2)$ for each action in one episode, where B denotes the batch size, c the sketch size, and d the dimensionality of inputs, satisfying $d < c < B$.
- We present two practical variants of our reward imputation approach, including the rate-scheduled version in which the imputation rate is set without tuning, and the version for nonlinear rewards.
- We carried out extensive experiments on the synthetic data, public benchmark, and the data collected from a real commercial product to demonstrate our performance, empirically analyzed the influence of different parameters, and verified the correctness of the theoretical results.

Related Work. Recently, batched bandit has become an active research topic in statistics and learning theory including 2-armed bandit (Perchet et al., 2016), multi-armed bandit (Gao et al., 2019; Zhang et al., 2020; Wang & Cheng, 2020), and contextual bandit (Han et al., 2020; Ren & Zhou, 2020; Gu et al., 2021). Han et al. (2020) defined linear contextual bandits, and designed UCB-type algorithms for both stochastic and adversarial contexts, where true rewards of different actions have the same parameters. Zhang et al. (2020) provided methods for inference on data collected in batches using bandits, and introduced a batched least squares estimator which is asymptotically normal for both multi-arm and contextual bandits. Recently, Esfandiari et al. (2021) proved refined regret upper bounds of batched bandits in stochastic and adversarial settings. There are several recent works that consider similar settings to CBB and propose novel approaches benefited from bandits. Jin et al. (2018) focused on the setting of episodic Markov decision process, and analyzed the regret of Q-learning equipped with bandit exploration. Wang & Cheng (2020) designed LASSO bandits with regret guarantees for online batch learning setting under high-dimensional context space.

2 Problem Formulation and Analysis

First, we introduce some notations. Let $[x] = \{1, 2, \ldots, x\}$, $\mathcal{S} \subseteq \mathbb{R}^d$ be the context space (could be adversarial), $\mathcal{A} = \{A_j\}_{j\in[M]}$ the action space, $[\boldsymbol{A}; \boldsymbol{B}] = [\boldsymbol{A}^\intercal, \boldsymbol{B}^\intercal]^\intercal$, $\|\boldsymbol{A}\|_{\mathrm{F}}$, $\|\boldsymbol{A}\|_1$ $\|\boldsymbol{A}\|_2$ denote the Frobenius norm, 1-norm, and spectral norm of a matrix $\boldsymbol{A}$, respectively, $\|\boldsymbol{a}\|_1$ and $\|\boldsymbol{a}\|_2$ be the ℓ_1-norm and the ℓ_2-norm of a vector $\boldsymbol{a}$, $\sigma_{\min}(\boldsymbol{A})$ and $\sigma_{\max}(\boldsymbol{A})$ denote the minimum and maximum of the of singular values of $\boldsymbol{A}$. In this paper, we focus on the setting of *Contextual Batched Bandits* (CBB) (see Algorithm 1), where the decision process is partitioned into N episodes, and in each episode, CBB consists of two phases: 1) the *policy updating* approximates the optimal policy based on the received contexts and rewards; 2) the *online decision* selects actions for execution following the updated and fixed policy $\boldsymbol{\pi}$ for B steps (also called the *batch size* is B), and stores the context-action pairs and the observed rewards of the executed actions into a data buffer $\mathcal{D}$. The reward R in CBB is a *partial-information feedback* where rewards are unobserved for the non-executed actions.

Different from existing batch bandit setting (Han et al., 2020; Esfandiari et al., 2021), where the true reward feedbacks for all actions are controlled by the same parameter vector while the contexts received differ by actions at each step, we assume that in the CBB setting, the mechanism of true reward feedback differs by actions and the received context is shared by actions. Formally, for any context $\boldsymbol{s}_i \in \mathcal{S} \subseteq \mathbb{R}^d$ and action $A \in \mathcal{A}$, we assume that the expectation of the true reward $R_{i,A}^{\mathrm{true}}$

Algorithm 1 Contextual Batched Bandit (CBB)

INPUT: Batch size B, number of episodes N, action space $\mathcal{A} = \{A_j\}_{j\in[M]}$, context space $\mathcal{S} \subseteq \mathbb{R}^d$
Initialize policy $\boldsymbol{\pi}_0 \leftarrow \mathbf{1}/M$, and sample data buffer $\mathcal{D}_1 = \{(\boldsymbol{s}_{0,b}, A_{I_{0,b}}, R_{0,b})\}_{b\in[B]}$ using $\boldsymbol{\pi}_0$
for $n = 1$ **to** N **do**
  Update the policy $\boldsymbol{\pi}_n$ on $\mathcal{D}_n$ {`Policy Updating`}
  for $b = 1$ **to** B **do**
    Observe context $\boldsymbol{s}_{n,b}$
    Choose $A_{I_{n,b}} \in \mathcal{A}$ following the updated policy $\boldsymbol{\pi}_n(\boldsymbol{s}_{n,b})$ {`Online Decision`}
  end for
  $\mathcal{D}_{n+1} \leftarrow \{(\boldsymbol{s}_{n,b}, A_{I_{n,b}}, R_{n,b}\}_{b\in[B]}$, where $R_{n,b}$ denotes the reward of action $A_{I_{n,b}}$ on context $\boldsymbol{s}_{n,b}$
end for

is determined by an unknown action-specific *reward parameter vector* $\boldsymbol{\theta}^*_A \in \mathbb{R}^d$: $\mathbb{E}\left[R^{\text{true}}_{i,A} \mid \boldsymbol{s}_i\right] = \langle \boldsymbol{\theta}^*_A, \boldsymbol{s}_i \rangle$ (the linear reward will be extended to the nonlinear case in Section 5). This setting for reward feedback matches many real-world applications, e.g., each action corresponds to a different category of candidate coupons in coupon recommendation, and the reward feedback mechanism of each category differs due to the different discount pricing strategies.

Next, we provide deeper understandings of the influence of unobserved feedbacks on the performance of policy updating in the CBB setting. We first conducted an empirical comparison by applying the batch UCB policy (Han et al., 2020) to environments under different proportions of received reward feedbacks. In particular, the agent under full-information feedback can receive all the rewards of the executed and non-executed actions, called *Full-Information CBB* (FI-CBB) setting. From Figure 1, we can observe that the partial-information feedbacks could be damaging in terms of hurting the policy updating, and batched bandit could benefit from more reward feedbacks, where the performance of 80% feedback is very close to that of FI-CBB. Then, we demonstrate the difference of instantaneous regrets between the CBB and FI-CBB settings as in Theorem 1. The detailed description and proof of Theorem 1 can be found in Appendix A.

Theorem 1. *For any action $A \in \mathcal{A}$ and context $\boldsymbol{s}_i \in \mathcal{S}$, let $\boldsymbol{\theta}^n_A$ be the reward parameter vector estimated by the batched UCB policy in the n-th episode. The upper bound of instantaneous regret (defined by $|\langle \boldsymbol{\theta}^n_A, \boldsymbol{s}_i \rangle - \langle \boldsymbol{\theta}^*_A, \boldsymbol{s}_i \rangle|$) in the FI-CBB setting is tighter than that in CBB setting (i.e., using the partial-information feedback).*

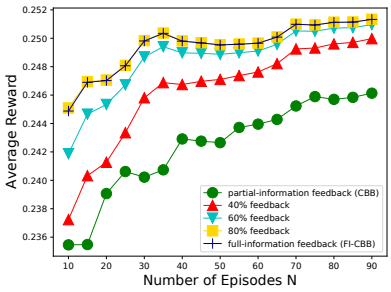


Figure 1: Average rewards of batch UCB policy (Han et al., 2020) under different proportions of received reward feedbacks, interacting with the synthetic environment in Section 6, where $x\%$ feedback means that $x\%$ of actions can receive their true rewards

From Theorem 1, we can conclude that the price paid for having access only to the partial-information feedbacks is the deterioration in the regret bound. Ideally, the policy would be updated using the full-information feedback. In CBB, however, the full-information feedback is unaccessible. Fortunately, in CBB, different reward parameter vectors need to be maintained and estimated separately for each action, and the potential reward structures of the non-executed actions have been somehow captured. So why don't we use these maintained reward parameters to estimate the unknown rewards for the non-executed actions? Next, we propose an efficient reward imputation approach that uses this additional reward structure information for improving the performance of the bandit policy.

3 Efficient Reward Imputation for Policy Updating

In this section, we present an efficient reward imputation approach tailored for policy updating in the CBB setting.

Formulation of Reward Imputation. Since the true reward parameters differ by actions in the CBB setting, we need to maintain a different estimated reward parameter vector for each action. As shown in Figure 2, in contrast to CBB that ignores the contexts and rewards of the non-executed steps of A_j, our reward imputation approach completes the missing values using the imputed contexts and rewards, approximating the full-information CBB setting. Specifically, in the $(n+1)$-th episode, for

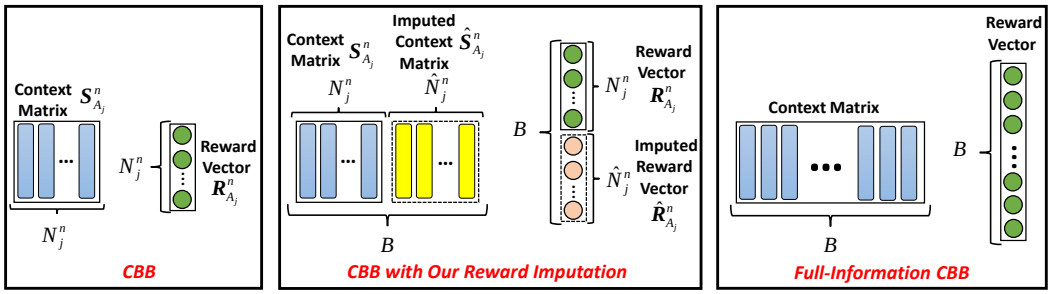


Figure 2: Comparison of the stored data corresponding to the action $A_j \in \mathcal{A}$ in CBB, CBB with our reward imputation, and full-information CBB, in the $(n+1)$-th episode

each action $A_j \in \mathcal{A}, j \in [M]$, we store context vectors and rewards corresponding to the steps in which the action A_j is executed, into a *context matrix* $\boldsymbol{S}^n_{A_j} \in \mathbb{R}^{N^n_j \times d}$ and a *reward vector* $\boldsymbol{R}^n_{A_j} \in \mathbb{R}^{N^n_j}$, respectively, where N^n_j denotes the number of steps (in episode $n+1$) in which the action A_j is executed. Additionally, for $\forall A_j \in \mathcal{A}, j \in [M]$, we store context vectors corresponding to the non-executed steps of action A_j (denote the number of non-executed steps by $\widehat{N}^n_j$, i.e., $\widehat{N}^n_j = B - N^n_j$), into an *imputed context matrix* $\widehat{\boldsymbol{S}}^n_{A_j} \in \mathbb{R}^{\widehat{N}^n_j \times d}$, and compute the *imputed reward vector* as follows:

$$\widehat{\boldsymbol{R}}^n_{A_j} = \{r_{n,1}(A_j), r_{n,2}(A_j), \ldots, r_{n,\widehat{N}^n_j}(A_j)\} \in \mathbb{R}^{\widehat{N}^n_j}, \quad j \in [M],$$

where $r_{n,b}(A_j) := \langle \bar{\boldsymbol{\theta}}^n_{A_j}, \boldsymbol{s}_{n,b} \rangle$ denotes the *imputed reward* for each step $b \in [\widehat{N}^n_j]$, and $\boldsymbol{s}_{n,b}$ is the b-th row of $\widehat{\boldsymbol{S}}^n_{A_j}$. In the $(n+1)$-th episode, the *estimated parameter vector* $\bar{\boldsymbol{\theta}}^{n+1}_{A_j}$ of the imputed reward for action A_j can be updated by solving the following *imputation regularized ridge regression*:

$$\bar{\boldsymbol{\theta}}^{n+1}_{A_j} = \underset{\boldsymbol{\theta} \in \mathbb{R}^d}{\arg\min} \underbrace{\left\| \boldsymbol{L}^n_{A_j} \boldsymbol{\theta} - \boldsymbol{T}^n_{A_j} \right\|_2^2}_{\text{Observed Term}} + \gamma \underbrace{\left\| \widehat{\boldsymbol{L}}^n_{A_j} \boldsymbol{\theta} - \widehat{\boldsymbol{T}}^n_{A_j} \right\|_2^2}_{\text{Imputation Term}} + \lambda \|\boldsymbol{\theta}\|_2^2, \quad n = 0, 1, \ldots, N-1, \tag{1}$$

where $\gamma \in [0,1]$ is the *imputation rate* which controls the degree of reward imputation, $\lambda > 0$ is the regularization parameter, $\widehat{\boldsymbol{T}}^n_{A_j}$ is estimated by $\{\bar{\boldsymbol{\theta}}^k_{A_j}\}_{k=0}^n$ from the previous episodes, and $\boldsymbol{L}^n_{A_j} = [\boldsymbol{S}^0_{A_j}; \boldsymbol{S}^1_{A_j}; \cdots; \boldsymbol{S}^n_{A_j}] \in \mathbb{R}^{L^n_j \times d}$, $\boldsymbol{T}^n_{A_j} = [\boldsymbol{R}^0_{A_j}; \boldsymbol{R}^1_{A_j}; \cdots; \boldsymbol{R}^n_{A_j}] \in \mathbb{R}^{L^n_j}$, $L^n_j = \sum_{k=0}^n N^k_j$, $\widehat{\boldsymbol{L}}^n_{A_j} = [\widehat{\boldsymbol{S}}^0_{A_j}; \widehat{\boldsymbol{S}}^1_{A_j}; \cdots; \widehat{\boldsymbol{S}}^n_{A_j}] \in \mathbb{R}^{\widehat{L}^n_j \times d}$, $\widehat{\boldsymbol{T}}^n_{A_j} = [\widehat{\boldsymbol{R}}^0_{A_j}; \widehat{\boldsymbol{R}}^1_{A_j}; \cdots; \widehat{\boldsymbol{R}}^n_{A_j}] \in \mathbb{R}^{\widehat{L}^n_j}$, $\widehat{L}^n_j = \sum_{k=0}^n \widehat{N}^k_j$, yielding

$$\bar{\boldsymbol{\theta}}^{n+1}_{A_j} = \left(\boldsymbol{\Psi}^{n+1}_{A_j}\right)^{-1} \left(\boldsymbol{b}^{n+1}_{A_j} + \gamma \hat{\boldsymbol{b}}^{n+1}_{A_j}\right), \tag{2}$$

where $\boldsymbol{\Psi}^{n+1}_{A_j} := \lambda \boldsymbol{I}_d + \boldsymbol{\Phi}^{n+1}_{A_j} + \gamma \widehat{\boldsymbol{\Phi}}^{n+1}_{A_j}$,

$$\boldsymbol{\Phi}^{n+1}_{A_j} = \boldsymbol{\Phi}^n_{A_j} + \boldsymbol{S}^{n\intercal}_{A_j} \boldsymbol{S}^n_{A_j}, \qquad \boldsymbol{b}^{n+1}_{A_j} = \boldsymbol{b}^n_{A_j} + \boldsymbol{S}^{n\intercal}_{A_j} \boldsymbol{R}^n_{A_j}, \tag{3}$$

$$\widehat{\boldsymbol{\Phi}}^{n+1}_{A_j} = \eta \widehat{\boldsymbol{\Phi}}^n_{A_j} + \widehat{\boldsymbol{S}}^{n\intercal}_{A_j} \widehat{\boldsymbol{S}}^n_{A_j}, \qquad \hat{\boldsymbol{b}}^{n+1}_{A_j} = \eta \hat{\boldsymbol{b}}^n_{A_j} + \widehat{\boldsymbol{S}}^{n\intercal}_{A_j} \widehat{\boldsymbol{R}}^n_{A_j}, \tag{4}$$

and $\eta \in (0,1)$ is the *discount parameter* which controls how fast the previous imputed rewards are forgotten, and can help guaranteeing the regret bound in Theorem 2. The imputation rate γ measures a trade-off between bias and variance which is analyzed in Remark 1.

Efficient Reward Imputation using Sketching. As shown in the first 4 columns in Table 1, the overall time complexity of the imputation for each action is $O(Bd^2)$ in each episode, where B represents the batch size, and d the dimensionality of the input. Thus, for all the M actions in one episode, reward imputation increases the time complexity from $O(Bd^2)$ of the approach without imputation to $O(MBd^2)$. To address this issue, we design an efficient reward imputation approach using sketching, which reduces the time complexity of each action in one episode from $O\left(Bd^2\right)$ to $O\left(cd^2\right)$, where c denotes the *sketch size* satisfying $d < c < B$ and $cd > B$. Specifically, in the $(n+1)$-th episode, the imputation regularized ridge regression Eq.(1) can be approximated by a *sketched ridge regression* as

$$\tilde{\boldsymbol{\theta}}^{n+1}_{A_j} = \underset{\boldsymbol{\theta} \in \mathbb{R}^d}{\arg\min} \left\| \boldsymbol{\Pi}^n_{A_j} \left(\boldsymbol{L}^n_{A_j} \boldsymbol{\theta} - \boldsymbol{T}^n_{A_j}\right) \right\|_2^2 + \gamma \left\| \widehat{\boldsymbol{\Pi}}^n_{A_j} \left(\widehat{\boldsymbol{L}}^n_{A_j} \boldsymbol{\theta} - \widehat{\boldsymbol{T}}^n_{A_j}\right) \right\|_2^2 + \lambda \|\boldsymbol{\theta}\|_2^2, \tag{5}$$

Table 1: The time complexities of the original reward imputation in Eq.(1) (first 4 columns) and the reward imputation using sketching in Eq.(5) (last 4 columns) for action A_j in the $(n+1)$-th episode, where N_j^n ($\widehat{N}_j^n$) denotes the number of steps in which the action A_j is executed (non-executed) in episode $n+1$, $\widehat{N}_j^n + N_j^n = B$, and the sketch size c satisfying $d < c < B$ and $cd > B$ (MM: matrix multiplication; MI: matrix inversion; Overall: overall time complexity for action A_j in one episode)

Original reward imputation in Eq.(1)				Reward imputation using sketching in Eq.(5)			
Item	Operation	Eq.	Time	Item	Operation	Eq.	Time
$\boldsymbol{\Phi}_{A_j}^{n+1}, \widehat{\boldsymbol{\Phi}}_{A_j}^{n+1}$	MM	(3), (4)	$O(Bd^2)$	$\boldsymbol{G}_{A_j}^{n+1}, \widehat{\boldsymbol{G}}_{A_j}^{n+1}$	MM	(7), (8)	$O(cd^2)$
$\boldsymbol{b}_{A_j}^{n+1}, \hat{\boldsymbol{b}}_{A_j}^{n+1}$	MM	(3), (4)	$O(Bd)$	$\boldsymbol{p}_{A_j}^{n+1}, \hat{\boldsymbol{p}}_{A_j}^{n+1}$	MM	(7), (8)	$O(cd)$
$(\boldsymbol{\Psi}_{A_j}^{n+1})^{-1}$	MI	(2)	$O(d^3)$	$(\boldsymbol{W}_{A_j}^{n+1})^{-1}$	MI	(6)	$O(d^3)$
	–			$\boldsymbol{\Gamma}_{A_j}^n, \boldsymbol{\Lambda}_{A_j}^n$	Sketching	–	$O(N_j^n d)$
	–			$\widehat{\boldsymbol{\Gamma}}_{A_j}^n, \widehat{\boldsymbol{\Lambda}}_{A_j}^n$	Sketching	–	$O(\widehat{N}_j^n d)$
Overall	–	–	$O(Bd^2)$	Overall	–	–	$O(cd^2)$

where $\tilde{\boldsymbol{\theta}}_A^{n+1}$ denotes the estimated parameter vector of the imputed reward using sketching for action $A \in \mathcal{A}$, $\boldsymbol{C}_{A_j}^n \in \mathbb{R}^{c \times N_j^n}$ and $\widehat{\boldsymbol{C}}_{A_j}^n \in \mathbb{R}^{c \times \widehat{N}_j^n}$ are the *sketch submatrices* for the observed term and the imputation term, respectively, and the *sketch matrices* for the two terms can be represented as

$$\boldsymbol{\Pi}_{A_j}^n = \left[\boldsymbol{C}_{A_j}^0, \boldsymbol{C}_{A_j}^1, \cdots, \boldsymbol{C}_{A_j}^n\right] \in \mathbb{R}^{c \times L_j^n}, \quad \widehat{\boldsymbol{\Pi}}_{A_j}^n = \left[\widehat{\boldsymbol{C}}_{A_j}^0, \widehat{\boldsymbol{C}}_{A_j}^1, \cdots, \widehat{\boldsymbol{C}}_{A_j}^n\right] \in \mathbb{R}^{c \times \widehat{L}_j^n}.$$

We denote the sketches of the context matrix and the reward vector by $\boldsymbol{\Gamma}_{A_j}^n := \boldsymbol{C}_{A_j}^n \boldsymbol{S}_{A_j}^n \in \mathbb{R}^{c \times d}$ and $\boldsymbol{\Lambda}_{A_j}^n := \boldsymbol{C}_{A_j}^n \boldsymbol{R}_{A_j}^n \in \mathbb{R}^c$, the sketches of the imputed context matrix and the imputed reward vector by $\widehat{\boldsymbol{\Gamma}}_{A_j}^n := \widehat{\boldsymbol{C}}_{A_j}^n \widehat{\boldsymbol{S}}_{A_j}^n \in \mathbb{R}^{c \times d}$ and $\widehat{\boldsymbol{\Lambda}}_{A_j}^n := \widehat{\boldsymbol{C}}_{A_j}^n \widehat{\boldsymbol{R}}_{A_j}^n \in \mathbb{R}^c$, and obtain the solution of Eq.(5):

$$\tilde{\boldsymbol{\theta}}_{A_j}^{n+1} = \left(\boldsymbol{W}_{A_j}^{n+1}\right)^{-1} \left(\boldsymbol{p}_{A_j}^{n+1} + \gamma \hat{\boldsymbol{p}}_{A_j}^{n+1}\right), \tag{6}$$

where $\eta \in (0, 1)$ denotes the discount parameter, $\boldsymbol{W}_{A_j}^{n+1} := \lambda \boldsymbol{I}_d + \boldsymbol{G}_{A_j}^{n+1} + \gamma \widehat{\boldsymbol{G}}_{A_j}^{n+1}$, and

$$\boldsymbol{G}_{A_j}^{n+1} = \boldsymbol{G}_{A_j}^n + \boldsymbol{\Gamma}_{A_j}^{n\intercal} \boldsymbol{\Gamma}_{A_j}^n, \qquad \boldsymbol{p}_{A_j}^{n+1} = \boldsymbol{p}_{A_j}^n + \boldsymbol{\Gamma}_{A_j}^{n\intercal} \boldsymbol{\Lambda}_{A_j}^n, \tag{7}$$

$$\widehat{\boldsymbol{G}}_{A_j}^{n+1} = \eta \widehat{\boldsymbol{G}}_{A_j}^n + \widehat{\boldsymbol{\Gamma}}_{A_j}^{n\intercal} \widehat{\boldsymbol{\Gamma}}_{A_j}^n, \quad \hat{\boldsymbol{p}}_{A_j}^{n+1} = \eta \hat{\boldsymbol{p}}_{A_j}^n + \widehat{\boldsymbol{\Gamma}}_{A_j}^{n\intercal} \widehat{\boldsymbol{\Lambda}}_{A_j}^n. \tag{8}$$

Using the parameter $\tilde{\boldsymbol{\theta}}_{A_j}^{n+1}$, we obtain the *sketched version of imputed reward* as $\tilde{r}_{n,b}(A_j) := \langle \tilde{\boldsymbol{\theta}}_{A_j}^n, \boldsymbol{s}_{n,b} \rangle$ at step $b \in [\widehat{N}_j^n]$. Finally, we specify that the sketch submatrices $\{\boldsymbol{C}_A^n\}_{A \in \mathcal{A}, n \in [N]}$ and $\{\widehat{\boldsymbol{C}}_{A_j}^n\}_{A \in \mathcal{A}, n \in [N]}$ are the block construction of Sparser Johnson-Lindenstrauss Transform (SJLT) (Kane & Nelson, 2014), where the sketch size c is divisible by the number of blocks D[1]. As shown in the last 4 columns in Table 1, sketching reduces the time complexity from $O(MBd^2)$ to $O(Mcd^2)$ for reward imputation of all M actions in one episode, where $c < B$. When $Mc \approx B$, the overall time complexity of our reward imputation using sketching is even comparable to that without reward imputation which has a $O(Bd^2)$ time complexity.

Updated Policy using Imputed Rewards. Inspired by the UCB strategy (Li et al., 2010), the updated policy for online decision of the $(n+1)$-th episode can be formulated using the imputed rewards (parameterized by $\bar{\boldsymbol{\theta}}_A^{n+1}$ in Eq.(2)) or the sketched version of imputed rewards (parameterized by $\tilde{\boldsymbol{\theta}}_A^{n+1}$ in Eq.(6)). Specifically, for a new context $\boldsymbol{s}$,
origin policy $\bar{\boldsymbol{\pi}}_{n+1}$ selects the action following $A \leftarrow \arg\max_{A \in \mathcal{A}} \langle \bar{\boldsymbol{\theta}}_A^{n+1}, \boldsymbol{s} \rangle + \omega [\boldsymbol{s}^\intercal (\boldsymbol{\Psi}_A^{n+1})^{-1} \boldsymbol{s}]^{\frac{1}{2}}$,
sketched policy $\tilde{\boldsymbol{\pi}}_{n+1}$ selects the action following $A \leftarrow \arg\max_{A \in \mathcal{A}} \langle \tilde{\boldsymbol{\theta}}_A^{n+1}, \boldsymbol{s} \rangle + \alpha [\boldsymbol{s}^\intercal (\boldsymbol{W}_A^{n+1})^{-1} \boldsymbol{s}]^{\frac{1}{2}}$,
where $\omega \geq 0$ and $\alpha \geq 0$ are the regularization parameters in policy and their theoretical values are given in Theorem 4. We summarize the reward imputation using sketching and the sketched policy into Algorithm 2, called SPUIR. Similarly, we call the updating of the original policy that uses reward imputation without sketching, the Policy Updating with Imputed Rewards (PUIR).

[1]Since we set the number of blocks of SJLT as $D < d$, we omit D in the complexity analysis.

Algorithm 2 Sketched Policy Updating with Imputed Rewards (SPUIR) in the $(n+1)$-th episode

INPUT: Policy $\tilde{\boldsymbol{\pi}}_n$, $\mathcal{D}_{n+1}$, $\mathcal{A} = \{A_j\}_{j\in[M]}$, $\alpha \geq 0, \eta \in (0,1)$, $\gamma \in [0,1]$, $\lambda > 0$, $\boldsymbol{W}_{A_j}^0 = \lambda \boldsymbol{I}_d$, $\boldsymbol{G}_{A_j}^0 = \widehat{\boldsymbol{G}}_{A_j}^0 = \boldsymbol{O}_d$, $\boldsymbol{p}_{A_j}^0 = \hat{\boldsymbol{p}}_{A_j}^0 = \boldsymbol{0}$, $\tilde{\boldsymbol{\theta}}_{A_j}^0 = \boldsymbol{0}$, $j \in [M]$, batch size B, sketch size c, number of block D

OUTPUT: Updated policy $\tilde{\boldsymbol{\pi}}_{n+1}$

1: For all $j \in [M]$, store context vectors and rewards corresponding to the steps in which the action A_j is executed, into $\boldsymbol{\Gamma}_{A_j}^n \in \mathbb{R}^{N_j^n \times d}$ and $\boldsymbol{\Lambda}_{A_j}^n \in \mathbb{R}^{N_j^n}$

2: For all $j \in [M]$, store context vectors corresponding to the steps in which the action A_j is not executed into $\widehat{\boldsymbol{\Gamma}}_{A_j}^n \in \mathbb{R}^{\widehat{N}_j^n \times d}$, where $\widehat{N}_j^n \leftarrow B - N_j^n$

3: $\tilde{r}_{n,b}(A_j) \leftarrow \langle \tilde{\boldsymbol{\theta}}_{A_j}^n, \boldsymbol{s}_{n,b} \rangle$, for all $A_j \in \mathcal{A}$ and $b \in [\widehat{N}_j^n]$, where $\boldsymbol{s}_{n,b}$ is the b-th row of $\widehat{\boldsymbol{\Gamma}}_{A_j}^n$

4: Compute imputed reward vector $\widehat{\boldsymbol{R}}_{A_j}^n \leftarrow \{\tilde{r}_{n,1}(A_j), \ldots, \tilde{r}_{n,\widehat{N}_j^n}(A_j)\} \in \mathbb{R}^{\widehat{N}_j^n}$ for $\forall j \in [M]$

5: **for all** action $A_j \in \mathcal{A}$ **do**

6: $\quad \boldsymbol{G}_{A_j}^{n+1} \leftarrow \boldsymbol{G}_{A_j}^n + \boldsymbol{\Gamma}_{A_j}^{n\intercal} \boldsymbol{\Gamma}_{A_j}^n, \quad \boldsymbol{p}_{A_j}^{n+1} \leftarrow \boldsymbol{p}_{A_j}^n + \boldsymbol{\Gamma}_{A_j}^{n\intercal} \boldsymbol{\Lambda}_{A_j}^n$ {Eq.(7)}

7: $\quad \widehat{\boldsymbol{G}}_{A_j}^{n+1} \leftarrow \eta \widehat{\boldsymbol{G}}_{A_j}^n + \widehat{\boldsymbol{\Gamma}}_{A_j}^{n\intercal} \widehat{\boldsymbol{\Gamma}}_{A_j}^n, \quad \hat{\boldsymbol{p}}_{A_j}^{n+1} \leftarrow \eta \hat{\boldsymbol{p}}_{A_j}^n + \widehat{\boldsymbol{\Gamma}}_{A_j}^{n\intercal} \widehat{\boldsymbol{\Lambda}}_{A_j}^n$ {Eq.(8)}

8: $\quad \boldsymbol{W}_{A_j}^{n+1} \leftarrow \lambda \boldsymbol{I}_d + \boldsymbol{G}_{A_j}^{n+1} + \gamma \widehat{\boldsymbol{G}}_{A_j}^{n+1}, \quad \tilde{\boldsymbol{\theta}}_{A_j}^{n+1} \leftarrow (\boldsymbol{W}_{A_j}^{n+1})^{-1} (\boldsymbol{p}_{A_j}^{n+1} + \gamma \hat{\boldsymbol{p}}_{A_j}^{n+1})$ {Eq.(6)}

9: **end for**

10: $\tilde{\boldsymbol{\pi}}_{n+1}(\boldsymbol{s})$ selects action $A \leftarrow \arg\max_{A\in\mathcal{A}} \langle \tilde{\boldsymbol{\theta}}_A^{n+1}, \boldsymbol{s} \rangle + \alpha [\boldsymbol{s}^\intercal \left(\boldsymbol{W}_A^{n+1}\right)^{-1} \boldsymbol{s}]^{\frac{1}{2}}$ for a new context $\boldsymbol{s}$

11: **return** $\{\tilde{\boldsymbol{\theta}}_A^{n+1}\}_{A\in\mathcal{A}}$, $\{\left(\boldsymbol{W}_A^{n+1}\right)^{-1}\}_{A\in\mathcal{A}}$

4 Theoretical Analysis

We provide the instantaneous regret bound, prove the approximation error of sketching, and analyze the regret in the CBB setting. The detailed proofs can be found in Appendix B. We first demonstrate the instantaneous regret bound of the original solution $\bar{\boldsymbol{\theta}}_A^n$ in Eq.(1).

Theorem 2 (Instantaneous Regret Bound)**.** *Let $\eta \in (0,1)$ be the discount parameter, $\gamma \in [0,1]$ the imputation rate. In the n-th episode, if the rewards $\{R_{n,b}\}_{b\in[B]}$ are independent*[2] *and bounded by C_R, then, for $\forall b \in [B]$ and $\forall A \in \mathcal{A}$, with probability at least $1-\delta$,*

$$\left|\left\langle \bar{\boldsymbol{\theta}}_A^n, \boldsymbol{s}_{n,b} \right\rangle - \langle \boldsymbol{\theta}_A^*, \boldsymbol{s}_{n,b} \rangle\right| \leq \left[\lambda \|\boldsymbol{\theta}_A^*\|_2 + \nu + C_{\boldsymbol{\theta}} [\gamma \eta^N / (1-\eta)]^{\frac{1}{2}}\right] [\boldsymbol{s}_{n,b}^\intercal \left(\boldsymbol{\Psi}_A^n\right)^{-1} \boldsymbol{s}_{n,b}]^{\frac{1}{2}}, \tag{9}$$

where $\boldsymbol{\Psi}_A^n = \lambda \boldsymbol{I}_d + \boldsymbol{\Phi}_A^n + \gamma \widehat{\boldsymbol{\Phi}}_A^n$, $\nu = [2C_R^2 \log(2MB/\delta)]^{\frac{1}{2}}$, and $C_{\boldsymbol{\theta}} = [2\|\bar{\boldsymbol{\theta}}_A^n\|_2^2 + 2\|\boldsymbol{\theta}_A^*\|_2^2]^{\frac{1}{2}}$. The first term on the right-hand side of Eq.(9) can be seen as the bias term for the reward imputation, while the second term is the variance term. The variance term of our algorithm is not larger than that without the reward imputation, i.e, for $\forall \boldsymbol{s} \in \mathbb{R}^d$, $[\boldsymbol{s}^\intercal \left(\boldsymbol{\Psi}_A^n\right)^{-1} \boldsymbol{s}]^{\frac{1}{2}} \leq [\boldsymbol{s}^\intercal \left(\lambda \boldsymbol{I}_d + \boldsymbol{\Phi}_A^n\right)^{-1} \boldsymbol{s}]^{\frac{1}{2}}$. Further, a larger imputation rate γ leads to a smaller variance term $[\boldsymbol{s}^\intercal \left(\boldsymbol{\Psi}_A^n\right)^{-1} \boldsymbol{s}]^{\frac{1}{2}}$.

Remark 1. *From Theorem 2, we can observe that the proposed reward imputation approach achieves a smaller variance term (the second term on the right-hand side of Eq.(9)) than that without the reward imputation, while incurring a bias term $C_{\boldsymbol{\theta}} \sqrt{\gamma \eta^N / (1-\eta)}$ in addition to the two bias terms $\lambda \|\boldsymbol{\theta}_A^*\|_2 + \nu$ that exist in every UCB-based policy. But this additional bias term decreases exponentially with the number of episodes N, which is close to 0 when N is large. Overall, CBB integrated with our reward imputation approach enjoys a lower instantaneous regret bound after a few number of episodes. Besides, the imputation rate γ controls a trade-off between the bias term and the variance term, and we will design a rate-scheduled approach for choosing the imputation rate γ in Section 5,.*

Although some approximation error bounds using SJLT have been proposed (Nelson & Nguyên, 2013; Kane & Nelson, 2014; Zhang & Liao, 2019), it is still unknown what is the lower bound of the sketch size while applying SJLT to the sketched ridge regression problem in our SPUIR. Next, we prove the approximation error as well as the lower bound of the sketch size for the sketched ridge regression problem. For convenience, we drop all the superscripts and subscripts in this result.

Theorem 3 (Approximation Error Bound of Imputation using Sketching)**.** *Denote the imputation regularized ridge regression function by $F(\boldsymbol{\theta})$ (defined in Eq.(1)) and the sketched ridge regression*

[2]The assumption about conditional independence of the rewards is commonly used in the bandits literature, which can be ensured using a master technology as a theoretical construction (Auer, 2002; Chu et al., 2011).

function by $F^{\mathrm{S}}(\boldsymbol{\theta})$ (defined in Eq.(5)) for reward imputation, whose solutions (i.e., the estimated reward parameter vectors) are $\bar{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^d} F(\boldsymbol{\theta})$ and $\tilde{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^d} F^{\mathrm{S}}(\boldsymbol{\theta})$. Let $\gamma \in [0,1]$ be the imputation rate, $\lambda > 0$ the regularization parameter, $\delta \in (0, 0.1]$, $\varepsilon \in (0,1)$, $\boldsymbol{L}_{\mathrm{all}} = [\boldsymbol{L}; \sqrt{\gamma}\widehat{\boldsymbol{L}}]$, and $\rho_\lambda = \|\boldsymbol{L}_{\mathrm{all}}\|_2^2/(\|\boldsymbol{L}_{\mathrm{all}}\|_2^2+\lambda)$. If $\boldsymbol{\Pi}$ and $\widehat{\boldsymbol{\Pi}}$ are SJLT, assuming that $D = \Theta(\varepsilon^{-1}\log^3(d\delta^{-1}))$ and the sketch size $c = \Omega\left(d \operatorname{polylog}\left(d\delta^{-1}\right)/\varepsilon^2\right)$, with probability at least $1-\delta$, $F(\tilde{\boldsymbol{\theta}}) \leq (1+\rho_\lambda\varepsilon)F(\bar{\boldsymbol{\theta}})$ and $\|\tilde{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}}\|_2 = O\left(\sqrt{\rho_\lambda\varepsilon}\right)$ hold.

To measure the convergence of approximating the optimal policy in sequential decision-making, we define the *regret* of SPUIR against the optimal policy as follows:

$$\mathrm{Reg}\left(\{A_{I_{n,b}}\}_{n\in[N],b\in[B]}\right) := \max_{A\in\mathcal{A}} \sum_{n\in[N],b\in[B]} [\langle \boldsymbol{\theta}^*_A, \boldsymbol{s}_{n,b}\rangle - \langle \boldsymbol{\theta}^*_{A_{I_{n,b}}}, \boldsymbol{s}_{n,b}\rangle],$$

where $I_{n,b}$ denotes the index of the executed action using the sketched policy $\tilde{\boldsymbol{\pi}}_n$ (parameterized by $\{\tilde{\boldsymbol{\theta}}^n_A\}_{A\in\mathcal{A}}$) at step b in the n-th episode. Finally, we demonstrate the regret bound of SPUIR.

Theorem 4 (Regret Bound of SPUIR)**.** *Let $T = BN$ be the overall number of steps, $\eta \in (0,1)$ be the discount parameter, $\gamma \in [0,1]$ the imputation rate, $\lambda > 0$ the regularization parameter, $C^{\max}_{\boldsymbol{\theta}^*} = \max_{A\in\mathcal{A}} \|\boldsymbol{\theta}^*_A\|_2$, $C^{\max}_{\bar{\boldsymbol{\theta}}} = \max_{A\in\mathcal{A},n\in[N]} \|\bar{\boldsymbol{\theta}}^n_A\|_2$, $C^{\max}_{\boldsymbol{\theta}} = [2(C^{\max}_{\boldsymbol{\theta}^*})^2 + 2(C^{\max}_{\bar{\boldsymbol{\theta}}})^2]^{\frac{1}{2}}$. Assume that the conditional independence assumption in Theorem 2 holds and the upper bound of rewards is C_R, $M = O(\mathrm{poly}(d))$, $T \geq d^2$, $\nu = [2C_R^2\log(2MB/\delta_1)]^{\frac{1}{2}}$ with $\delta_1 \in (0,1)$,*

$$\omega = \lambda C^{\max}_{\boldsymbol{\theta}^*} + \nu + C^{\max}_{\boldsymbol{\theta}}[\gamma\eta^N/(1-\eta)]^{\frac{1}{2}}, \quad \alpha = \omega C_\alpha,$$

where $C_\alpha > 0$ which decreases with increase of $1/\varepsilon$ and $\varepsilon \in (0,1)$. Let $\delta_2 \in (0, 0.1]$, $\rho_\lambda < 1$ be the constant defined in Theorem 3, and C_{reg} be a positive constant that decreases with increase of $1/\varepsilon$. For the sketch matrices $\{\boldsymbol{\Pi}^n_A\}_{A\in\mathcal{A},n\in[N]}$ and $\{\widehat{\boldsymbol{\Pi}}\}_{A\in\mathcal{A},n\in[N]}$, assuming that the number of blocks in SJLT $D = \Theta(\varepsilon^{-1}\log^3(d\delta_2^{-1}))$, and the sketch size satisfying $c = \Omega\left(d \operatorname{polylog}\left(d\delta_2^{-1}\right)/\varepsilon^2\right)$, then, for an arbitrary sequence of contexts $\{\boldsymbol{s}_{n,b}\}_{n\in[N],b\in[B]}$, with probability at least $1 - N(\delta_1+\delta_2)$,

$$\mathrm{Reg}(\{A_{I_{n,b}}\}_{n\in[N],b\in[B]}) \leq 2\alpha C_{\mathrm{reg}}\sqrt{10M}\log(T+1)(\sqrt{dT}+dB) + O\left(T\sqrt{\rho_\lambda\epsilon d}/B\right). \quad (10)$$

Remark 2. *Setting $B = O(\sqrt{T/d})$ and $\rho_\lambda\epsilon = 1/d$ yields a sublinear upper bound of regret of order $\widetilde{O}(\sqrt{MdT})$[3] provided that the sketch size $c = \Omega(\rho_\lambda^2 d^3 \operatorname{polylog}(d\delta_2^{-1}))$. We can observe that the lower bound of c is independent of the overall number of steps T, and a theoretical value of the batch size is $B = C_B\sqrt{T/d} = C_B^2 N/d$, where setting the constant $C_B \approx 25$ is a suitable choice that has been verified in the experiments in Section 6. In particular, when $\rho_\lambda = O(1/d)$, the sketch size of order $c = \Omega(d \operatorname{polylog} d)$ is sufficient to achieve a sublinear regret, which has been demonstrated in our experimental results. Since the lower bound of regret for contextual batched bandit in (Han et al., 2020) assumes that there are only two actions and both the actions share the same true reward model, it can not be applied to our CBB setting where each action corresponds to a different reward model. Despite the lack of the lower bound in our CBB setting, from the theoretical results of regret, we can observe that our SPUIR admits several advantages: (a) the order of our regret bound is not higher than those in the literature of contextual bandits in the fully-online setting (i.e., $B = 1$) (Li et al., 2019; Dimakopoulou et al., 2019), which is a more simple setting than ours; (b) The first term on the right-hand side of the regret bound Eq.(10) measures the performance of policy updating using imputed rewards (called "policy error"). From Theorem 2 and Remark 1, we obtain that, in each episode, our policy updating has a smaller variance than the policy without the reward imputation, and incurs an exponentially-decaying additional bias, leading to a tighter regret (i.e., smaller policy error) after a few number of episodes. (c) The second term on the right-hand side of Eq.(10) is of order $O(T\sqrt{\rho_\lambda\epsilon d}/B)$, which is incurred by the sketching approximation using SJLT (called "sketching error"). This sketching error does not have any influence on the order of regret of SPUIR, which may even have a lower order with a suitable choice of $\rho_\lambda\varepsilon$, e.g., setting $\rho_\lambda\varepsilon = T^{-1/4}d^{-1}$ yields a sketching error of order $O(T^{3/8}d^{1/2})$ provided that $c = \Omega(\rho_\lambda^2 d^3 \operatorname{polylog}(d\delta_2^{-1})\sqrt{T})$.*

5 Extensions of Our Approach

To make the proposed reward imputation approach more feasible and practical, we tackle the following two research questions by designing variants of our approach following the theoretical results:

[3] We use the notation of $\widetilde{O}$ to suppress logarithmic factors in the overall number of steps T.

RQ1 (Parameter Selection): *Can we set the imputation rate γ without tuning?*
RQ2 (Nonlinear Reward): *Can we apply the proposed reward imputation approach to the case where the expectation of true rewards is nonlinear?*
Rate-Scheduled Approach. For RQ1, we equip PUIR and SPUIR with a rate-scheduled approach, called PUIR-RS and SPUIR-RS, respectively. From Remark 1, a larger imputation rate γ leads to a smaller variance while increasing the bias, while this bias term decreases exponentially with the number of episodes N. Therefore, instead of using a fixed imputation rate, we can gradually increase γ with the number of episodes, avoiding the large bias at the beginning of reward imputation. Specifically, we set $\gamma = X\%$ for episodes from $(X-10)\% \times N$ to $X\% \times N$, where $X \in [10, 100]$.
Application to Nonlinear Rewards. For RQ2, we provide nonlinear versions of reward imputation. We use linearization technologies of nonlinear rewards, rather than directly setting the rewards as nonlinear functions (Valko et al., 2013; Chatterji et al., 2019), avoiding the linear regret or curse of kernelization. Specifically, instead of using the linear imputed reward $\tilde{r}_{n,b}(A_j) := \langle \tilde{\boldsymbol{\theta}}^n_{A_j}, \boldsymbol{s}_{n,b} \rangle$, we use the following linearized nonlinear imputed rewards, denotes by $\mathcal{T}_{n,b}(\boldsymbol{\theta}, A)$:
1) SPUIR-Exp. We assume that the expected reward is an exponential function as $G_{\mathrm{E}}(\boldsymbol{\theta}, \boldsymbol{s}) = \exp\left(\boldsymbol{\theta}^\intercal \boldsymbol{s}\right)$. Then $\mathcal{T}_{n,b}(\boldsymbol{\theta}, A) = \langle \boldsymbol{\theta}, \nabla_{\boldsymbol{\theta}} G_{\mathrm{E}}(\boldsymbol{\theta}, \boldsymbol{s}_{n,b}) \rangle$, where $\nabla_{\boldsymbol{\theta}} G_{\mathrm{E}}(\boldsymbol{\theta}, \boldsymbol{s}_{n,b}) = \exp\left(\boldsymbol{\theta}^\intercal \boldsymbol{s}_{n,b}\right) \boldsymbol{s}_{n,b}$.
2) SPUIR-Poly. When the expected reward is a polynomial function as $G_{\mathrm{P}}(\boldsymbol{\theta}, \boldsymbol{s}) = \left(\boldsymbol{\theta}^\intercal \boldsymbol{s}\right)^2$. Then $\mathcal{T}_{n,b}(\boldsymbol{\theta}, A) = \langle \boldsymbol{\theta}, \nabla_{\boldsymbol{\theta}} G_{\mathrm{P}}(\boldsymbol{\theta}, \boldsymbol{s}_{n,b}) \rangle$, where $\nabla_{\boldsymbol{\theta}} G_{\mathrm{P}}(\boldsymbol{\theta}, \boldsymbol{s}_{n,b}) = 2\left(\boldsymbol{\theta}^\intercal \boldsymbol{s}_{n,b}\right) \boldsymbol{s}_{n,b}$.
3) SPUIR-Kernel. Consider that the underlying expected reward in a Gaussian reproducing kernel Hilbert space (RKHS). We use $\mathcal{T}_{n,b}(\boldsymbol{\theta}, A) = \langle \boldsymbol{\theta}, \phi(\boldsymbol{s}_{n,b}) \rangle$ in random feature space, where the random feature mapping ϕ can be explicitly defined as in (Rahimi & Recht, 2007).
For SPUIR-Exp and SPUIR-Poly, combining the linearization of convex functions (Shalev-Shwartz, 2011) with Theorem 4 yields the regret bounds of the same order. For SPUIR-Kernel, using the approximation error of random features (Rahimi & Recht, 2008), we can obtain a regret bound with an additional error of order $O(B/\sqrt{d_{\mathrm{r}}})$, where d_{r} is the dimension of the random features.

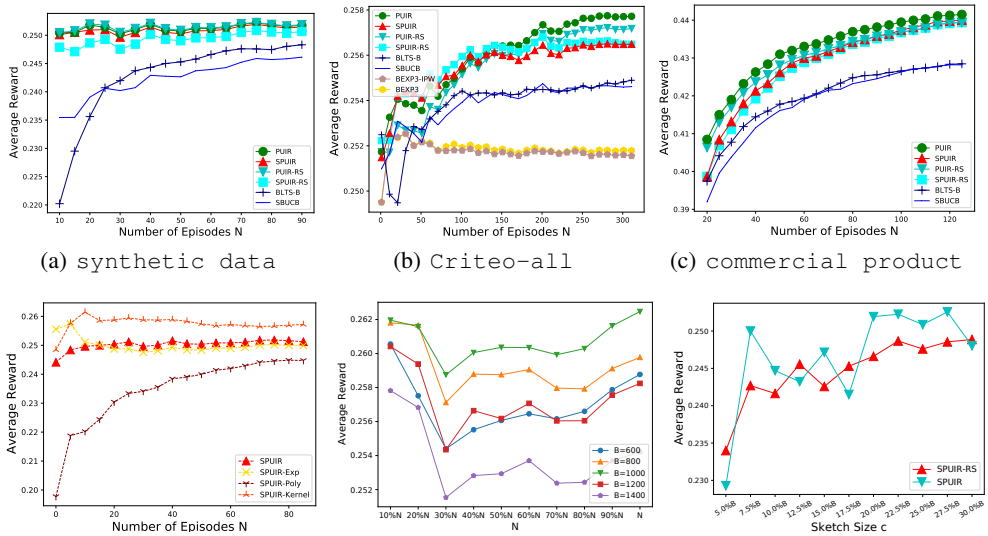
6 EXPERIMENTS

We empirically evaluated the performance of our algorithms on 3 datasets: the synthetic dataset, publicly available Criteo dataset[4] (`Criteo-recent`, `Criteo-all`), and dataset collected from a real commercial app for coupon recommendation (`commercial product`).
Experimental Settings. We compared our algorithms with: Sequential Batch UCB (SBUCB) (Han et al., 2020), Batched linear EXP3 (BEXP3) (Neu & Olkhovskaya, 2020), Batched linear EXP3 using Inverse Propensity Weighting (BEXP3-IPW) (Bistritz et al., 2019), Batched Balanced Linear Thompson Sampling (BLTS-B) (Dimakopoulou et al., 2019), and Sequential version of Delayed Feedback Model (DFM-S) (Chapelle, 2014). We implemented all the algorithms on Intel(R) Xeon(R) Silver 4114 CPU@2.20GHz, and repeated the experiments 20 times. We tested the performance of algorithms in streaming recommendation scenarios, where the reward is represented by a linear combination of the click and conversion behaviors of users. According to Remark 2, we set the batch size as $B = C_B^2 N/d$, the constant $C_B \approx 25$, and the sketch size $c = 150$ on all the datasets. The average reward was used to evaluate the accuracy of algorithms.
Performance Evaluation. Figure 3(a)–(c) reports the average reward of SPUIR with its variants and the baselines. We observed that SPUIR and its variants achieved higher average rewards, demonstrating the effectiveness of our reward imputation. Moreover, SPUIR and its rate-scheduled version SPUIR-RS had similar performances compared with the origin PUIR, which indicates the practical effectiveness of our variants and verifies the correctness of the theoretical analyses. The results on `commercial product` in Table 2 indicate that SPUIR outperformed the second-best baseline with the improvements of 1.07% CVR (conversion rate) and 1.12% CTCVR (post-view click-through&conversion rate). Besides, our reward imputation approaches were more efficient than DFM-S, BLTS-B. The variants using sketching of our algorithms (SPUIR, SPUIR-RS) significantly reduced the time costs of reward imputation, and took less than twice as long to run compared to the baselines without reward imputation (SBUCB, BEXP3, BEXP3-IPW). Figure 3(d) illustrates performances of SPUIR and its nonlinear variants, where SPUIR-Kernel achieved the highest rewards indicating the effectiveness of the nonlinear generalization of our approach. For different decision tasks, a suitable nonlinear reward model needs to be selected for better performances.
Parameter Influence. From the regret bound Eq.(10), we can observe that a larger batch size B results in a larger first term (of order $O(B)$, called policy error) but a smaller second term (of order

[4]https://labs.criteo.com/2013/12/conversion-logs-dataset/

Table 2: Performance comparison of coupon recommendation on `commercial product`

Algorithm	CVR (mean $\pm$ std)	CTCVR (mean $\pm$ std)	Time (sec., mean $\pm$ std)
DFM-S	0.8656 ± 0.0473	0.3317 ± 0.0218	302.3140 ± 8.3045
SBUCB	0.8569 ± 0.0037	0.4277 ± 0.0084	43.5435 ± 0.3659
BEXP3	0.4846 ± 0.0205	0.2425 ± 0.0116	53.5001 ± 0.9220
BEXP3-IPW	0.4862 ± 0.0187	0.2436 ± 0.0113	56.0101 ± 1.4142
BLTS-B	0.8663 ± 0.0178	0.4285 ± 0.0157	218.2109 ± 1.8198
PUIR	0.8807 ± 0.0053	0.4411 ± 0.0029	184.3575 ± 2.2346
SPUIR	0.8770 ± 0.0059	0.4397 ± 0.0032	81.5753 ± 1.5879
PUIR-RS	0.8763 ± 0.0056	0.4389 ± 0.0030	180.4999 ± 1.7763
SPUIR-RS	0.8758 ± 0.0058	0.4391 ± 0.0031	80.8003 ± 2.9030



(a) `synthetic data` (b) `Criteo-all` (c) `commercial product`

(d) nonlinear reward (e) batch size B (f) sketch size c

Figure 3: (a), (b), (c): Average rewards of the compared algorithms, the proposed SPUIR and its variants on synthetic dataset, Criteo dataset, and the real commercial product data, where we omitted the curves of algorithms whose average rewards are 5% lower than the highest reward; (d): SPUIR and its three nonlinear variants on synthetic dataset; (e): SPUIR with different batch sizes on `Criteo-recent`; (f): SPUIR and SPUIR-RS with different sketch sizes on synthetic dataset

$O(1/B)$, called sketching error), indicating that a suitable batch size B needs to be set. This conclusion was empirically verified in Figure 3(e), where $B = 1,000$ ($C_B = 25$) yields better empirical performance in terms of the average reward. Similar phenomenon can also be observed on Criteo dataset and `commercial product`. All of the results verified the theoretical results in Remark 2: $B = C_B\sqrt{T/d} = C_B^2 N/d$ is a suitable choice while setting $C_B \approx 25$. From the results in Figure 3(f) we observe that, for our SPUIR and SPUIR-RS, the performances significantly increased when the sketch size c reached $10\%B$ ($\approx d \log d$), which demonstrates the conclusion in Remark 2 that only the sketch size of order $c = \Omega(d \operatorname{polylog} d)$ is needed for satisfactory performance.

7 Conclusion

Partial-information feedback is ubiquitous in real-world applications, where reward feedback is usually underutilized for learning. This paper proposes a theoretically sound and computationally efficient reward imputation approach for contextual batched bandits, which mimics the reward generation mechanism of the environment approximating the setting of full-information feedback. The proposed reward imputation approach reduces the time complexity of imputation on large batches of data using sketching, achieves a relative-error bound for sketching approximation, has an instantaneous regret with an exponentially-decaying bias and a smaller variance, and enjoys a sublinear regret bound against the optimal policy. The theoretical formulation and algorithmic implementation may provide an efficient reward imputation scheme for online learning under limited feedback.

ETHICS STATEMENT

To verify the effectiveness and efficiency of our algorithms on real products, we conducted experiments on a real dataset collected from a commercial social app. We call this dataset `commercial product`, where the data were collected after the users gave consent, and did not contain any personally identifiable information or offensive content.

REPRODUCIBILITY STATEMENT

The source code of the proposed algorithms is submitted as supplementary materials. For theoretical results, clear explanations of any assumptions and a complete proof of the claims are included as an appendix. The detailed experimental settings are also provided in the appendix. Since the dataset `commercial product` is non-public, we did not provide a URL. We will release this non-public dataset after the publication of this paper, and the link to download this non-public dataset is to be included in the final paper.

REFERENCES

Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3:397–422, 2002.

Ilai Bistritz, Zhengyuan Zhou, Xi Chen, Nicholas Bambos, and Jose Blanchet. Online EXP3 learning in adversarial bandits with delayed feedback. In *Advances in Neural Information Processing Systems 32*, pp. 11349–11358, 2019.

Jean Bourgain, Sjoerd Dirksen, and Jelani Nelson. Toward a unified theory of sparse dimensionality reduction in Euclidean space. In *Proceedings of the 47th Annual ACM on Symposium on Theory of Computing*, pp. 499–508, 2015.

Olivier Chapelle. Modeling delayed feedback in display advertising. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1097–1105, 2014.

Niladri S. Chatterji, Aldo Pacchiano, and Peter L. Bartlett. Online learning with kernel losses. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 971–980, 2019.

Ku-Chun Chou, Hsuan-Tien Lin, Chao-Kai Chiang, and Chi-Jen Lu. Pseudo-reward algorithms for contextual bandits with linear payoff functions. In *Proceedings of the 6th Asian Conference on Machine Learning*, pp. 344–359, 2015.

Wei Chu, Lihong Li, Lev Reyzin, and Robert E. Schapire. Contextual bandits with linear payoff functions. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, pp. 208–214, 2011.

Maria Dimakopoulou, Zhengyuan Zhou, Susan Athey, and Guido Imbens. Balanced linear contextual bandits. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, pp. 3445–3453, 2019.

Hossein Esfandiari, Amin Karbasi, Abbas Mehrabian, and Vahab S. Mirrokni. Regret bounds for batched bandits. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, pp. 7340–7348, 2021.

Zijun Gao, Yanjun Han, Zhimei Ren, and Zhengqing Zhou. Batched multi-armed bandits problem. In *Advances in Neural Information Processing Systems 32*, pp. 501–511, 2019.

Quanquan Gu, Amin Karbasi, Khashayar Khosravi, Vahab Mirrokni, and Dongruo Zhou. Batched neural bandits. *arXiv preprint arXiv:2102.13028*, 2021.

Yanjun Han, Zhengqing Zhou, Zhengyuan Zhou, Jose H. Blanchet, Peter W. Glynn, and Yinyu Ye. Sequential batch learning in finite-action linear contextual bandits. *CoRR*, abs/2004.06321, 2020.

Chi Jin, Zeyuan Allen-Zhu, Sébastien Bubeck, and Michael I. Jordan. Is q-learning provably efficient? In *Advances in Neural Information Processing Systems 31*, 2018.

Daniel M Kane and Jelani Nelson. Sparser Johnson-Lindenstrauss transforms. *Journal of the ACM*, 61(1):4:1–4:23, 2014.

Gi-Soo Kim and Myunghee Cho Paik. Doubly-robust lasso bandit. In *Advances in Neural Information Processing Systems 32*, pp. 5869–5879, 2019.

Andrew S. Lan and Richard G. Baraniuk. A contextual bandits framework for personalized learning action selection. In *Proceedings of the 9th International Conference on Educational Data Mining*, pp. 424–429, 2016.

Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, pp. 661–670, 2010.

Shuai Li, Wei Chen, Shuai Li, and Kwong-Sak Leung. Improved algorithm on online clustering of bandits. In Sarit Kraus (ed.), *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, volume Li2019Improved, pp. 2923–2929, 2019.

Jelani Nelson and Huy L Nguyên. OSNAP: Faster numerical linear algebra algorithms via sparser subspace embeddings. In *Proceedings of the 54th Annual Symposium on Foundations of Computer Science*, pp. 117–126, 2013.

Gergely Neu and Julia Olkhovskaya. Efficient and robust algorithms for adversarial linear contextual bandits. In *Proceedings of the 33rd Conference on Learning Theory*, pp. 3049–3068, 2020.

Vianney Perchet, Philippe Rigollet, Sylvain Chassang, and Erik Snowberg. Batched bandit problems. *The Annals of Statistics*, 44(2):660–681, 2016.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems 20*, pp. 1177–1184, 2007.

Ali Rahimi and Benjamin Recht. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *Advances in Neural Information Processing Systems 21*, pp. 1313–1320, 2008.

Zhimei Ren and Zhengyuan Zhou. Dynamic batch learning in high-dimensional sparse linear contextual bandits. *arXiv preprint arXiv:2008.11918*, 2020.

Yuta Saito, Gota Morishita, and Shota Yasui. Dual learning algorithm for delayed conversions. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1849–1852, 2020.

Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2011.

Michal Valko, Nathaniel Korda, Rémi Munos, Ilias N. Flaounas, and Nello Cristianini. Finite-time analysis of kernelised contextual bandits. In *Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence*, 2013.

Chi-Hua Wang and Guang Cheng. Online batch decision-making with high-dimensional covariates. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, pp. 3848–3857, 2020.

Shusen Wang, Alex Gittens, and Michael W. Mahoney. Sketched ridge regression: Optimization perspective, statistical perspective, and model averaging. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 3608–3616, 2017.

Jiaqi Yang, Wei Hu, Jason D. Lee, and Simon Shaolei Du. Impact of representation learning in linear bandits. In *Proceedings of the 9th International Conference on Learning Representations*, 2021.

E. Yom-Tov, G. Feraru, M. Kozdoba, S. Mannor, and I. Hochberg. Encouraging physical activity in patients with diabetes: Intervention using a reinforcement learning system. *Journal of Medical Internet Research*, 19(10):e338, 2017.

Yuya Yoshikawa and Yusaku Imai. A nonparametric delayed feedback model for conversion rate prediction. *arXiv:1802.00255v1*, 2018.

Kelly W. Zhang, Lucas Janson, and Susan A. Murphy. Inference for batched bandits. In *Advances in Neural Information Processing Systems 33*, pp. 9818–9829, 2020.

Xiao Zhang and Shizhong Liao. Incremental randomized sketching for online kernel learning. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 7394–7403, 2019.

A Detailed Problem Formulation and Detailed Proof in Problem Analysis

In this part, we give the detailed problem formulation and the detailed proof of Theorem 1 in problem analysis in Section 2.

A.1 Detailed Problem Formulation of CBB

In this paper, we focus on the setting of contextual batched bandits (CBB), which can be formulated as a 6-tuple $\langle \mathcal{S}, \mathcal{A}, \boldsymbol{\pi}, R, N, B \rangle$:
Context space $\mathcal{S} \subseteq \mathbb{R}^d$ means a vector space containing the context information received at each step, e.g., context summarizes the information of both the user and items in recommendation scenarios.
Action space $\mathcal{A} = \{A_j\}_{j \in [M]}$ contains M candidate actions for execution. As an example, in recommender systems, each action corresponds to a candidate item, and selecting an action means that the corresponding item is recommended.
Policy $\boldsymbol{\pi}$ determines which action to take at each step, which is a function of the context $\boldsymbol{s} \in \mathcal{S}$ and outputs an action for execution (or a selection distribution over action space $\mathcal{A}$).
Reward R in CBB is a *partial-information feedback* where rewards are unobserved for the non-executed actions. Consider a stochastic bandit setting, where the expectation of the true reward is assumed to be a function of the context $\boldsymbol{s} \in \mathcal{S}$. In particular, different from the shared expectation function of true rewards in existing batch bandits (Han et al., 2020), we assume that the expectation functions of true rewards are different for each action, where each expectation function corresponds to an unknown parameter vector $\boldsymbol{\theta}_A^* \in \mathbb{R}^d$, $A \in \mathcal{A}$. This setting for rewards matches many real-world applications, e.g., each action corresponds to a different category of candidate coupons in coupon recommendation.
Number of episodes N. The decision process in CBB is partitioned into N episodes. Within one episode, the agent updates the policy using the collected data, and then interacts with the environment for multiple steps using the updated and fixed policy.
Batch size B is the number of steps in each episode. That is, in each episode, the agent interacts with the environment B times using a fixed policy, and stores the contexts, executed actions, and observed rewards into a data buffer $\mathcal{D}$ at the end of each episode.

A.2 Detailed Description and Proof of Theorem 1 in Problem Analysis

We present some theoretical findings about the regret difference between the partial-information feedback and the full-information feedback. Assuming that the agent in the CBB setting can observe the rewards of all the candidate actions from the environment at each step, we apply the batched UCB policy (Han et al., 2020) to this setting (see Algorithm 3), We demonstrate an instantaneous regret bound in Theorem 5, where Theorem 5 is a detailed version of Theorem 1 in Section 2.

Algorithm 3 Batch UCB Policy Updating in the $(n+1)$-th episode in Full-Information CBB Setting

INPUT: Policy $\boldsymbol{\pi}_n$, data buffer $\mathcal{D}_{n+1}$, action space $\mathcal{A} = \{A_j\}_{j \in [M]}$, $\boldsymbol{\theta}_{A_j}^0 = \mathbf{0}$, $j \in [M]$, batch size B
OUTPUT: Updated policy $\boldsymbol{\pi}_{n+1}$

1: Let $\widetilde{\boldsymbol{L}}_n \in \mathbb{R}^{(n+1)B \times d}$ be the matrix that stores all the context vectors till the n-th episode as the row vectors
2: For $\forall A \in \mathcal{A}$, let $\widetilde{\boldsymbol{T}}_A^n \in \mathbb{R}^{(n+1)B}$ be the reward vector that stores all the rewards of action $A \in \mathcal{A}$ till the n-th episode
3: `// Policy Updating`
4: $\boldsymbol{\Upsilon}_{n+1} \leftarrow \widetilde{\boldsymbol{L}}_n^\intercal \widetilde{\boldsymbol{L}}_n$
5: **for all** action $A \in \mathcal{A}$ **do**
6: $\quad \boldsymbol{\theta}_A^{n+1} \leftarrow (\boldsymbol{I}_d + \boldsymbol{\Upsilon}_{n+1})^{-1} \widetilde{\boldsymbol{L}}_n^\intercal \widetilde{\boldsymbol{T}}_A^n$
7: **end for**
8: For a new context $\boldsymbol{s}$, $\boldsymbol{\pi}_{n+1}(\boldsymbol{s})$ is to choose the action following: $A \leftarrow \arg\max_{A \in \mathcal{A}} \langle \boldsymbol{\theta}_A^{n+1}, \boldsymbol{s} \rangle$
9: **return** $\{\boldsymbol{\theta}_A^{n+1}\}_{A \in \mathcal{A}}$

Theorem 5 (Instantaneous Regret Bound in Full-Information CBB Setting, Detailed Version of Theorem 1)**.** *Let $\widetilde{\boldsymbol{L}}_{n-1} \in \mathbb{R}^{nB\times d}$ be the matrix that stores all the context vectors till the $(n-1)$-th episode as the row vectors, and $\widetilde{\boldsymbol{T}}_A^{n-1} \in \mathbb{R}^{nB}$ be the reward vector that stores all the rewards of action $A \in \mathcal{A}$ till the $(n-1)$-th episode. Given the action space $\mathcal{A} = \{A_j\}_{j\in[M]}$, in the n-th episode, assume that the rewards are independent and bounded by C_R. Then, with probability at least $1-\delta$, for $\forall b \in [B]$ and $\forall A \in \mathcal{A}$, we have the following instantaneous regret bound in the n-th episode*

$$|\langle \boldsymbol{\theta}_A^n, \boldsymbol{s}_{n,b}\rangle - \langle \boldsymbol{\theta}_A^*, \boldsymbol{s}_{n,b}\rangle| \leq \left[\|\boldsymbol{\theta}_A^*\|_2 + \sqrt{2C_R^2 \log(2MB/\delta)}\right] \sqrt{\boldsymbol{s}_{n,b}^\intercal \left(\boldsymbol{I}_d + \boldsymbol{\Upsilon}_n\right)^{-1} \boldsymbol{s}_{n,b}}, \tag{11}$$

where $\boldsymbol{\Upsilon}_n = \widetilde{\boldsymbol{L}}_{n-1}^\intercal \widetilde{\boldsymbol{L}}_{n-1}$ and the parameter of reward model $\boldsymbol{\theta}_A^n$ in the batched UCB policy is obtained by

$$\boldsymbol{\theta}_A^n := \underset{\boldsymbol{\theta}\in\mathbb{R}^d}{\arg\min} \left\|\widetilde{\boldsymbol{L}}_{n-1}\boldsymbol{\theta} - \widetilde{\boldsymbol{T}}_A^{n-1}\right\|_2^2 + \|\boldsymbol{\theta}\|_2^2 = \left(\boldsymbol{I}_d + \boldsymbol{\Upsilon}_n\right)^{-1} \widetilde{\boldsymbol{L}}_{n-1}^\intercal \widetilde{\boldsymbol{T}}_A^{n-1}.$$

Further, the instantaneous regret bound Eq.(11) in FI-CBB setting is tighter than that in CBB setting (i.e., using the partial-information feedback). In particular, the variance term $\sqrt{\boldsymbol{s}_{n,b}^\intercal \left(\boldsymbol{I}_d + \boldsymbol{\Upsilon}_n\right)^{-1} \boldsymbol{s}_{n,b}}$ is smaller than that in CBB setting.

Proof of Theorem 1. By the formulation of $\boldsymbol{\theta}_A^n$ and the triangular inequality, we first obtain that

$$\begin{aligned}
&|\langle \boldsymbol{\theta}_A^n, \boldsymbol{s}_{n,b}\rangle - \langle \boldsymbol{\theta}_A^*, \boldsymbol{s}_{n,b}\rangle| \\
&= \left|\boldsymbol{s}_{n,b}^\intercal \left(\boldsymbol{I}_d + \boldsymbol{\Upsilon}_n\right)^{-1} \widetilde{\boldsymbol{L}}_{n-1}^\intercal \widetilde{\boldsymbol{T}}_A^{n-1} - \boldsymbol{s}_{n,b}^\intercal \boldsymbol{\theta}_A^*\right| \\
&= \left|\boldsymbol{s}_{n,b}^\intercal \left(\boldsymbol{I}_d + \boldsymbol{\Upsilon}_n\right)^{-1} \left[\widetilde{\boldsymbol{L}}_{n-1}^\intercal \widetilde{\boldsymbol{T}}_A^{n-1} - \left(\boldsymbol{I}_d + \boldsymbol{\Upsilon}_n\right)\boldsymbol{\theta}_A^*\right]\right| \\
&= \left|\boldsymbol{s}_{n,b}^\intercal \left(\boldsymbol{I}_d + \boldsymbol{\Upsilon}_n\right)^{-1} \left[\widetilde{\boldsymbol{L}}_{n-1}^\intercal \widetilde{\boldsymbol{T}}_A^{n-1} - \left(\boldsymbol{I}_d + \widetilde{\boldsymbol{L}}_{n-1}^\intercal \widetilde{\boldsymbol{L}}_{n-1}\right)\boldsymbol{\theta}_A^*\right]\right| \\
&= \left|\boldsymbol{s}_{n,b}^\intercal \left(\boldsymbol{I}_d + \boldsymbol{\Upsilon}_n\right)^{-1} \widetilde{\boldsymbol{L}}_{n-1}^\intercal \left(\widetilde{\boldsymbol{T}}_A^{n-1} - \widetilde{\boldsymbol{L}}_{n-1}\boldsymbol{\theta}_A^*\right) - \boldsymbol{s}_{n,b}^\intercal \left(\boldsymbol{I}_d + \boldsymbol{\Upsilon}_n\right)^{-1} \boldsymbol{\theta}_A^*\right| \\
&\leq \left|\boldsymbol{s}_{n,b}^\intercal \left(\boldsymbol{I}_d + \boldsymbol{\Upsilon}_n\right)^{-1} \widetilde{\boldsymbol{L}}_{n-1}^\intercal \left(\widetilde{\boldsymbol{T}}_A^{n-1} - \widetilde{\boldsymbol{L}}_{n-1}\boldsymbol{\theta}_A^*\right)\right| + \left|\boldsymbol{s}_{n,b}^\intercal \left(\boldsymbol{I}_d + \boldsymbol{\Upsilon}_n\right)^{-1} \boldsymbol{\theta}_A^*\right|
\end{aligned} \tag{12}$$

Next, we bound the two terms in the last row of Eq.(12).

Bounding $\left|\boldsymbol{s}_{n,b}^\intercal \left(\boldsymbol{I}_d + \boldsymbol{\Upsilon}_n\right)^{-1} \widetilde{\boldsymbol{L}}_{n-1}^\intercal \left(\widetilde{\boldsymbol{T}}_A^{n-1} - \widetilde{\boldsymbol{L}}_{n-1}\boldsymbol{\theta}_A^*\right)\right|$:

Since $\mathrm{E}\left[\widetilde{\boldsymbol{T}}_A^{n-1}\right] = \widetilde{\boldsymbol{L}}_{n-1}\boldsymbol{\theta}_A^*$ and the received rewards are independent, by the Azuma-Hoeffding bound, we have

$$\begin{aligned}
&\Pr\left\{\left|\boldsymbol{s}_{n,b}^\intercal \left(\boldsymbol{I}_d + \boldsymbol{\Upsilon}_n\right)^{-1} \widetilde{\boldsymbol{L}}_{n-1}^\intercal \left(\widetilde{\boldsymbol{T}}_A^{n-1} - \widetilde{\boldsymbol{L}}_{n-1}\boldsymbol{\theta}_A^*\right)\right| \geq \nu\sqrt{\boldsymbol{s}_{n,b}^\intercal \left(\boldsymbol{I}_d + \boldsymbol{\Upsilon}_n\right)^{-1} \boldsymbol{s}_{n,b}}\right\} \\
&\leq 2\exp\left\{-\frac{\nu^2 \boldsymbol{s}_{n,b}^\intercal \left(\boldsymbol{I}_d + \boldsymbol{\Upsilon}_n\right)^{-1} \boldsymbol{s}_{n,b}}{2C_R^2 \|\widetilde{\boldsymbol{L}}_{n-1}\left(\boldsymbol{I}_d + \boldsymbol{\Upsilon}_n\right)^{-1} \boldsymbol{s}_{n,b}\|_2^2}\right\},
\end{aligned} \tag{13}$$

where $\nu > 0$ is some constant. Since

$$\begin{aligned}
\|\widetilde{\boldsymbol{L}}_{n-1}\left(\boldsymbol{I}_d + \boldsymbol{\Upsilon}_n\right)^{-1} \boldsymbol{s}_{n,b}\|_2^2 &= \boldsymbol{s}_{n,b}^\intercal \left(\boldsymbol{I}_d + \boldsymbol{\Upsilon}_n\right)^{-1} \widetilde{\boldsymbol{L}}_{n-1}^\intercal \widetilde{\boldsymbol{L}}_{n-1}\left(\boldsymbol{I}_d + \boldsymbol{\Upsilon}_n\right)^{-1} \boldsymbol{s}_{n,b} \\
&\leq \boldsymbol{s}_{n,b}^\intercal \left(\boldsymbol{I}_d + \boldsymbol{\Upsilon}_n\right)^{-1} \left(\boldsymbol{I}_d + \widetilde{\boldsymbol{L}}_{n-1}^\intercal \widetilde{\boldsymbol{L}}_{n-1}\right)\left(\boldsymbol{I}_d + \boldsymbol{\Upsilon}_n\right)^{-1} \boldsymbol{s}_{n,b} \\
&\leq \boldsymbol{s}_{n,b}^\intercal \left(\boldsymbol{I}_d + \boldsymbol{\Upsilon}_n\right)^{-1} \left(\boldsymbol{I}_d + \boldsymbol{\Upsilon}_n\right)\left(\boldsymbol{I}_d + \boldsymbol{\Upsilon}_n\right)^{-1} \boldsymbol{s}_{n,b} \\
&= \boldsymbol{s}_{n,b}^\intercal \left(\boldsymbol{I}_d + \boldsymbol{\Upsilon}_n\right)^{-1} \boldsymbol{s}_{n,b},
\end{aligned}$$

combing with Eq.(13) implies the following results

$$\begin{aligned}
&\Pr\left\{\left|\boldsymbol{s}_{n,b}^\intercal \left(\boldsymbol{I}_d + \boldsymbol{\Upsilon}_n\right)^{-1} \widetilde{\boldsymbol{L}}_{n-1}^\intercal \left(\widetilde{\boldsymbol{T}}_A^{n-1} - \widetilde{\boldsymbol{L}}_{n-1}\boldsymbol{\theta}_A^*\right)\right| \geq \nu\sqrt{\boldsymbol{s}_{n,b}^\intercal \left(\boldsymbol{I}_d + \boldsymbol{\Upsilon}_n\right)^{-1} \boldsymbol{s}_{n,b}}\right\} \\
&\leq 2\exp\left\{-\frac{\nu^2}{2C_R^2}\right\}.
\end{aligned} \tag{14}$$

Combing Eq.(14) with the union bound, yields that, with probability at least $1-\delta$, for $\forall b \in [B]$ and $\forall A \in \mathcal{A}$,

$$\left|\boldsymbol{s}_{n,b}^{\intercal}\left(\boldsymbol{I}_d+\boldsymbol{\Upsilon}_n\right)^{-1}\widetilde{\boldsymbol{L}}_{n-1}^{\intercal}\left(\widetilde{\boldsymbol{T}}_A^{n-1}-\widetilde{\boldsymbol{L}}_{n-1}\boldsymbol{\theta}_A^*\right)\right| \leq \nu\sqrt{\boldsymbol{s}_{n,b}^{\intercal}\left(\boldsymbol{I}_d+\boldsymbol{\Upsilon}_n\right)^{-1}\boldsymbol{s}_{n,b}}, \tag{15}$$

where the failure probability is

$$\delta = 2MB\exp\left\{-\frac{\nu^2}{2C_R^2}\right\},$$

yielding that $\nu = \sqrt{2C_R^2\log(2MB/\delta)}$.

Bounding $\left|\boldsymbol{s}_{n,b}^{\intercal}\left(\boldsymbol{I}_d+\boldsymbol{\Upsilon}_n\right)^{-1}\boldsymbol{\theta}_A^*\right|$:

Since $\boldsymbol{\Upsilon}_n$ is positive semi-definite, combining with the Hölder inequality, we obtain

$$\begin{aligned}\left|\boldsymbol{s}_{n,b}^{\intercal}\left(\boldsymbol{I}_d+\boldsymbol{\Upsilon}_n\right)^{-1}\boldsymbol{\theta}_A^*\right| &\leq \left\|\left(\boldsymbol{I}_d+\boldsymbol{\Upsilon}_n\right)^{-1}\boldsymbol{s}_{n,b}\right\|_2 \\ &= \sqrt{\boldsymbol{s}_{n,b}\left(\boldsymbol{I}_d+\boldsymbol{\Upsilon}_n\right)^{-1}\left(\boldsymbol{I}_d+\boldsymbol{\Upsilon}_n\right)^{-1}\boldsymbol{s}_{n,b}} \\ &\leq \sqrt{\boldsymbol{s}_{n,b}\left(\boldsymbol{I}_d+\boldsymbol{\Upsilon}_n\right)^{-1}\left(\boldsymbol{I}_d+\boldsymbol{\Upsilon}_n\right)\left(\boldsymbol{I}_d+\boldsymbol{\Upsilon}_n\right)^{-1}\boldsymbol{s}_{n,b}} \\ &= \sqrt{\boldsymbol{s}_{n,b}\left(\boldsymbol{I}_d+\boldsymbol{\Upsilon}_n\right)^{-1}\boldsymbol{s}_{n,b}}.\end{aligned} \tag{16}$$

Combing Eq.(15) and Eq.(16) concludes the proof.

Similarly to the proof of Eq.(20), we can obtain that the variance term in full-information setting is smaller than that in partial-information setting. □

B Detailed Proofs in Theoretical Analysis

In this section, we provide the instantaneous regret bound in each episode, prove the approximation error of sketching, and analyze the regret for policy updating in the CBB setting. Figure 4 describes the dependence structure of our theoretical results.

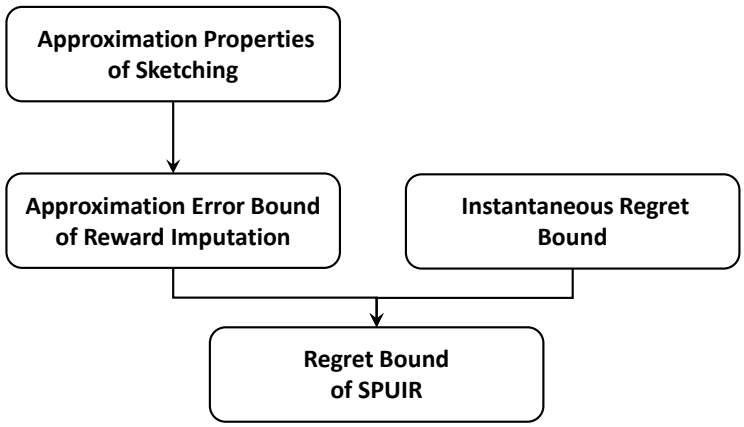


Figure 4: The dependence structure of our theoretical results, where the proof of instantaneous regret bound (Theorem 2) is provided in Appendix B.1, the analysis of approximation properties of sketching (Theorem 6) is given in Appendix B.2, the approximation error bound of reward imputation (Theorem 3) is proven in Appendix B.3, and the regret of SPUIR (Theorem 4) is analyzed in Appendix B.4

B.1 PROOF OF THEOREM 2

Proof of Theorem 2. From the formulation of $\boldsymbol{\theta}_A$ and the triangular inequality, we can obtain that, for each action $A \in \mathcal{A}$,

$$
\begin{aligned}
&|\langle \boldsymbol{\theta}_A^n, \boldsymbol{s}_{n,b}\rangle - \langle \boldsymbol{\theta}_A^*, \boldsymbol{s}_{n,b}\rangle| \\
&= \left|\boldsymbol{s}_{n,b}^\intercal \left(\boldsymbol{\Psi}_A^n\right)^{-1}\left(\boldsymbol{b}_A^n + \gamma \hat{\boldsymbol{b}}_A^n\right) - \boldsymbol{s}_{n,b}^\intercal \boldsymbol{\theta}_A^*\right| \\
&= \left|\boldsymbol{s}_{n,b}^\intercal \left(\boldsymbol{\Psi}_A^n\right)^{-1}\left[\left(\boldsymbol{L}_A^{n-1}\right)^\intercal \boldsymbol{T}_A^{n-1} + \gamma\left(\widehat{\boldsymbol{L}}_A^{n-1}\right)^\intercal \widehat{\boldsymbol{T}}_A^{n-1} - \boldsymbol{\Psi}_A^n \boldsymbol{\theta}_A^*\right]\right| \\
&= \left|\boldsymbol{s}_{n,b}^\intercal \left(\boldsymbol{\Psi}_A^n\right)^{-1}\left[\left(\boldsymbol{L}_A^{n-1}\right)^\intercal \boldsymbol{T}_A^{n-1} + \gamma\left(\widehat{\boldsymbol{L}}_A^{n-1}\right)^\intercal \widehat{\boldsymbol{T}}_A^{n-1} - \left(\lambda \boldsymbol{I}_d + \boldsymbol{\Phi}_A^n + \gamma \widehat{\boldsymbol{\Phi}}_A^n\right)\boldsymbol{\theta}_A^*\right]\right| \\
&= \Big|\boldsymbol{s}_{n,b}^\intercal \left(\boldsymbol{\Psi}_A^n\right)^{-1}\Big\{\left(\boldsymbol{L}_A^{n-1}\right)^\intercal \boldsymbol{T}_A^{n-1} + \gamma\left(\widehat{\boldsymbol{L}}_A^{n-1}\right)^\intercal \widehat{\boldsymbol{T}}_A^{n-1} - \\
&\quad \left[\lambda \boldsymbol{I}_d + \left(\boldsymbol{L}_A^{n-1}\right)^\intercal \boldsymbol{L}_A^{n-1} + \gamma\left(\widehat{\boldsymbol{L}}_A^{n-1}\right)^\intercal \widehat{\boldsymbol{L}}_A^{n-1}\right]\boldsymbol{\theta}_A^*\Big\}\Big| \\
&= \Big|\boldsymbol{s}_{n,b}^\intercal \left(\boldsymbol{\Psi}_A^n\right)^{-1}\left(\boldsymbol{L}_A^{n-1}\right)^\intercal\left(\boldsymbol{T}_A^{n-1} - \boldsymbol{L}_A^{n-1}\boldsymbol{\theta}_A^*\right) - \lambda \boldsymbol{s}_{n,b}^\intercal \left(\boldsymbol{\Psi}_A^n\right)^{-1}\boldsymbol{\theta}_A^* + \\
&\quad \boldsymbol{s}_{n,b}^\intercal \left(\boldsymbol{\Psi}_A^n\right)^{-1}\gamma\left(\widehat{\boldsymbol{L}}_A^{n-1}\right)^\intercal\left(\widehat{\boldsymbol{T}}_A^{n-1} - \widehat{\boldsymbol{L}}_A^{n-1}\boldsymbol{\theta}_A^*\right)\Big| \\
&\le \underbrace{\left|\boldsymbol{s}_{n,b}^\intercal \left(\boldsymbol{\Psi}_A^n\right)^{-1}\left(\boldsymbol{L}_A^{n-1}\right)^\intercal\left(\boldsymbol{T}_A^{n-1} - \boldsymbol{L}_A^{n-1}\boldsymbol{\theta}_A^*\right)\right|}_{X_A^{(1)}} + \underbrace{\lambda\left|\boldsymbol{s}_{n,b}^\intercal \left(\boldsymbol{\Psi}_A^n\right)^{-1}\boldsymbol{\theta}_A^*\right|}_{X_A^{(2)}} + \\
&\quad \underbrace{\left|\boldsymbol{s}_{n,b}^\intercal \left(\boldsymbol{\Psi}_A^n\right)^{-1}\gamma\left(\widehat{\boldsymbol{L}}_A^{n-1}\right)^\intercal\left(\widehat{\boldsymbol{T}}_A^{n-1} - \widehat{\boldsymbol{L}}_A^{n-1}\boldsymbol{\theta}_A^*\right)\right|}_{X_A^{(3)}}.
\end{aligned}
$$

Next, we bound $X_A^{(1)}$, $X_A^{(2)}$, and $X_A^{(3)}$. For convenience, we drop all the superscripts and subscripts about n and b. Similarly to the proof of Theorem 1, we bound $X_A^{(1)} + X_A^{(2)}$ as follows: with probability at least $1-\delta$,

$$
X_A^{(1)} + X_A^{(2)} \le (\lambda\|\boldsymbol{\theta}_A^*\|_2 + \nu)\sqrt{\boldsymbol{s}^\intercal \boldsymbol{\Psi}_A^{-1}\boldsymbol{s}}, \tag{17}
$$

where $\nu = \sqrt{2C_R^2 \log(2MB/\delta)}$. For $X_A^{(3)}$, using the Cauchy-Schwarz inequality, we have

$$
\begin{aligned}
X_A^{(3)} &\le \gamma \left\|\widehat{\boldsymbol{L}}_A \boldsymbol{\Psi}_A^{-1}\boldsymbol{s}\right\|_2 \left\|\widehat{\boldsymbol{T}}_A - \widehat{\boldsymbol{L}}_A\boldsymbol{\theta}_A^*\right\|_2 \\
&= \sqrt{\gamma}\sqrt{\boldsymbol{s}^\intercal \boldsymbol{\Psi}_A^{-1}\left(\gamma \widehat{\boldsymbol{L}}_A^\intercal \widehat{\boldsymbol{L}}_A\right)\boldsymbol{\Psi}_A^{-1}\boldsymbol{s}}\ \left\|\widehat{\boldsymbol{T}}_A - \widehat{\boldsymbol{L}}_A\boldsymbol{\theta}_A^*\right\|_2 \\
&\le \sqrt{\gamma}\sqrt{\boldsymbol{s}^\intercal \boldsymbol{\Psi}_A^{-1}\boldsymbol{s}}\ \left\|\widehat{\boldsymbol{T}}_A - \widehat{\boldsymbol{L}}_A\boldsymbol{\theta}_A^*\right\|_2.
\end{aligned} \tag{18}
$$

Now we need to bound the term $\left\|\widehat{\boldsymbol{T}}_A - \widehat{\boldsymbol{L}}_A\boldsymbol{\theta}_A^*\right\|_2$. Since using the discount parameter η in Eq.(4) is equivalent to multiplying both the imputed contexts and the imputed rewards by the parameter $\sqrt{\eta}$ in each episode, we have

$$
\begin{aligned}
\left\|\widehat{\boldsymbol{T}}_A - \widehat{\boldsymbol{L}}_A\boldsymbol{\theta}_A^*\right\|_2 &\le \sqrt{2\|\boldsymbol{\theta}_A\|_2^2 + 2\|\boldsymbol{\theta}_A^*\|_2^2}\sqrt{\sum_{i=0}^{N}\eta^{N-i}} \\
&= \sqrt{2\|\boldsymbol{\theta}_A\|_2^2 + 2\|\boldsymbol{\theta}_A^*\|_2^2}\sqrt{\frac{\eta^N(1-\eta^{N+1})}{1-\eta}} \\
&\le \sqrt{2\|\boldsymbol{\theta}_A\|_2^2 + 2\|\boldsymbol{\theta}_A^*\|_2^2}\sqrt{\frac{\eta^N}{1-\eta}}.
\end{aligned} \tag{19}
$$

Substituting Eq.(19) into Eq.(18) yields the upper bound of $X_A^{(3)}$.

Then, we prove that

$$\sqrt{\boldsymbol{s}^\intercal \left(\boldsymbol{\Psi}_A^n\right)^{-1} \boldsymbol{s}} \leq \sqrt{\boldsymbol{s}^\intercal \left(\lambda \boldsymbol{I}_d + \boldsymbol{\Phi}_A^n\right)^{-1} \boldsymbol{s}}. \tag{20}$$

holds, which is equivalent to

$$\boldsymbol{s}^\intercal \left(\lambda \boldsymbol{I}_d + \boldsymbol{\Phi} + \gamma \widehat{\boldsymbol{\Phi}}\right)^{-1} \boldsymbol{s} \leq \boldsymbol{s}^\intercal \left(\lambda \boldsymbol{I}_d + \boldsymbol{\Phi}\right)^{-1} \boldsymbol{s}. \tag{21}$$

Letting $\boldsymbol{\Theta} = \lambda \boldsymbol{I}_d + \boldsymbol{\Phi}$, by Sherman-Morrison-Woodbury formula, we have

$$\begin{aligned} \left(\boldsymbol{\Theta} + \gamma \widehat{\boldsymbol{\Phi}}\right)^{-1} = \left(\boldsymbol{\Theta} + \gamma \widehat{\boldsymbol{S}}^\intercal \widehat{\boldsymbol{S}}\right)^{-1} &= \boldsymbol{\Theta}^{-1} - \gamma \boldsymbol{\Theta}^{-1} \widehat{\boldsymbol{S}}^\intercal \left(\boldsymbol{I}_d + \gamma \widehat{\boldsymbol{S}} \boldsymbol{\Theta}^{-1} \widehat{\boldsymbol{S}}^\intercal\right)^{-1} \widehat{\boldsymbol{S}} \boldsymbol{\Theta}^{-1} \\ &= \boldsymbol{\Theta}^{-1} - \boldsymbol{\Theta}^{-1} \widehat{\boldsymbol{S}}^\intercal \left(\frac{\boldsymbol{I}_d}{\gamma} + \widehat{\boldsymbol{S}} \boldsymbol{\Theta}^{-1} \widehat{\boldsymbol{S}}^\intercal\right)^{-1} \widehat{\boldsymbol{S}} \boldsymbol{\Theta}^{-1}, \end{aligned} \tag{22}$$

yielding that Eq.(21) is equivalent to

$$\boldsymbol{s}^\intercal \boldsymbol{\Gamma} \boldsymbol{s} \geq 0, \tag{23}$$

where

$$\boldsymbol{\Gamma} = \boldsymbol{\Theta}^{-1} \widehat{\boldsymbol{S}}^\intercal \left(\boldsymbol{I}_d / \gamma + \widehat{\boldsymbol{S}} \boldsymbol{\Theta}^{-1} \widehat{\boldsymbol{S}}^\intercal\right)^{-1} \widehat{\boldsymbol{S}} \boldsymbol{\Theta}^{-1}.$$

Let $\boldsymbol{S} = \boldsymbol{U}_d \boldsymbol{\Sigma}_d^{1/2} \boldsymbol{V}_d^\intercal$, $\widehat{\boldsymbol{S}} = \widehat{\boldsymbol{U}}_d \widehat{\boldsymbol{\Sigma}}_d^{1/2} \widehat{\boldsymbol{V}}_d^\intercal$ be the Singular Value Decomposition (SVD) of $\boldsymbol{S}$ and $\widehat{\boldsymbol{S}}$, respectively. Note that $\boldsymbol{\Phi} = \boldsymbol{V}_d \boldsymbol{\Sigma}_d \boldsymbol{V}_d^\intercal$, $\widehat{\boldsymbol{\Phi}} = \widehat{\boldsymbol{V}}_d \widehat{\boldsymbol{\Sigma}}_d \widehat{\boldsymbol{V}}_d^\intercal$. We can obtain that $\boldsymbol{\Gamma}$ is a square symmetric positive semi-definite matrix, since $\boldsymbol{\Gamma}$ can be decomposed into

$$\boldsymbol{\Gamma} = \boldsymbol{Q}^\intercal \boldsymbol{Q},$$

where $\boldsymbol{P}_\gamma \boldsymbol{\Lambda}_\gamma \boldsymbol{P}_\gamma^\intercal$ is the SVD of $\boldsymbol{I}_d / \gamma + \widehat{\boldsymbol{S}} \boldsymbol{\Theta}^{-1} \widehat{\boldsymbol{S}}^\intercal$ and

$$\boldsymbol{Q} = \boldsymbol{\Lambda}_\gamma^{-1/2} \boldsymbol{P}_\gamma^\intercal \widehat{\boldsymbol{S}} \boldsymbol{\Theta}^{-1}.$$

Thus, $\forall \boldsymbol{s} \neq \boldsymbol{0}$, Eq.(23) holds, yielding that Eq.(21) also holds.

Finally, we prove that a larger imputation rate γ leads to a smaller variance term $\sqrt{\boldsymbol{s}^\intercal \left(\boldsymbol{\Psi}\right)^{-1} \boldsymbol{s}}$. From Eq.(22), the variance term can be represented as follows:

$$\sqrt{\boldsymbol{s}^\intercal \left(\boldsymbol{\Psi}\right)^{-1} \boldsymbol{s}} = \left[\boldsymbol{s}^\intercal \boldsymbol{\Theta}^{-1} \boldsymbol{s} - \boldsymbol{s}^\intercal \boldsymbol{\Theta}^{-1} \widehat{\boldsymbol{S}}^\intercal \boldsymbol{M}_\gamma^{-1} \widehat{\boldsymbol{S}} \boldsymbol{\Theta}^{-1} \boldsymbol{s}\right]^{1/2}, \tag{24}$$

where $\boldsymbol{M}_\gamma = \boldsymbol{I}_d / \gamma + \widehat{\boldsymbol{S}} \boldsymbol{\Theta}^{-1} \widehat{\boldsymbol{S}}^\intercal$. Letting $\boldsymbol{M}_\gamma = \boldsymbol{U}_{\boldsymbol{M}_\gamma} \boldsymbol{\Lambda}_{\boldsymbol{M}_\gamma} \boldsymbol{U}_{\boldsymbol{M}_\gamma}^\intercal$ be the SVD of $\boldsymbol{M}_\gamma$, and $\boldsymbol{z} = \boldsymbol{U}_{\boldsymbol{M}_\gamma}^\intercal \widehat{\boldsymbol{S}} \boldsymbol{\Theta}^{-1} \boldsymbol{s}$, from Eq.(24) we can written the variance term as follows:

$$\sqrt{\boldsymbol{s}^\intercal \left(\boldsymbol{\Psi}\right)^{-1} \boldsymbol{s}} = \left[\boldsymbol{s}^\intercal \boldsymbol{\Theta}^{-1} \boldsymbol{s} - \boldsymbol{z}^\intercal \boldsymbol{\Lambda}_{\boldsymbol{M}_\gamma}^{-1} \boldsymbol{z}\right]^{1/2}. \tag{25}$$

In Eq.(25), we can observed that

$$\boldsymbol{z}^\intercal \boldsymbol{\Lambda}_{\boldsymbol{M}_\gamma} \boldsymbol{z} = \|(\boldsymbol{\Lambda}_{\boldsymbol{M}_\gamma})^{-1/2} \boldsymbol{z}\|_2^2 \in \left[\frac{1}{\sigma_{\max}(\boldsymbol{M}) + 1/\gamma} \|\boldsymbol{z}\|_2^2,\ \frac{1}{\sigma_{\min}(\boldsymbol{M}) + 1/\gamma} \|\boldsymbol{z}\|_2^2\right]$$

where $\boldsymbol{M} = \widehat{\boldsymbol{S}} \boldsymbol{\Theta}^{-1} \widehat{\boldsymbol{S}}^\intercal$, which indicates that a larger imputation rate γ leads to a smaller variance term. □

B.2 Approximation Properties of Sketching

Although some error bounds of approximation using SJLT have been proposed (Nelson & Nguyên, 2013; Kane & Nelson, 2014; Bourgain et al., 2015), it is still unknown what is the lower bound of the sketch size while applying SJLT to the sketched ridge regression problem in our SPUIR. To address this issue, we first prove two approximation properties of SJLT which are necessary to achieve approximation error bound of the sketched ridge regression using SJLT. For convenience, we drop all the superscripts and subscripts in these theoretical results.

Lemma 1 ((Nelson & Nguyên, 2013)). *Let $\boldsymbol{U} \in \mathbb{R}^{L\times d}$ be a matrix with orthonormal columns, $\boldsymbol{\Pi} \in \mathbb{R}^{c\times L}$ the SJLT. Assuming that $D = \Theta(\varepsilon_\sigma^{-1}\log^3(d\delta_0^{-1}))$ for $\boldsymbol{\Pi}$, $\varepsilon_\sigma \in (0,1)$ and $d \le c$, with probability at least $1-\delta_0$ all singular values of $\boldsymbol{\Pi U}$*

$$\sigma_i(\boldsymbol{\Pi U}) = 1 \pm \varepsilon_\sigma, \quad i \in [d],$$

as long as

$$c \ge \frac{d\log^8\left(d\delta_0^{-1}\right)}{\varepsilon_\sigma^2}.$$

Further, this holds if the hash function h and σ defining the $\boldsymbol{\Pi}$ is $\Omega\left(\log(d\delta_0^{-1})\right)$-wise independent.

Theorem 6 (Approximation Properties of SJLT). *Let $\boldsymbol{U} \in \mathbb{R}^{L\times d}$ be a matrix with orthonormal columns, and $\boldsymbol{A}$ be a matrix of any proper size. If $\boldsymbol{\Pi} \in \mathbb{R}^{c\times L}$ is the SJLT satisfying the assumptions in Lemma 1, and $d \le c \le L$, then $\boldsymbol{\Pi}$ has the following two properties:*

1) Subspace embedding property: set $c = \Omega\left(d\ \mathrm{polylog}\left(d\delta_{\mathrm{s}}^{-1}\right)/\varepsilon_{\mathrm{s}}^2\right)$, for $\varepsilon_{\mathrm{s}} \in (0,1)$, with probability at least $1-\delta_{\mathrm{s}}$,

$$\left\|\boldsymbol{U}^\intercal\boldsymbol{\Pi}^\intercal\boldsymbol{\Pi U} - \boldsymbol{I}_d\right\|_2 \le \varepsilon_{\mathrm{s}};$$

2) Matrix multiplication property: set $c = \Omega(d/(\varepsilon_{\mathrm{m}}\delta_{\mathrm{m}}))$, for $\varepsilon_{\mathrm{m}} \in (0,1)$, with probability at least $1-\delta_{\mathrm{m}}$,

$$\|\boldsymbol{U}^\intercal\boldsymbol{\Pi}^\intercal\boldsymbol{\Pi A} - \boldsymbol{U}^\intercal\boldsymbol{A}\|_{\mathrm{F}}^2 \le \varepsilon_{\mathrm{m}}\|\boldsymbol{A}\|_{\mathrm{F}}^2.$$

Proof of Theorem 6. 1) From Lemma 1, by setting $c = \Omega\left(d\ \mathrm{polylog}\left(d\delta_{\mathrm{s}}^{-1}\right)/\varepsilon_0^2\right)$, we can obtain the upper bounds of eigenvalues: with probability at least $1-\delta_{\mathrm{s}}$,

$$\lambda_i\left(\boldsymbol{U}^\intercal\boldsymbol{\Pi}^\intercal\boldsymbol{\Pi U}\right) = \sigma_i^2(\boldsymbol{\Pi U}) \in [(1-\varepsilon_\sigma)^2, (1+\varepsilon_\sigma)^2] \subseteq [1-2\varepsilon_\sigma, 1+3\varepsilon_\sigma], \tag{26}$$

which yields that

$$\left|\lambda_i\left(\boldsymbol{U}^\intercal\boldsymbol{\Pi}^\intercal\boldsymbol{\Pi U} - \boldsymbol{I}_d\right)\right| \le 3\varepsilon_\sigma. \tag{27}$$

Eq.(28) is equivalent to

$$\left\|\boldsymbol{U}^\intercal\boldsymbol{\Pi}^\intercal\boldsymbol{\Pi U} - \boldsymbol{I}_d\right\|_2 \le 3\varepsilon_\sigma.$$

Letting $\varepsilon_{\mathrm{s}} = 3\varepsilon_\sigma$ and $\varepsilon_\sigma \in (0, 1/3)$ yields the subspace embedding property.

2) From Lemma 1 in (Zhang & Liao, 2019), we have

$$\mathbb{E}\left[\|\boldsymbol{U}^\intercal\boldsymbol{\Pi}^\intercal\boldsymbol{\Pi A} - \boldsymbol{U}^\intercal\boldsymbol{A}\|_{\mathrm{F}}^2\right] \le \frac{2}{c}\|\boldsymbol{U}\|_{\mathrm{F}}^2\|\boldsymbol{A}\|_{\mathrm{F}}^2 = \frac{2d}{c}\|\boldsymbol{A}\|_{\mathrm{F}}^2. \tag{28}$$

Combining Eq.(28) with the Markov's inequality, we obtain that, with probability at least $1-\delta_{\mathrm{m}}$,

$$\|\boldsymbol{U}^\intercal\boldsymbol{\Pi}^\intercal\boldsymbol{\Pi A} - \boldsymbol{U}^\intercal\boldsymbol{A}\|_{\mathrm{F}}^2 \le \frac{2d}{\delta_{\mathrm{m}}c}\|\boldsymbol{A}\|_{\mathrm{F}}^2.$$

Letting $\varepsilon_{\mathrm{m}} = \dfrac{2d}{\delta_{\mathrm{m}}c}$ yields the matrix multiplication property.

□

B.3 Proof of Theorem 3

Next, using the approximation properties of SJLT in Theorem 6, we prove that the objective function value of the imputation regularized ridge regression problem for reward imputation can be approximated well with a relative-error bound. Moreover, we prove that the solution solving the sketched ridge regression problem for reward imputation is also a good approximation of the solution solving the imputation regularized ridge regression. The following theorem is a detailed version of Theorem 3.

Theorem 7 (Approximation Error Bound of Imputation using Sketching, Detailed Version of Theorem 3). *Let $\gamma \in [0,1]$ be the imputation rate, $\lambda > 0$ the regularization parameter, $\boldsymbol{\Pi} \in \mathbb{R}^{c\times L}$ and $\widehat{\boldsymbol{\Pi}} \in \mathbb{R}^{c\times\widehat{L}}$ be the SJLT, and $\boldsymbol{L} \in \mathbb{R}^{L\times d}, \widehat{\boldsymbol{L}} \in \mathbb{R}^{\widehat{L}\times d}, \boldsymbol{T} \in \mathbb{R}^L, \widehat{\boldsymbol{T}} \in \mathbb{R}^{\widehat{L}}, \boldsymbol{\theta} \in \mathbb{R}^d$. Denote the*

imputation regularized ridge regression function F and sketched ridge regression function F^{S} for reward imputation by

$$F(\boldsymbol{\theta}) = \|\boldsymbol{L}\boldsymbol{\theta} - \boldsymbol{T}\|_2^2 + \gamma \left\|\widehat{\boldsymbol{L}}\boldsymbol{\theta} - \widehat{\boldsymbol{T}}\right\|_2^2 + \lambda\|\boldsymbol{\theta}\|_2^2,$$

$$F^{\mathrm{S}}(\boldsymbol{\theta}) = \|\boldsymbol{\Pi}\left(\boldsymbol{L}\boldsymbol{\theta} - \boldsymbol{T}\right)\|_2^2 + \gamma \left\|\widehat{\boldsymbol{\Pi}}\left(\widehat{\boldsymbol{L}}\boldsymbol{\theta} - \widehat{\boldsymbol{T}}\right)\right\|_2^2 + \lambda\|\boldsymbol{\theta}\|_2^2,$$

and the solutions of these regression problems by

$$\bar{\boldsymbol{\theta}} = \operatorname*{arg\,min}_{\boldsymbol{\theta}\in\mathbb{R}^d} F(\boldsymbol{\theta}) \quad \textit{and} \quad \tilde{\boldsymbol{\theta}} = \operatorname*{arg\,min}_{\boldsymbol{\theta}\in\mathbb{R}^d} F^{\mathrm{S}}(\boldsymbol{\theta}).$$

Let $\delta \in (0, 0.1]$, $\varepsilon \in (0, 1)$, $\rho_\lambda = \|\boldsymbol{L}_{\mathrm{all}}\|_2^2/(\|\boldsymbol{L}_{\mathrm{all}}\|_2^2 + \lambda)$. For $\boldsymbol{\Pi}$ and $\widehat{\boldsymbol{\Pi}}$, assuming that $D = \Theta(\varepsilon^{-1}\log^3(d\delta^{-1}))$ and

$$c = \Omega\left(d\,\mathrm{polylog}\left(d\delta^{-1}\right)/\varepsilon^2\right),$$

with probability at least $1-\delta$,

$$F(\tilde{\boldsymbol{\theta}}) \le (1 + \rho_\lambda\varepsilon)F(\bar{\boldsymbol{\theta}}), \tag{29}$$

$$\|\tilde{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}}\|_2 \le \frac{\sqrt{\rho_\lambda\varepsilon F\left(\bar{\boldsymbol{\theta}}\right)}}{\sigma_{\min}\left(\boldsymbol{L}_{\mathrm{all}}^\lambda\right)}, \tag{30}$$

where $\boldsymbol{L}_{\mathrm{all}}^\lambda = \left[\boldsymbol{L}; \sqrt{\gamma}\widehat{\boldsymbol{L}}; \sqrt{\lambda}\boldsymbol{I}_d\right] \in \mathbb{R}^{(L+\widehat{L}+d)\times d}$. Furthermore, if there is a constant fraction of the norm of $\boldsymbol{T}_{\mathrm{all}}^0$ lies in the column space of L_{all}^λ, then Eq.(30) can be strengthened. Formally, assuming that a mild structural assumption on the context matrix and the reward vector is satisfied, i.e., $\|\boldsymbol{U}_{\mathrm{all}}\boldsymbol{U}_{\mathrm{all}}^\intercal\boldsymbol{T}_{\mathrm{all}}^0\|_2 \ge \xi\|\boldsymbol{T}_{\mathrm{all}}^0\|_2$ with a constant $\xi \in (0, 1]$, then with probability at least $1-\delta$,

$$\|\tilde{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}}\|_2 \le \left(\kappa(\boldsymbol{L}_{\mathrm{all}}^\lambda)\sqrt{\xi^{-2} - 1}\right)\sqrt{\rho_\lambda\epsilon}\|\bar{\boldsymbol{\theta}}\|_2, \tag{31}$$

where $\kappa(\boldsymbol{A})$ denotes the condition number of $\boldsymbol{A}$, $\boldsymbol{T}_{\mathrm{all}}^0 = [\boldsymbol{T}; \widehat{\boldsymbol{T}}; \boldsymbol{0}_d] \in \mathbb{R}^{(L+\widehat{L}+d)}$, and $\boldsymbol{L}_{\mathrm{all}}^\lambda = \boldsymbol{U}_{\mathrm{all}}\boldsymbol{\Sigma}_{\mathrm{all}}\boldsymbol{V}_{\mathrm{all}}^\intercal$ is the SVD of $\boldsymbol{L}_{\mathrm{all}}^\lambda$.

Proof of Theorem 7. We first introduce some more notation of block matrices that will simplify the proof of the theorem:

$$\boldsymbol{\Pi}_{\mathrm{all}} = \begin{pmatrix} \boldsymbol{\Pi} & \boldsymbol{O} \\ \boldsymbol{O} & \widehat{\boldsymbol{\Pi}} \end{pmatrix}, \quad \boldsymbol{L}_{\mathrm{all}} = \begin{pmatrix} \boldsymbol{L} \\ \sqrt{\gamma}\widehat{\boldsymbol{L}} \end{pmatrix}, \quad \boldsymbol{T}_{\mathrm{all}} = \begin{pmatrix} \boldsymbol{T} \\ \widehat{\boldsymbol{T}} \end{pmatrix}. \tag{32}$$

Then the regression functions can be rewritten as follows:

$$F(\boldsymbol{\theta}) = \|\boldsymbol{L}_{\mathrm{all}}\boldsymbol{\theta} - \boldsymbol{T}_{\mathrm{all}}\|_2^2 + \lambda\|\boldsymbol{\theta}\|_2^2, \qquad F^{\mathrm{S}}(\boldsymbol{\theta}) = \|\boldsymbol{\Pi}_{\mathrm{all}}\left(\boldsymbol{L}_{\mathrm{all}}\boldsymbol{\theta} - \boldsymbol{T}_{\mathrm{all}}\right)\|_2^2 + \lambda\|\boldsymbol{\theta}\|_2^2.$$

Obviously, $\boldsymbol{\Pi}_{\mathrm{all}}$ is still an SJLT. Combining Theorem 6 with theorem 19 in (Wang et al., 2017), we can obtain, setting

$$c = \Omega\left(\max\{d\,\mathrm{polylog}\left(d\delta_{\mathrm{s}}^{-1}\right)/\varepsilon_{\mathrm{s}}^2,\ d/(\varepsilon_{\mathrm{m}}\delta_{\mathrm{m}})\}\right),$$

with probability at least $1 - (\delta_{\mathrm{s}} + \delta_{\mathrm{m}})$,

$$F(\tilde{\boldsymbol{\theta}}) - F(\bar{\boldsymbol{\theta}}) \le \rho_\lambda\tau F(\bar{\boldsymbol{\theta}}), \tag{33}$$

where $\rho_\lambda = \frac{\|\boldsymbol{L}_{\mathrm{all}}\|_2^2}{\|\boldsymbol{L}_{\mathrm{all}}\|_2^2+\lambda}$ and $\tau = \frac{2\max\{\varepsilon_{\mathrm{s}}^2, \varepsilon_{\mathrm{m}}\}}{1-\varepsilon_{\mathrm{s}}}$. Letting $\varepsilon_{\mathrm{s}} = \varepsilon_{\mathrm{m}} := \varepsilon_0$, Eq.(33) can be rewritten as

$$F(\tilde{\boldsymbol{\theta}}) - F(\bar{\boldsymbol{\theta}}) \le \frac{2\rho_\lambda\varepsilon_0}{1-\varepsilon_0}F(\bar{\boldsymbol{\theta}}), \tag{34}$$

Assuming that $\delta_{\mathrm{s}} = \delta_{\mathrm{m}} := \delta/2 \in (0, 0.1]$ and $\varepsilon_0 \in (0, 1/3)$, setting $\epsilon = \frac{2\varepsilon_0}{1-\varepsilon_0} \in (0, 1)$, from Eq.(34) we obtain the upper bound Eq.(29).

Next, we bound the difference between the solutions solving the sketched ridge regression problem and the original regression problem. Since $\sigma_{\min}^2(\boldsymbol{A})\|\boldsymbol{x}\|_2^2 \le \|\boldsymbol{A}\boldsymbol{x}\|_2^2$ for any $\boldsymbol{A}$ and $\boldsymbol{x}$ with proper sizes, we have

$$\sigma_{\min}^2(\boldsymbol{L}_{\mathrm{all}})\|\tilde{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}}\|_2^2 \le \left\|\boldsymbol{L}_{\mathrm{all}}(\tilde{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}})\right\|_2^2. \tag{35}$$

The key ingredient of bounding $\|\tilde{\boldsymbol{\theta}}-\bar{\boldsymbol{\theta}}\|_2$ is to bound $\|\boldsymbol{L}_{\text{all}}(\tilde{\boldsymbol{\theta}}-\bar{\boldsymbol{\theta}})\|_2$. Let $\boldsymbol{L}_{\text{all}}^{\lambda} = \left[\boldsymbol{L}; \sqrt{\gamma}\widehat{\boldsymbol{L}}; \sqrt{\lambda}\boldsymbol{I}_d\right] \in \mathbb{R}^{(L+\widehat{L}+d)\times d}$, $\boldsymbol{T}_{\text{all}}^0 = [\boldsymbol{T}; \widehat{\boldsymbol{T}}; \boldsymbol{0}_d] \in \mathbb{R}^{(L+\widehat{L}+d)}$, $\boldsymbol{L}_{\text{all}}^{\lambda} = \boldsymbol{U}_{\text{all}}\boldsymbol{\Sigma}_{\text{all}}\boldsymbol{V}_{\text{all}}^{\intercal}$ be the SVD of $\boldsymbol{L}_{\text{all}}^{\lambda}$, and denote a matrix with orthonormal columns by $\boldsymbol{U}_{\text{all}}^{\perp} \in \mathbb{R}^{(L+\widehat{L}+d)\times(L+\widehat{L})}$ which satisfies

$$\boldsymbol{U}_{\text{all}}\boldsymbol{U}_{\text{all}}^{\intercal} + \boldsymbol{U}_{\text{all}}^{\perp}(\boldsymbol{U}_{\text{all}}^{\perp})^{\intercal} = \boldsymbol{I}_{L+\widehat{L}+d} \quad \text{and} \quad \boldsymbol{U}_{\text{all}}^{\intercal}\boldsymbol{U}_{\text{all}}^{\perp} = \boldsymbol{O}.$$

Then, we can rewrite the solution $\bar{\boldsymbol{\theta}}$ as follows:

$$\begin{aligned}\bar{\boldsymbol{\theta}} &= \underset{\boldsymbol{\theta}\in\mathbb{R}^d}{\arg\min}\, F(\boldsymbol{\theta}) = \underset{\boldsymbol{\theta}\in\mathbb{R}^d}{\arg\min} \left\|\boldsymbol{L}_{\text{all}}^{\lambda}\boldsymbol{\theta} - \boldsymbol{T}_{\text{all}}^0\right\|_2^2 \\ &= (\boldsymbol{L}_{\text{all}}^{\lambda})^{\dagger}\boldsymbol{T}_{\text{all}}^0 = \boldsymbol{V}_{\text{all}}\boldsymbol{\Sigma}_{\text{all}}^{-1}\boldsymbol{U}_{\text{all}}^{\intercal}\boldsymbol{T}_{\text{all}}^0,\end{aligned}$$

which yields that

$$\begin{aligned}\boldsymbol{T}_{\text{all}}^0 - \boldsymbol{L}_{\text{all}}^{\lambda}\bar{\boldsymbol{\theta}} &= \boldsymbol{T}_{\text{all}}^0 - \boldsymbol{L}_{\text{all}}^{\lambda}\boldsymbol{V}_{\text{all}}\boldsymbol{\Sigma}_{\text{all}}^{-1}\boldsymbol{U}_{\text{all}}^{\intercal}\boldsymbol{T}_{\text{all}}^0 \\ &= \boldsymbol{T}_{\text{all}}^0 - \boldsymbol{U}_{\text{all}}\boldsymbol{\Sigma}_{\text{all}}\boldsymbol{V}_{\text{all}}^{\intercal}\boldsymbol{V}_{\text{all}}\boldsymbol{\Sigma}_{\text{all}}^{-1}\boldsymbol{U}_{\text{all}}^{\intercal}\boldsymbol{T}_{\text{all}}^0 \\ &= \boldsymbol{T}_{\text{all}}^0 - \boldsymbol{U}_{\text{all}}\boldsymbol{U}_{\text{all}}^{\intercal}\boldsymbol{T}_{\text{all}}^0 \\ &= \boldsymbol{U}_{\text{all}}^{\perp}(\boldsymbol{U}_{\text{all}}^{\perp})^{\intercal}\boldsymbol{T}_{\text{all}}^0.\end{aligned} \tag{36}$$

Thus, $\boldsymbol{T}_{\text{all}}^0 - \boldsymbol{L}_{\text{all}}^{\lambda}\bar{\boldsymbol{\theta}}$ is orthogonal to $\boldsymbol{U}_{\text{all}}$, and consequently to $\boldsymbol{L}_{\text{all}}^{\lambda}(\tilde{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}})$, and we can obtain the following equality by Pythagoras's theorem:

$$\left\|\boldsymbol{L}_{\text{all}}^{\lambda}(\tilde{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}})\right\|_2^2 = \left\|\boldsymbol{L}_{\text{all}}^{\lambda}\tilde{\boldsymbol{\theta}} - \boldsymbol{T}_{\text{all}}^0\right\|_2^2 - \left\|\boldsymbol{L}_{\text{all}}^{\lambda}\bar{\boldsymbol{\theta}} - \boldsymbol{T}_{\text{all}}^0\right\|_2^2. \tag{37}$$

Combining Eq.(37) with Eq.(29) yields that

$$\left\|\boldsymbol{L}_{\text{all}}^{\lambda}(\tilde{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}})\right\|_2^2 = F(\tilde{\boldsymbol{\theta}}) - F(\bar{\boldsymbol{\theta}}) \leq \rho_{\lambda}\varepsilon F(\bar{\boldsymbol{\theta}}). \tag{38}$$

Substituting Eq.(38) into Eq.(35) concludes the proof of Eq.(30).

If we make a mild structural assumption on the context matrix and the reward vector, we can provide a stronger bound of $\|\tilde{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}}\|_2$. Specifically, assuming that $\|\boldsymbol{U}_{\text{all}}\boldsymbol{U}_{\text{all}}^{\intercal}\boldsymbol{T}_{\text{all}}^0\|_2 \geq \xi\|\boldsymbol{T}_{\text{all}}^0\|_2$ with a constant $\xi \in (0, 1]$, from Eq.(36) and Pythagoras's theorem we have

$$\begin{aligned}F(\bar{\boldsymbol{\theta}}) &= \|\boldsymbol{L}_{\text{all}}^{\lambda}\bar{\boldsymbol{\theta}} - \boldsymbol{T}_{\text{all}}^0\|_2^2 \\ &= \|\boldsymbol{T}_{\text{all}}^0\|_2^2 - \|\boldsymbol{U}_{\text{all}}\boldsymbol{U}_{\text{all}}^{\intercal}\boldsymbol{T}_{\text{all}}^0\|_2^2 \\ &\leq (\xi^{-2} - 1)\|\boldsymbol{U}_{\text{all}}\boldsymbol{U}_{\text{all}}^{\intercal}\boldsymbol{T}_{\text{all}}^0\|_2^2 \\ &= (\xi^{-2} - 1)\|\boldsymbol{L}_{\text{all}}^{\lambda}\bar{\boldsymbol{\theta}}\|_2^2 \\ &\leq (\xi^{-2} - 1)\|\boldsymbol{L}_{\text{all}}^{\lambda}\|_2^2\, \|\bar{\boldsymbol{\theta}}\|_2^2 \\ &\leq (\xi^{-2} - 1)\sigma_{\max}^2(\boldsymbol{L}_{\text{all}}^{\lambda})\|\bar{\boldsymbol{\theta}}\|_2^2.\end{aligned} \tag{39}$$

Combining Eq.(39) with Eq.(30) yields Eq.(31). □

B.4 Proof of Theorem 4

Proof of Theorem 4. In our sketched policy, letting $C_{\boldsymbol{\theta}^*}^{\max} = \max_{A\in\mathcal{A}} \|\boldsymbol{\theta}_A^*\|_2$, $C_{\bar{\boldsymbol{\theta}}}^{\max} = \max_{A\in\mathcal{A}, n\in[N]} \|\bar{\boldsymbol{\theta}}_A^n\|_2$, $C_{\boldsymbol{\theta}}^{\max} = \sqrt{2(C_{\boldsymbol{\theta}^*}^{\max})^2 + 2(C_{\bar{\boldsymbol{\theta}}}^{\max})^2}$, $\nu = \sqrt{2C_R^2\log(2MB/\delta)}$, and

$$\omega = \lambda C_{\boldsymbol{\theta}^*}^{\max} + \nu + C_{\boldsymbol{\theta}}^{\max}\sqrt{\frac{\gamma\eta^N}{1-\eta}},$$

from Eq.(9) in Theorem 2 we obtain that

$$\left|\left\langle\bar{\boldsymbol{\theta}}_A^n, \boldsymbol{s}_{n,b}\right\rangle - \left\langle\boldsymbol{\theta}_A^*, \boldsymbol{s}_{n,b}\right\rangle\right| \leq \omega\sqrt{\boldsymbol{s}_{n,b}^{\intercal}\left(\boldsymbol{\Psi}_A^n\right)^{-1}\boldsymbol{s}_{n,b}}. \tag{40}$$

Before proving the upper bound of $\left|\left\langle \tilde{\boldsymbol{\theta}}_A^n, \boldsymbol{s}_{n,b}\right\rangle - \langle \boldsymbol{\theta}_A^*, \boldsymbol{s}_{n,b}\rangle\right|$, we need to provide a technical tool as follows. For convenience, we also drop all the superscripts and subscripts. The goal is to find a constant C_α such that

$$\sqrt{\boldsymbol{s}^\intercal \boldsymbol{\Psi}^{-1}\boldsymbol{s}} \leq C_\alpha \sqrt{\boldsymbol{s}^\intercal \boldsymbol{W}^{-1}\boldsymbol{s}}, \tag{41}$$

which is equivalent to the condition that the matrix $C_\alpha^2 \boldsymbol{W}^{-1} - \boldsymbol{\Psi}^{-1}$ is positive semidefinite. Let $\boldsymbol{L}_{\text{all}}$ and $\boldsymbol{\Pi}_{\text{all}}$ be the matrices defined in Eq.(32), $\boldsymbol{L}_{\text{all}} = \widetilde{\boldsymbol{U}}_{\text{all}}\widetilde{\boldsymbol{\Sigma}}_{\text{all}}\widetilde{\boldsymbol{V}}_{\text{all}}^\intercal$ be the SVD of $\boldsymbol{L}_{\text{all}}$, and $\tilde{\sigma}_1 \geq \tilde{\sigma}_2 \cdots \geq \tilde{\sigma}_d$ be the singular values of $\boldsymbol{L}_{\text{all}}$. Then the i-th eigenvalue of $\boldsymbol{\Psi}^{-1} = (\lambda \boldsymbol{I}_d + \boldsymbol{L}_{\text{all}}^\intercal \boldsymbol{L}_{\text{all}})^{-1}$ can be represented as $\lambda_i(\boldsymbol{\Psi}^{-1}) = 1/(\tilde{\sigma}_i^2 + \lambda)$, and the i-th eigenvalue of $\boldsymbol{W}^{-1} = (\lambda \boldsymbol{I}_d + \boldsymbol{L}_{\text{all}}^\intercal \boldsymbol{\Pi}_{\text{all}}^\intercal \boldsymbol{\Pi}_{\text{all}} \boldsymbol{L}_{\text{all}})^{-1}$ is $\lambda_i(\boldsymbol{W}^{-1}) = 1/(\hat{\lambda}_i + \lambda)$, where $\hat{\lambda}_i$ is the i-th eigenvalue of $\widetilde{\boldsymbol{\Sigma}}_{\text{all}}\widetilde{\boldsymbol{U}}_{\text{all}}^\intercal \boldsymbol{\Pi}_{\text{all}}^\intercal \boldsymbol{\Pi}_{\text{all}} \widetilde{\boldsymbol{U}}_{\text{all}} \widetilde{\boldsymbol{\Sigma}}_{\text{all}}$.

From the Lidskii's theorem and Eq.(26), we have

$$\hat{\lambda}_i \in [\tilde{\sigma}_d^2(1 - 2\varepsilon_\sigma), \tilde{\sigma}_1^2(1 + 3\varepsilon_\sigma)]. \tag{42}$$

Assuming that the positive semi-definiteness of $C_\alpha^2 \boldsymbol{W}^{-1} - \boldsymbol{\Psi}^{-1}$ is satisfied, we obtain that $C_\alpha^2 \lambda_i(\boldsymbol{W}^{-1}) - \lambda_i(\boldsymbol{\Psi}^{-1}) \geq 0$ for $i \in [d]$, and combining this inequality with Eq.(42) yields that

$$C_\alpha = \sqrt{[\tilde{\sigma}_1^2(1 + 3\varepsilon_\sigma) + \lambda]/(\tilde{\sigma}_d^2 + \lambda)}.$$

From the proof of Theorem 6 and Theorem 3, we can obtain that $\varepsilon_\sigma = \varepsilon/(6 + 3\varepsilon)$, yielding that

$$C_\alpha = \sqrt{\frac{\tilde{\sigma}_1^2[1 + \varepsilon/(2 + \varepsilon)] + \lambda}{\tilde{\sigma}_d^2 + \lambda}},$$

which decreases with increase of $1/\varepsilon$. Similarly to the proof of C_α satisfying Eq.(41), we can obtain that

$$\sqrt{\boldsymbol{s}^\intercal \boldsymbol{W}^{-1}\boldsymbol{s}} \leq C_{\text{reg}} \sqrt{\boldsymbol{s}^\intercal \boldsymbol{\Psi}^{-1}\boldsymbol{s}}, \tag{43}$$

provided that

$$C_{\text{reg}} = \sqrt{\frac{\tilde{\sigma}_1^2 + \lambda}{\tilde{\sigma}_d^2[1 - 2\varepsilon/(6 + 3\varepsilon)] + \lambda}}.$$

Obviously, C_{reg} also decreases with increase of $1/\varepsilon$.

Then, letting $\alpha = \omega C_\alpha$, from Eq.(40) and Eq.(41) we have

$$\begin{aligned}
\left|\left\langle \tilde{\boldsymbol{\theta}}_A^n, \boldsymbol{s}_{n,b}\right\rangle - \langle \boldsymbol{\theta}_A^*, \boldsymbol{s}_{n,b}\rangle\right| &\leq \left|\left\langle \tilde{\boldsymbol{\theta}}_A^n, \boldsymbol{s}_{n,b}\right\rangle - \left\langle \bar{\boldsymbol{\theta}}_A^n, \boldsymbol{s}_{n,b}\right\rangle\right| + \left|\left\langle \bar{\boldsymbol{\theta}}_A^n, \boldsymbol{s}_{n,b}\right\rangle - \langle \boldsymbol{\theta}_A^*, \boldsymbol{s}_{n,b}\rangle\right| \\
&\leq \left|\left\langle \tilde{\boldsymbol{\theta}}_A^n - \bar{\boldsymbol{\theta}}_A^n, \boldsymbol{s}_{n,b}\right\rangle\right| + \omega\sqrt{\boldsymbol{s}_{n,b}^\intercal \left(\boldsymbol{\Psi}_A^n\right)^{-1} \boldsymbol{s}_{n,b}} \\
&\leq Y_{n,b} + \alpha\sqrt{\boldsymbol{s}_{n,b}^\intercal \left(\boldsymbol{W}_A^n\right)^{-1} \boldsymbol{s}_{n,b}},
\end{aligned} \tag{44}$$

where $Y_{n,b}$ denotes the upper bound of $\left|\left\langle \tilde{\boldsymbol{\theta}}_A^n - \bar{\boldsymbol{\theta}}_A^n, \boldsymbol{s}_{n,b}\right\rangle\right|$ for $\forall A \in \mathcal{A}$.

Next, using the compatibility of norm, we give a specific representation of the sum of $Y_{n,b}$ as follows:

$$\begin{aligned}
\sum_{b \in [B]} Y_{n,b} &= \max_{A \in \mathcal{A}} \|\boldsymbol{S}_A^n(\tilde{\boldsymbol{\theta}}_A^n - \bar{\boldsymbol{\theta}}_A^n)\|_1 \\
&\leq \max_{A \in \mathcal{A}} \|\boldsymbol{S}_A^n\|_1 \|\tilde{\boldsymbol{\theta}}_A^n - \bar{\boldsymbol{\theta}}_A^n\|_1 \\
&\leq \max_{A \in \mathcal{A}} \|\boldsymbol{S}_A^n\|_1 \sqrt{d} \|\tilde{\boldsymbol{\theta}}_A^n - \bar{\boldsymbol{\theta}}_A^n\|_2.
\end{aligned} \tag{45}$$

Further, we give a more specific upper bound in Eq.(45) under mild structural assumption in Theorem 3. Let $\kappa_{\text{all}}^{\max}$ denote the maximum of the condition numbers of $\boldsymbol{L}_{\text{all}}^\lambda(A, n)$ for $A \in \mathcal{A}, n \in [N]$, and $\boldsymbol{L}_{\text{all}}^\lambda(A, n) = \left[\boldsymbol{L}_A^n; \sqrt{\gamma}\widehat{\boldsymbol{L}}_A^n; \sqrt{\lambda}\boldsymbol{I}_d\right]$, and $\boldsymbol{U}_{\text{all}}(A, n)$ be the left singular matrix of $\boldsymbol{L}_{\text{all}}^\lambda(A, n)$. Letting $\boldsymbol{T}_{\text{all}}^0(A, n) = [\boldsymbol{T}_A^n; \widehat{\boldsymbol{T}}_A^n; \boldsymbol{0}_d]$, assuming that $\|\boldsymbol{U}_{\text{all}}(A, n)\boldsymbol{U}_{\text{all}}(A, n)^\intercal \boldsymbol{T}_{\text{all}}^0(A, n)\|_2 \geq$

$\xi \|\boldsymbol{T}^0_{\text{all}}(A,n)\|_2$ with a constant $\xi \in (0,1]$, substituting the upper bound Eq.(31) in Theorem 3 into Eq.(45) yields that

$$\sum_{b\in[B]} Y_{n,b} \leq \left(\kappa^{\max}_{\text{all}} \sqrt{\xi^{-2}-1}\right) C_{\boldsymbol{S}} C^{\max}_{\bar{\boldsymbol{\theta}}} \sqrt{\rho_\lambda \epsilon d}, \tag{46}$$

where $C_{\boldsymbol{S}} = \max_{n\in[N], A\in\mathcal{A}} \|\boldsymbol{S}^n_A\|_1$.

From Eq.(43), Eq.(44), Eq.(46) and the definition of our sketched policy, letting $C_Y = \left(\kappa^{\max}_{\text{all}} \sqrt{\xi^{-2}-1}\right) C_{\boldsymbol{S}} C^{\max}_{\bar{\boldsymbol{\theta}}}$, we obtain that

$$\begin{aligned}
&\text{Reg}(\{A_{I_{n,b}}\}_{n\in[N],b\in[B]}) \\
&= \sum_{n\in[N],b\in[B]} \left[\max_{A\in\mathcal{A}} \langle \boldsymbol{\theta}^*_A, \boldsymbol{s}_{n,b}\rangle - \left\langle \boldsymbol{\theta}^*_{A_{I_{n,b}}}, \boldsymbol{s}_{n,b}\right\rangle\right] \\
&\leq \sum_{n\in[N],b\in[B]} \left[\max_{A\in\mathcal{A}} \left(\left\langle \tilde{\boldsymbol{\theta}}^n_A, \boldsymbol{s}_{n,b}\right\rangle + \alpha\sqrt{\boldsymbol{s}^\intercal_{n,b} \left(\boldsymbol{W}^n_A\right)^{-1} \boldsymbol{s}_{n,b}}\right) + Y_{n,b} - \left\langle \boldsymbol{\theta}^*_{A_{I_{n,b}}}, \boldsymbol{s}_{n,b}\right\rangle\right] \\
&= \sum_{n\in[N],b\in[B]} \left[\left\langle \tilde{\boldsymbol{\theta}}^n_{A_{I_{n,b}}}, \boldsymbol{s}_{n,b}\right\rangle + \alpha\sqrt{\boldsymbol{s}^\intercal_{n,b} \left(\boldsymbol{W}^n_{A_{I_{n,b}}}\right)^{-1} \boldsymbol{s}_{n,b}} + Y_{n,b} - \left\langle \boldsymbol{\theta}^*_{A_{I_{n,b}}}, \boldsymbol{s}_{n,b}\right\rangle\right] \\
&\leq 2\alpha \sum_{n\in[N],b\in[B]} \sqrt{\boldsymbol{s}^\intercal_{n,b} \left(\boldsymbol{W}^n_{A_{I_{n,b}}}\right)^{-1} \boldsymbol{s}_{n,b}} + 2 \sum_{n\in[N],b\in[B]} Y_{n,b} \\
&\leq 2\alpha C_{\text{reg}} \sqrt{B} \sum_{n\in[N]} \sqrt{\sum_{b\in[B]} \boldsymbol{s}^\intercal_{n,b} \left(\boldsymbol{\Psi}^n_{A_{I_{n,b}}}\right)^{-1} \boldsymbol{s}_{n,b}} + 2NC_Y\sqrt{\rho_\lambda \epsilon d} \\
&= 2\alpha C_{\text{reg}} \sqrt{B} \sum_{n\in[N]} \sqrt{\sum_{b\in[B]} \left\langle \boldsymbol{s}_{n,b}\, \boldsymbol{s}^\intercal_{n,b}, \left(\boldsymbol{\Psi}^n_{A_{I_{n,b}}}\right)^{-1}\right\rangle} + 2NC_Y\sqrt{\rho_\lambda \epsilon d} \\
&= 2\alpha C_{\text{reg}} \sqrt{B} \sum_{n\in[N]} \sqrt{\sum_{A\in\mathcal{A}} \left\langle \boldsymbol{S}^{n\intercal}_A \boldsymbol{S}^n_A, (\boldsymbol{\Psi}^n_A)^{-1}\right\rangle} + 2NC_Y\sqrt{\rho_\lambda \epsilon d} \\
&= 2\alpha C_{\text{reg}} \sqrt{B} \sum_{n\in[N]} \sqrt{\sum_{A\in\mathcal{A}} \text{tr}\left(\boldsymbol{S}^{n\intercal}_A \boldsymbol{S}^n_A (\boldsymbol{\Psi}^n_A)^{-1}\right)} + O(N\sqrt{\rho_\lambda \epsilon d}), \\
&\leq 2\alpha C_{\text{reg}} \sqrt{BM} \sum_{n\in[N]} \sqrt{\max_{A\in\mathcal{A}} \left\{\text{tr}\left(\boldsymbol{S}^{n\intercal}_A \boldsymbol{S}^n_A (\boldsymbol{\Psi}^n_A)^{-1}\right)\right\}} + O(N\sqrt{\rho_\lambda \epsilon d}).
\end{aligned} \tag{47}$$

When the structural assumption in Theorem 3 is not satisfied, from Eq.(30), we can obtain that the second term in Eq.(47) is also of order $O(\sqrt{\rho_\lambda \epsilon} d)$, which does not influence the order of the final regret bound. Finally, combining Eq.(47) with lemma 3 in (Han et al., 2020) gives the final regret bound. □

C Detailed Experimental Settings and More Experimental Results

In this section, we provide more details and results in the experiments.

C.1 Description of Datasets

Table 3 summarizes the description of datasets used in the experiments.

Next, we provide more details about the three datasets.

Synthetic Data. Inspired by the experiments in (Saito et al., 2020), the synthetic data generation procedure was formulated as follows, which simulates the streaming recommendation environment.

Table 3: Description of datasets in the experiments (T: number of instances; B: batch size; N: number of episodes; d: dimensionality of context; M: number of actions; C_B satisfying $B = C_B^2 N/d$)

Dataset	T	B	N	d	M	C_B
`synthetic data`	126,000	1,400	90	40	5	25.00
`Criteo-recent`	75,000	1,000	75	50	5	25.82
`Criteo-all`	1,276,000	4,000	319	50	15	25.04
`commercial product`	216,568	1,700	128	50	5	25.83

- Context $\boldsymbol{s}_i \in \mathbb{R}^d$: we drew elements of $\boldsymbol{s}_i$ independently from a Gaussian distribution $\mathcal{N}(0.1, 0.2^2)$, where $d = 40$;
- Click-Through-Rate (CTR): the CTRs for the 5 actions were respectively set as $\{10\%, 15\%, 25\%, 20\%, 30\%\}$;
- The indicator variables of click events:

$$C_i = \begin{cases} 1 & \text{a click occurs in context } \boldsymbol{s}_i, \\ 0 & \text{otherwise.} \end{cases}$$

  We sampled the click index set according to the uniform distribution.
- Conversion rate (CVR) in context $\boldsymbol{s}_i$: when $C_i = 1$,

$$\mathrm{CVR}(\boldsymbol{s}_i) := \mathrm{sigmoid}(\langle \boldsymbol{w}_{\mathrm{c}}, \boldsymbol{s}_i \rangle), = \frac{1}{1 + \exp(-\langle \boldsymbol{w}_{\mathrm{c}}, \boldsymbol{s}_i \rangle)},$$

  where the coefficient vector $\boldsymbol{w}_{\mathrm{c}} \in \mathbb{R}^d$ is sampled according to a Gaussian distribution as $\boldsymbol{w}_{\mathrm{c}} \sim \mathcal{N}(\kappa_{\mathrm{c}} \boldsymbol{1}_d, \sigma_{\mathrm{c}}^2 \boldsymbol{I}_d)$, and we set different means and standard deviations for different action with $\kappa_{\mathrm{c}} \in [0 : -0.2 : -0.8]$ and $\sigma_{\mathrm{c}} \in [0.01 : +0.01 : 0.05]$;

Criteo Data. We used the publicly available Criteo dataset[5], consisting of Criteo's traffic on display ads over a period of two months (Chapelle, 2014), where each context consists of 8 integer features and 9 categorical features. Following the experiments in (Yoshikawa & Imai, 2018), the categorical features were represented as one-hot vectors and then concatenated to the integer features. We reduced the dimensionality of the feature vectors to 50 using principal component analysis (PCA). All of the algorithms were tested in a simulated online environment that was trained on users' logs in the Criteo dataset. Specifically, we chose several campaigns from the Criteo dataset, where each campaign represents a category of items and corresponds to an action. This online environment contains a prediction model for the CVR, which was well trained by applying DFM (Chapelle, 2014) using the true user feedbacks. This environment model was trained for each chosen campaign, whose AUCs are ranging from 70% to 90%, assuring that the online environment can provide nearly realistic feedbacks. To simulate the uncertainty of user behaviors, Gaussian noises with zero-mean were added to the model parameters. At each step, the online environment randomly selected a campaign and samples one context from this campaign, and revealed the context to the agent with a preset CTR. To generate a reasonable sequence of instances, the environment kept the order of timestamps of the contexts in each campaign. We tested our algorithms and the baselines with the following two online environments on the Criteo dataset: `Criteo-recent` contains 5 campaigns ($75,000$ instances) chosen from the recent campaigns, corresponding to 5 actions; `Criteo-all` contains 15 campaigns ($1,276,000$ instances) chosen from all the campaigns, corresponding to 15 actions.

Data Collected from a Real Commercial App for Coupon Recommendation. To verify the effectiveness and efficiency of our algorithms on real products, we conducted experiments on a real dataset collected from a commercial social app. We call this dataset `commercial product`, where the data were collected after the users gave consent, and did not contain any personally identifiable information or offensive content. Since this dataset from a commercial app is proprietary,

[5]https://labs.criteo.com/2013/12/conversion-logs-dataset/

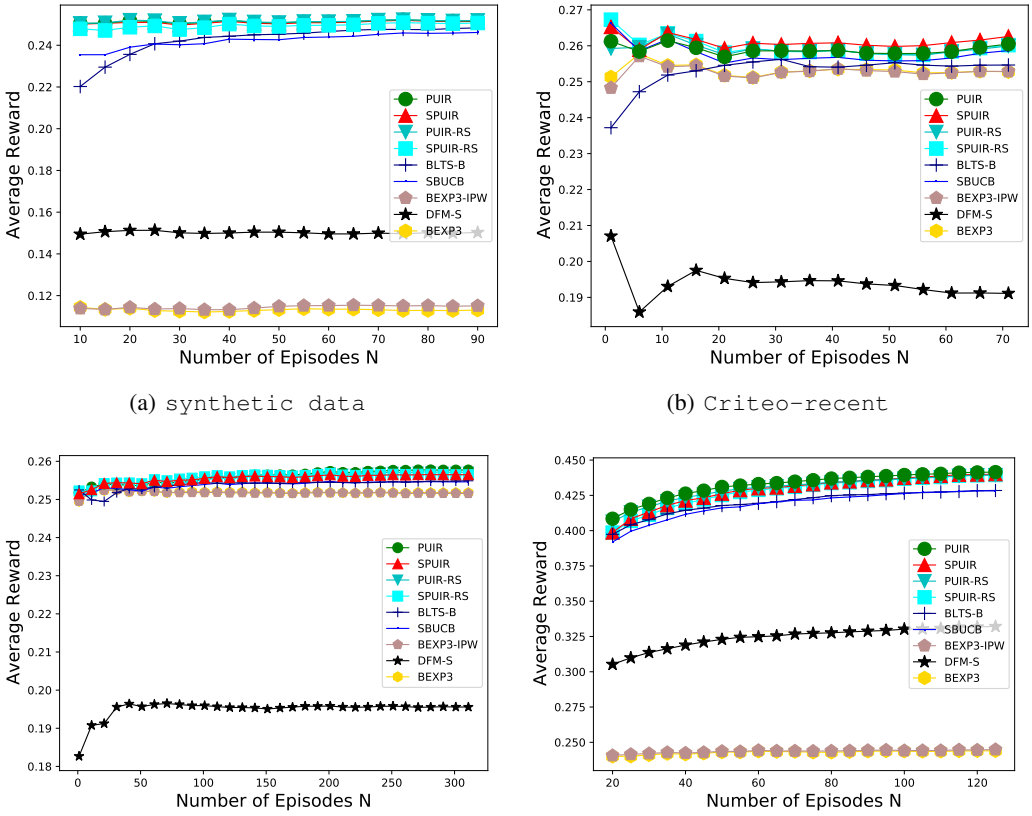
we did not provide a URL. We will release this dataset after the publication of this paper. In this commercial app, after clicking a recommended coupon, a user may convert the coupon after some time, or just leave it there. The dataset was collected during a 1-month period with a subsampling, and consists of $216,568$ instances from 5 categories of coupons, where each context is described by 86 numerical features and 16 categorical features. The timestamps of clicks and conversions were also recorded. Following the settings on the Criteo data, we also represented the categorical features as a one-hot vector, reduced the dimensionality of the feature vectors to 50 by PCA. The action space contains 5 actions, where each corresponding to one coupon category. Due to the limitation of real online experiments, in this experiment, we still trained DFM using the true user feedbacks as the online environment, where AUCs range from 75% to 90%.

C.2 Detailed Specification of Hyperparameters

In these experiments, the true reward is defined by $R_{i,A}^{\text{true}} = \lambda_{\text{c}} C_{i,A} + (1 - \lambda_{\text{c}}) V_{i,A}$ ($C_{i,A}$ and $V_{i,A}$ denote true binary variables of user click and conversion when executing action A given context $\boldsymbol{s}_i$), where $\lambda_{\text{c}} = 0.01$ on the synthetic data, Criteo Data, and commercial product data, respectively. As in most contextual bandit literature (Li et al., 2010; Chu et al., 2011), we set the regularization parameter $\lambda = 1$ in the Euclidean regularization. According to theoretical analysis in Remark 2, we set the batch size as $B = C_B^2 N/d$, set the constant $C_B \approx 25$ and the sketch size $c = 150$ on all the datasets ($B = 1400, 1000, 4000, 1700$ for `synthetic data`, `Criteo-recent`, `Criteo-all`, and `commercial product`). The regularization parameters ω, α in our policy and that in the batch UCB policy were tuned in $[0.2 : +0.2 : 1.2]$. For the SJLT in SPUIR and its variants, sketch size was set as $c = 150$ and the number of block D was selected in $\{1, 2, 4, 6\}$. Except for the rate-scheduled variants of our approaches, the imputation rate γ was selected in $[0.1 : +0.2 : 0.9]$. Besides, the discount parameter η was tuned in $[0.1 : +0.2 : 0.9]$. In the nonlinear variant of our approach SPUIR-Kernel, we selected the dimension of the random features d_{r} in $\{50, 100, 200\}$ and the kernel width of Gaussian kernel in $\{2^{-(i+1)/2}, i = [-12 : 2 : 12]\}$.

C.3 More Experimental Results

For better illustration, in Figure 3 of the manuscript, we omitted the curves of algorithms whose average rewards are 5% lower than the highest reward. Now we provide the curves of all the algorithms in Figure 5.



(a) synthetic data

(b) Criteo-recent

(c) Criteo-all

(d) commercial product

Figure 5: Average rewards of the compared algorithms, the proposed SPUIR and its variants on synthetic dataset, Criteo dataset, and the real commercial product data