# Privacy Preserving Task Push in Spatial Crowdsourcing with Unknown Popularity

Yin Xu, *Student Member, IEEE*, Mingjun Xiao, *Member, IEEE*, Jie Wu, *Fellow, IEEE*, and He Sun

**Abstract**—In this paper, we investigate the privacy-preserving task push problem with unknown popularity in Spatial Crowdsourcing (SC), where the platform needs to select some tasks with unknown popularity and push them to workers. Meanwhile, the preferences of workers and the popularity values of tasks might involve some sensitive information, which should be protected from disclosure. To address these concerns, we propose a Privacy Preserving Auction-based Bandit scheme, termed PPAB. Specifically, on the basis of the Combinatorial Multi-armed Bandit (CMAB) game, we first construct a Differentially Private Auction-based CMAB (DPA-CMAB) model. Under the DPA-CMAB model, we design a privacy-preserving arm-pulling policy based on Diffie-Hellman (DH), Differential Privacy (DP), and upper confidence bound, which includes the DH-based encryption mechanism and the hybrid DP-based protection mechanism. The policy not only can learn the popularity of tasks and make online task push decisions, but also can protect the popularity as well as workers' preferences from being revealed. Meanwhile, we design an auction-based incentive mechanism to determine the payment for each selected task. Furthermore, we conduct an in-depth analysis of the security and online performance of PPAB, and prove that PPAB satisfies some desired properties (i.e., truthfulness, individual rationality, and computational efficiency). Finally, the significant performance of PPAB is confirmed through extensive simulations on the real-world dataset.

**Index Terms**—Spatial crowdsourcing, Combinatorial multi-armed bandit, Privacy preservation, Incentive mechanism.

## 1 Introduction

With the fast-paced development of wireless networks and smart mobile devices in day-to-day life, Spatial Crowdsourcing (SC) has become an attractive paradigm in utilizing the crowd power to address a large-scale of complex tasks [1]–[4]. A typical SC system consists of three parties: requesters, a crowd of workers, and a platform on the cloud. Requesters outsource location-dependent tasks to workers via the platform, and then workers will physically move to specified locations to accomplish the corresponding tasks [5]–[7]. Nowadays, SC applications are ubiquitous, such as information collection (e.g., Waze), transportation (e.g., Uber), and micro-tasks (e.g., gMission).

In this paper, we focus on SC systems based on task push, where requesters submit various tasks with different popularity to the platform, the platform continually pushes these tasks to workers, and workers will select their preferred tasks to perform. Much effort has been devoted to designing such systems in recent years [8]–[12]. However, most of the existing works assume that the popularity of tasks is known by the platform in advance. This assumption is unrealistic in many real-world applications, such as taxi hailing and food/express delivery. We take the taxi-hailing systems as an example. As shown in Fig. 1, many passengers (i.e., requesters) generate massive location-related orders (i.e., SC tasks), and the platform will push orders to some drivers (i.e., workers). These orders can be classified into different types according to their districts (e.g., each colored dotted box represents a task type) or other properties. Drivers may exhibit diverse preferences towards different types of orders. Accordingly, each type of orders has a popularity characteristic, which can be viewed as a statistic on the overall preferences of workers. The orders with high popularity will be pushed to drivers preferentially. Since preferences are generally private information, drivers are reluctant to reveal them to others. Consequently, the popularity of orders is *unknown* to passengers and the platform. Meanwhile, the popularity might also imply passengers' private information (e.g., statistics [13], [14] indicate that the popularity of an order is usually associated with the travel distance, destination distribution, and so on). Then, a significant problem for the platform is how to select appropriate tasks while taking the unknown popularity and privacy into consideration simultaneously. Such a Privacy-preserving Task Push problem with Unknown Popularity (PTP-UP) suffers three major challenges:

First, the PTP-UP problem involves the privacy preserving issue. In the above SC systems, the preferences and popularity might imply the sensitive private information of workers and requesters, respectively. As a result, when workers make decisions to accept or reject the tasks pushed by the platform according to their preferences, they are reluctant to unveil their personal decisions to the platform. Likewise, each requester desires to conceal its own task popularity values from other requesters. However, the platform must depend on the popularity values of tasks to make the decisions of task push, and some requesters might

- *Y. Xu, M. Xiao (Corresponding Author), and He Sun are with the School of Computer Science and Technology / Suzhou Institute for Advanced Research / State Key Laboratory of Cognitive Intelligence, University of Science and Technology of China (USTC), Hefei, China. Email: {xuyin218@mail., xiaomj@, hesun@mail.}ustc.edu.cn*
- *J. Wu is with the Center for Networked Computing, Temple University, 1925 N. 12th Street, Philadelphia, PA 19122. Email: jiewu@temple.edu*

*This research was supported in part by the National Natural Science Foundation of China (NSFC) under Grant Nos. 62172386 and 61872330, the Natural Science Foundation of Jiangsu Province in China under Grant Nos. BK20231212 and BK20191194, and the National Natural Science Foundation of USA under Grant Nos. CNS 2128378, CNS 2107014, CNS 1824440, and CNS 1828363.*


Fig. 1. Illustration of the SC scenario

eavesdrop on others' popularity based on the task selection results. Hence, the first challenge is how to safeguard the privacy of workers' preferences and the popularity of tasks from being disclosed, meanwhile enabling the platform to make the appropriate task push decisions.

Second, the PTP-UP problem entails the privacy-preserving online learning issue. The platform will make online decisions to select and push appropriate tasks to workers according to their popularity. The underlying objective is to align these pushed tasks with workers' preferences as closely as possible, thereby optimizing the reward of the platform. However, the popularity values of tasks are unknown to the platform. Therefore, the platform needs to estimate the popularity value of each task through a series of online learning processes. Then, the second challenge is how to iteratively learn the popularity of each task while concurrently exploiting these learned results to make the best online task push decision. What's more, the platform also needs to protect the learned popularity privacy from being disclosed during this process, making the problem much more challenging.

Third, the PTP-UP problem also involves the incentive issue under the circumstance of privacy protection and unknown popularity. When the platform makes the task push decisions, it also needs to take the rewards into account besides the popularity. Each requester hopes that its task can be pushed to some workers and is willing to provide a payment as an incentive, so as to compete for the opportunity of task push. Hence, the third challenge is how to design an efficient incentive mechanism for the online decisions of task push. In this design, we need to seamlessly integrate the online learning of unknown popularity and the privacy considerations, since they might affect the determination of payments. Additionally, the incentive mechanism also needs to satisfy some vital economic properties, including individual rationality and truthfulness.

So far, a wide spectrum of studies have investigated various task assignment or task push issues in SC systems [8], [10]–[12], [15]–[22], in which several works have addressed the problems arisen from different unknown characteristics (e.g., unknown worker's quality or unknown preferences, etc.) by utilizing the online learning techniques of Multi-armed Bandit (MAB) [18]–[24]. Nevertheless, only a few works discussed the privacy-preserving issues [20]–[22] or incentive mechanism designs [23]–[28] together with the MAB-based online learning. For example, the authors developed two differentially private arm-pulling algorithms to learn the unknown workers' qualities in SC systems [20]. The work in [24] learned the uncertainties over time with the help of an auction to incentivize consumers to reduce electricity. However, none of these approaches take the online learning on unknown characteristics, privacy protection, and incentives into consideration at the same time. Consequently, they are not applicable to our system for dealing with the PTP-UP problem.

In this paper, we propose a Privacy-Preserving Auction-based Bandit (PPAB) scheme to address our SC challenges. Specifically, we design a novel Differentially Private Auction-based Combinatorial MAB (DPA-CMAB) model, in which each task is seen as an arm, its popularity is regarded as the corresponding reward, and selecting tasks is equivalent to pulling arms. By means of this model, the platform can learn the popularity of tasks and make online task push decisions. Under the DPA-CMAB model, we then design a privacy-preserving arm-pulling policy based on Diffie-Hellman (DH), Differential Privacy (DP), and Upper Confidence Bound (UCB), which can protect the preferences of workers and the popularity information from being disclosed. Next, we design an auction-based incentive mechanism to determine the payment of each selected task. Meanwhile, we analyze the regret and prove that PPAB satisfies DP and some critical economic properties. Finally, we conduct extensive simulations on the real-world trace to corroborate the significant performance of PPAB.

To sum up, our multi-fold contributions are as follows.

- We introduce the PTP-UP problem for SC systems based on task push and propose the PPAB scheme to solve it. Unlike existing studies, PPAB takes online learning, privacy protection, and incentives simultaneously, so as to form a reliable SC system.
- We formulate the PTP-UP problem as a novel DPA-CMAB model. Under DPA-CMAB, we design the DH-based encryption mechanism and the hybrid DP-based protection mechanism for PPAB to protect the popularity as well as workers' preferences from being revealed. Moreover, we prove that PPAB meets the DP property.
- We design a privacy-preserving arm-pulling policy based on UCB for the proposed PPAB scheme. This policy can learn the unknown popularity of each task to facilitate online task push decisions without privacy concerns. Furthermore, we conduct an in-depth analysis of the online performance of PPAB, deriving an upper bound on regret (i.e., the expected loss in rewards).
- An incentive mechanism in PPAB is designed to determine the payments corresponding to the pushed tasks. In addition, we prove that the PPAB scheme guarantees some essential properties such as truthfulness, individual rationality, and computational efficiency.

The remainder of the paper is organized as follows. In Sec. 2, we introduce the system model, security model, and the problem formulation. The detailed scheme design of PPAB is elaborated in Sec. 3. In Sec. 4, we present the theoretical analysis. Sec. 5 shows the simulations and evaluations in great detail. We review the related works in Sec. 6. After discussing open issues and the potential future research directions in Sec. 7, we conclude the paper in Sec. 8. Note that some proofs are moved to the Appendix.

TABLE 1
Description of major notations

| Variable | Description |
|---|---|
| $i, j, t$ | the indexes of task, worker, and period. |
| $M, T$ | the number of tasks/requesters; the total periods. |
| $K, \mathcal{M}^t$ | the number and set of recruited workers. |
| $\mu_i, \mu_i^t$ | the expected popularity; the estimated popularity. |
| $x_{i,j}^t$ | the acceptance decision of $w_{i,j}^t$ about task $i$. |
| $\mathcal{W}_i^t, \mathcal{X}_i^t$ | the group of workers for $i$ and the decision vector. |
| $\vartheta_i, b_i, p_i$ | the true valuation, the bid, and the payment of $i$. |
| $A_i^t, D$ | the staleness of task $i$; the staleness constraint. |
| $\mathrm{U}_i^t, \overline{\mathrm{U}}_i^t$ | the final popularity; the accumulative popularity. |
| $\widetilde{\mathrm{U}}_i^t$ | the confused accumulative reward in period $t$. |
| $\epsilon, \varphi_i^t$ | the privacy budget and the confidence bound. |
| $n_i^t$ | the number of $i$ being selected until period $t$. |
| $\chi_i^t, \mathcal{M}^*$ | the counter of task $i$ in period $t$; the optimal selected tasks based on optimal policy. |
| $\widehat{\mathrm{U}}_i^t$ | the DP-based combinatorial UCB index of task $i$. |
| $\Psi_i^t$ | the utility of the requester $r_i$ who requests task $i$. |
| $\phi^t, \delta$ | the upper bound of noise and a small number. |

## 2 MODEL AND PROBLEM

### 2.1 System Overview

We consider a typical SC system, including a platform, lots of workers (i.e., a sizeable worker pool), and some task requesters. With the aid of the task push service provided by the platform, requesters can recruit workers to perform their tasks. We give some related definitions as follows:

**Definition 1** (Tasks with unknown popularity)**.** Let $\mathcal{M} = \{1, ..., i, ..., M\}$ be a set of tasks proposed by the corresponding requesters $\mathcal{R} = \{r_1, ..., r_i, ..., r_M\}$. We consider that the types of these tasks are heterogeneous. For example, the locations specified by various tasks belong to different districts, and each requester only publishes one task. Additionally, these long-term tasks operate in a time-slotted manner, with the entire process divided into $T$ periods.

In each period $t$, we use $\mu_i^t \in [0, 1]$ to denote the estimated popularity of task $i$, which can be regarded as the probability that task $i$ is accepted by any worker. In this paper, we adopt a simple numerical value to model the overall preferences of workers, which can be easily extended to some complex popularity models. The values of $\{\mu_i^t | i \in \mathcal{M}, t \in [1, T]\}$ follow an independent and identically unknown distribution with an unknown expectation $\mu_i$, i.e., $\mu_i \triangleq \mathbb{E}(\mu_i^t)$. Since the distribution and mean are unknown, we thus say that the popularity of task $i$ is unknown. The requester $r_i$ will pay the platform if he/she receives a completion result of task $i$ from a worker. Let $\vartheta_i$ denote the true valuation of task $i$ for each completion and let $b_i$ be the value claimed by requester $r_i$. Here, $b_i$ is called "bid" in auctions. Since many requesters are selfish and rational, they may attempt to increase their utilities (i.e., net profit) by taking some strategic behaviors, e.g., requester $r_i$ may strategically misreport its true valuation (i.e., $b_i \neq \vartheta_i$). Hence, $b_i$ will not always be equal to $\vartheta_i$.

**Definition 2** (Platform)**.** As a profit-making intermediate agent, the platform provides a task push service for requesters. In order to reap more profits, the platform prefers to select the tasks with the popularity as high as possible. Considering the dynamic nature of workers, the platform first randomly picks $N$ workers from the worker pool in each period. Subsequently, the platform decides which tasks to proactively push to these $N$ workers. If a worker is willing to accept a task and return its completion result to the corresponding requester, the platform will charge a monetary reward from the requester.

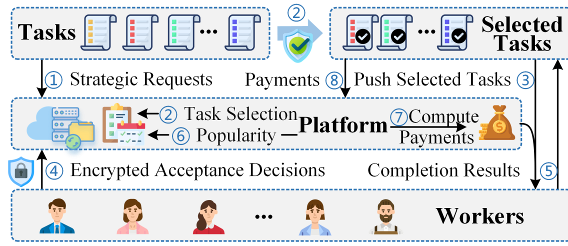



Fig. 2. The workflow in the $t$-th period

**Definition 3** (Workers)**.** The system contains an unknown number of workers, which may have different preferences for diverse tasks. After picking out $N$ workers at random in the $t$-th period, the platform pushes each selected task $i$ to the $N$ workers, denoted by $\mathcal{W}_i^t = \{w_{i,1}^t, ..., w_{i,j}^t, ..., w_{i,N}^t\}$. Let $\mathcal{X}_i^t = \{x_{i,1}^t, ..., x_{i,j}^t, ..., x_{i,N}^t\} \in \{0, 1\}^N$ be a binary vector to denote the acceptance decisions of these workers, where $x_{i,j}^t \in \{0, 1\}$. Specifically, $x_{i,j}^t = 1$ indicates that worker $w_{i,j}^t$ accepts and completes task $i$ while $x_{i,j}^t = 0$ means rejection.

**Definition 4** (Staleness of each task)**.** The staleness of each task is the time elapsed since the task is pushed by the platform. At time $t$, with $E_i^t$ as the time of the most recent pushing event of task $i$, we use $A_i^t = t - E_i^t$ to define the staleness of task $i$. To appeal to more requesters, the platform sets a peak staleness constraint $D$ for each task. This constraint can prevent some tasks from not being performed for a long time, thereby ensuring the information freshness of tasks. Since $A_i^t$ is related to the time span, $D$ can be set as an increasing function of the total period $T$, i.e., $D = f(T)$.

**System workflow:** We illustrate the detailed workflow in Fig. 2. First, requesters publish their tasks with bids to the platform (Step 1). Then, the platform selects some tasks with as high popularity as possible, which will be conducted periodically until the given time is exhausted (Step 2). During the task selection process, some strategic requesters might misreport their valuations and the task popularity of each requester should be protected from being revealed to other requesters. Next, each selected task is pushed by the platform to a group of random workers (Step 3). Each worker who received the task recommendation makes an acceptance decision (i.e., accept or reject the task). In order to protect the privacy of each worker's precise preference, each worker uploads an encrypted acceptance decision rather than its original decision (Step 4). After that, the worker with an affirmative reply performs the task and returns its completion result to the corresponding requester so as to attain a remuneration (Step 5). Meanwhile, the platform updates the popularity value of each selected task according to encrypted acceptance decisions, and determines its payment (Steps 6 and 7). Finally, each selected requester pays the designated monetary reward (the payment multiplied by the number of completion results) to the platform (Step 8).

For ease of reference, Table 1 lists the major notations and descriptions used throughout this paper.

### 2.2 Security Model

In this paper, we consider a typical security model, i.e., the widely-adopted semi-honest model [29]. Under the model, our focus lies in protecting the privacy of workers and requesters. Specifically, each worker hopes to prevent its acceptance decision from being revealed to the platform. After estimating the task popularity of each requester based on workers' decisions, the platform also needs to protect the popularity values from being leaked to other requesters. To better illuminate these privacy concerns, we delineate the potential behaviors of both the platform and requesters.

1) *Platform:* The platform is semi-honest. That is, on one hand, the platform dutifully adheres to the designed scheme for completing the task push process, showing the honest aspect. On the other hand, the platform may also try to infer each worker's preference from the received data, reflecting the dishonest aspect. Like in [30]–[32], we do not consider the case that the platform may collude with requesters and workers as well as other types of potential attacks (e.g., hacker attacks).
2) *Requester:* Each requester will also follow the whole process to benefit from participating in the system, signifying the honest aspect. On the other hand, the requester could potentially eavesdrop on other requesters' popularity values based on the task selection outcomes and leverage the statistics to infer some sensitive information, showcasing the dishonest aspect.

DP techniques and encryption approaches have been widely adopted in both academia and industry [33]–[36]. Since the homomorphic encryption/decryption technology may require a trusted third party or involve some time-consuming operations, we take advantage of the Diffie-Hellman (DH) protocol to ensure that each worker's acceptance decision is not leaked to the platform. The security of the DH protocol can be guaranteed by:

**Definition 5** (Diffie-Hellman protocol, DH [33])**.** All workers agree on a large prime number and an element $g$ that generates a cyclic subgroup. The key pair of each worker $w$ is created as $(g^{Key_w}, Key_w)$, where $g^{Key_w}$ is the public key and $Key_w$ is the secret key. Then, given the public key $g^{Key_{w'}}$ of worker $w'$, $w$ can determine the shared secret key between $w$ and $w'$ by calculating $(g^{Key_{w'}})^{Key_w}$.

Since the platform and workers have access to tasks, we adopt DP to protect the popularity sequence of each requester from being revealed to other requesters, except for the platform and workers. The DP-based security needs to satisfy the following property.

**Definition 6** (Differential Privacy, DP [34])**.** A randomized mechanism $\mathcal{A}$ has $\epsilon$-DP if for any two input sets $D_1$ and $D_2$ differing on at most one element, and for any set of outcomes $O \subseteq Range(\mathcal{A})$, we have $Pr[\mathcal{A}(D_1) \in O] \leq exp(\epsilon) \times Pr[\mathcal{A}(D_2) \in O]$. $\epsilon > 0$ is the privacy budget: the smaller $\epsilon$, the stricter protection and lower data availability.

### 2.3 Problem Formulation

We introduce the privacy-preserving task selection problem. The aim is to maximize the sum of popularity of all tasks over all periods. Let $\mathcal{M}^t \subset \mathcal{M}$ denote the selected tasks in period $t$. Moreover, we define the final popularity of a task according to $\mathcal{M}^t$ in period $t$, denoted by $\mathrm{U}_i^t$. Specifically, if $i \in \mathcal{M}^t$, we set $\mathrm{U}_i^t = \mu_i^t$; otherwise, there is $\mathrm{U}_i^t = 0$.

Our objective is to determine $\{\mathcal{M}^1, ..., \mathcal{M}^t, ..., \mathcal{M}^T\}$ in each period, so that the total expected popularity of all tasks can be maximized under some constraints. Thus, the privacy-preserving task selection problem is:

$$Maximize: \quad \mathbb{E}[\textstyle\sum_{t=1}^{T}\sum_{i=1}^{M}\mathrm{U}_i^t] \quad (1)$$

$$Subject\ to: \quad A_i^t \leq D, \forall i \in \mathcal{N}, \forall t \in [1, T] \quad (2)$$

$$|\mathcal{M}^t| = K, \forall t \in [2, T] \quad (3)$$

$$Security: \quad \text{Protect } \mathcal{X}_i^t \text{ and } \mathrm{U}_i^t, \forall i \in \mathcal{N}, \forall t \in [1, T] \quad (4)$$

Here, Eq. (2) indicates the staleness constraint. Eq. (3) represents the quantity constraint, i.e., there are $K$ tasks selected in each period. Note that $K \in \{1, ..., M\}$ is a preset value decided by the platform based on its capacity. Eq. (4) means that the values of $\mathcal{X}_i^t$ and $\mathrm{U}_i^t$ should be protected.

Finally, the payment determination problem aims to compute the payment (denoted by $p_i^t$) of the requester $r_i$. Auction is one of the most efficient ways to solve this problem, and the auction should meet the following properties.

**Definition 7** (Truthfulness [37])**.** We let $b$ be an arbitrary bid for task $i$ that is selected, and $p_i^t(b)$ is the corresponding payment determined by the auction algorithm. Then, if there is $\vartheta_i^t - p_i^t(b) \leq \vartheta_i^t - p_i^t$, the auction algorithm is truthful. Here, $p_i^t$ is the payment when requester $r_i$ claims its true value as its bid, i.e., $p_i^t = p_i^t(\vartheta_i^t)$. Moreover, we define the utility of requester $r_i$ as $\Psi_i^t = \vartheta_i^t - p_i^t(b)$.

**Definition 8** (Individual Rationality)**.** If requester $r_i$ with the winning bid $b_i$ has a nonnegative payoff, i.e., $\vartheta_i^t - p_i^t(b_i) \geq 0$, then the auction algorithm holds individual rationality.

**Definition 9** (Computational Efficiency [37])**.** If an auction algorithm can generate results and terminate in a polynomial time, the algorithm is computationally efficient.

## 3 Scheme Design

In this section, we propose a Privacy Preserving Auction-based Bandit scheme, called PPAB. We first formulate the PTP-UP problem as a novel MAB model that comprehensively integrates privacy preservation and incentives. Under this model, the PPAB scheme adopts a privacy-preserving arm-pulling policy to select appropriate tasks, incorporating a DH-based encryption mechanism to safeguard the privacy of workers' preferences and a hybrid DP-based protection mechanism to shield the popularity information of tasks. Additionally, PPAB synergistically combines the auction into CMAB to determine the payments for chosen requesters. To offer a more in-depth understanding, we provide a detailed exposition of our model and the two mechanisms, subsequently followed by a comprehensive outline of the scheme and an illustrative example.

### 3.1 Differentially Private Auction-based CMAB model

The task push process with unknown popularity is actually an online learning and sequential decision-making problem. CMAB [19], [38] is a reinforcement learning model which is a promising answer to cope with the problem. Basically, a typical CMAB model is comprised of a slot machine with multiple arms. Pulling an arm will earn a reward drawn from an unknown distribution. A player will pull a set of arms (called a super arm) together period by period based on a bandit policy to learn the distribution, so as to maximize the cumulative reward.

By taking privacy protection and incentives into consideration, we construct a Differentially Private Auction-based CMAB (DPA-CMAB) model to address the PTP-UP problem. In this model, each task is treated as an arm, the popularity of each task is regarded as the corresponding reward, and selecting $K$ tasks is equivalent to pulling a super arm. Recall that, the number of selected tasks, $K \in \{1, ..., M\}$, is a preset value decided by the platform based on its capacity. During each period, the popularity of each selected task $i$ is estimated by computing $\mu_i^t = \sum_{j=1}^{N} x_{i,j}^t / N$, allowing the platform to learn the popularity of the $K$ selected tasks. Crucially, based on the DH-based encryption mechanism, the platform only receives the sum of decisions $\sum_{j=1}^{N} x_{i,j}^t$ of task $i$ without knowing each worker $j$'s original decision $x_{i,j}^t$. At the same time, the popularity values of tasks, i.e., $\{\mu_i^t | i \in \mathcal{M}^t, t \in [1, T]\}$, which contain rich sensitive information, can also be safeguarded via the DP-based protection mechanism. Besides, it is a remarkable fact that arms are not mindless machines but subjectively conscious. In other words, each arm might strategically report its value to manipulate the payment determination process. Therefore, the DPA-CMAB model incorporates auction techniques to incentivize each arm to bid truthfully.

### 3.2 Diffie-Hellman-based Encryption Mechanism

When a selected task $i$ is pushed to $N$ random workers $\mathcal{W}_i^t = \{w_{i,1}^t, ..., w_{i,j}^t, ..., w_{i,N}^t\}$, the platform will attach these workers' IDs (denoted by $\{ID_{w_{i,j}^t} | w_{i,j}^t \in \mathcal{W}_i^t\}$). Suppose that all workers' IDs in our system are ordered and there exists a key distribution center. The center needs to create a pair with a public key and a private key for each worker, and broadcast all public keys. We use $(g^{Key_{w_{i,j}^t}}, Key_{w_{i,j}^t})$ to denote the key pair of worker $w_{i,j}^t$. In order to conceal the original decision $x_{i,j}^t$, each worker $w_{i,j}^t \in \mathcal{W}_i^t$ encrypts $x_{i,j}^t$ using the following steps:

Step 1: worker $w_{i,j}^t$ computes a series of shared secret numbers based on the DH protocol. That is, for any other worker $w_{i,q}^t \in \mathcal{W}_i^t$, the shared secret number between $w_{i,j}^t$ and $w_{i,q}^t$, denoted by $\Phi(w_{i,j}^t, w_{i,q}^t)$, can be computed by

$$\Phi(w_{i,j}^t, w_{i,q}^t) = (g^{Key_{w_{i,q}^t}})^{Key_{w_{i,j}^t}}, \forall w_{i,q}^t \in \mathcal{W}_i^t, q \neq j. \tag{5}$$

Step 2: worker $w_{i,j}^t$ takes $\Phi(w_{i,j}^t, w_{i,q}^t)$ as the seed of Cryptographically Secure Pseudo Random Number Generator (CSPRNG), and then can generate a random number (a.k.a., salt): $\tilde{\Phi}(w_{i,j}^t, w_{i,q}^t) = \text{CSPRNG}(\Phi(w_{i,j}^t, w_{i,q}^t))$.

Step 3: worker $w_{i,j}^t$ compares its own ID with other workers' IDs. If $ID_{w_{i,q}^t} > ID_{w_{i,j}^t}$, $w_{i,j}^t$ will add the salt $\tilde{\Phi}(w_{i,j}^t, w_{i,q}^t)$ into $x_{i,j}^t$; otherwise, the worker will subtract the salt. Thus, $x_{i,j}^t$ can be encrypted as $\tilde{x}_{i,j}^t$, i.e.,

$$\tilde{x}_{i,j}^t = x_{i,j}^t + \sum_{w_{i,q}^t \in \mathcal{W}_i^t : ID_{w_{i,q}^t} > ID_{w_{i,j}^t}} \tilde{\Phi}(w_{i,j}^t, w_{i,q}^t) - \sum_{w_{i,q}^t \in \mathcal{W}_i^t : ID_{w_{i,q}^t} < ID_{w_{i,j}^t}} \tilde{\Phi}(w_{i,j}^t, w_{i,q}^t). \tag{6}$$

By following the above steps, each worker $w_{i,j}^t$ uploads an encrypted decision $\tilde{x}_{i,j}^t$ to the platform. Finally, the platform accumulates all encrypted decisions to attain the aggregated decision: $\sum_{j=1}^{N} x_{i,j}^t = \sum_{j=1}^{N} \tilde{x}_{i,j}^t$ and then evaluates the popularity of task $i$: $\mu_i^t = \sum_{j=1}^{N} x_{i,j}^t / N$.

### 3.3 Hybrid DP-based Protection Mechanism

For the task selection process, the platform will execute a series of arm-pulling operations periodically under the DPA-CMAB model. In each period, the platform will opt for a group of tasks according to the accumulative reward of each task. If a task $i$ is selected, the final popularity $\mathrm{U}_i^t = \mu_i^t$ will be added into the accumulative reward of task $i$; otherwise, the final popularity $\mathrm{U}_i^t$ is equal to zero, i.e., the accumulative reward of task $i$ remains unchanged. If the platform directly uses the accumulative rewards, the selection results might cause the popularity of some tasks to be leaked. Thus, we leverage the hybrid DP-based protection mechanism to shield the popularity of any task from being exposed to other requesters.

Specifically, the platform injects a hybrid Laplace noise into the accumulative reward of each task, even though the popularity of the task in the current period is equal to zero. We take task $i$ as a concrete example to illustrate the mechanism. In the $(t+1)$-th period, the platform has recorded the final popularity of task $i$ from the period 1 to the period $t$, i.e., $\{\mathrm{U}_i^1, \mathrm{U}_i^2, ..., \mathrm{U}_i^t\}$. Before the task selection process, the platform first computes the accumulative reward, denoted by $\overline{\mathrm{U}}_i^t = \sum_{\tau=1}^{t} \mathrm{U}_i^\tau$. Then, the platform perturbs the accumulative reward $\overline{\mathrm{U}}_i^t$ into a confused accumulative reward (denoted by $\widetilde{\mathrm{U}}_i^t$) according to the following equation:

$$\widetilde{\mathrm{U}}_i^t = \overline{\mathrm{U}}_i^t + (\varrho^t - 1) Lap(\frac{2M\lfloor \log t \rfloor}{\epsilon}) + Lap(\frac{2M}{\epsilon}). \tag{7}$$

Here, $Lap(\frac{2M\lfloor \log t \rfloor}{\epsilon})$ is the Laplace noise, which is actually drawn from a Laplace distribution with mean zero and scale $\frac{2M\lfloor \log t \rfloor}{\epsilon}$. The corresponding probability density function is denoted by $f(x) = \frac{\epsilon}{4M\lfloor \log t \rfloor} exp(-\frac{\epsilon}{2M\lfloor \log t \rfloor}|x|)$. Similarly, $Lap(\frac{2M}{\epsilon})$ is also the Laplace noise. $\varrho^t$ is the number of 1's in the binary expression of $t$. In this way, the platform will select $K$ tasks by virtue of the confused accumulative reward $\widetilde{\mathrm{U}}_i^t$, so that the value of $\overline{\mathrm{U}}_i^t$ will not be known to other requesters.

### 3.4 A Privacy Preserving Auction-based Bandit Scheme

Under the DPA-CMAB model, our PTP-UP problem can be divided into two key sub-problems in each period: a privacy-preserving task selection problem and a payment determination problem. For the former, we aim to maximize the total popularity while protecting the privacy of workers' preferences and task popularity. For the latter, we need an incentive mechanism to determine the appropriate payments for selected tasks, so that each requester will tell the truth. Actually, the two sub-problems are inextricably linked, because only when requesters are truthful can the platform carry out the efficient task selection to achieve the maximization of total popularity.

To balance the trade-off between exploration (i.e., trying some sub-optimal tasks to find the potential optimal tasks) and exploitation (i.e., choosing the current best tasks based on the learned results) in the privacy-preserving task selection problem, we first design a DP-based combinatorial UCB index, taking privacy into consideration. More specifically, we introduce $n_i^t$ for $i \in \mathcal{M}, t \in [1, T]$ to record the number of times that task $i$'s popularity has been learned. Recall that we use $\overline{\mathrm{U}}_i^t$ to denote the accumulative popularity of task $i$. After the workers return the encrypted decisions, the values of $n_i^t$, $A_i^t$, and $\overline{\mathrm{U}}_i^t$ at the end of $t$ are updated as follows.

$$n_i^t = \begin{cases} n_i^{t-1}+1, & if\ i \in \mathcal{M}^t \\ n_i^{t-1}, & if\ i \notin \mathcal{M}^t \end{cases} \quad A_i^t = \begin{cases} 0, & i \text{ is pushed} \\ A_i^{t-1}+1, & \text{Otherwise} \end{cases} \tag{8}$$

$$\overline{\mathrm{U}}_i^t = \begin{cases} \overline{\mathrm{U}}_i^{t-1} + \mu_i^t = \overline{\mathrm{U}}_i^{t-1} + \sum_{j=1}^{N} \frac{\tilde{x}_{i,j}^t}{N}, & if\ i \in \mathcal{M}^t \\ \overline{\mathrm{U}}_i^{t-1}, & if\ i \notin \mathcal{M}^t \end{cases} \tag{9}$$

Now, we bring in the DP-based combinatorial UCB index, denoted by $\widehat{\mathrm{U}}_i^t$, to signify the perturbed average reward for task selection. $\widehat{\mathrm{U}}_i^t$ consists of the empirical popularity (indicating the learned knowledge from observed results), a confidence bound (indicating the uncertainty of empiricism), and the perturbation term (indicating the Laplace noise), which is shown as follows:

$$\widehat{\mathrm{U}}_i^t = \frac{\widetilde{\mathrm{U}}_i^t}{n_i^t} + \varphi_i^t + \frac{\phi^t}{n_i^t},\ \varphi_i^t = \sqrt{(K+1)\ln(\sum_{j=1}^{M} n_j^t)/n_i^t}, \tag{10}$$

where $\widetilde{\mathrm{U}}_i^t$ is derived by adding the hybrid Laplace noise into $\overline{\mathrm{U}}_i^t$ based on Eq. (7). $\phi^t = \frac{2\sqrt{2}}{\epsilon} log(\frac{4}{\delta})(\log t + 1)$ means an upper bound on the total Laplace noise with a high probability $1-\delta$. The second term $\varphi_i^t$ is the upper confidence bound, which takes an optimism principle in the face of uncertainty. With the increase of $n_i^t$, $\varphi_i^t$ decreases rapidly. Therefore, tasks that have been selected less frequently may have more opportunities to be pushed in the next period.

Next, we explain how to conduct the task selection and payment determination processes. In the initial period (i.e., $t = 1$), the platform will select all tasks (i.e., $\mathcal{M}^1 = \mathcal{M}$) to learn the initial values of $\overline{\mathrm{U}}_i^t, \widetilde{\mathrm{U}}_i^t, n_i^t$, and $\widehat{\mathrm{U}}_i^t$. Subsequently, the platform will always select $K$ tasks in each period. More specifically, in the $t$-th period, the platform sorts all tasks in a non-increasing order of $b_i \widehat{\mathrm{U}}_i^{t-1}$. Since $\vartheta_i \mu_i N$ can be seen as the expected valuation of task $i$ for per push, we adopt $b_i \widehat{\mathrm{U}}_i^{t-1}$ as the selection criterion. According to this order, the platform selects the top $K$ tasks to form the winning set $\mathcal{M}^t$, and then randomly picks $N$ workers to construct the set $\mathcal{W}_i^t$ for task recommendation. After recommending each selected task in $\mathcal{M}^t$ to workers in $\mathcal{W}_i^t$, the platform proceeds to determine the payments for the corresponding requesters. Based on the auction theory, we design an incentive mechanism to find out the critical payment for each selected task in the $t$-th period, i.e.,

$$p_i^t = \max\{b_{K+1}\widehat{\mathrm{U}}_{K+1}^{t-1}/\widehat{\mathrm{U}}_i^{t-1}, \vartheta_{min}\}. \tag{11}$$

Here, we consider that the valuation $\vartheta_i$ ($\forall i \in \mathcal{M}$) and the bid $b_i$ ($\forall i \in \mathcal{M}$) belong to an acknowledged range $[\vartheta_{min}, \vartheta_{max}]$. The critical payment is $(b_{K+1}\widehat{\mathrm{U}}_{K+1}^{t-1})/\widehat{\mathrm{U}}_i^{t-1}$, and the function $\max\{\cdot\}$ can be used for restricting the minimal payment. In other words, the critical payment represents that a selected task will not win in the auction if its bid is larger than the critical payment. In addition, the payment will not be lower than $\vartheta_{min}$, which can ensure the property of individual rationality.

**Algorithm 1:** The Proposed PPAB Scheme

**Input:** $\mathcal{M}, \{b_i\}, T, N, K, \vartheta_{min}, \epsilon$, and $\delta$
**Output:** $\{\mathcal{M}^t | t \in [1, T]\}$ and $\{p_i^t | i \in \mathcal{M}, t \in [1, T]\}$

```
Initialization: n_i^t = 0, A_i^t = 0, Ū_i^t = 0, Ũ_i^t = 0, Û_i^t = 0,
    p_i^t = 0, M^t = ∅, ∀i∈M, t∈[1,T];
Initial exploration phase:
for each task i ∈ M and t = 1 do
    The platform randomly picks N workers, pushes
      task i to the worker set W_i^1 = {w_{i,1}^1, ..., w_{i,j}^1,
      ..., w_{i,N}^1}, receives a series of encrypted decisions
      {x̃_{i,j}^1 | j ∈ W_i^1} from the worker set W_i^1, computes
      μ_i^1 = Σ_{j=1}^N x̃_{i,j}^1 / N, and determines p_i^1 = ϑ_min;
    Update the parameters: Ū_i^1, Ũ_i^1, n_i^1, and Û_i^1;
Privacy-preserving task selection and incentives:
while each period t < T do
    Sort the tasks according to the value b_i Û_i^t:
    b_1 Û_1^t ≥ b_2 Û_2^t ≥ ... ≥ b_M Û_M^t; t ⇐ t + 1;
    Select the top K tasks as winners, denoted as M^t;
    Pick a group of workers at random, denoted as W_i^t;
    for each task i ∈ M do
        if the task i ∈ M^t then
            The platform pushes task i to the selected
              workers in W_i^t and determines the pay-
              ment for the corresponding requester r_i,
              i.e., p_i^t = max{ b_{K+1} Û_{K+1}^{t-1} / Û_i^{t-1}, ϑ_min };
            Receive encrypted decisions {x̃_{i,j}^t | j ∈ W_i^t};
            Compute μ_i^t = (Σ_{j=1}^N x̃_{i,j}^t)/N;
        Update Ū_i^t and perturb it to get Ũ_i^t (i.e., Eq. (7));
        Update the parameters: n_i^t, Û_i^t, and A_i^t accord-
          ing to Eqs. (8), (9), and (10);
        if the task i ∉ M^t and A_i^t > D then
            The platform pushes task i to W_i^t and deter-
              mines the payment for r_i: p_i^t = ϑ_min;
            Reset the value of A_i^t, i.e., A_i^t = 0;
Return {M^t | t ∈ [1,T]} and {p_i^t | i ∈ M, t ∈ [1,T]}.
```

Then, the platform checks whether each task meets the peak staleness constraint $D$. Since $D$ is a preset parameter, we apply a monotonic increasing function $D = T/\ln(T+2)$ to determine the value of $D$ in this paper, which will not affect the correctness of the algorithm design. Based on Eq. (8), the staleness values of all tasks will be updated after the task selection process. If the staleness of task $i$ exceeds the preset constraint (i.e., $A_i^t > D, \forall i \notin \mathcal{M}^t$), the platform will push task $i$ to ensure its freshness. In this way, each task will not be completely ignored by the platform. Meanwhile, task $i$ needs to give the lowest payment $\vartheta_{min}$ to the platform for each received completion result.

Finally, the workers in $\mathcal{W}_i^t$ upload their encrypted decisions (i.e., $\{\tilde{x}_{i,j}^t | \forall i \in \mathcal{M}^t, j \in \mathcal{W}_i^t\}$), and the platform evaluates the popularity of each task $i$ using $\mu_i^t = \sum_{j=1}^{N} \tilde{x}_{i,j}^t / N$. Therefore, some related parameters (i.e., $\overline{\mathrm{U}}_i^t, \widetilde{\mathrm{U}}_i^t, n_i^t$, and $\widehat{\mathrm{U}}_i^t$) can be updated according to Eqs. (8), (9), and (10). The unknown task selection and the payment determination processes will continue until the given time is exhausted (i.e., $t=T$).

**Detailed Algorithm.** According to the above solution, the proposed PPAB scheme is elaborated in Algorithm 1. At the beginning, we initialize some parameters and sets, e.g., $n_i^t=0$, $A_i^t=0$, $\overline{\mathrm{U}}_i^t=\widetilde{\mathrm{U}}_i^t=\widehat{\mathrm{U}}_i^t=0$, $p_i^t=0$, and $\mathcal{M}^t=\emptyset$ (Step 1). Then, the initial exploration phase begins (i.e., $t=1$). The platform will tentatively select all tasks to learn and estimate their popularity values (Steps 2-4). At the end of the first period, we can compute the popularity of each task $\mu_i^1$ and then update $\overline{\mathrm{U}}_i^1, \widetilde{\mathrm{U}}_i^1, n_i^1$, and $\widehat{\mathrm{U}}_i^1$ (Step 5).

After the initial exploration, we use the learned popularity information to select tasks and determine the corresponding payments (i.e., exploitation). Specifically, the platform first sorts the tasks in a non-increasing order of $b_i\widehat{\mathrm{U}}_i^t$ and updates the index of the current period (Steps 8-9). According to the order, the top $K$ tasks will be chosen as winners to constitute the set $\mathcal{M}^t$, and then the platform will choose a worker set $\mathcal{W}_i^t$ at random (Steps 10-11). Next, we judge whether a task belongs to $\mathcal{M}^t$ or not (Steps 12-16). If task $i$ is selected in the current period, the platform will push the task $i$ to the selected workers in $\mathcal{W}_i^t$ and determine the payment $p_i^t$ for requester $r_i$ (Step 14). After the platform receives the set of workers' encrypted decisions $\{\tilde{x}_{i,j}^t\}$ and computes $\mu_i^t=(\sum_{j=1}^N \tilde{x}_{i,j}^t)/N$, the related parameters of all tasks need to be updated according to Eqs. (7), (8), (9), and (10) (i.e., exploration). Note that, the DP-based combinatorial UCB index $\widehat{\mathrm{U}}_i^t$ uses the confused accumulative reward $\widetilde{\mathrm{U}}_i^t$ instead of $\overline{\mathrm{U}}_i^t$. To control the staleness of each task, the platform pushes some "old" tasks (i.e., $A_i^t>D$), updates the value of $A_i^t$, and charges the lowest payment $\vartheta_{min}$ (Steps 19-21). The PPAB scheme alternates the exploitation process and the exploration process until $t=T$.

### 3.5 A Straightforward Example

For a better understanding of the PPAB scheme, we provide an example. There are three tasks requested by $\{r_1, r_2, r_3\}$, respectively. In each round, the platform can push two tasks to suitable workers (i.e., $M=3$, $K=2$, and $N=30$). We assume that the popularity of each task follows the Gaussian distribution, and the expected popularity of tasks is set as $\{\mu_1=0.3, \mu_2=0.6, \mu_3=0.8\}$. Meanwhile, three requesters submit their bids $\{b_1=4, b_2=6, b_3=5\}$ to the platform, respectively. The information of tasks and the number of accepted workers are illustrated in Fig. 3.

At the beginning, the platform selects all tasks to learn their initial popularity according to $\mu_i^t=(\sum_{j=1}^N \tilde{x}_{i,j}^t)/N$: $\mu_1^1=9/30=0.3$, $\mu_2^1=15/30=0.5$, $\mu_3^1=27/30=0.9$, and then updates $\widehat{\mathrm{U}}_i^t$: $\widehat{\mathrm{U}}_1^1=0.3+\sqrt{3*\ln(3)}\approx 2.12$, $\widehat{\mathrm{U}}_2^1\approx 2.32$, $\widehat{\mathrm{U}}_3^1\approx 2.72$. According to the non-increasing order: $b_2*\widehat{\mathrm{U}}_2^1=6*2.32>b_3*\widehat{\mathrm{U}}_3^1>b_1*\widehat{\mathrm{U}}_1^1$, task-2 and task-3 will be selected as winners in the 2nd period. Correspondingly, $\widehat{\mathrm{U}}_i^t$ will be updated as: $\widehat{\mathrm{U}}_1^2=0.3+\sqrt{3*\ln(5)}\approx 2.5$, $\widehat{\mathrm{U}}_2^2=\frac{0.5+21/30}{1+1}+\sqrt{\frac{3*\ln(5)}{2}}\approx 2.15$, $\widehat{\mathrm{U}}_3^2=\frac{0.9+24/30}{1+1}+\sqrt{\frac{3*\ln(5)}{2}}\approx 2.4$. Owing to the non-increasing order $b_2*\widehat{\mathrm{U}}_2^2>b_3*\widehat{\mathrm{U}}_3^2>b_1*\widehat{\mathrm{U}}_1^2$, task-2 and task-3 will be winners in the third period. Repeating the above steps until $t=T$, the whole process would be over completely. Note that the platform provides an extra push service for task-1 due to the staleness constraint in the ninth period. Here, we remove the encryption mechanism and the

| (Task 1, Task 2, Task 3): | Expected Popularity = (0.3, 0.6, 0.8) | Bid = (4, 6, 5) |
|---|---|---|

| Sum of $x_{i,j}^t$ | Period 1 | Period 2 | Period 3 | Period 4 | Period 5 | Period 6 | Period 7 | Period 8 | Period 9 | Period 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Task 1 | 9 | | | | 9 | | | | 9 | 12 |
| Task 2 | 15 | 21 | 15 | 15 | 21 | 15 | 18 | 15 | 18 | 18 |
| Task 3 | 27 | 24 | 24 | 21 | | 27 | 27 | 21 | 24 | |

Fig. 3. Task information and $\sum_{j=1}^N \tilde{x}_{i,j}^t$ in different periods

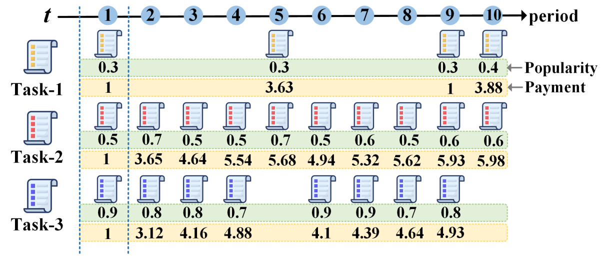



Fig. 4. The whole process of PPAB

Laplace noise for simplicity. Due to the limited space, we only present the first ten periods, and the task push order of the platform is $\{1,2,3\}, \{2,3\}, \{2,3\}, \{2,3\}, \{1,2\}, \{3,2\}, \{3,2\}, \{3,2\}, \{3,2,1\}, \{1,2\}$. The whole process and some related values are displayed in Fig. 4.

## 4 Performance Analysis

In this section, we prove the security of PPAB, and analyze the regret bound and some critical economic properties.

### 4.1 Security Analysis

**Theorem 1.** *The PPAB scheme guarantees the correctness of the aggregated decision and satisfies the $\epsilon$-DP.*

*Proof Sketch:* We first prove that there is $\sum_{j=1}^N x_{i,j}^t=\sum_{j=1}^N \tilde{x}_{i,j}^t$ based on the properties of DH and CSPRNG. Then, we consider an arbitrary pair of popularity sequences with at most one different popularity vector and prove that PPAB meets Definition 6, thereby completing the proof.

Please see Appendix for proof details. Theorem 1 ensures the security of workers' preferences and task popularity.

### 4.2 Regret Analysis

We analyze the regret performance, which denotes the difference between the total popularity of PPAB and the optimal solution. The optimal solution is achieved when the platform knows the popularity of all tasks in advance, i.e., $\mu_i, \forall i\in\mathcal{M}$. Meanwhile, the bid is the extremely-critical payment, i.e., $b_i=p_i$. Let $\mathcal{M}^*$ be the optimal selected task set based on the optimal policy. Obviously, if the platform knows the order: $b_i\mu_{r_1^*}\geq\cdots b_K\mu_{r_K^*}\cdots\geq b_M\mu_{r_M^*}$, it will always choose the top $K$ tasks in all periods, i.e., $\mathcal{M}^*=\{r_1^*, ..., r_K^*\}$. Here, we directly use the notation of requesters to represent the corresponding tasks since they have the one-to-one correspondence. Now, we introduce the concept of regret [19] as follows:

$$Rgt=\sum_{t=1}^{T}\sum_{i=1}^{K}\mu_{r_i^*}-\sum_{t=1}^{T}\mathbb{E}[\sum_{i\in\mathcal{M}^t}\mu_i^t]+\lfloor\frac{T-1}{D}\rfloor\sum_{i=K+1}^{M}\mu_{r_i^*}-\sum_{t=1}^{T}\mathbb{E}[\sum_{A_i^t>D}\mu_i^t].$$

Note that, $*$ denotes the identification of the optimal tasks under the optimal policy. The regret mainly arises from two parts: task selection and staleness control. In the following regret analysis, we consider that each requester is truthful, i.e., $b_i = \vartheta_i, \forall i \in \mathcal{M}$, and we will prove the truthfulness in the next subsection. For simplicity of the following description, we first define the largest and smallest possible difference about the total popularity value among all non-optimal tasks, i.e., $\mathcal{M}^t \neq \mathcal{M}^*$:

$$\Delta_{max} = \sum\nolimits_{i\in\mathcal{M}^*} \mu_i - min_{\mathcal{M}^t\neq\mathcal{M}^*}(\sum\nolimits_{i\in\mathcal{M}^t} \mu_i), \quad (12)$$

$$\Delta_{min} = \sum\nolimits_{i\in\mathcal{M}^*} b_i\mu_i - max_{\mathcal{M}^t\neq\mathcal{M}^*}(\sum\nolimits_{i\in\mathcal{M}^t} b_i\mu_i). \quad (13)$$

Next, we introduce $\chi_i^t$ as the counter of task $i$ after the initial exploration period. In each period $t$ ($t \geq 2$), $\chi_i^t$ is updated according to the following rules: 1) Case 1: when the optimal task set is determined, i.e., $\mathcal{M}^* = \mathcal{M}^t$, $\chi_i^t$ will be unchanged; 2) Case 2: when a non-optimal task set is selected, i.e., $\mathcal{M}^* \neq \mathcal{M}^t$, there must exist one task $i'$ with the minimum counter $\chi_{i'}^{t-1}$, and we let $\chi_{i'}^t = \chi_{i'}^{t-1} + 1$. That is,

$$\begin{cases} \chi_i^t \ \ remains \ \ unchanged, \quad case\,1 \\ i = argmin_{i'\in\mathcal{M}^t}\chi_{i'}^{t-1}, \chi_i^t = \chi_i^{t-1}+1, \quad case\,2 \end{cases} \quad (14)$$

Here, if multiple tasks have the same minimum counter, we choose any one task arbitrarily. Since there always exists a counter to be increased by 1 in Case 2, the sum of the counter $\chi_i^t$ is equal to the total number of the non-optimal task sets. Therefore, before analyzing the regret, we need to derive the upper bound of the expected counter $\mathbb{E}[\chi_i^T]$ by using the contradiction proof technique as follows.

**Lemma 1.** *At the end of the period $T$, the expected counter $\chi_i^T$ has an upper bound for any task $i \in \mathcal{M}$, i.e.,*

$$\mathbb{E}[\chi_i^T] \leq \max\big(\frac{(K+1)\Lambda^2\ln(TK)}{4\theta^2}, \frac{\sqrt{2}\Lambda\log(4/\delta)(\log t+1)}{(1-\theta)\epsilon}\big) + 1 + 2K\pi^2/3, \text{ where } \Lambda = 4K\vartheta_{max}/\Delta_{min}. \quad (15)$$

The complete proof can be found in the Appendix. Based on Lemma 1, we can derive an upper bound on the regret of the PPAB scheme, which is displayed as follows.

**Theorem 2.** *For any the number of periods $T > 0$, The worst regret of PPAB is bounded by $O(MK^3\epsilon^{-1}\ln(TK))$.*

*Proof Sketch:* We mainly consider the reward loss of PPAB incurred by the unknown popularity of tasks. First, we define the notation $\chi_i^t$ for task $i$ to count the number of times that PPAB has not produced the optimal task selection solution until the $t$-th period. Then, we derive the upper bound of the expected counter $\mathbb{E}[\chi_i^T]$ by using the contradiction proof technique (see Lemma 1). Next, we estimate the largest possible difference of rewards between PPAB and the optimal strategy in any period, i.e., $\Delta_{max}$. Therefore, the worst regret can be bounded by $\Delta_{max}\sum_{i=1}^{M}\mathbb{E}[\chi_i^T]$. Finally, we further make a deduction to obtain a sub-linear regret bound about $T$, i.e., $O(MK^3\epsilon^{-1}\ln(TK))$.

### 4.3 Truthfulness, Individual Rationality, and Efficiency

In order to guarantee that each requester will bid using its true valuation, we prove the truthfulness of PPAB.

**Theorem 3.** *The PPAB scheme has truthfulness in each period, i.e., each requester will bid truthfully.*

Please refer to the Appendix for the detailed proof. Next, we should ensure that each requester's utility is greater than zero so as to motivate the participation of requesters.

**Theorem 4.** *In each period, PPAB is individually rational.*

TABLE 2
Evaluation Settings

| Parameter name | Values |
|---|---|
| number of tasks, $M$ | [60, 500] (100 in default) |
| number of periods, $T$ | [20,000, 30,000] (20,000 in default) |
| number of selected tasks, $K$ | [5, 250] (10 in default) |
| privacy budget, $\epsilon$ | 0.1, 0.4, 0.7, **1** |

*Proof.* In the $t$-th period, if a task $i$ is not selected, the utility of the corresponding requester is zero. Otherwise, if the task $i$ is picked, the requester $r_i$'s utility is $\Psi_i^t = \vartheta_i - p_i^t(b_i)$. When $p_i^t(b_i) = \vartheta_{min}$, there is $\Psi_i^t \geq 0$. According to $b_i\widehat{\mathrm{U}}_i^{t-1} > b_{K+1}\widehat{\mathrm{U}}_{K+1}^{t-1}$ and the incentive mechanism in Eq. (11), we obtain $b_i > b_{K+1}\widehat{\mathrm{U}}_{K+1}^{t-1}/\widehat{\mathrm{U}}_i^{t-1} = p_i^t(b_i)$. Based on Theorem 3, each requester will bid truthfully (i.e., $b_i = \vartheta_i$). Hence, we further derive that there is $\Psi_i^t = b_i - b_{K+1}\widehat{\mathrm{U}}_{K+1}^{t-1}/\widehat{\mathrm{U}}_i^{t-1} > 0$. In short, the utility of each requester is nonnegative, so that the PPAB scheme can satisfy the individual rationality. □

Finally, we confirm that the PPAB scheme can be executed in polynomial time and give the following theorem.

**Theorem 5.** *The PPAB scheme is computationally efficient.*

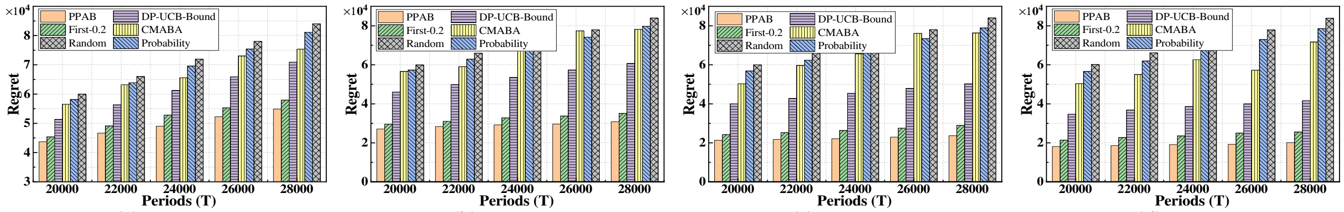
*Proof.* According to the process of PPAB illustrated in Algorithm 1, the PPAB scheme is mainly comprised of the initial exploration phase, privacy-preserving task selection, payment determination, and staleness constraint check. In each period, the platform needs to compute the initial popularity according to the feedback of $N$ workers, so the computational complexity of the initial exploration phase is $O(NM)$. When the platform conducts the operations of sorting, pricing, and checking, Lines 8-21 are enclosed in a loop that iterates $T$ times having the worst-case complexity of $O(TNKM\log(M))$. Note that the computational overhead of sorting workers is $O(M\log(M))$. Therefore, we can conclude that the computational overhead of Algorithm 1 is $O(TNKM\log(M))$. Based on Definition 9, the PPAB scheme satisfies computational efficiency. □

## 5 Performance Evaluation

In this section, we evaluate the performance of the PPAB scheme with extensive simulations. It starts with the introduction of basic simulation settings and the compared algorithms, followed by the detailed evaluation results. Additionally, we conduct the simulations on a computer with Inter(R) Core(TM) i5-10400 CPU @2.9GHz and 16GB RAM under a Windows platform, and all experiments are implemented in Matlab language.

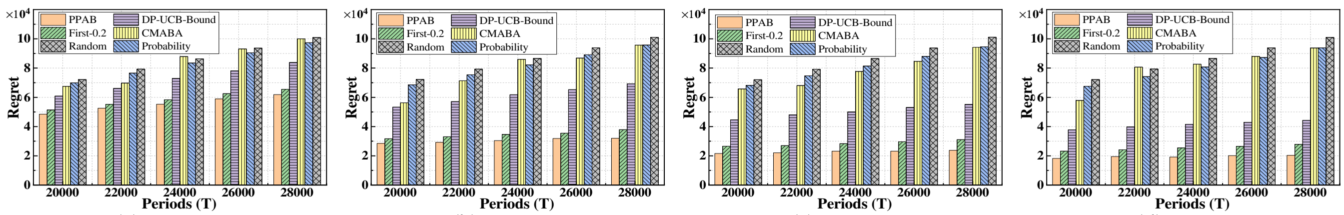
### 5.1 Evaluation Methodology

*Simulation Setup.* We meticulously select a real data trace of Taxi Trips [39] for experimental validation, which is provided by the city of Chicago. Each entry in this dataset records the trip ID, taxi ID, time stamp, trip miles, pickup/dropoff community area, etc. In our simulations, we treat the taxi-hailing requests (i.e., trips) as tasks and regard the taxis as workers. We first classify these tasks according to the pickup community area, i.e., trips belonging to the same area can be seen as a type of task and have the same popularity. Since the number of pickup community



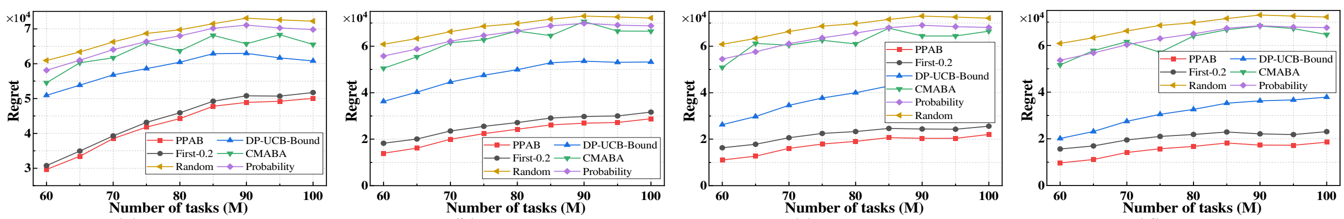

(a) $\epsilon = 0.1$ (b) $\epsilon = 0.4$ (c) $\epsilon = 0.7$ (d) $\epsilon = 1$

Fig. 5. Performance Comparison with $\mu_{min} = 0.05, \mu_{max} = 0.8$: Regret vs. the number of periods ($T$) under different privacy budgets



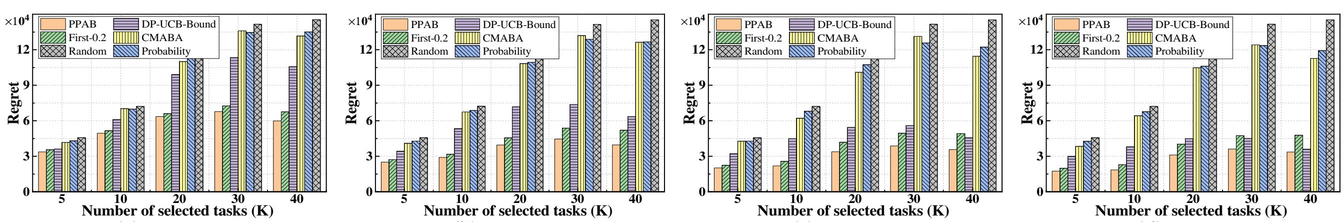

(a) $\epsilon = 0.1$ (b) $\epsilon = 0.4$ (c) $\epsilon = 0.7$ (d) $\epsilon = 1$

Fig. 6. Performance Comparison with $\mu_{min} = 0.1, \mu_{max} = 1$: Regret vs. the number of periods ($T$) under different privacy budgets




(a) $\epsilon = 0.1$ (b) $\epsilon = 0.4$ (c) $\epsilon = 0.7$ (d) $\epsilon = 1$

Fig. 7. Performance Comparison: Regret vs. the number of tasks ($M$) under different privacy budgets




(a) $\epsilon = 0.1$ (b) $\epsilon = 0.4$ (c) $\epsilon = 0.7$ (d) $\epsilon = 1$

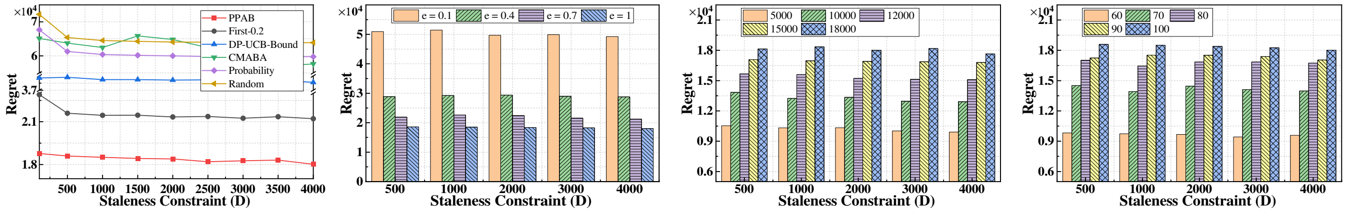
Fig. 8. Performance Comparison: Regret vs. the number of selected tasks ($K$) under different privacy budgets

areas is only 77, we choose $M$ from [60, 500] with the aid of a synthetic dataset. We make a statistical analysis about the number of occurrences of each area, so as to simulate the expected popularity of each task. Thus, the number of workers who accept one task in each period can be generated from a truncated Gaussian distribution with a fixed expected value. Subsequently, we assume that the true valuation $\vartheta_i$ for each task $i$ is directly proportional to the average trip distance, spanning from 1 to 10. The privacy budget $\epsilon$ is then generated incrementally from 0.1 to 1 with a step of 0.3. Furthermore, the total number of periods $T$ varies in the range of 20,000 to 30,000 with an increment of 2000. Simultaneously, the quantity of selected tasks $K$ is set within the range of 5 to 250. Besides, we set $T = 20,000$, $K = 10$, and $N = 30$ by default. Table 2 lists some simulation parameters, where default values are in bold fonts. All experimental results are averaged on 100 random repetitions under the same setting.

*Algorithms for Comparison.* In the experiments, we compare our proposed PPAB scheme with several typical and state-of-the-art algorithms, including DP-UCB-bound [40], First-0.2 [41], Probability [20], and CMABA [27]. The DP-UCB-bound algorithm pulls the arm with the maximal value $b_i(\frac{\widetilde{\mathrm{U}}_i^t}{n_i^t} + \frac{4\sqrt{8}\log t(\log_2 n_i^t+1)}{n_i^t\epsilon})$. "First-0.2" randomly selects $K$ tasks in the first $0.2T$ periods and greedily chooses the top $K$ tasks based on the value $b_i\widehat{\mathrm{U}}_i^t$ in the remaining periods. The algorithms in [20], [27] cannot be directly applied for comparison, we borrow their basic ideas: "Probability" pulls an arm according to the probability distribution of $b_i\widehat{\mathrm{U}}_i^t/\sum_{i=1}^M b_i\widehat{\mathrm{U}}_i^t$. The CMABA algorithm uses $0.5T$ periods for the exploration and directly adopts the learned results in the remaining exploitation phase. Additionally, we implement the optimal algorithm as a baseline in which the popularity of all tasks is known in advance, and the random algorithm refers to selecting tasks randomly in each period.

### 5.2 Evaluation Results

1) *Evaluation of learning performance under different privacy budgets*: We first investigate the influence of the number of periods $T$ by varying it within the range of 20,000 to 28,000. According to the statistical results of the real-world dataset, we set a constraint on the expected popularity of each task, ensuring it remains within the range of [$\mu_{min} = 0.05$, $\mu_{max} =$ 0.8]. As depicted in Fig. 5, the regrets of all implemented algorithms increase along with the growth of $T$. Furthermore, when we increase the value of the privacy budget $\epsilon$, some




(a) $\epsilon = 1, T = 20000$ (b) different $\epsilon$ (c) different $T$ (d) different $M$

Fig. 9. Impact of staleness constraint: Regret vs $D$ under different privacy budgets, periods, and the number of tasks

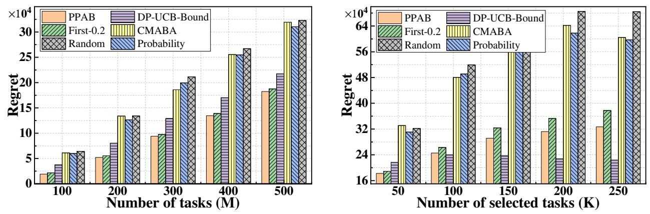



Fig. 10. Impact of the number of tasks and the number of selected tasks

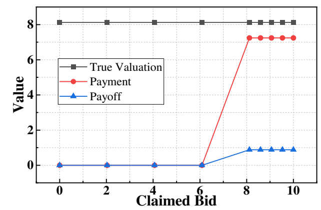



Fig. 11. Truthfulness

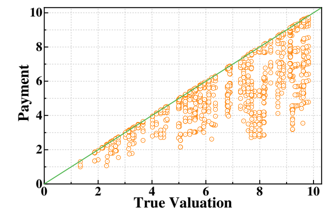



Fig. 12. Individual Rationality

regrets will have a decrease. This phenomenon is attributed to the fact that a higher $\epsilon$ implies a lower privacy protection level. That is, the platform will add less Laplace noise for each task and the selection results are closer to the optimal scenario. It is noteworthy that increasing the privacy budget does not impact the random algorithm, as this algorithm is not subject to privacy concerns. When $T$=28, 000 and $\epsilon = 1$, we calculate that the achieved regrets of PPAB are about 52% and 72% lower than those of the DP-UCB-Bound algorithm and the CMABA algorithm, respectively. This is because PPAB takes the confidence bound and the perturbation into account and explores tasks to a greater extent. Additionally, the algorithm introduced in [20] places a stronger emphasis on exploration, potentially resulting in more periods for exploring low-popularity tasks. As a result, the regret of PPAB is much lower than these compared algorithms.

Then, we enlarge the range of expected popularity by using synthetic data to observe the effect of $T$, as shown in Fig. 6. It is observed that the regrets across all algorithms exhibit a slight growth when $\mu_{max} = 1$. This observation can be attributed to the fact that a larger $\mu_{max}$ indicates a larger gap of popularity among tasks. Consequently, the difference between the optimal solution and PPAB increases. Notably, we compute that the regrets of PPAB are about 27% and 80% lower than those of the First-0.2 algorithm and the Random algorithm, respectively. This reason is that the First-0.2 algorithm might choose many low-popularity tasks in the first $0.2T$ periods. Besides, the random mechanism consistently selects tasks in a random fashion, thereby yielding the maximum regret. Additionally, we also illustrate the correlation between privacy budgets and regrets, in which the variation tendency is consistent with Fig. 5.

Next, we evaluate the performance of PPAB by changing the number of tasks from 60 to 100 with a step of 10, as illustrated in Fig. 7. Although a higher $M$ may introduce additional low-popularity tasks, PPAB can consistently outperform the compared algorithms, particularly the random policy. The random algorithm has the maximal regret since it might choose plenty of low-popularity tasks. Besides, the gap between PPAB and the First-0.2 algorithm is widening along with the increase of the privacy budget. Meanwhile, we study the performance of the implemented algorithms by changing the number of selected tasks in the range of [5, 40]. As illustrated in Fig. 8, a larger value of $K$ corresponds to a higher level of regret. Nevertheless, when the number of selected tasks reaches 40, the regret does not exhibit a distinct increase. This is because $K$ and $M$ are getting closer and the learned popularity will have an attenuate impact.

Furthermore, given $\epsilon$ and $T$, we present the changes of the regret when the peak staleness constraint $D$ varies, as depicted in Fig. 9(a). The figure demonstrates that the regret will decrease slightly when improving $D$, since the staleness values of many tasks have been updated if $D$ is large. When we alter the privacy budget $\epsilon$ from 0.1 to 1 in Fig. 9(b), the regret will have a notable decrease. Hence, a trade-off arises between the level of privacy protection and the extent of the regret. Subsequently, we analyze the influence of $D$ under different numbers of periods and diverse numbers of tasks, where we ignore the relationship between $T$ and $D$. According to Figs. 9(c) and 9(d), we find that $D$ has a little influence on the regret compared with $T$ and $M$.

Finally, we further enlarge the number of tasks $M$ and the number of selected tasks $K$ to verify the scalability of PPAB. Here, we first increase the value of $M$ from 100 to 500 while setting $K = M * 0.1$. Afterwards, we vary the value of $K$ within the range of [50, 250] under $M = 500$. According to Fig. 10, we observe that our scheme can still work when the number of tasks is large. More importantly, PPAB consistently exhibits the lowest regret when compared to other algorithms. These results signify the ability of PPAB to handle different data scales, demonstrating its reliability and performance stability across various data sizes.

2) *Evaluation of incentive performance*: In order to demonstrate the economic characteristics of the PPAB scheme, we consider the three metrics: truthfulness, individual rationality, and underpayment ratio.

Truthfulness: We randomly pick a winning bid, change its claimed valuation, and recalculate the payment. As illustrated in Fig. 11, the requester's utility remains unchanged when the claimed bid exceeds its true valuation. However, we also observe that if the claimed bid falls below the critical payment, the payoff becomes zero. Therefore, submitting a true bid is the optimal choice for the requester, which can showcase the truthfulness of PPAB.

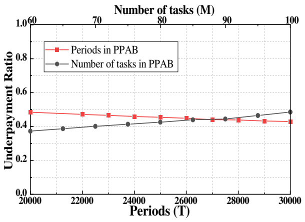



Fig. 13. Underpayment Ratio

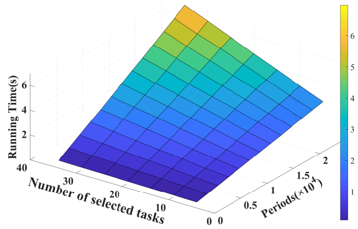



Fig. 14. Computational Efficiency

TABLE 3
Computational Time (second) under Different Parameters

| Params \ Periods | 5000 | 9000 | 13000 | 17000 | 21000 | 25000 |
|---|---|---|---|---|---|---|
| $M = 60$ | 0.412 | 0.752 | 1.082 | 1.413 | 1.770 | 2.105 |
| $M = 70$ | 0.455 | 0.826 | 1.205 | 1.571 | 1.968 | 2.352 |
| $M = 80$ | 0.501 | 0.906 | 1.345 | 1.735 | 2.131 | 2.562 |
| $M = 90$ | 0.554 | 1.001 | 1.463 | 1.908 | 2.375 | 2.842 |
| $M = 100$ | 0.624 | 1.138 | 1.621 | 2.144 | 2.662 | 3.169 |
| $N = 10$ | 0.601 | 1.096 | 1.583 | 2.083 | 2.580 | 3.101 |
| $N = 20$ | 0.610 | 1.108 | 1.616 | 2.107 | 2.601 | 3.142 |
| $N = 30$ | 0.621 | 1.138 | 1.648 | 2.137 | 2.687 | 3.201 |

Individual Rationality: We record the payments of selected tasks and their corresponding true valuations in several periods, as shown in Fig. 12. It can be observed that each requester's payment will be less than its true valuation. In other words, the utility of each requester will not be less than zero (i.e., $\Psi_i^t = \vartheta_i^t - p_i^t(b) \geq 0$), which verifies the property of individual rationality.

Underpayment Ratio: We calculate the underpayment ratio of PPAB, which can be defined as $(\sum_{t=1}^{T}\sum_{p_i^t \neq 0}(\vartheta_i - p_i^t))/\sum_{t=1}^{T}\sum_{p_i^t \neq 0}\vartheta_i$. As depicted in Fig. 13, the underpayment ratio remains approximately stable and is always less than 0.6 when we vary $T$ (or $M$). Therefore, the platform will achieve a satisfactory income and some requesters can make a profit, so as to maintain the system functioning.

Computational Efficiency: We present the running time of the PPAB scheme to demonstrate its computational efficiency, as depicted in Fig. 14. When we alter the number of periods $T$ and the number of selected tasks $K$, it can be observed that the running time gradually increases. It is noteworthy that the execution time remains below 7 seconds under the setting of $K = 35$ and $T = 25,000$, which is significantly shorter than the auction cycle. To further explore the impact of different factors on computational efficiency, we record the running time of the PPAB scheme under various parameters in Table 3. We observe that increasing the number of tasks $M$ leads to a significant rise in running time. Concurrently, the running time increases slowly as the number of workers $N$ increases. This is because the number of workers mainly affects the summation process of the acceptance decisions and does not impact other steps of the PPAB scheme. These simulation outcomes consistently align with our theoretical analysis results (i.e., Theorem 5).

## 6 Related Work

We mainly review the related works from three aspects and highlight the differences compared to our work.

**Task Push in SC**: Recommending suitable tasks to workers stands as a pivotal marketing strategy for the SC platform, and extensive researches have been conducted on this issue [8]–[12], [42]. For example, the authors in [11] inferred user ratings on tasks from their interactive behaviors. Zhao *et al.* [12] proposed a preference-aware spatial task assignment system. Nevertheless, most of these studies assume that the knowledge of workers' preferences can be directly acquired from historical records. A handful of studies have investigated task push without prior information about workers, e.g., Kang *et al.* [43] learned a worker's interests and reliabilities for different categories of tasks. However, the paper solely considered a single worker and aimed to achieve the personalized task recommendation. The framework proposed by Ding *et al.* [44] operates within a bandit learning structure to probe users' interests in the item features. The authors in [10] designed two learning-based recommendation algorithms based on the attractiveness scores between workers and tasks. These task push mechanisms focus on the precise matching between workers and tasks with an omission of privacy considerations. Only a few works investigate the privacy-preserving task matching in SC systems [45]–[50]. For instance, Shu *et al.* [45] developed a proxy-free matching approach while safeguarding both task privacy and worker privacy. Song *et al.* [46] leveraged matrix decomposition and proxy re-encryption to achieve privacy-preserving task matching with threshold similarity search. Different from these studies, we take the online learning on unknown popularity and privacy protection into consideration simultaneously.

**Auction-based CMAB Mechanisms**: In the area of MAB mechanisms, considerable studies have been done on making online decisions under uncertain environments [18], [19], [51]. For example, some recent studies in SC systems have adopted MAB to learn workers' unknown qualities or costs [52]. Wang *et al.* [18] proposed a multi-round user recruitment strategy based on CMAB and the graph theory, which finds the optimal group of unknown workers without considering the incentive issue. Traditional CMAB models simply assume that all arms are feelingless machines. However, in many practical applications, arms exhibit self-interested rational behaviors. Consequently, each requester may strategically manipulate its reported valuation to maximize its individual utility. To make requesters have no incentive to lie, only a few studies investigated the combination of auctions and CMAB problems [24], [26]–[28]. Among them, Jain *et al.* [24] employed the MAB and auction techniques to elicit a truthful mechanism for demand response. The work in [26] applied sponsored search auctions to design a deterministic allocation and payment rule. Xiao *et al.* [27] proposed an adaptive incentive mechanism based on CMAB and the reverse auction to tackle the budget-feasible unknown worker recruitment problem. Nevertheless, all of them do not take the privacy issue into account. Actually, after integrating DP into CMAB, how to determine each selected requester's payment and how to guarantee the truthfulness of requesters will become more challenging. Thus, considering both privacy protection and incentives makes our DPA-CMAB model completely different from existing auction-based or DP-based CMAB models.

**DP-based CMAB Mechanisms**: Although the homomorphic encryption technique can provide theoretically provable high-level security guarantees, it may rely on a trusted third party and suffers from high computational overheads. In contrast, DP is a light-weight yet robust tool with a rigorous mathematical foundation and useful properties [34], [37]. So far, a handful of studies on the DP-based bandit problem have emerged [20]–[22], [40]. For instance, the authors in [40] developed a UCB-based differentially private algorithm that can possess an optimal regret in the stochastic bandit scenario. Zhao *et al.* [20] investigated the privacy-preserving unknown worker recruitment issue and proposed two differentially private arm-pulling algorithms to learn the qualities of workers. Chen *et al.* [21] designed private algorithms with theoretical bounds to achieve the nearly optimal regret. In [22], the MAB problem was considered under the shuffle model of DP, leading to the derivation of distribution-dependent/independent regret. Unfortunately, most of these existing efforts concentrate solely on embedding DP into the MAB model, ignoring the intricate incentive aspect. Furthermore, they do not take the strategic behaviors of arms and the staleness constraint into account together. These neglected factors will complicate algorithmic design and theoretical analysis. Overall, none of the existing studies tightly integrate the online learning, privacy, and incentives to solve the privacy-preserving task push problem with unknown popularity for SC systems.

## 7 Discussion

In this section, we present various discussions on possible extensions of PPAB for more practical scenarios and then point out the potential future research directions.

First, we discuss a dynamic scenario where new tasks may arrive dynamically. In the task selection phase, we assume that all long-term tasks have been pre-published, so that the number of tasks and the number of selected tasks are already fixed and known. However, for a multi-task-oriented SC system, new tasks may be published anytime online. Moreover, some requesters may not participate in the system during certain periods, i.e., several tasks sometimes do not need to be pushed. Therefore, how to address the dynamic arrival of new tasks is a critical challenge. This challenge could lead to new potential research directions, such as the selection problem with unavailable arms, which will be investigated in our future work.

Then, we explore the extensions of PPAB that can adapt to more complex task popularity models. For simplicity, we adopt a coarse-grained model of task popularity to reflect the overall preferences of workers. Actually, PPAB can be easily extended to a fine-grained model of task popularity. For example, we can use some vectors to describe the properties of each task and the preference characteristics of each worker. Moreover, task popularity can be defined using various metrics, such as task acceptance probability, task completion frequency, and task sharing count. Hence, establishing a sophisticated and practical task popularity model is a very complex research issue in itself, which may result in a completely new research work.

Finally, we intend to study distributed scenarios with PPAB to support more realistic applications. In this paper, we consider a typical centralized SC system with a semi-honest platform. Actually, the platform may be untrusted and vulnerable in real-world applications. Exploring the integration of the blockchain technology and smart contracts may be required to address it. In our future work, we attempt to construct a decentralized SC system with the PPAB scheme to enhance transparency, credibility, and security. In addition, we primarily emphasize the task popularity as a key factor in the task push model. In future, we will take more practical factors into consideration, such as geographical locations of workers and social network influence.

## 8 Conclusion

In this study, we investigate the privacy-preserving task push problem with unknown popularity for SC and propose a privacy-preserving auction-based bandit scheme, termed PPAB. We formalize the problem as a novel DPA-CMAB model and design a secure arm-pulling policy, which can protect the privacy of workers' preferences and the popularity information, balance the exploration-exploitation trade-off, and ensure the freshness of tasks. Based on the auction theory, we develop an incentive mechanism to determine the payments for selected tasks. Meanwhile, we derive an upper bound on regret and conduct a security analysis of PPAB. Furthermore, we prove that PPAB meets some delightful economic properties. Rigorous simulations based on real-world data corroborate the marked performance improvements brought about by our proposed scheme.

## References


[1] Y. Tong, Z. Zhou, Y. Zeng, L. Chen, and C. Shahabi, "Spatial crowdsourcing: a survey," *The VLDB Journal*, vol. 29, no. 1, pp. 217–250, 2020.

[2] W. R. Ouyang, L. M. Kaplan, A. Toniolo, M. B. Srivastava, and T. J. Norman, "Parallel and streaming truth discovery in large-scale quantitative crowdsourcing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 10, pp. 2984–2997, 2016.

[3] W. Wang, Z. He, P. Shi, W. Wu, Y. Jiang, B. An, Z. Hao, and B. Chen, "Strategic social team crowdsourcing: Forming a team of truthful workers for crowdsourcing in social networks," *IEEE Transactions on Mobile Computing*, vol. 18, no. 6, pp. 1419–1432, 2019.

[4] C. Chai, J. Fan, G. Li, J. Wang, and Y. Zheng, "Crowdsourcing database systems: Overview and challenges," in *IEEE ICDE*, 2019, pp. 2052–2055.

[5] J. Jiang, B. An, Y. Jiang, P. Shi, Z. Bu, and J. Cao, "Batch allocation for tasks with overlapping skill requirements in crowdsourcing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 8, pp. 1722–1737, 2019.

[6] Y. Zhao, K. Zheng, Y. Li, H. Su, J. Liu, and X. Zhou, "Destination-aware task assignment in spatial crowdsourcing: A worker decomposition approach," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 12, pp. 2336–2350, 2020.

[7] Y. Xu, M. Xiao, Y. Zhu, J. Wu, S. Zhang, and J. Zhou, "Aoi-guaranteed incentive mechanism for mobile crowdsensing with freshness concerns," *IEEE Transactions on Mobile Computing*, pp. 1–17, 2023.

[8] X. Yin, H. Wang, W. Wang, and K. Zhu, "Task recommendation in crowdsourcing systems: A bibliometric analysis," *Technology in Society*, vol. 63, p. 101337, 2020.

[9] L. Wang, D. Yang, Z. Y, F. Xiong, L. Han, S. Pan, and B. Guo, "Task scheduling in three-dimensional spatial crowdsourcing: A social welfare perspective," *IEEE Transactions on Mobile Computing*, vol. 22, no. 9, pp. 5555–5567, 2023.

[10] M. S. Safran and D. Che, "Efficient learning-based recommendation algorithms for top-$N$ tasks and top-$N$ workers in large-scale crowdsourcing systems," *ACM Transactions on Information Systems*, vol. 37, no. 1, pp. 2:1–2:46, 2019.

[11] M. Yuen, I. King, and K. Leung, "Taskrec: A task recommendation framework in crowdsourcing systems," *Neural Processing Letters*, vol. 41, no. 2, pp. 223–238, 2015.

[12] Y. Zhao, K. Zheng, H. Yin, G. Liu, J. Fang, and X. Zhou, "Preference-aware task assignment in spatial crowdsourcing: From individuals to groups," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 7, pp. 3461–3477, 2022.

[13] M. Pan, W. Huang, Y. Li, X. Zhou, Z. Liu, R. Song, H. Lu, Z. Tian, and J. Luo, "DHPA: dynamic human preference analytics framework: A case study on taxi drivers' learning curve analysis," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 11, no. 1, pp. 8:1–8:19, 2020.

[14] H. Wang and H. Yang, "Ridesourcing systems: A framework and review," *Transportation Research Part B: Methodological*, vol. 129, pp. 122–155, 2019.

[15] Y. Li, Y. Zhao, and K. Zheng, "Preference-aware group task assignment in spatial crowdsourcing: A mutual information-based approach," in *IEEE ICDM*, 2021, pp. 350–359.

[16] Y. Zhao, J. Xia, G. Liu, H. Su, D. Lian, S. Shang, and K. Zheng, "Preference-aware task assignment in spatial crowdsourcing," in *AAAI*, 2019, pp. 2629–2636.

[17] B. Zhao, P. Xu, Y. Shi, Y. Tong, Z. Zhou, and Y. Zeng, "Preference-aware task assignment in on-demand taxi dispatching: An online stable matching approach," in *AAAI*, 2019, pp. 2245–2252.

[18] H. Wang, Y. Yang, E. Wang, W. Liu, Y. Xu, and J. Wu, "Combinatorial multi-armed bandit based user recruitment in mobile crowdsensing," in *IEEE SECON*, 2020, pp. 1–9.

[19] W. Chen, Y. Wang, and Y. Yuan, "Combinatorial multi-armed bandit: General framework and applications," in *ICML*, vol. 28, 2013, pp. 151–159.

[20] H. Zhao, M. Xiao, J. Wu, Y. Xu, H. Huang, and S. Zhang, "Differentially private unknown worker recruitment for mobile crowdsensing using multi-armed bandits," *IEEE Transactions on Mobile Computing*, vol. 20, no. 9, pp. 2779–2794, 2021.

[21] X. Chen, K. Zheng, Z. Zhou, Y. Yang, W. Chen, and L. Wang, "(locally) differentially private combinatorial semi-bandits," in *ICML*, vol. 119, 2020, pp. 1757–1767.

[22] J. Tenenbaum, H. Kaplan, Y. Mansour, and U. Stemmer, "Differentially private multi-armed bandits in the shuffle model," in *NeurIPS*, 2021, pp. 24 956–24 967.

[23] H. Zhao and W. Chen, "Online second price auction with semi-bandit feedback under the non-stationary setting," in *AAAI*, 2020, pp. 6893–6900.

[24] S. Jain and S. Gujar, "A multiarmed bandit based incentive mechanism for a subset selection of customers for demand response in smart grids," in *AAAI*, 2020, pp. 2046–2053.

[25] M. Braverman, J. Mao, J. Schneider, and S. M. Weinberg, "Multi-armed bandit problems with strategic arms," in *COLT*, ser. Proceedings of Machine Learning Research, vol. 99, 2019, pp. 383–416.

[26] D. Padmanabhan, S. Bhat, K. Prabuchandran, S. Shevade, and Y. Narahari, "Dominant strategy truthful, deterministic multi-armed bandit mechanisms with logarithmic regret for sponsored search auctions," *Applied Intelligence*, pp. 1–18, 2021.

[27] M. Xiao, J. Wang, H. Zhao, and G. Gao, "Unknown worker recruitment in mobile crowdsensing using CMAB and auction," in *IEEE ICDCS*, 2020, pp. 1145–1155.

[28] Y. Wu, F. Li, L. Ma, Y. Xie, T. Li, and Y. Wang, "A context-aware multiarmed bandit incentive mechanism for mobile crowd sensing systems," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7648–7658, 2019.

[29] M. Xiao, G. Gao, J. Wu, S. Zhang, and L. Huang, "Privacy-preserving user recruitment protocol for mobile crowdsensing," *IEEE/ACM Transactions on Networking*, vol. 28, no. 2, pp. 519–532, 2020.

[30] C. Ying, H. Jin, X. Wang, and Y. Luo, "Double insurance: Incentivized federated learning with differential privacy in mobile crowdsensing," in *IEEE SRDS*, 2020, pp. 81–90.

[31] Y. Hu and R. Zhang, "A spatiotemporal approach for secure crowdsourced radio environment map construction," *IEEE/ACM Transactions on Networking*, vol. 28, no. 4, pp. 1790–1803, 2020.

[32] J. Lin, D. Yang, M. Li, J. Xu, and G. Xue, "Frameworks for privacy-preserving mobile crowdsensing incentive mechanisms," *IEEE Transactions on Mobile Computing*, vol. 17, no. 8, pp. 1851–1864, 2018.

[33] G. Xu, H. Li, S. Liu, K. Yang, and X. Lin, "Verifynet: Secure and verifiable federated learning," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 911–926, 2020.

[34] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography conference*, 2006, pp. 265–284.

[35] D. Yuan, Q. Li, G. Li, Q. Wang, and K. Ren, "Priradar: A privacy-preserving framework for spatial crowdsourcing," *IEEE transactions on information forensics and security*, vol. 15, pp. 299–314, 2020.

[36] Y. Xu, M. Xiao, J. Wu, H. Tan, and G. Gao, "A personalized privacy preserving mechanism for crowdsourced federated learning," *IEEE Transactions on Mobile Computing*, vol. 23, no. 2, pp. 1568–1585, 2024.

[37] Y. Xu, M. Xiao, A. Liu, and J. Wu, "Edge resource prediction and auction for distributed spatial crowdsourcing with differential privacy," *IEEE Internet of Things Journal*, vol. 9, no. 17, pp. 15 554–15 569, 2022.

[38] G. Gao, S. Huang, H. Huang, M. Xiao, J. Wu, Y. Sun, and S. Zhang, "Combination of auction theory and multi-armed bandits: Model, algorithm, and application," *IEEE Transactions on Mobile Computing*, vol. 22, no. 11, pp. 6343–6357, 2023.

[39] "Chicago Taxi Trips dataset (aug 2023)," Downloaded from https://data.cityofchicago.org/Transportation/Taxi-Trips/wrvz-psew.

[40] A. C. Y. Tossou and C. Dimitrakakis, "Algorithms for differentially private multi-armed bandits," in *AAAI*, 2016, pp. 2087–2093.

[41] L. Tran-Thanh, A. C. Chapman, E. M. de Cote, A. Rogers, and N. R. Jennings, "Epsilon-first policies for budget-limited multi-armed bandits," in *AAAI*, 2010.

[42] J. Shu, X. Jia, K. Yang, and H. Wang, "Privacy-preserving task recommendation services for crowdsourcing," *IEEE Transactions on Services Computing*, vol. 14, no. 1, pp. 235–247, 2021.

[43] Q. Kang and W. P. Tay, "Task recommendation in crowdsourcing based on learning preferences and reliabilities," *IEEE Transactions on Services Computing*, 2020.

[44] Q. Ding, Y. Liu, C. Miao, F. Cheng, and H. Tang, "A hybrid bandit framework for diversified recommendation," in *AAAI*, 2021, pp. 4036–4044.

[45] J. Shu, K. Yang, X. Jia, X. Liu, C. Wang, and R. H. Deng, "Proxy-free privacy-preserving task matching with efficient revocation in crowdsourcing," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 1, pp. 117–130, 2021.

[46] F. Song, Z. Qin, D. Liu, J. Zhang, X. Lin, and X. Shen, "Privacy-preserving task matching with threshold similarity search via vehicular crowdsourcing," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 7, pp. 7161–7175, 2021.

[47] Y. Guo, H. Xie, Y. Miao, C. Wang, and X. Jia, "Fedcrowd: A federated and privacy-preserving crowdsourcing platform on blockchain," *IEEE Transactions on Services Computing*, vol. 15, no. 4, pp. 2060–2073, 2022.

[48] W. Tang, K. Zhang, J. Ren, Y. Zhang, and X. S. Shen, "Privacy-preserving task recommendation with win-win incentives for mobile crowdsourcing," *Information Sciences*, vol. 527, pp. 477–492, 2020.

[49] C. Zhang, Y. Guo, X. Jia, C. Wang, and H. Du, "Enabling proxy-free privacy-preserving and federated crowdsourcing by using blockchain," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6624–6636, 2021.

[50] Y. Yang, Y. Cheng, Y. Yuan, G. Wang, L. Chen, and Y. Sun, "Privacy-preserving cooperative online matching over spatial crowdsourcing platforms," *Proceedings of the VLDB Endowment*, vol. 16, no. 1, pp. 51–63, 2022.

[51] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine Learning*, vol. 47, no. 2-3, pp. 235–256, 2002.

[52] S. Klos, C. Tekin, M. van der Schaar, and A. Klein, "Context-aware hierarchical online learning for performance maximization in mobile crowdsourcing," *IEEE/ACM Transactions on Networking*, vol. 26, no. 3, pp. 1334–1347, 2018.

**Yin Xu** is currently an associate researcher in the School of Computer Science and Technology at the University of Science and Technology of China (USTC). She received her Ph.D. degree from the School of Computer Science and Technology at USTC, Hefei, China, in 2024. She also received her B.S. degree from the School of Computer Science and Technology at the Anhui University, Hefei, China, in 2019. Her research interests include spatial crowdsensing, federated learning, privacy preservation, game theory, edge computing, and incentive mechanism design. She has published 20 articles in referred journals and conferences, including IEEE TMC, TII, ICDE, INFOCOM, DASFAA, etc.

**Mingjun Xiao** is a professor in the School of Computer Science and Technology at the University of Science and Technology of China (USTC). He received his Ph.D. from USTC in 2004. His research interests include mobile crowdsensing, edge computing, federated learning, auction theory, data security and privacy. He has published more over 100 papers in referred journals and conferences, including TMC, TC, TPDS, TON, TKDE, TSC, INFOCOM, ICDE, ICNP, etc. He served as the TPC member of INFOCOM'23, INFOCOM'22, IJCAI'22, INFOCOM'21, IJCAI'21, INFOCOM'20, INFOCOM'19, ICDCS'19, DASFAA'19, INFOCOM'18, etc. He is on the reviewer board of several top journals such as TMC, TON, TPDS, TSC, TVT, TCC, etc.

**Jie Wu** is the Director of the Center for Networked Computing and Laura H. Carnell professor at Temple University. He also serves as the Director of International Affairs at College of Science and Technology. He served as Chair of Department of Computer and Information Sciences from the summer of 2009 to the summer of 2016 and Associate Vice Provost for International Affairs from the fall of 2015 to the summer of 2017. Prior to joining Temple University, he was a program director at the National Science Foundation and was a distinguished professor at Florida Atlantic University. His current research interests include mobile computing and wireless networks, routing protocols, network trust and security, distributed algorithms, applied machine learning, and cloud computing. Dr. Wu regularly publishes in scholarly journals, conference proceedings, and books. He serves on several editorial boards, including IEEE Transactions on Mobile Computing, IEEE Transactions on Service Computing, Journal of Parallel and Distributed Computing, and Journal of Computer Science and Technology. Dr. Wu is/was general chair/co-chair for IEEE IPDPS'08, IEEE DCOSS'09, IEEE ICDCS'13, ACM MobiHoc'14, ICPP'16, IEEE CNS'16, WiOpt'21, and ICDCN'22 as well as program chair/cochair for IEEE MASS'04, IEEE INFOCOM'11, CCF CNCC'13, and ICCCN'20. He was an IEEE Computer Society Distinguished Visitor, ACM Distinguished Speaker, and chair for the IEEE Technical Committee on Distributed Processing (TCDP). Dr. Wu is a Fellow of the AAAS and a Fellow of the IEEE. He is the recipient of the 2011 China Computer Federation (CCF) Overseas Outstanding Achievement Award.

**He Sun** received his B.S. and B.A. degrees from the School of Computer Science and Technology and School of Foreign Languages respectively, at Qingdao University (QDU), Qingdao, China, in 2020. He is currently working toward the PhD degree in the School of Computer Science and Technology at University of Science and Technology of China (USTC), Hefei, China. His research interests include reinforcement learning, game theory, data collection, data trading, mobile computing, and privacy preservation.

# APPENDIX

In this appendix, we provide the detailed proofs of Theorem 1, Lemma 1, Theorem 2, and Theorem 3. Specifically, we first prove the security of PPAB, and then analyze the upper bound of regret and the truthfulness of PPAB.

## 0.1 Security Analysis

**Theorem 1.** *The PPAB scheme guarantees the correctness of the aggregated decision and satisfies the $\epsilon$-DP.*

*Proof.* In order to protect the privacy of each worker's preference, each worker is allowed to upload an encrypted decision $\tilde{x}_{i,j}^t$ rather than the original decision $x_{i,j}^t$. Thus, we need to prove the correctness of the aggregated decision, i.e., $\sum_{j=1}^N x_{i,j}^t = \sum_{j=1}^N \tilde{x}_{i,j}^t$. Without loss of generality, we assume that the worker set is $\mathcal{W}_i^t = \{w_{i,1}^t, w_{i,2}^t, w_{i,3}^t\}$ with assigned IDs: $ID_{w_{i,1}^t} = 1, ID_{w_{i,2}^t} = 2, ID_{w_{i,3}^t} = 3$. According to Section 3.2, three workers respectively calculate their encrypted decisions: $\tilde{x}_{i,1}^t = x_{i,1}^t + \tilde{\Phi}(w_{i,1}^t, w_{i,2}^t) + \tilde{\Phi}(w_{i,1}^t, w_{i,3}^t)$, $\tilde{x}_{i,2}^t = x_{i,2}^t + \tilde{\Phi}(w_{i,2}^t, w_{i,3}^t) - \tilde{\Phi}(w_{i,2}^t, w_{i,1}^t)$, and $\tilde{x}_{i,3}^t = x_{i,3}^t - \tilde{\Phi}(w_{i,3}^t, w_{i,1}^t) - \tilde{\Phi}(w_{i,3}^t, w_{i,2}^t)$. Based on the properties of the DH protocol and CSPRNG, the output of $\tilde{\Phi}$ remains unchanged as long as its input is identical, e.g., $\tilde{\Phi}(w_{i,1}^t, w_{i,2}^t) = \tilde{\Phi}(w_{i,2}^t, w_{i,1}^t)$. Therefore, we can achieve the correctness of the aggregated decisions: $\tilde{x}_{i,1}^t + \tilde{x}_{i,2}^t + \tilde{x}_{i,3}^t = x_{i,1}^t + x_{i,2}^t + x_{i,3}^t$.

Next, the privacy of the popularity sequences of tasks is safeguarded via the platform. We consider an arbitrary period $t$ and a pair of popularity sequences $\vec{\mathrm{U}} = \{\mathrm{U}^1, \cdots, \mathrm{U}^t\}$ and $\vec{\mathrm{U}}' = \{\mathrm{U}'^1, \cdots, \mathrm{U}'^t\}$ with at most one different popularity vector, where $\mathrm{U}^t = \{\mathrm{U}_1^t, \cdots, \mathrm{U}_M^t\}$. Thus, there exists at most one period $\tau \leq t$ in which $\{\mathrm{U}_1^\tau, \cdots, \mathrm{U}_i^\tau, \cdots, \mathrm{U}_M^\tau\}$ is perturbed to $\{\mathrm{U}_1'^\tau, \cdots, \mathrm{U}_i'^\tau, \cdots, \mathrm{U}_M'^\tau\}$. That is, for any task $i$, $\{\mathrm{U}_i^1, ..., \mathrm{U}_i^t\}$ and $\{\mathrm{U}_i'^1, ..., \mathrm{U}_i'^t\}$ differ in at most one item. Since all popularity values belong to [0, 1], there is $|\sum_{\tau=1}^t \mathrm{U}_i^\tau - \sum_{\tau=1}^t \mathrm{U}_i'^\tau| \leq 1$. Let $\tilde{U}$ be an arbitrary confused accumulative reward. According to Section 3.3, we can get the following inequalities:

$$\frac{\mathbf{Pr}[\tilde{U} = \sum_{\tau=1}^t \mathrm{U}_i^\tau + (\varrho^t - 1)Lap(\frac{2M\lfloor \log t \rfloor}{\epsilon}) + Lap(\frac{2M}{\epsilon})]}{\mathbf{Pr}[\tilde{U} = \sum_{\tau=1}^t \mathrm{U}_i'^\tau + (\varrho^t - 1)Lap(\frac{2M\lfloor \log t \rfloor}{\epsilon}) + Lap(\frac{2M}{\epsilon})]}$$
$$= \frac{\{\frac{\epsilon}{4M\lfloor \log t \rfloor} exp(-\frac{\epsilon|\tilde{U} - \sum_{\tau=1}^t \mathrm{U}_i^\tau|}{2M\lfloor \log t \rfloor})\}^{\varrho^t - 1} \cdot \{\frac{\epsilon}{4M} exp(-\frac{\epsilon|\tilde{U} - \sum_{\tau=1}^t \mathrm{U}_i^\tau|}{2M})\}}{\{\frac{\epsilon}{4M\lfloor \log t \rfloor} exp(-\frac{\epsilon|\tilde{U} - \sum_{\tau=1}^t \mathrm{U}_i'^\tau|}{2M\lfloor \log t \rfloor})\}^{\varrho^t - 1} \cdot \{\frac{\epsilon}{4M} exp(-\frac{\epsilon|\tilde{U} - \sum_{\tau=1}^t \mathrm{U}_i'^\tau|}{2M})\}}$$
$$= \frac{\{exp([\epsilon|\tilde{U} - \sum_{\tau=1}^t \mathrm{U}_i'^\tau| - \epsilon|\tilde{U} - \sum_{\tau=1}^t \mathrm{U}_i^\tau|]/2M\lfloor \log t \rfloor)\}^{\varrho^t - 1}}{exp([\epsilon|\tilde{U} - \sum_{\tau=1}^t \mathrm{U}_i'^\tau| - \epsilon|\tilde{U} - \sum_{\tau=1}^t \mathrm{U}_i^\tau|]/2M)}$$
$$\leq exp(\frac{\epsilon(\varrho^t - 1)|\sum_{\tau=1}^t \mathrm{U}_i^\tau - \sum_{\tau=1}^t \mathrm{U}_i'^\tau|}{2M\lfloor \log t \rfloor}) \cdot exp(\frac{\epsilon|\sum_{\tau=1}^t \mathrm{U}_i^\tau - \sum_{\tau=1}^t \mathrm{U}_i'^\tau|}{2M})$$
$$\leq exp(\frac{\epsilon|\sum_{\tau=1}^t \mathrm{U}_i^\tau - \sum_{\tau=1}^t \mathrm{U}_i'^\tau|}{2M} + \frac{\epsilon|\sum_{\tau=1}^t \mathrm{U}_i^\tau - \sum_{\tau=1}^t \mathrm{U}_i'^\tau|}{2M}) \leq exp(\frac{\epsilon}{M}).$$

Here, the second inequality follows since there is $\varrho^t - 1 \leq \lfloor \log t \rfloor$. From the above analysis, the hybrid DP-based protection mechanism conducted by the platform can guarantee that the popularity sequence of each task meets the $exp(\frac{\epsilon}{M})$-DP. Then, we consider all tasks and harness the composition property of differential privacy so as to obtain:

$$\frac{\prod_{i=1}^M Pr[\tilde{U} = \sum_{\tau=1}^t \mathrm{U}_i^\tau + (\varrho^t - 1)Lap(\frac{2M\lfloor \log t \rfloor}{\epsilon}) + Lap(\frac{2M}{\epsilon})]}{\prod_{i=1}^M Pr[\tilde{U} = \sum_{\tau=1}^t \mathrm{U}_i'^\tau + (\varrho^t - 1)Lap(\frac{2M\lfloor \log t \rfloor}{\epsilon}) + Lap(\frac{2M}{\epsilon})]}$$
$$\leq \prod_{i=1}^M exp(\epsilon/M) \leq exp(\epsilon). \quad (1)$$

Based on the above analysis, PPAB ensures the correctness of the aggregated decision and satisfies the $\epsilon$-DP. □

## 0.2 Regret Analysis

Before analyzing the regret of PPAB, we need to derive the upper bound of the expected counter $\mathbb{E}[\chi_i^T]$ by using the contradiction proof technique as follows.

**Lemma 1.** *At the end of the period $T$, the expected counter $\chi_i^T$ has an upper bound for any task $i \in \mathcal{M}$, i.e.,*

$$\mathbb{E}[\chi_i^T] \leq \max(\frac{(K+1)\Lambda^2 \ln(TK)}{4\theta^2}, \frac{\sqrt{2}\Lambda \log(4/\delta)(\log t + 1)}{(1-\theta)\epsilon})$$
$$+1 + 2K\pi^2/3, \textit{ where } \Lambda = 4K\vartheta_{max}/\Delta_{min}. \quad (2)$$

*Proof.* Let $\mathbb{I}(\cdot)$ be an indicator function, i.e., $\mathbb{I}\{true\} = 1$ and $\mathbb{I}\{false\} = 0$, which will be used to denote whether the selected task set is optimal. According to the update rule of the counter $\chi_i^T$, we attain:

$$\chi_i^T = \sum_{t=2}^T \mathbb{I}\{case\,2\} \leq \gamma + \sum_{t=2}^T \Pi\{case\,2, \chi_i^t \geq \gamma\}$$
$$\leq \gamma + \sum_{t=2}^T \mathbb{I}\{\sum_{i \in \mathcal{M}^t} b_i \widehat{\mathrm{U}}_i^{t-1} \geq \sum_{i \in \mathcal{M}^*} b_i \widehat{\mathrm{U}}_i^{t-1}, \chi_i^t \geq \gamma\}$$
$$\leq \gamma + \sum_{t=2}^T \mathbb{I}\{\max_{\gamma \leq n_{r_1}^t \leq \ldots \leq n_{r_K}^t \leq t-1} \sum_{i=1}^K b_{r_i} \widehat{\mathrm{U}}_{r_i}^{t-1}$$
$$\geq \min_{1 \leq n_{r_1}^* \leq \ldots \leq n_{r_K}^* \leq t-1} \sum_{i=1}^K b_{r_i^*} \widehat{\mathrm{U}}_{r_i^*}^{t-1}\}$$
$$\leq \gamma + \sum_{t=1}^T \sum_{n_{r_1}^t = \gamma}^{t-1} \cdots \sum_{n_{r_K}^t = \gamma}^{t-1} \sum_{n_{r_1}^* = 1}^{t-1} \cdots \sum_{n_{r_K}^* = 1}^{t-1} \mathbb{I}\{\sum_{i=1}^K b_{r_i} \widehat{\mathrm{U}}_{r_i}^t \geq \sum_{i=1}^K b_{r_i^*} \widehat{\mathrm{U}}_{r_i^*}^t\}.$$

To further derive the upper bound of $\chi_i^T$, we need to compute the probability that the event $\sum_{i=1}^K b_{r_i} \widehat{\mathrm{U}}_{r_i}^t \geq \sum_{i=1}^K b_{r_i^*} \widehat{\mathrm{U}}_{r_i^*}^t$ occurs. Based on the proof by contradiction, one of the following three conditions must have happened:

$$\text{Condition 1: } \sum_{i=1}^K b_{r_i} \frac{\widetilde{\mathrm{U}}_{r_i}^t}{n_{r_i}^t} \geq \sum_{i=1}^K b_{r_i}(\mu_{r_i} + \varphi_{r_i}^t + \frac{\phi^t}{n_{r_i}^t}); \quad (3)$$

$$\text{Condition 2: } \sum_{i=1}^K b_{r_i^*} \frac{\widetilde{\mathrm{U}}_{r_i^*}^t}{n_{r_i^*}^t} \leq \sum_{i=1}^K b_{r_i^*}(\mu_{r_i^*} - \varphi_{r_i^*}^t - \frac{\phi^t}{n_{r_i^*}^t}); \quad (4)$$

$$\text{Condition 3: } \sum_{i=1}^K b_{r_i^*} \mu_{r_i^*} < \sum_{i=1}^K b_{r_i}(\mu_{r_i} + 2\varphi_{r_i}^t + \frac{2\phi^t}{n_{r_i}^t}). \quad (5)$$

Condition 1 implies a drastic overestimate of the sub-optimal task set. We first deduce the upper bound of the probability of condition 1 as follows:

$$\mathbf{Pr}\{\sum_{i=1}^K b_{r_i} \frac{\widetilde{\mathrm{U}}_{r_i}^t}{n_{r_i}^t} \geq \sum_{i=1}^K b_{r_i}(\mu_{r_i} + \varphi_{r_i}^t + \frac{\phi^t}{n_{r_i}^t})\}$$
$$\leq \mathbf{Pr}\{\sum_{i=1}^K \widetilde{\mathrm{U}}_{r_i}^t \geq \sum_{i=1}^K (\overline{\mathrm{U}}_{r_i}^t + \phi^t)\}$$
$$+\mathbf{Pr}\{\sum_{i=1}^K \overline{\mathrm{U}}_{r_i}^t \geq \sum_{i=1}^K (n_{r_i}^t \mu_{r_i} + n_{r_i}^t \varphi_{r_i}^t)\}. \quad (6)$$

Given the accumulative reward $\overline{\mathrm{U}}_i^t$ of an arbitrary task and the confused accumulative reward $\widetilde{\mathrm{U}}_i^t$ perturbed by the hybrid DP-based protection mechanism, we have $\mathbf{Pr}\{|\widetilde{\mathrm{U}}_i^t - \overline{\mathrm{U}}_i^t| \geq \phi^t\} \leq \delta$ according to the lemma in [38]. Here, $\delta$ is a small number which is close to 0, and $|\widetilde{\mathrm{U}}_i^t - \overline{\mathrm{U}}_i^t|$ denotes the total Laplace noise injected into $\overline{\mathrm{U}}_i^t$. Therefore, we can continue to get the upper bound of Eq. (6).

$$\sum_{i=1}^K \mathbf{Pr}\{\widetilde{\mathrm{U}}_{r_i}^t \geq (\overline{\mathrm{U}}_{r_i}^t + \phi^t)\} + \sum_{i=1}^K \mathbf{Pr}\{\overline{\mathrm{U}}_{r_i}^t \geq (n_{r_i}^t \mu_{r_i} + n_{r_i}^t \varphi_{r_i}^t)\}$$

$$\leq K\cdot\delta+\sum_{i=1}^{K} e^{-2(n_{r_i}^t\varphi_{r_i}^t)^2/n_{r_i}^t} \leq K\cdot\delta+\sum_{i=1}^{K} e^{-2[(K+1)\ln(\sum_{i'}^{M} n_{i'}^t)]}$$
$$\leq K\cdot\delta+\sum_{i=1}^{K} e^{-2[(K+1)\ln(Mt)]} \leq K\cdot\delta+K\cdot t^{-2(K+1)}. \quad (7)$$

Note that, the deduction of Eq. (7) makes use of the Chernoff-Hoeffding bound, which has been widely-used in previous works [19, 49]. If we let $\delta = t^{-2(K+1)}$, there is:

$$\mathbf{Pr}\{\sum_{i=1}^{K} b_{r_i}\frac{\widetilde{\mathrm{U}}_{r_i}^t}{n_{r_i}^t} \geq \sum_{i=1}^{K} b_{r_i}(\mu_{r_i}+\varphi_{r_i}^t+\frac{\phi^t}{n_{r_i}^t})\} \leq 2K\cdot t^{-2(K+1)}.$$

Condition 2 signifies an underestimate of the optimal task set. The upper bound of the probability of condition 2 is similar to the above analysis of condition 1, i.e.,

$$\mathbf{Pr}\{\sum_{i=1}^{K} b_{r_i^*}\frac{\widetilde{\mathrm{U}}_{r_i^*}^t}{n_{r_i^*}^t} \leq \sum_{i=1}^{K} b_{r_i^*}(\mu_{r_i^*}-\varphi_{r_i^*}^t-\frac{\phi^t}{n_{r_i^*}^t})\} \leq 2K\cdot t^{-2(K+1)}.$$

Condition 3 implies an overlap in the confidence intervals of the optimal and sub-optimal task set. Now, we need to choose a suitable value $\gamma$ so that condition 3 is not true.

$$\sum_{i=1}^{K} b_{r_i^*}\mu_{r_i^*} - \sum_{i=1}^{K} b_{r_i}\mu_{r_i} - \sum_{i=1}^{K} b_{r_i}(2\varphi_{r_i}^t + 2\phi^t/n_{r_i}^t)$$
$$\geq \Delta_{min} - \sum_{i=1}^{K} b_{r_i}(2\sqrt{(K+1)\ln(\sum_{i'}^{M} n_{i'}^t)/n_{r_i}^t} + 2\phi^t/n_{r_i}^t)$$
$$\geq \Delta_{min} - K\vartheta_{max}(2\sqrt{(K+1)\ln(TK)/\gamma} + 2\phi^t/\gamma) \geq 0, \quad (8)$$

where the second step is due to the fact that $n_{r_i}^t \geq \chi_i^t \geq \gamma$ and $b_{r_i} \leq \vartheta_{max}$. For any $\theta \in (0,1)$, if the following two inequalities hold, we conclude that $\Delta_{min} \geq K\vartheta_{max}(2\sqrt{((K+1)\ln(TK))/\gamma}+2\phi^t/\gamma)$ is true.

$$\theta\Delta_{min} \geq 2K\vartheta_{max}\sqrt{\frac{(K+1)\ln(TK)}{\gamma}}, \quad (9)$$

$$(1-\theta)\Delta_{min} \geq \frac{2K\vartheta_{max}\phi^t}{\gamma}. \quad (10)$$

Through solving Eq. (9) and Eq. (10), we can obtain:

$$\gamma \geq \frac{4(K+1)K^2\vartheta_{max}^2\ln(TK)}{\theta^2\Delta_{min}^2} \text{ and } \gamma \geq \frac{2K\vartheta_{max}\phi^t}{(1-\theta)\Delta_{min}}. \quad (11)$$

We substitute $\phi^t = \frac{2\sqrt{2}}{\epsilon}log(\frac{4}{\delta})(\log t+1)$ into the second inequality in Eq. (11) and further have the bound:

$$\gamma \geq \frac{4\sqrt{2}K\vartheta_{max}\log(4/\delta)(\log t+1)}{(1-\theta)\epsilon\Delta_{min}}. \quad (12)$$

After acquiring the lower bound of $\gamma$ and analyzing the above three conditions, we can derive the upper bound of the expected counter:

$$\mathbb{E}[\chi_i^T] \leq \lceil \max\{\frac{(K+1)\Lambda^2\ln(TK)}{4\theta^2}, \frac{\sqrt{2}\Lambda\log(4/\delta)(\log T+1)}{(1-\theta)\epsilon}\}\rceil$$
$$+\sum_{t=1}^{+\infty}(t-\gamma)^K(t-1)^K 4K\cdot t^{-2(K+1)}$$
$$\leq \max\{\frac{(K+1)\Lambda^2\ln(TK)}{4\theta^2}, \frac{\sqrt{2}\Lambda\log(4/\delta)(\log T+1)}{(1-\theta)\epsilon}\}$$
$$+1+4K\sum_{t=1}^{+\infty}t^{-2} \leq \Xi+1+\frac{2K\pi^2}{3},$$
$$\Xi \triangleq \max\{\frac{(K+1)\Lambda^2\ln(TK)}{4\theta^2}, \frac{\sqrt{2}\Lambda\log(4/\delta)(\log T+1)}{(1-\theta)\epsilon}\},$$

where $\Lambda = 4K\vartheta_{max}/\Delta_{min}$ and the last inequality follows from $\sum_{t=1}^{+\infty}t^{-2} = \pi^2/6$. Hence, the lemma holds. □

Based on Lemma 1, we can derive an upper bound on the regret of the PPAB scheme, which is displayed as follows.

**Theorem 2.** *For any the number of periods $T > 0$, The worst regret of PPAB is bounded by $O(MK^3\epsilon^{-1}\ln(TK))$.*

*Proof.* We mainly consider the reward loss of PPAB incurred by the unknown popularity of tasks. According to the definition of the regret and Lemma 1, we acquire the following upper bound of the regret:

$$rgt \leq \sum_{t=1}^{T}\sum_{i=1}^{K}\mu_{r_i^*} - \sum_{t=1}^{T}\mathbb{E}[\sum_{i\in\mathcal{M}^t}\mu_i^t] + \frac{T}{D}\sum_{i=K+1}^{M}\mu_{r_i^*}$$
$$\leq \Delta_{max}\sum_{i=1}^{M}\mathbb{E}[\chi_i^T] + T(M-K)/D$$
$$\leq \Delta_{max}M(1+2K\pi^2/3) + \Delta_{max}M\Xi + M\ln(T+2)$$
$$\leq \Delta_{max}M\max(\frac{(K+1)\Lambda^2\ln(TK)}{4\theta^2}, \frac{\sqrt{2}\Lambda\log(4/\delta)(\log T+1)}{(1-\theta)\epsilon})$$
$$+\Delta_{max}M(1+2K\pi^2/3) + M\ln(T+2)$$
$$= O(MK^3\epsilon^{-1}\ln(TK)).$$

This is a sub-linear regret bound about $T$. Now, the proof of the theorem is completed. □

### 0.3 Truthfulness Analysis

In order to guarantee that each requester will bid using its true valuation, we prove the truthfulness of PPAB.

**Theorem 3.** *The PPAB scheme has truthfulness in each period, i.e., each requester will bid truthfully.*

*Proof.* We analyze the scenario when a requester $r_i$ does not bid its true valuation, i.e., $b_i \neq \vartheta_i$. In the initial exploration phase, the payment is irrespective of the bid, i.e., each requester pays the platform $\vartheta_{min}$. Hence, the property of truthfulness is satisfied because the requester cannot manipulate the bid $b_i$ to increase its utility. After the initial exploration phase (i.e., $t \geq 2$), we consider two cases:
Case 1: Requester $r_i$ submits its true valuation $\vartheta_i$ and task $i$ is selected (i.e., $i \in \mathcal{M}^t$).

This implies that when the requester $r_i$ bids its true valuation, task $i$ can be selected. Therefore, there must be $\vartheta_i\widehat{\mathrm{U}}_i^{t-1} > b_{K+1}\widehat{\mathrm{U}}_{K+1}^{t-1}$ and $p_i^t = b_{K+1}\widehat{\mathrm{U}}_{K+1}^{t-1}/\widehat{\mathrm{U}}_i^{t-1}$. When the requester $r_i$ submits a forged bid, i.e.,
*Underbid* ($b_i < \vartheta_i$): If $b_i$ is such that $b_i\widehat{\mathrm{U}}_i^{t-1} < b_{K+1}\widehat{\mathrm{U}}_{K+1}^{t-1}$, task $i$ will be not selected according to the greedy selection policy. If $b_i$ is such that $b_{K+1}\widehat{\mathrm{U}}_{K+1}^{t-1} < b_i\widehat{\mathrm{U}}_i^{t-1} < \vartheta_i\widehat{\mathrm{U}}_i^{t-1}$, task $i$ will be selected. Nevertheless, the payment paid by requester $r_i$ still remains the same, that is, there is $p_i^t(b_i) = b_{K+1}\widehat{\mathrm{U}}_{K+1}^{t-1}/\widehat{\mathrm{U}}_i^{t-1}$. Thus, the requester will not get extra benefits from the underbid.
*Overbid* ($b_i > \vartheta_i$): According to the selection criteria $b_i\widehat{\mathrm{U}}_i^{t-1}$, task $i$ continues to win and the payment is still the same. That is, the utility of requester $r_i$ remains unchanged from such an overbid. Therefore, the overbid is inoperative.
Case 2: Requester $r_i$ submits its true valuation $\vartheta_i$ and task $i$ is not selected (i.e., $i \notin \mathcal{M}^t$).

This indicates that when requester $r_i$ bids its true valuation, task $i$ will not be selected, i.e., $\vartheta_i\widehat{\mathrm{U}}_i^{t-1} < b_{K+1}\widehat{\mathrm{U}}_{K+1}^{t-1}$. Under this circumstance, the utility of requester $r_i$ is zero, i.e., $\Psi_i^t = 0$. When the requester submits a forged bid:
*Underbid* ($b_i < \vartheta_i$): Owing to $b_i\widehat{\mathrm{U}}_i^{t-1} < b_{K+1}\widehat{\mathrm{U}}_{K+1}^{t-1}$, task $i$ continues to fail in the auction. Therefore, this behavior would not bring any advantage to the requester $r_i$.
*Overbid* ($b_i > \vartheta_i$): When the bid $b_i$ is such that $b_i\widehat{\mathrm{U}}_i^{t-1} >$

$b_{K+1}\widehat{\mathrm{U}}_{K+1}^{t-1} > \vartheta_i \widehat{\mathrm{U}}_i^{t-1}$, requester $r_i$ wins and pays $p_i^t(b_i) = b_{K+1}\widehat{\mathrm{U}}_{K+1}^{t-1}/\widehat{\mathrm{U}}_i^{t-1}$. However, the utility of requester $r_i$ will be less than zero (i.e., $\vartheta_i - p_i^t(b_i) < 0$). If $b_i$ is such that $b_{K+1}\widehat{\mathrm{U}}_{K+1}^{t-1} > b_i \widehat{\mathrm{U}}_i^{t-1} > \vartheta_i \widehat{\mathrm{U}}_i^{t-1}$, the overbid is not sufficient to make requester $r_i$ win. Therefore, bidding truthfully is a better choice for the requester $r_i$.

From the Myerson's theorem [51, 52], an auction mechanism is truthful if and only if it satisfies: the selection process is monotonic and the payment of a winner is the critical value. Due to the greedy selection policy, it is easy to prove the monotonicity. According to the analysis of two cases, we can also prove that $b_{K+1}\widehat{\mathrm{U}}_{K+1}^{t-1}/\widehat{\mathrm{U}}_i^{t-1}$ is the critical payment. Therefore, bidding the true valuation $\vartheta_i$ is the best strategy of requester $r_i$. In other words, the auction process proposed in the PPAB scheme is truthful for each period. The proof of the theorem is finished. □