Online Incentive Protocol Design for Reposting Service in Online Social Networks

HAORAN GU, College of Computer Science, Chongqing University, Chongqing, China
SHIYUAN ZHENG, The Chinese University of Hong Kong, Hong Kong, Hong Kong
XUDONG LIU, The Chinese University of Hong Kong, Hong Kong, Hong Kong
HONG XIE, School of Computer Science and Technology, University of Science and Technology of China, Hefei, China
JOHN C.S. LUI, The Chinese University of Hong Kong, Hong Kong, Hong Kong

Reposting plays an essential role in boosting visibility on online social networks (OSNs). In this paper, we study the problem of designing "reposting service" in an OSN to incentivize "transactions" between requesters (users who seek to enhance visibility) and suppliers (users who are willing to repost if certain incentives are given), and maximize the *welfare increase* accumulated through a given time horizon. We formulate a mathematical model for reposting which captures various factors like click-through rates (CTRs), requesters' valuations and suppliers' costs. We formulate the problem of maximizing the welfare increase via judiciously assigning suppliers to requesters from two aspects: (a) "user-centric" and (b) "platform-centric". The user-centric aspect deals with situations where requesters and suppliers collaborate and share valuations and costs. To address the challenge of unknown CTRs, we propose an online learning protocol and achieve a sub-linear regret. The platform-centric aspect corresponds to the scenario where users keep their valuations or costs private. To address the challenges of unknown CTRs, valuations and costs, we design an "explore-then-commit" online protocol. We prove the truthfulness of the proposed online protocol, and we also prove that this protocol has a sub-linear regret. Lastly, we conduct extensive experiments on six public datasets to evaluate the effectiveness and scalability of the proposed protocols.

CCS Concepts: • **Theory of computation** → Social networks;

Additional Key Words and Phrases: Social visibility, social network, mechanism design, online learning algorithms

ACM Reference Format:
Haoran Gu, Shiyuan Zheng, Xudong Liu, Hong Xie, and John C.S. Lui. 2025. Online Incentive Protocol Design for Reposting Service in Online Social Networks. *ACM Trans. Web* 19, 2, Article 20 (May 2025), 30 pages.
https://doi.org/10.1145/3696473

Authors' Contact Information: Haoran Gu, College of Computer Science, Chongqing University, Chongqing, Sichuan, China; e-mail: davidghr95@gmail.com; Shiyuan Zheng, The Chinese University of Hong Kong, Hong Kong, Hong Kong; e-mail: shira0905@gmail.com; Xudong Liu, The Chinese University of Hong Kong, Hong Kong, Hong Kong; e-mail: liuxd1215@gmail.com; Hong Xie (Corresponding author), School of Computer Science and Technology, University of Science and Technology of China, Hefei, China; e-mail: xiehong2018@foxmail.com; John C. S. Lui, The Chinese University of Hong Kong, Hong Kong, Hong Kong; e-mail: cslui@cse.cuhk.edu.hk.

ACM 1559-1131/2025/05-ART20
https://doi.org/10.1145/3696473

1 Introduction

Online Social Networks (OSNs) have become popular venues for users to create contents and share contents with friends, followers, and the like. The evolution of mobile social networks has further transformed the way that people get along with online contents. On one hand, the popularization of 4G and the birth of 5G make faster connections available and more affordable. Not only pictures and texts but also high-quality videos can be consumed during daily commutes. On the other hand, emerging mobile social platforms such as TikTok and Instagram lower the bar to create contents and make consuming and sharing of contents more convenient. With such a context, boosting the visibility of contents for users, i.e., making the contents reach a larger audience on the OSN has been a heated topic [1, 14, 21, 43]. The importance of large visibility can be well illustrated by the example of incentivized social advertising [4, 31], an emerging marketing model which provides monetization opportunities to users. For instance, a user can post an article or a video containing ads as a small portion to help a company advertise new products and get a "cut" from the advertising revenue. Then the larger visibility the content gets, the better reach of this product can be made, which indicates larger sales for the company and more revenue for content creators. However, one of the main obstacles to getting large visibility is that a content has to compete with many other contents [5] and the effect of "the strong get stronger" makes less popular contents buried by the most popular ones.

Reposting, a content-sharing behavior, plays an essential role in visibility boosting. For example, recently, an Atlanta-based e-sports organization Ghost Gaming posted a Tweet on X (formerly known as Twitter) "Enter to win a $2,000 NZXT and a $100 gift card to the Ghost Gaming shop!".[1] They also added that users who retweet this Tweet would be able to join the lucky draw which offers above valuable prizes to the winner. That Tweet got 8k retweets and 7.5k likes. By contrast, the previous Tweets posted by Ghost Gaming not involved in prize-giving reposting only got about 0.1k likes on average. Obviously, this Tweet successfully stimulated many users to repost, so that the content could finally reach many more users and get many more likes. One can refer to [25] for more examples that show the power of reposting on Twitter.

Reposting services prevail in OSNs in a variety of forms. For example, RepostExchange[2] is a platform powered by requesters and suppliers from SoundCloud,[3] where requesters are SoundCloud artists who will pay other users for reposting, since they can benefit from boosting the visibility of music records, and suppliers are SoundCloud users who will repost music if certain rewards are given. The payment and the reward are in a form called "credit", which serves as the currency on RepostExchange. Another example is Sina Weibo in China, a microblogging platform similar to X. The platform offers an official service named WeiTask,[4] where Weibo users can opt to participate. Requesters can launch recruitment to call for reposting, and the platform would select suppliers. If selected suppliers repost the content, they will receive rewards.

Note that the above-mentioned reposting services use simple heuristic pricing mechanisms, whose efficiency and effectiveness have no theoretical guarantees. Having no theoretical guarantees may lead the system to a "socially sub-optimal" state, which hurts the ecosystem of OSNs. Thus, it is important to design theoretically efficient and effective mechanisms. To the best of our knowledge, few works have studied this problem (please refer to Section 7 for more elaboration). In this paper, we *design protocols to incentivize reposting of contents with theoretical guarantees*. We assume all the users are rational. Specifically, there is a set of requesters who are content

[1] https://twitter.com/GhostEsports/status/1161381978073116673
[2] https://repostexchange.com
[3] https://soundcloud.com. This is a music platform powered by a global community of music artists and listeners.
[4] https://weirenwu.weibo.com

creators and want to increase their content visibility by requesting other users to repost. There is also a set of suppliers who are willing to repost requesters' contents when a certain reward is offered. A content is visible to the neighbors of the user who posts or reposts it. If a supplier s reposts requester r's contents, the visibility increase of r's content is the marginal increase of the number of users to whom the content is visible. *Click* is the actual action that users would take after viewing the post, which eventually contributes to the revenue of content creators. Without loss of generality, we use *click* to model a range of beneficial behaviors after viewing the content. For example, it can model a thumbs up to show agreement on the opinion of a post, or purchasing behavior after viewing an advertising post. We use **click-through rate (CTR)** to denote the probability of a click after viewing the post. A requester has a personal unit valuation for each click received and a supplier has a personal unit cost for each click contributed by his reposting. The objective is to maximize the social welfare increase accumulated throughout $T \in \mathbb{N}_+$ rounds, which is the total valuations increase of requesters minus the total cost of suppliers. The decision variable is the *assignment* from suppliers to requesters.

We formulate the decision problem from two different aspects: (a) "user-centric" and (b) "platform-centric". The user-centric aspect deals with the situation where requesters and suppliers collaborate to search for the optimal assignment among them. In this setting, requesters and suppliers would share their valuations and costs, e.g., they can set up social groups and report their valuations or costs. One challenge in searching for the optimal assignment is that the CTRs are unknown, making the social welfare increase unable to be calculated. We propose an online learning protocol to address this challenge. Our online learning protocol is built on the observation that the outcome of each assignment from suppliers to requesters reveals samples on the CTRs of requesters who are assigned at least one supplier. Note that for those requesters who are assigned no supplier at all, no sample on their CTRs is revealed. This leads to the challenge of balancing the exploration vs. exploitation tradeoff. In our online learning protocol, we employ a combinatorial multi-armed bandit algorithm to address this challenge. We also prove that our proposed protocol enjoys a sub-linear regret. The platform-centric aspect corresponds to the setting where requesters and suppliers do not collaborate and there exists a service like RepostExchange to determine the assignments. In this setting, requesters and suppliers would be asked to report (not necessarily truthfully) their unit valuations and costs to the service. Compared to the user-centric aspect, besides the challenge of unknown CTRs, one extra challenge is that requesters and suppliers usually do not have the incentive to report truthfully. We design a truthful "explore-then-commit" online protocol to address these challenges. In the exploration phase, our protocol uses several rounds to estimate the CTRs. In the commit phase, our protocol uses a charging and rewarding scheme which can incentivize requesters and suppliers to report their true unit valuations and costs. We also prove the regret for this truthful online protocol. Lastly, we conduct extensive experiments on real-world datasets to evaluate the efficiency and effectiveness of the proposed protocols. The highlights of our contributions include:

— We formulate the mathematical model and the welfare increase maximization problem for the reposting service from both the user-centric and platform-centric aspects.
— We propose an online learning protocol with a probable sub-linear regret to address the challenge of unknown CTRs for the user-centric aspect.
— We propose a truthful "explore-then-commit" online protocol with a sub-linear regret to address the challenge of unknown unit valuations and costs as well as unknown CTRs for the platform-centric aspect.
— Lastly, we conduct extensive experiments on six public datasets to evaluate the effectiveness and scalability of the proposed protocols. We also reveal fundamental insights into how

the social network structure influences the effectiveness and scalability of the proposed protocols.

The remainder of this paper is organized as follows. Section 2 presents the mathematical model of reposting service in OSNs and problem formulation. Section 3 presents the online protocol for user-centric welfare increase maximization. Section 4 presents the online incentive protocol for platform-centric welfare increase maximization. Section 5 shows experiments on real-world OSNs. Section 7 gives discussions on related literatures. Section 8 concludes the paper.

2 Model and Problem Formulation

In this section, we first present the mathematical model of the reposting service which qualities the visibility, valuations, costs, CTRs, welfare increase, and so on. Then we formulate the problem of assigning suppliers to requesters from both user-centric and platform-centric aspects.

2.1 The Model of Reposting Service

Consider an OSN characterized by a directed and unweighted graph $\mathcal{G} \triangleq (\mathcal{U}, \mathcal{E})$, where $\mathcal{U} \triangleq \{1, \ldots, N\}$ denotes a set of $N \in \mathbb{N}_+$ users and $\mathcal{E} \subseteq \mathcal{U} \times \mathcal{U}$ denotes a set of edges between users. On X-like OSNs, a direct edge (u', u) represents user u' follows user u. On Facebook like OSNs, a friendship link between u and u' can be modeled by two directed edges (u, u') and (u', u). The set of incoming neighbors of user $u \in \mathcal{U}$ is denoted by

$$\mathcal{N}_u \triangleq \{u' | u' \in \mathcal{U}, (u', u) \in \mathcal{E}\}.$$

Content posted by user u is visible to his incoming neighbors $\mathcal{N}_u$, and this content can also be visible to the incoming neighbors of user u', i.e., $\mathcal{N}_{u'}$, if user u' reposts it. However, the reposting behavior should not be taken for granted, since reposting is associated with some cost, e.g., the time, the social pressure of potentially annoying their incoming neighbors, and so on. Let $\mathcal{R} \subseteq \mathcal{U}$ denote a set of requesters who call for reposting to boost the visibility and are willing to pay for that. Let $\mathcal{S} \subseteq \mathcal{U}$ denote a set of suppliers who are willing to repost a content if some financial reward is provided. We then give the formal definition of visibility.

Definition 1 (Visible Set and Visibility). If supplier $s \in \mathcal{S} \cup \{0\}$ reposts a content created by requester $r \in \mathcal{R}$, the visible set of this content is defined as

$$\mathcal{V}_r(s) \triangleq \mathcal{N}_r \cup \mathcal{N}_s,$$

where $s = 0$ is to model that no supplier reposts this content and correspondingly, $\mathcal{N}_0 = \emptyset$. The visibility of this content is the cardinality of the visible set $|\mathcal{V}_r(s)|$.

Among the visible set of r's content, some users would click on this content. This fraction also can be interpreted as the probability of receiving a click once the content is viewed, and it is called the *click-through rate (CTR)*. We assume that this probability only depends on the content creator. Let θ_r denote the CTR of a content created by requester r, which reflects the overall content quality (i.e., attractiveness) of r. Each requester r has a normalized unit valuation $v_r \in [0, 1]$ for each click of his content. Each supplier s has a normalized unit cost $c_s \in [0, 1]$ for each increased click from his incoming neighbors. For ease of presentation, we denote the CTR vector, valuation vector and cost vector respectively as

$$\boldsymbol{\theta} \triangleq (\theta_r : r \in \mathcal{R}), \quad \boldsymbol{v} \triangleq (v_r : r \in \mathcal{R}), \quad \boldsymbol{c} \triangleq (c_s : s \in \mathcal{S}).$$

We consider a total number of $T \in \mathbb{N}_+$ rounds of assigning suppliers to requesters. Each round $t \in [T] \triangleq \{1, \ldots, T\}$ is associated with the same set of requesters $\mathcal{R}$ and the same set of suppliers $\mathcal{S}$. For the ease of presentation, we assume that users who are not suppliers, i.e., $\mathcal{U} \setminus \mathcal{S}$, do not repost contents when no payment is made. This assumption only simplifies the calculation of the visible set of content. If users who are not suppliers repost contents, then a content will spread faster in an OSN. In each round t, each requester creates and posts a new content. Without loss of generality, in each round, each requester is allowed to have at most one supplier repost his content, and a supplier is allowed to repost at most one requester's content. Note that this assumption is not a restriction, because the following trick of virtual requesters or virtual suppliers can handle the setting that a requester has multiple contents to repost or a supplier would like to repost multiple contents. In a situation where a requester has multiple contents, one can create multiple virtual copies of this requester while satisfying that each virtual requester has only one content to repost. In a situation where a supplier would like to repost multiple contents, one can create multiple virtual copies of this supplier while satisfying that each virtual supplier would repost only one content.

Let $a_{t,r} \in \mathcal{S} \cup \{0\}$ denote the supplier assigned to requester r in round t. Here $a_{t,r} = 0$ models that no supplier is assigned to requester r in round t. Denote the requesters who are assigned a supplier in round t as

$$\mathcal{R}'_t = \left\{ r | r \in \mathcal{R}, a_{t,r} \neq 0 \right\}.$$

Denote the *assignment profile* in round t as

$$\boldsymbol{a}_t = (a_{t,r} : r \in \mathcal{R}).$$

We define a valid assignment profile as follows.

Definition 2 (Valid Assignment Profiles). An assignment profile $\boldsymbol{a}_t$ is valid if it satisfies:

$$|\{r | a_{t,r} = s, r \in \mathcal{R}\}| \leq 1, \quad \forall s \in \mathcal{S}, \forall t \in [T]. \tag{1}$$

Condition (1) states that in a valid assignment profile, each supplier is assigned to at most one requester in each time step. We denote the set of all valid assignment profiles by

$$\mathcal{A} \triangleq \{\boldsymbol{a} | \boldsymbol{a} \text{ satisfies Equation (1)}\}.$$

Given time step t, we use $S_{(r, a_{t,r})}$ to denote the visibility increase of requester r due to supplier $a_{t,r}$'s reposting, formally

$$S_{(r, a_{t,r})} \triangleq |\mathcal{N}_{a_{t,r}} \setminus \mathcal{N}_r|.$$

Let $V(r, a_{t,r})$ denote the corresponding expected valuation increase of requester r, formally

$$V(r, a_{t,r}) \triangleq S_{(r, a_{t,r})} \theta_r v_r.$$

Let $C(r, a_{t,r})$ denote the corresponding expected cost of supplier $a_{t,r}$ for reposting r's content, formally

$$C(r, a_{t,r}) \triangleq S_{(r, a_{t,r})} \theta_r c_{a_{t,r}}.$$

Then, $V(r, a_{t,r}) - C(r, a_{t,r})$ quantifies the overall expected benefit of the requester-supplier pair $(r, a_{t,r})$ due to assigning supplier $a_{t,r}$ to requester r. Thus, the expected welfare increase associated with assignment profile $\boldsymbol{a}_t$ is denoted by

$$W(\boldsymbol{a}_t) = \sum_{r \in \mathcal{R}} \left(V(r, a_{t,r}) - C(r, a_{t,r}) \right).$$

Finally, we denote the expected *welfare increase* accumulated throughout T time steps associated with the assignment profile sequence $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_T$ as

$$W_T = \sum_{t \in [T]} W(\boldsymbol{a}_t),$$

which is the objective to maximize in the reposting service.

2.2 Problem Formulation

We formulate the assignment problem to maximize the cumulative welfare increase from both the user-centric and platform-centric aspects.

User-centric welfare increase maximization. We consider the problem that requesters and suppliers collaborate to maximize their total benefits, i.e., the welfare increase. They would share their unit valuations and unit costs, but they do not know the CTR vector $\boldsymbol{\theta}$.

Ensuring that suppliers and requesters share their costs and valuations is an economic issue. In particular, it is about how to divide the welfare increase among requesters and suppliers. Sharing costs and valuations can improve the assignment, which in turn improve the total welfare increase benefiting all requesters and suppliers. In other words, requesters and suppliers have the economic incentive to collaborate. To ensure they tell the truth, i.e., report their true valuations or costs, one needs to design a proper reward sharing rule. This is possible, as indicated by the platform-centric welfare increase problem. One can extend the rule to achieve this.

The challenge is to estimate the CTRs and maximize the cumulative welfare increase simultaneously. The optimal assignment profile can be stated as

$$\boldsymbol{a}^* \in \arg\max_{\boldsymbol{a}\in\mathcal{A}} W(\boldsymbol{a}).$$

Our objective is to design a protocol to select a sequence of assignment profiles $(\boldsymbol{a}_t, t \in [T])$ to minimize the regret, which is defined as follows:

$$R_T \triangleq \sum_{t=1}^{T} \left[W(\boldsymbol{a}^*) - W(\boldsymbol{a}_t)\right]. \tag{2}$$

Minimizing regret R_T is equivalent to maximizing welfare increase W_T, since regret R_T quantifies how well our protocols compared with the optimal assignment which has full knowledge. One can directly compare different methods using regret which can contribute to a better theoretical conclusion.

Platform-centric welfare increase maximization. We consider the problem that requesters and suppliers do not collaborate, i.e., they do not share unit valuations and unit costs. In this case, we design a reposting service for the platform to determine the assignment profiles. Compared with the user-centric setting, one additional challenge is that requesters' valuations and suppliers' costs are unknown to the platform. Our objective is to design a protocol to solicit the unit valuations and unit costs and also estimate the CTRs so that we can maximize the cumulative welfare increase or equivalently minimize the regret. Formally, the protocol is parameterized by the charging scheme

$$\boldsymbol{p}_t \triangleq (p_{t,r} : r \in \mathcal{R}),$$

and reward scheme

$$\boldsymbol{q}_t \triangleq (q_{t,s} : s \in \mathcal{S}),$$

where $p_{t,r}, q_{t,s} \in [0,1], \forall t \in [T]$. More specifically, at time step t, each requester is charged $p_{t,r}$ for each of his clicks increased due to the assignment, and each supplier is rewarded $q_{t,s}$ for each increased click contributed by him. The goal is to design a charging and reward scheme to incentivize requesters and suppliers to report their unit valuations and unit costs truthfully.

To provide a clear understanding of the notations used throughout this study, Table 1 presents a comprehensive description of each notation. This table includes the notation names and their definitions.

Table 1. Main Notations

Notation	Definition
$\mathcal{G}$	a directed and unweighted graph
$\mathcal{U}$	a set of $N \in \mathbb{N}_+$ users
$\mathcal{E}$	a set of edges between users
$\mathcal{N}_u$	the set of incoming neighbors of user $u \in \mathcal{U}$
$\mathcal{R}$	a set of requesters who call for reposting
$\mathcal{S}$	a set of suppliers who are willing to repost a content
$\boldsymbol{\theta}$	the click-through rate vector
$\boldsymbol{v}$	the valuation vector
$\boldsymbol{c}$	the cost vector
$\boldsymbol{a}$	assignment profile
$S_{(r,a_{t,r})}$	the visibility increase of requester r due to supplier $a_{t,r}$'s reposting
$V_{(r,a_{t,r})}$	the expected valuation increase of requester r at round t
$C_{(r,a_{t,r})}$	the expected cost increase of supplier $a_{t,r}$ at round t
$W(\boldsymbol{a}_t)$	the expected welfare increase with assignment profile $\boldsymbol{a}_t$
$\boldsymbol{a}^*$	the optimal assignment profile
$\boldsymbol{p}_t$	the charging scheme
$\boldsymbol{q}_t$	the reward scheme
$\boldsymbol{b}\|_{\mathcal{X}}$	the reported valuation profile of a subset of requesters $\mathcal{X} \subseteq \mathcal{R}$
$\boldsymbol{b}\|_{\mathcal{Y}}$	the reported cost profile of a subset of requesters $\mathcal{Y} \subseteq \mathcal{S}$
$\boldsymbol{b}\|_{\mathcal{R}}$	the reported unit valuation
$\boldsymbol{b}\|_{\mathcal{S}}$	the reported unit cost
U_r	the utility of a requester $r \in \mathcal{R}$
U_s	the utility of a supplier $s \in \mathcal{S}$
$N_{(r,s)}$	the increased number of clicks of r contributed by s's reposting

2.3 Discussion on Model Assumptions

Our model makes certain simplifications in order to strike a delicate balance between the expressiveness and mathematical tractability. One simplification is that the visibility is modeled by one-hop neighbors. In practice, neighbors of suppliers may repost interesting contents, leading to multi-hop spreading of contents. This can be captured by multi-hop neighbors [66]. Our method can be straightforwardly extended to this setting. Another simplification is the model of virtual requester or virtual supplier. Note that this modeling trick mainly simplifies the presentation. It does not increase algorithmic complexity, since the algorithmic problem is to assign each content to a supplier and assigning multiple contents to the same supplier is equivalent to assigning these contents to the corresponding virtual suppliers in a one-to-one manner.

A requester creates and posts different contents across different time slots. Our model treats different contents independently in quantifying social visibility. Given two requesters r_1 and r_2 in time slot t, it may happen that the intersection $\mathcal{N}_{a_{t,r_1}} \cap \mathcal{N}_{a_{t,r_2}}$ is non-empty. Each user in the intersection $\mathcal{N}_{a_{t,r_1}} \cap \mathcal{N}_{a_{t,r_2}}$ would see contents created by requester r_1 and r_2, respectively, and its clicking behavior across these two contents is modeled as an independent click. We are aware that the object (or types) of contents would have a subtle impact, i.e., complementary or competition, on the click behavior and thus affect the social visibility qualification. We leave it as an interesting future work for further study.

When CTRs change infrequently over time, one can divide the time slots into different intervals according to the change of CTRs, such that each interval does not contain any change point of

CTRs. Our model and algorithms can be applied to handle each interval independently. To further improve the learning efficiency, one can cluster intervals based on CTRs such that in each cluster the CTRs have a slight difference. Our model and algorithms can be applied to handle each cluster independently. When CTRs change frequently, one can apply non-stationary combinatorial bandits to handle them [30, 38, 65]. We leave it as an interesting future work.

3 User-centric Welfare Increase Maximization

This section studies the problem of user-centric welfare increase maximization, where requesters and suppliers share their unit valuations and unit cost. We start from a warm-up case where the CTR vector $\boldsymbol{\theta}$ is accessible, then design a protocol to address the challenge of unknown $\boldsymbol{\theta}$.

3.1 Offline Optimal Assignment Protocol

Optimal assignment. We first consider the setting that the CTR vector $\boldsymbol{\theta}$ is known to users and the goal is to design a protocol that can find out the optimal assignment $\boldsymbol{a}^*$ to maximize the welfare increase. This protocol serves as a building block to study the setting that $\boldsymbol{\theta}$ is unknown to users. Note that under this setting with full knowledge there is no need to vary the assignment with time step t, and the welfare increase in different time steps are identical. Thus, we omit the subscript or superscript of t for ease of presentation.

To facilitate the protocol design, we first construct a complete weighted and undirected bipartite graph denoted by $\mathcal{B} = (\mathcal{R}, \mathcal{S}, \boldsymbol{E})$, where $\mathcal{R}$ and $\mathcal{S}$ are two disjoint node sets representing requesters and suppliers respectively, and $\boldsymbol{E} \triangleq [e_{r,s} : r \in \mathcal{R}, s \in \mathcal{S}]$ denotes the weights. The weight of edge (r, s) is set as

$$e_{r,s} = V(r,s) - C(r,s),$$

which is the expected welfare increase of requester-supplier pair (r, s). We define a matching in the bipartite graph $\mathcal{B}$ as follows.

Definition 3 (Matching). The edge set $\mathcal{M} \subseteq \mathcal{R} \times \mathcal{S}$ of graph $\mathcal{B}$ is a matching if it satisfies that $|\{s|(r,s) \in \mathcal{M}\}| \leq 1, \forall r \in \mathcal{R}$ and $|\{r|(r,s) \in \mathcal{M}\}| \leq 1, \forall s \in \mathcal{S}$.

The following lemma states the connection between an assignment profile $\boldsymbol{a}$ and a matching in graph $\mathcal{B}$.

Lemma 1. *An assignment profile $\boldsymbol{a}$ can be mapped into a matching $\mathcal{M}(\boldsymbol{a}) = \{(r, a_r)|r \in \mathcal{R}, a_r \neq 0\}$ which satisfies*

$$W(\boldsymbol{a}) = \sum_{(r,s)\in\mathcal{M}(\boldsymbol{a})} e_{r,s}.$$

A matching $\mathcal{M}$ in $\mathcal{B}$ can be mapped into an assignment profile $\boldsymbol{a}(\mathcal{M})$ with $a_r(\mathcal{M}) = \sum_{s\in\mathcal{S}} s\mathbb{I}_{\{(r,s)\in\mathcal{M}\}}$ which satisfies

$$\sum_{(r,s)\in\mathcal{M}} e_{r,s} = W(\boldsymbol{a}(\mathcal{M})).$$

Lemma 1 implies that an effective approach to find the optimal assignment profile is to locate the maximum weighted matching and then map it to an assignment profile $\boldsymbol{a}^*$. Based on this observation, Protocol 1 outlines the protocol `OptAssign` to find the optimal assignment profile. In protocol `OptAssign`, we first construct the complete weighted bipartite graph $\mathcal{B}$. Then we find the maximum weighted matching via the algorithm `MaxWeightMatching`$(\mathcal{R}, \mathcal{S}, \boldsymbol{E})$. There are a variety of implementations of `MaxWeightMatching` and one example is the Hungarian algorithm [29]. Lastly, we map the maximum weighted matching into the optimal assignment.

Approximate protocol. Since it is computationally expensive to find a maximum weight matching for a large-scale graph using exact algorithms, e.g., $O(N^3)$ if using the Hungarian

Protocol 1: `OptAssign`$(\mathcal{R}, \mathcal{S}, \boldsymbol{\theta}, \boldsymbol{v}, \boldsymbol{c})$.

Construct complete undirected bipartite graph $\mathcal{B} = (\mathcal{R}, \mathcal{S}, \boldsymbol{E})$
$\mathcal{M}^* \leftarrow$`MaxWeightMatching`$(\mathcal{B})$
$a_r(\mathcal{M}^*) = \sum_{s \in \mathcal{S}} s \mathbb{I}_{\{(r,s) \in \mathcal{M}^*\}}$
return assignment profile $(a_r(\mathcal{M}^*) : r \in \mathcal{R})$

Protocol 2: `GreedyAssign`$(\mathcal{R}, \mathcal{S}, \boldsymbol{\theta}, \boldsymbol{v}, \boldsymbol{c})$.

Construct complete undirected bipartite graph $\mathcal{B} = (\mathcal{R}, \mathcal{S}, \boldsymbol{E})$
Initialize $\mathcal{M} = \emptyset$
for all $(r, s) \in \boldsymbol{E}$ in descending order by weights **do**
    if (r, s) is still in $\mathcal{B}$ **then**
        add (r, s) to $\mathcal{M}$.
        delete r and s from $\mathcal{B}$.
    else
        skip this edge and continue.
    end if
end for
$a_r(\mathcal{M}) = \sum_{s \in \mathcal{S}} s \mathbb{I}_{\{(r,s) \in \mathcal{M}\}}$
return assignment profile $(a_r(\mathcal{M}) : r \in \mathcal{R})$

algorithm, we propose an approximation protocol to improve computational efficiency. Protocol 2 outlines `GreedyAssign`, which uses a greedy approach for approximating the maximum welfare increase. After constructing the bipartite graph, we rank all the edges in descending order by their weights and traverse them according to this order. If an edge (r, s) still exists in graph $\mathcal{B}$, we add it into the matching. Then delete the corresponding requester and supplier, as well as their adjacent edges from graph $\mathcal{B}$. We repeat until there is no edge left in $\mathcal{B}$. Finally, we get a matching and we map it to the corresponding assignment profile which is the result of the approximate protocol.

Theorem 1. *Protocol* `GreedyAssign` *(Protocol 2) guarantees an approximation ratio of* $1/2$*. The running time complexity of* `GreedyAssign` *is bounded by* $O(|\boldsymbol{E}| \log |\boldsymbol{E}|)$.

Remark. Theorem 1 states that `GreedyAssign` can achieve a guaranteed approximation ratio of at least $1/2$ for user-centric welfare increase with much smaller time complexity than `OptAssign` which uses an exact search algorithm like the Hungarian algorithm. The technical proof of Theorem 1 is presented in supplementary file.

3.2 Online Learning Protocol

Protocol design. Now we use Protocol 1 as a building block to study the assignment problem in the user-centric aspect where the CTR vector $\boldsymbol{\theta}$ is unknown to requesters and suppliers. In this setting, one needs to balance the exploration and exploitation tradeoff. We achieve this tradeoff via the **upper confidence bound (UCB)** method [6]. At time step t, for each requester, an unbiased estimator of θ_r can be

$$\widehat{\theta}_r = \frac{\sum_{\tau \in [t]} \frac{N^{(\tau)}_{(r, a_{\tau,r})}}{S_{(r, a_{\tau,r})}} \mathbb{I}\{a_{\tau,r} \neq 0\}}{\sum_{\tau \in [t]} \mathbb{I}\{a_{\tau,r} \neq 0\}}, \tag{3}$$

where $N^{(\tau)}_{(r, a_{\tau,r})}$ is the observed number of increased clicks of r received due to $a_{\tau,r}$'s reposting at time step τ. Based on Equation (3), we apply Hoeffding inequality to derive UCB index for θ_r. Then, we apply these UCB indexes to Protocol 1 for selecting the assignment profile.

Protocol 3: `OnlineAssign`$(\mathcal{R}, \mathcal{S}, \boldsymbol{v}, \boldsymbol{c})$.

Input: $\boldsymbol{v}, \boldsymbol{c}, (\mathcal{N}_u : \forall u \in \mathcal{U})$
for $t = 1$ to $|\mathcal{R}|$ **do**
  Select r from $\{r | r \in \mathcal{R}, N_r = 0\}$
  Select a random supplier s
  Select an assignment profile $\boldsymbol{a}_t$:
  $(a_{t,r} = s; a_{t,\bar{r}} = 0, \forall \bar{r} \in \mathcal{R} \setminus \{r\})$
  Observe number of increased clicks $N^{(t)}_{(r,s)}$
  Initialize $N_r = 1$
  Initialize $\bar{\theta}_r = N^{(t)}_{(r,s)} / S_{(r,s)}$
end for
for $t = |\mathcal{R}| + 1$ to T **do**
  Denote $\bar{\boldsymbol{\theta}}^+ = (\bar{\theta}^+_r : r \in \mathcal{R})$
  $\boldsymbol{a}_t \leftarrow$`OptAssign`$(\mathcal{R}, \mathcal{S}, \bar{\boldsymbol{\theta}}^+, \boldsymbol{v}, \boldsymbol{c})$
  Observe $N^{(t)}_{(r, a_{t,r})}$ for $\forall r \in \mathcal{R}'_t$
  for $r \in \mathcal{R}'_t$ **do**
    Update $N_r = N_r + 1$
    Update $\bar{\theta}_r = ((N_r - 1)\bar{\theta}_r + N^{(t)}_{(r, a_{t,r})} / S_{(r, a_{t,r})}) / N_r$
    Update $\bar{\theta}^+_r = \bar{\theta}_r + \sqrt{3 \log t / 2N_r}$
  end for
end for

Protocol 3 outlines details of the above UCB-based method, leading to our protocol `OnlineAssign`. Initially, we use $|\mathcal{R}|$ rounds of forced exploration to obtain samples of the CTR for each requester r. In the meantime, we keep track of some statistics of the assignment history: N_r which is the number of times the requester r has been assigned a supplier till the current round; $\bar{\theta}_r$ which is the empirical mean of sampled CTRs till the current round; and the UCB index $\bar{\theta}^+_r$ which adds an adjustment term to $\bar{\theta}_r$. In the remaining rounds after the forced exploration, we take the UCB index vector $\bar{\boldsymbol{\theta}}^+ = (\bar{\theta}^+_r : r \in \mathcal{R})$ as the input CTR vector of oracle `OptAssign` (Protocol 1) to decide the best assignment profile for this round, and we update N_r, $\bar{\theta}_r$ and $\bar{\theta}^+_r$ for each requester according to the observed clicks.

Protocol analysis. First, the time complexity of Protocol 3 mainly depends on the number of calls of oracle `OnlineAssign` which is smaller than T, as well as the implementation and the time complexity of `OnlineAssign`. Besides, the following theorem states the performance guarantee of Protocol 3.

Theorem 2. *For all $T > |\mathcal{R}|$, protocol* `OptAssign` *(Protocol 3) achieves a sub-linear regret*

$$\mathbb{E}[R_T] \leq O(k \log(T)),$$

where k is defined as:

$$\begin{aligned} k &\triangleq \max_{(r,s) \in \mathcal{R} \times \mathcal{S}} S^2_{(r,s)} \frac{6 \min(|\mathcal{R}|, |\mathcal{S}|)^3}{\Delta^2_{min}} \Delta_{max}, \\ \Delta_{max} &\triangleq \max_{\boldsymbol{a} \in \mathcal{A} \setminus \{\boldsymbol{a}^*\}} W(\boldsymbol{a}^*) - W(\boldsymbol{a}), \\ \Delta_{min} &= \min_{\boldsymbol{a} \in \mathcal{A} \setminus \{\boldsymbol{a}^*\}} W(\boldsymbol{a}^*) - W(\boldsymbol{a}). \end{aligned}$$

Remark. Theorem 2 states that the regret of protocol `OnlineAssign` is sub-linear to T, which implies the average single-round regret decreases with T and the average single-round welfare increase asymptotically approaches the optimal $W(\boldsymbol{a}^*)$.

The constant parameter 6 is due to tuning the confidence interval in the proof. In particular, its precise value is smaller than 6 but in a complicated form. We make it 6 for the purpose of making the expression clean. The technical proof of Theorem 2 is presented in supplementary file [20].

4 Platform-centric Welfare Increase Maximization

In this section, we consider the problem of platform-centric welfare increase maximization where unit valuations and unit costs are private information and not accessible to the platform. We first study a setting where the CTR vector $\boldsymbol{\theta}$ is known to the platform, then we generalize it to the setting where $\boldsymbol{\theta}$ is unknown.

4.1 Truthful Offline Protocol

We first consider the setting that the CTR vector $\boldsymbol{\theta}$ is known, but the requesters' unit valuations and suppliers' unit costs are unknown to the platform. The protocol for this setting serves as a building block for the setting where $\boldsymbol{\theta}$ is also unknown. The core idea is to design a protocol to solicit true unit valuations from requesters and true unit costs from suppliers, and then apply Protocol 1 to find the optimal assignment profile to optimize welfare increase. Note that such protocol design is independent of time step t. Thus, we omit the subscript or superscript of t for ease of presentation.

Protocol design. Before assigning suppliers to requesters, the platform asks each requester $r \in \mathcal{R}$ to report their unit valuation b_r, and asks each supplier $s \in \mathcal{S}$ to report their unit cost b_s. We define the reported valuation profile of a subset of requesters $\mathcal{X} \subseteq \mathcal{R}$ as

$$\boldsymbol{b}|_{\mathcal{X}}=(b_u : u \in \mathcal{X}),$$

and the reported cost profile of a subset of suppliers $\mathcal{Y} \subseteq \mathcal{S}$ as

$$\boldsymbol{b}|_{\mathcal{Y}}=(b_u : u \in \mathcal{Y}).$$

We then use `OptAssign` (Protocol 1) as an oracle to get the optimal assignment profile. Note that here we use the reported unit valuations $\boldsymbol{b}|_{\mathcal{R}}$ and reported unit costs $\boldsymbol{b}|_{\mathcal{S}}$ as the input unit valuations and unit costs of `OptAssign`. One can easily see that if requesters and suppliers report truthfully, i.e., $\boldsymbol{b}|_{\mathcal{R}} = \boldsymbol{v}$ and $\boldsymbol{b}|_{\mathcal{S}} = \boldsymbol{c}$, then the output assignment profile is exactly the optimal assignment profile. However, the challenge lies in designing a protocol to stimulate requesters and suppliers to report truthfully. Before diving into the protocol design, we define the following optimal assignment function to assist our presentation.

Definition 4 (Optimal Assignment Function and Pseudo Welfare Increase). Suppose only a subset of requesters $\mathcal{X} \subseteq \mathcal{R}$ and a subset of suppliers $\mathcal{Y} \subseteq \mathcal{S}$ are available. Define the corresponding optimal assignment function as $\boldsymbol{A}^*(\mathcal{X}, \mathcal{Y}, \boldsymbol{\theta}) \triangleq (A^*_u(\mathcal{X}, \mathcal{Y}, \boldsymbol{\theta}) : u \in \mathcal{X})$, where

$$\boldsymbol{A}^*(\mathcal{X}, \mathcal{Y}, \boldsymbol{\theta}) = \texttt{OptAssign}(\mathcal{X}, \mathcal{Y}, \boldsymbol{\theta}, \boldsymbol{b}|_{\mathcal{X}}, \boldsymbol{b}|_{\mathcal{Y}}).$$

Furthermore, we define the corresponding pseudo welfare increase with respect to $\mathcal{X}$ and $\mathcal{Y}$ as

$$W^*(\mathcal{X}, \mathcal{Y}, \boldsymbol{\theta}) = \sum_{u \in \mathcal{X}} S_{(u, A^*_u(\mathcal{X}, \mathcal{Y}, \boldsymbol{\theta}))} \theta_u (b_u - b_{A^*_u(\mathcal{X}, \mathcal{Y}, \boldsymbol{\theta})}).$$

Namely, $\boldsymbol{A}^*(\mathcal{X}, \mathcal{Y}, \boldsymbol{\theta})$ is the optimal assignment profile when only a subset of requesters $\mathcal{X} \subseteq \mathcal{R}$ and a subset of suppliers $\mathcal{Y} \subseteq \mathcal{S}$ are available and their unit valuations and unit costs are assumed to be $\boldsymbol{b}|_{\mathcal{X}}$ and $\boldsymbol{b}|_{\mathcal{Y}}$.

Protocol 4: `TruthOfflineAssign`$(\mathcal{R}, \mathcal{S}, \boldsymbol{\theta})$.

```
1: Select assignment a = A*(R, S, θ)
2: for r ∈ R do
3:     Requester r is charged by p_r(b|_R, b|_S, θ)
4:     s ← a_r
5:     Supplier s gets reward q_s(b|_R, b|_S, θ)
6: end for
```

Based on the **Vickrey-Clarke-Groves (VCG)** mechanism [13, 19, 56], we design the charging scheme and reward scheme where each requester (resp., supplier) is charged (resp., rewarded) for the *externality*, which is the difference between the welfare increase in the absence of him and the welfare increase in the presence of him. In the presence of requester $r \in \mathcal{R}$, the expected welfare increase of other requesters $\mathcal{R} \setminus \{r\}$ and suppliers $\mathcal{S}$ can be calculated as

$$\underbrace{W^*(\mathcal{R}, \mathcal{S}, \boldsymbol{\theta})}_{\text{welfare increase of } \mathcal{S} \text{ and } \mathcal{R}} \quad - \quad \underbrace{S_{(r, A^*_r(\mathcal{R}, \mathcal{S}, \boldsymbol{\theta}))} \theta_r b_r}_{\text{valuation increase of } r},$$

In the absence of requester $r \in \mathcal{R}$, the expected welfare increase of other requesters $\mathcal{R} \setminus \{r\}$ and suppliers $\mathcal{S}$ can be calculated as $W^*(\mathcal{R} \setminus \{r\}, \mathcal{S}, \boldsymbol{\theta})$. To stimulate the requester $r \in \mathcal{R}$ to report his true unit valuations, we design a charging scheme where the platform charges r by the marginal deduction on the welfare increase of other requesters $\mathcal{R} \setminus \{r\}$ and suppliers $\mathcal{S}$, formally,

$$p_r(\boldsymbol{b}|_{\mathcal{R}}, \boldsymbol{b}|_{\mathcal{S}}, \boldsymbol{\theta}) = \underbrace{W^*(\mathcal{R} \setminus \{r\}, \mathcal{S}, \boldsymbol{\theta})}_{\text{welfare increase in the absence of } r} - \underbrace{\left[W^*(\mathcal{R}, \mathcal{S}, \boldsymbol{\theta}) - S_{(r, A^*_r(\mathcal{R}, \mathcal{S}, \boldsymbol{\theta}))} \theta_r b_r\right]}_{\text{welfare increase in the presence of } r}. \tag{4}$$

The above can be interpreted as the loss of other participating users' welfare increase due to the existence of r. Note that this charging scheme $p_r(\boldsymbol{b}|_{\mathcal{R}}, \boldsymbol{b}|_{\mathcal{S}}, \boldsymbol{\theta})$ applies to all requesters $r \in \mathcal{R}$.

Similarly, we calculate the reward for suppliers as follows. In the presence of the supplier $s \in \mathcal{S}$, the welfare increase to requesters $\mathcal{R}$ and other suppliers $\mathcal{S} \setminus \{s\}$ can be calculated as

$$\underbrace{W^*(\mathcal{R}, \mathcal{S}, \boldsymbol{\theta})}_{\text{welfare increase of } \mathcal{R} \text{ and } \mathcal{S}} \quad + \underbrace{S_{(A^{*-1}_s(\mathcal{R}, \mathcal{S}, \boldsymbol{\theta}), s)} \theta_{A^{*-1}_s(\mathcal{R}, \mathcal{S}, \boldsymbol{\theta})} b_s}_{\text{cost of supplier } s}.$$

where $A^{*-1}_s(\mathcal{R}, \mathcal{S}, \boldsymbol{\theta})$ denotes the requester who is assigned to supplier s under the assignment profile $\boldsymbol{A}^*(\mathcal{R}, \mathcal{S}, \boldsymbol{\theta})$, and we set $A^{*-1}_s(\mathcal{R}, \mathcal{S}, \boldsymbol{\theta}) = 0$ by default when s is not assigned to any requester. In the absence of supplier $s \in \mathcal{S}$, the welfare increase to requesters $\mathcal{R}$ and other suppliers $\mathcal{S} \setminus \{s\}$ can be calculated as $W^*(\mathcal{R}, \mathcal{S} \setminus \{s\}, \boldsymbol{\theta})$. To stimulate the supplier $s \in \mathcal{S}$ to report his true unit costs, we design a reward scheme where the platform rewards s by the marginal contribution to the welfare increase of requesters $\mathcal{R}$ and other suppliers $\mathcal{S} \setminus \{s\}$, formally,

$$q_s(\boldsymbol{b}|_{\mathcal{R}}, \boldsymbol{b}|_{\mathcal{S}}, \boldsymbol{\theta}) = \underbrace{W^*(\mathcal{R}, \mathcal{S}, \boldsymbol{\theta}) + S_{(A^{*-1}_s(\mathcal{R}, \mathcal{S}, \boldsymbol{\theta}), s)} \theta_{A^{*-1}_s(\mathcal{R}, \mathcal{S}, \boldsymbol{\theta})} b_s}_{\text{welfare increase in the presence of } s} - \underbrace{W^*(\mathcal{R}, \mathcal{S} \setminus \{s\}, \boldsymbol{\theta})}_{\text{welfare increase in the absence of } s}. \tag{5}$$

The above can be interpreted as the gain of other participating users' welfare increase due to the existence of s. Note that this reward scheme $q_s(\boldsymbol{b}|_{\mathcal{R}}, \boldsymbol{b}|_{\mathcal{S}}, \boldsymbol{\theta})$ applies to all suppliers $s \in \mathcal{S}$. To summarize the above derivations, Protocol 4 states our truthful offline protocol.

Protocol analysis. We analyze the properties of our protocol specified by the charging scheme $p_r(\boldsymbol{b}|_{\mathcal{R}}, \boldsymbol{b}|_{\mathcal{S}}, \boldsymbol{\theta}), \forall r \in \mathcal{R}$ and the reward scheme $q_s(\boldsymbol{b}|_{\mathcal{R}}, \boldsymbol{b}|_{\mathcal{S}}, \boldsymbol{\theta}), \forall s \in \mathcal{S}$. We first define the utility for requesters and suppliers.

Definition 5 (Utility). Given the reported profile $(\boldsymbol{b}|_{\mathcal{R}}, \boldsymbol{b}|_{\mathcal{S}})$, the CTR vector $\boldsymbol{\theta}$, and the assignment profile $A^*(\mathcal{R}, \mathcal{S}, \boldsymbol{\theta})$ (i.e., the result of `OptAssign`$(\mathcal{R}, \mathcal{S}, \boldsymbol{\theta}, \boldsymbol{b}|_{\mathcal{R}}, \boldsymbol{b}|_{\mathcal{S}})$), then the utility of a requester $r \in \mathcal{R}$ is defined as:

$$U_r(\boldsymbol{b}|_{\mathcal{R}}, \boldsymbol{b}|_{\mathcal{S}}; \boldsymbol{\theta}) \triangleq v_r N_{(r, A_r^*(\mathcal{R}, \mathcal{S}, \boldsymbol{\theta}))} - p_r(\boldsymbol{b}|_{\mathcal{R}}, \boldsymbol{b}|_{\mathcal{S}}, \boldsymbol{\theta}),$$

and the utility of a supplier $s \in \mathcal{S}$ is also defined as his marginal gain:

$$U_s(\boldsymbol{b}|_{\mathcal{R}}, \boldsymbol{b}|_{\mathcal{S}}; \boldsymbol{\theta}) \triangleq q_s(\boldsymbol{b}|_{\mathcal{R}}, \boldsymbol{b}|_{\mathcal{S}}, \boldsymbol{\theta}) - c_s N_{(A_s^{*-1}(\mathcal{R}, \mathcal{S}, \boldsymbol{\theta}), s)},$$

where $N_{(r,s)}$ is the increased number of clicks of requester r contributed by s's reposting, i.e., the number of clicks from users in $\mathcal{N}_s \setminus \mathcal{N}_r$.

In the following, we introduce some conceptions of mechanism design from the perspective of reposting service.

Definition 6 (Efficient). A protocol is efficient if the selected assignment profile $\boldsymbol{a}$ maximizes the welfare increase, i.e., $\boldsymbol{a} \in \arg\max_{\boldsymbol{a} \in \mathcal{A}} W(\boldsymbol{a})$.

The efficient property states that by using this protocol, the assignment profile achieves the maximum welfare increase.

Definition 7 (Dominant Strategy Incentive Compatible (DSIC) [40]). A protocol is DSIC (or truthful) if it satisfies the following conditions. For $\forall r \in \mathcal{R}, \forall b_r, \forall \boldsymbol{b}|_{\mathcal{R} \setminus \{r\}}, \forall \boldsymbol{b}|_{\mathcal{S}}$ we have

$$\mathbb{E}[U_r((v_r, \boldsymbol{b}|_{\mathcal{R} \setminus \{r\}}), \boldsymbol{b}|_{\mathcal{S}}; \boldsymbol{\theta})] \geq \mathbb{E}[U_r((b_r, \boldsymbol{b}|_{\mathcal{R} \setminus \{r\}}), \boldsymbol{b}|_{\mathcal{S}}; \boldsymbol{\theta})],$$

and for $\forall s \in \mathcal{S}, \forall b_s, \forall \boldsymbol{b}|_{\mathcal{R}}, \forall \boldsymbol{b}|_{\mathcal{S} \setminus \{s\}}$ we have

$$\mathbb{E}[U_s(\boldsymbol{b}|_{\mathcal{R}}, (c_s, \boldsymbol{b}|_{\mathcal{S} \setminus \{s\}}); \boldsymbol{\theta})] \geq \mathbb{E}[U_s(\boldsymbol{b}|_{\mathcal{R}}, (b_s, \boldsymbol{b}|_{\mathcal{S} \setminus \{s\}}); \boldsymbol{\theta})].$$

The DISC property states that reporting unit valuation (resp., cost) truthfully is a weakly-dominant strategy for each requester (resp., supplier). Given that users are all rational, DSIC implies that all requesters and suppliers will report truthfully.

Definition 8 (Ex-interim Individually Rational (EIIR)). A protocol is EIIR if for $\forall r \in \mathcal{R}$ we have

$$\mathbb{E}[U_r((v_r, \boldsymbol{b}|_{\mathcal{R} \setminus \{r\}}), \boldsymbol{b}|_{\mathcal{S}}; \boldsymbol{\theta})] \geq 0,$$

and for $\forall s \in \mathcal{S}$ we have

$$\mathbb{E}[U_s(\boldsymbol{b}|_{\mathcal{R}}, (c_s, \boldsymbol{b}|_{\mathcal{S} \setminus \{s\}}); \boldsymbol{\theta})] \geq 0.$$

The EIIR property states that participating in and reporting unit valuation/cost truthfully will not lead to a negative expected utility. This guarantees that requesters and suppliers have the incentive to participate in the reposting service.

THEOREM 3. *The* `TruthOfflineAssign` *(Protocol 4) is efficient, DSIC and EIIR.*

Remark. Theorem 3 states that our proposed protocol can guarantee that requesters and suppliers would achieve the optimal and non-negative utilities by reporting their unit valuations/costs truthfully. As a result, the protocol can guarantee the maximum welfare increase even without given unit valuations and unit costs at first. The detailed proof is presented in supplementary file.

Protocol 5: `StrategicOnlineAssign`$(\mathcal{R}, \mathcal{S})$.

Input: $(\mathcal{N}_u : \forall u \in \mathcal{U})$
Solicit unit valuations $\boldsymbol{b}|_{\mathcal{R}}$ and unit costs $\boldsymbol{b}|_{\mathcal{S}}$
Initialize $N_r = 0, \bar{\theta}_r = 0, \forall r \in \mathcal{R}$
for $t = 1$ to $\max\{\gamma, \frac{|\mathcal{R}|\gamma}{|\mathcal{S}|}\}$ **do**
  Select an assignment $\boldsymbol{a}_t$, s.t. $|\mathcal{M}(\boldsymbol{a}_t)| = \min\{|\mathcal{R}|, |\mathcal{S}|\}$
  for $r \in \mathcal{R}'_t$ **do**
    Observe $N^{(t)}_{(r, a_{t,r})}$
    Update $N_r = N_r + 1$
    Update $\bar{\theta}_r = ((N_r - 1)\bar{\theta}_r + N^{(t)}_{(r, a_{t,r})}/S_{(r, a_{t,r})})/N_r$
  end for
end for
Denote $\bar{\boldsymbol{\theta}} = (\bar{\theta}_r : r \in \mathcal{R})$
Denote $\hat{\boldsymbol{a}}^* \leftarrow$`OptAssign`$(\mathcal{R}, \mathcal{S}, \bar{\boldsymbol{\theta}}, \boldsymbol{b}|_{\mathcal{R}}, \boldsymbol{b}|_{\mathcal{S}})$
for $t = \max\{\gamma, \frac{|\mathcal{R}|\gamma}{|\mathcal{S}|}\} + 1$ to T **do**
  Select assignment $\boldsymbol{a}_t = \boldsymbol{A}^*(\mathcal{R}, \mathcal{S}, \bar{\boldsymbol{\theta}})$
  for $r \in \mathcal{R}'_t$ **do**
    r is charged $p_{t,r} = p_r(\boldsymbol{b}|_{\mathcal{R}}, \boldsymbol{b}|_{\mathcal{S}}, \bar{\boldsymbol{\theta}})$
    $s = a_{t,r}$ is rewarded $q_{t,s} = q_s(\boldsymbol{b}|_{\mathcal{R}}, \boldsymbol{b}|_{\mathcal{S}}, \bar{\boldsymbol{\theta}})$
  end for
end for

4.2 Truthful Online Protocol

Now we study the most challenging setting where the CTRs $\boldsymbol{\theta}$, requesters' unit valuations $\boldsymbol{v}$ and suppliers' unit costs $\boldsymbol{c}$ are all unknown to the platform.

Protocol design. Since $\boldsymbol{\theta}$ is unknown, the charging scheme $p_r(\boldsymbol{b}|_{\mathcal{R}}, \boldsymbol{b}|_{\mathcal{S}}, \boldsymbol{\theta})$ and the reward scheme $q_s(\boldsymbol{b}|_{\mathcal{R}}, \boldsymbol{b}|_{\mathcal{S}}, \boldsymbol{\theta})$ which depend on $\boldsymbol{\theta}$ are also unable to calculate. To address this challenge, we propose an "explore-then-commit" online protocol `StrategicOnlineAssign`, which is outlined in Protocol 5. This protocol has two phases: the exploration phase and the commit phase. The exploration phase runs for $\max\{\gamma, \gamma|\mathcal{R}|/|\mathcal{S}|\}$ rounds so that each requester is selected for at least γ rounds. Exploration rounds do not involve charges and rewards. After the exploration phase, we use $\bar{\boldsymbol{\theta}}$, the empirical mean of CTR samples, to estimate $\boldsymbol{\theta}$. Then, the protocol goes into the commit phase. In this phase, we use $\bar{\boldsymbol{\theta}}$ to estimate the optimal assignment profile, i.e., $\boldsymbol{A}^*(\mathcal{R}, \mathcal{S}, \bar{\boldsymbol{\theta}})$, and this assignment profile is fixed for the remaining rounds. We also use $\bar{\boldsymbol{\theta}}$ to calculate the charging scheme $p_r(\boldsymbol{b}|_{\mathcal{R}}, \boldsymbol{b}|_{\mathcal{S}}, \bar{\boldsymbol{\theta}})$ and the reward scheme $q_s(\boldsymbol{b}|_{\mathcal{R}}, \boldsymbol{b}|_{\mathcal{S}}, \bar{\boldsymbol{\theta}})$ in each round in the commit phase.

Protocol analysis. We analyze the properties of Protocol `StrategicOnlineAssign`. With the conceptions defined in Section 4.1, we have the following theorem for Protocol `StrategicOnlineAssign`.

THEOREM 4. *Protocol* `StrategicOnlineAssign` *(Protocol 5) is DSIC and EIIR.*

Remark. Theorem 4 implies that Protocol 4 can stimulate requesters/suppliers to report their unit valuations/costs truthfully. Note that `StrategicOnlineAssign` cannot be guaranteed to be efficient, since it uses estimated CTRs in the commit phase to get the estimated optimal assignment profile. We also have the following theorem to state the performance on regret. The technical proof of Theorem 4 is presented in supplementary file.

THEOREM 5. *Protocol* `StrategicOnlineAssign` *(Protocol 5) has a worst-case expected regret*

$$\mathbb{E}[R_T] \leq \max\left\{\gamma, \frac{|\mathcal{R}|\gamma}{|\mathcal{S}|}\right\} S_{max} \min\{|\mathcal{R}|, |\mathcal{S}|\}$$
$$+\left(T - \max\left\{\gamma, \frac{|\mathcal{R}|\gamma}{|\mathcal{S}|}\right\}\right) S_{max} \min\{|\mathcal{R}|, |\mathcal{S}|\} \sqrt{\frac{2\log T}{\gamma}} + 2S_{max}, \tag{6}$$

where $S_{max} = \max_{(r,s)\in\mathcal{R}\times\mathcal{S}} S_{(r,s)}$. *To achieve the tightest upper bound, one can choose the value of* γ *such that the right-hand side of* (6) *is minimized.*

Remark. Theorem 5 indicates a higher regret than Theorem 2. The reason is that to guarantee DSIC, the protocol needs to perform exploration-exploitation separately according to the main result Theorem 1.3 in [7], and this constraint would cause higher regret than classic multi-armed bandit algorithms. Moreover, Theorem 4.1 in [7] derives a lower bound $\mathbb{E}[R_T] \geq \Omega(\max_{r\in\mathcal{R}} v_r |\mathcal{R}|^{1/3} T^{2/3})$ on regret. It indicates that any online algorithm must suffer this extra regret to guarantee the DSIC property. The detailed proof is presented in a supplementary file.

5 Evaluating Welfare Increase on Real-world Datasets

In this section, we conduct experiments on four real-world datasets to evaluate the welfare increase of our proposed protocols.

5.1 Experimental Settings

The real-world datasets we use for evaluation are described as follows.

- **Google+ Social Network [37]:** It is a sub-network of the Google+ user-user following network which contains 23,628 nodes and 39,242 directed edges. A node represents a user and a directed edge denotes that one user has the other user in his circles.
- **X Social Network [37]:** It is a sub-network of the X user-user following network which contains 23,370 nodes and 33,101 directed edges. A node represents a user and a directed edge indicates that a user follows another user.
- **GitHub Social Network [45]:** It is a sub-network of the GitHub user-user following network which contains 37,700 nodes and 289,003 undirected edges. Nodes are developers who have starred at least 10 repositories and edges are mutual follower relationships between them.
- **Deezer Europe Social Network [46]:** It is a sub-network of Deezer which contains 28,281 nodes and 92,752 undirected edges. Nodes are Deezer users from European countries and edges are mutual follower relationships between them.

In the above four datasets, there are no requesters or suppliers. Thus, we sample four subsets of users uniformly at random as requesters and suppliers in each dataset. Specifically, we sample 0.25% users as requesters and 0.25% users as suppliers. Besides, the datasets do not contain CTRs $\boldsymbol{\theta}$. The CTR of each requester is independently sampled from [0, 1] uniformly at random. We synthesize the unit valuation of a requester with degree d as

$$\frac{(1 + d/d_{max})^{\lambda}}{2^{\lambda}} \in [0, 1], \tag{7}$$

and synthesize the unit cost of a supplier with degree d as

$$1 - \frac{(1 + d/d_{max})^{\lambda}}{2^{\lambda}} \in [0, 1], \tag{8}$$

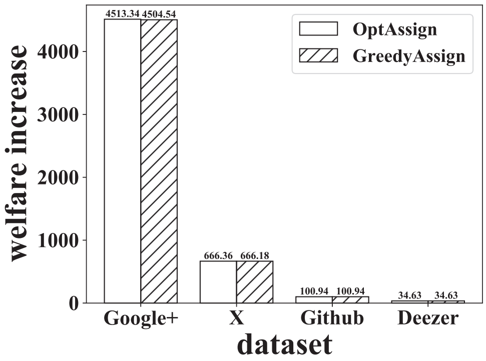


Fig. 1. Compare the welfare of OptAssign and GreedyAssign across different datasets.

where d_{max} is the largest degree in the network and $\lambda > 0$ is a parameter of the function to control the relationship between degree and unit valuation/cost. Specifically, Equation (7) models that a requester with a larger degree tends to have a larger unit valuation v_r and Equation (8) models that a supplier with a smaller degree tends to have a larger unit cost c_s. We use $\lambda = 0.8$ in the experiments unless otherwise specified.

We compare our protocols with other baselines. We use `Optimal` to refer to the optimal result that can be achieved under the setting with full knowledge, i.e., CTRs, unit valuations and unit costs are all accessible. We also have the following three heuristic methods: `DegradedAssignI`, `DegradedAssignII` and `DegradedAssignIII`. They use a framework similar to `OptAssign` to determine the assignment in each round. The only difference is the step in Line 1 of Protocol 1, which is to compute the weight $e_{r,s}$ of each edge (r, s) when constructing the bipartite graph. More specifically, `DegradedAssignI` use $e_{r,s} = S_{(r,s)}(v_r - c_s)$, `DegradedAssignII` use $e_{r,s} = S_{(r,s)}\theta_r$, and `DegradedAssignIII` use $e_{r,s} = S_{(r,s)}$.

5.2 Evaluate OptAssign and GreedyAssign

We compare two protocols, i.e., `OptAssign` which uses the Hungarian algorithm to find the maximum weighted matching in Protocol 1, and `GreedyAssign` which uses the greedy algorithm stated in Protocol 2 to find the maximum weighted matching, concerning welfare increase and running time. We evaluate these two protocols on datasets in the setting with full knowledge. From Figure 1, one can observe that for all the datasets, the welfare increase achieved by `GreedyAssign` is only slightly less than that achieved by `OptAssign`. Moreover, one can observe that the Google+ can achieve much more welfare increase. The reason is that the requesters and suppliers in Google+ network have a larger average degree.

5.3 Evaluate OnlineAssign

We now evaluate `OnlineAssign` (Protocol 3) which is designed for the user-centric aspect with unknown CTRs. We run $T = 200$ rounds. We assume that for requester r, the observed samples of CTRs across different rounds are independent and identically distributed Gaussian distribution with mean θ_r and variance 1. For comparison, we also show the cumulative welfare increase of heuristic method `DegradedAssignI`, which can be interpreted as a situation where the CTRs of all the requesters are assumed to be 1 seeing as CTRs are unknown. Figure 2 shows the cumulative welfare increase of different methods. For all the datasets, one can observe that the slope of the `OnlineAssign` curve increases with time and converges to the slope of `Optimal`. This observation verifies the logarithmic growth of regret stated in Theorem 2. In other words, protocol `OnlineAssign` can achieve a near-optimal single-round welfare increase in the later rounds, even

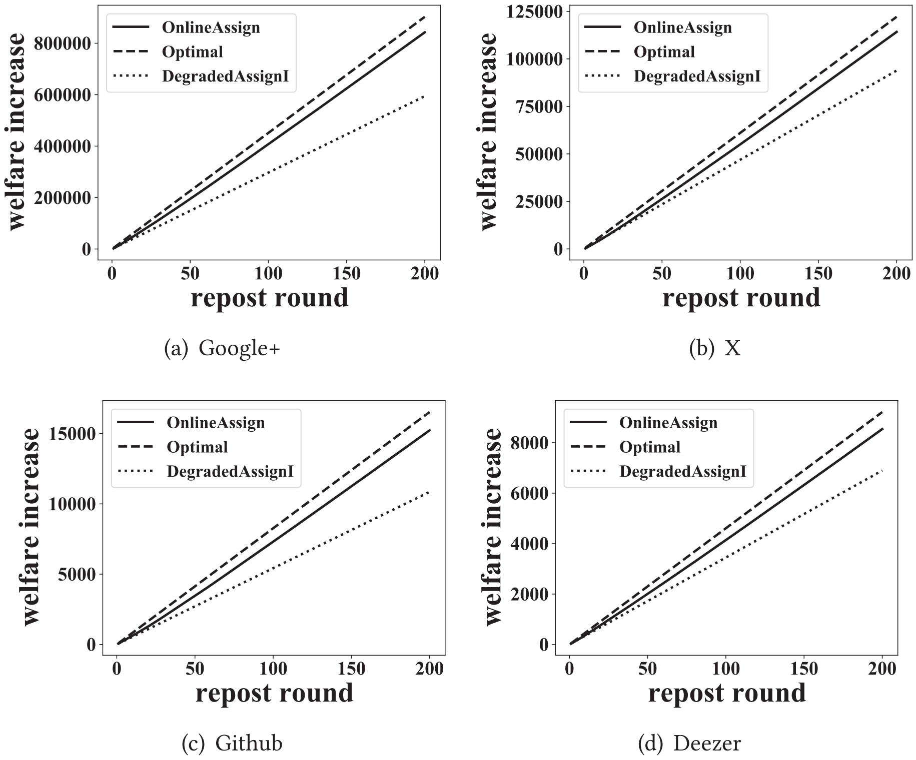


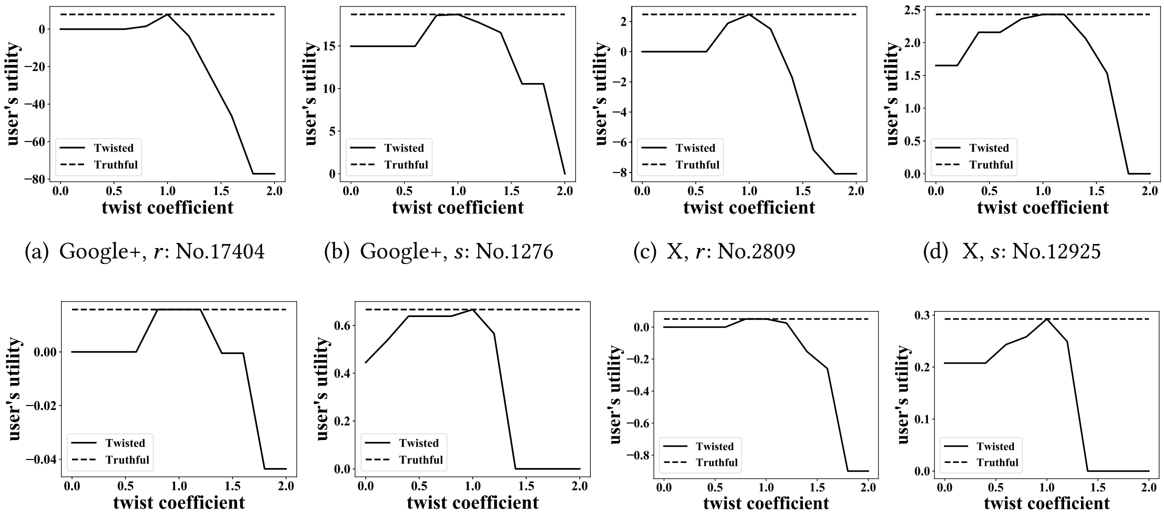
Fig. 2. Compare welfare increase of different methods.

though the exact values of CTRs are unknown at the beginning. Besides, in each dataset, after round 20 or so, the curve of `OnlineAssign` is always above `DegradedAssignI` and steeper than `DegradedAssignI`, which shows `OnlineAssign` performs much better than the heuristic method when the time horizon is large.

5.4 Evaluate Truthful Offline Protocol

We evaluate our truthful protocol proposed in Section 4.1 for the platform-centric aspect where the platform knows CTRs but does not know the unit valuations and unit costs. Recall that the protocol charges requesters according to Equation (4) and rewards suppliers according to Equation (5). We compare users' utilities achieved by truthful reporting and untruthful reporting to verify the DSIC property. For untruthful reporting, the reports of requesters and suppliers are twisted as $b_r = \alpha v_r$ and $b_s = \beta c_s$ respectively, where $\alpha, \beta \in \mathbb{R}_+$ are twist coefficients. Specifically, we sample one requester and one supplier uniformly at random from each dataset. The values of twist coefficients α and β are ranged from 0 to 2, with step size 0.2. Figure 3 shows the utilities of sampled requesters and suppliers when they use different reporting strategies. For all the datasets and all the requesters and suppliers, the utilities achieved by untruthful reporting (i.e., twist coefficients are not 1) are lower than or equal to the utilities achieved by truthful reporting. It implies that reporting truthfully is a weakly-dominant strategy which maximizes one's utility.

To show the performance associated with welfare increase, we compare our proposed protocol with heuristic method `DegradedAssignII`, which can be interpreted as a situation where the unit valuations of all the requesters are assumed to be 1 and unit costs of all the suppliers are assumed to be 0 seeing as unit valuations and unit costs CTRs are unknown. Figure 4 shows the single-round welfare increase achieved by different methods. For all the datasets, one can observe that the welfare increases achieved by our proposed protocol are the same as the welfare increase achieved



(a) Google+, r: No.17404 (b) Google+, s: No.1276 (c) X, r: No.2809 (d) X, s: No.12925

(e) GitHub. r: No.13806 (f) GitHub. s: No.5796 (g) Deezer. r: No.2025 (h) Deezer. s: No.7816

Fig. 3. User's utility under different reporting strategies.

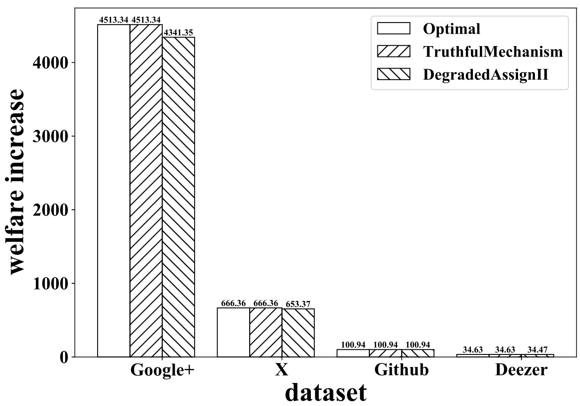
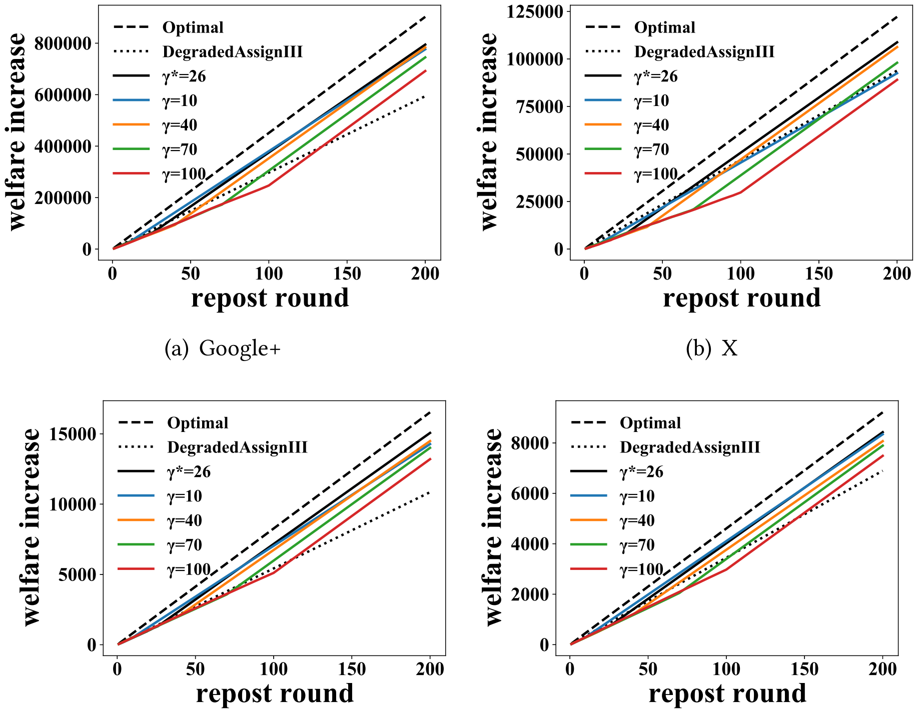


Fig. 4. Compare welfare increase of different methods.

by `OptAssign` (results in Figure 1). This observation verifies the efficient property of our protocol. Besides, the welfare increase achieved by our proposed protocol are higher than the heuristic method. Note that the difference between `DegradedAssignII` and optimum is not too much. The reason is that the unit valuation of many requesters are closed to 1 and the unit valuation of many suppliers are closed to 0, which make the assumed situation in `DegradedAssignII` relatively closed to real situation.

5.5 Evaluate `StrategicOnlineAssign`

We evaluate the protocol `StrategicOnlineAssign` proposed in Section 4.2 for the platform-centric aspect with unknown CTRs, i.e., the setting where the platform does not know the CTRs, unit valuations and unit costs. We apply protocol `StrategicOnlineAssign` on all the datasets for $T = 200$ rounds. We vary γ, the number of rounds in the exploration phase, to study its impact on cumulative welfare increase. We also compute the value γ^* that minimizes the regret upper bound of Protocol 5 (please refer to supplementary file for the regret upper bound) and ensures the tightest upper bound given $T = 200$. For all the datasets, the solution are $\gamma^* = 26$. For com-



(a) Google+ (b) X

(c) Github (d) Deezer

Fig. 5. Compare cumulative welfare increase of different methods and different values of γ.

parison, we compare it with heuristic method `DegradedAssignIII` which can be interpreted as a situation where the CTRs of all the requesters are assumed to be 1, the unit valuations of all the requesters are assumed to be 1 and unit costs of all the suppliers are assumed to be 0, seeing as all of these are unknown to the platform. Figure 5 shows the cumulative welfare increase when we use `DegradedAssignIII` and `StrategicOnlineAssign` with different values of γ. We can observe that for all the datasets, the protocol using $\gamma = \gamma^*$ achieves a larger cumulative welfare increase in the end than using other γ values. Besides, compared with `DegradedAssignIII`, `StrategicOnlineAssign` ($\gamma = \gamma^*$) achieves a larger cumulative welfare increase in the end in all the datasets. However, if T is relatively small (e.g., 50), we can find that the performance of `DegradedAssignIII` is also relatively good, especially for the X dataset. We can also observe that, for all the datasets, the slope of `StrategicOnlineAssign` ($\gamma = \gamma^*$) in round 200 is steeper than `DegradedAssignIII` and similar to `Optimal`, which implies a near-optimal single-round welfare increase in the later rounds.

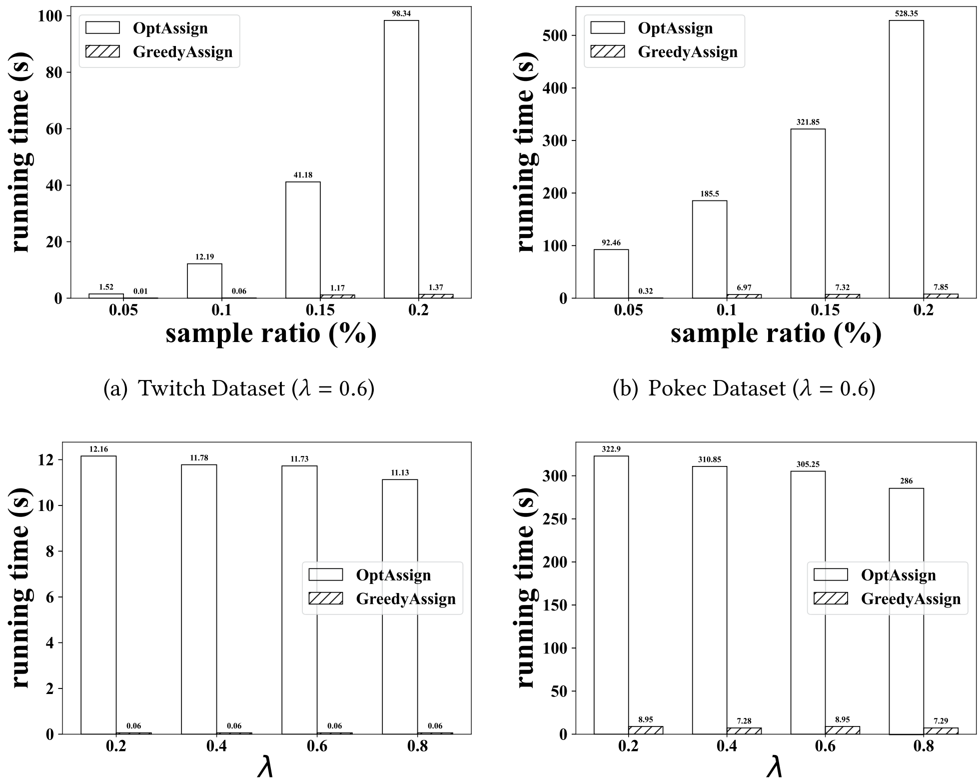
6 Evaluating Scalability and Parameter Sensitivity

In this section, we conduct experiments on two large scale datasets to to evaluate the scalability and parameter sensitivity of our proposed protocols.

6.1 Experimental Settings

The larger scale datasets we use for evaluation are described as follows.

— **Twitch Gamers Social Network [47]:** It is a sub-network of Twitch users which contains 168,114 nodes and 6,797,557 undirected edges. Nodes are Twitch users and edges are mutual follower relationships between them.



(a) Twitch Dataset (λ = 0.6)

(b) Pokec Dataset (λ = 0.6)

(c) Twitch dataset (sample ratio = 0.15%)

(d) Pokec Dataset (sample ratio = 0.15%)

Fig. 6. Evaluating the running time of `OptAssign` and `GreedyAssign`.

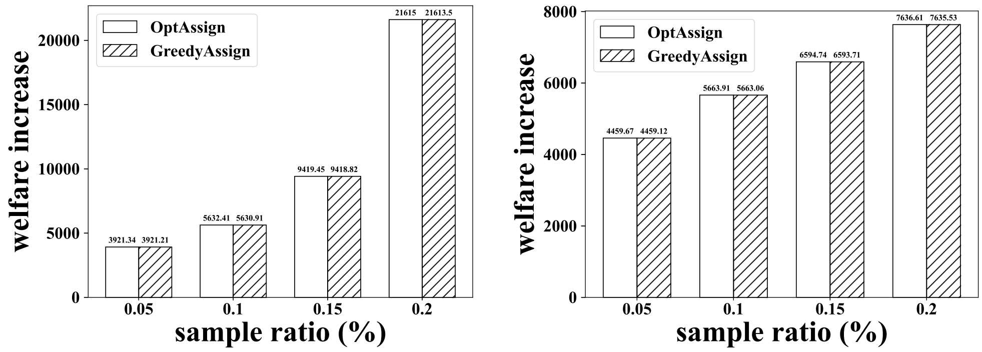
— **Pokec Social Network [51]:** It is a sub-network of the Pokec user-user following network which contains 1,632,803 nodes and 30,622,564 directed edges. A node represents a user and a directed edge indicates that a user follows another user.

The CTRs θ of the two datasets are defined same as Section 6.1. And the unit valuation of a requester is defined same as Equation (7). The unit cost of a supplier is defined same as Equation (8). In this section, we conduct our experiments in three parts. The part one is to compare two protocols, i.e., `OptAssign` and `GreedyAssign` in welfare and running time. The part two and part three are to compare protocol `OnlineAssign` and protocol `StrategicOnlineAssign` with two baselines methods (`POSTA-TSM` [23] and `ICT` [64]).

6.2 Evaluating Scalability

Note that the computational complexity of our proposed protocols are dominated by the assignment algorithms, i.e., `OptAssign` and `GreedyAssign`. Therefore, evaluating the scalability of proposed protocols boils down to evaluating the running time of `OptAssign` and `GreedyAssign`.

We evaluate the running time of them on Twitch dataset and Pokec dataset with full knowledge. In this part, we set the sample ratio = 0.15% as default and vary sample ratio = [0.05%, 0.1%, 0.15%, 0.2%]. We set parameter $\lambda = 0.6$ as default and vary $\lambda = [0.2, 0.4, 0.6, 0.8]$ in Equations (7) and (8). We use control variable method to study the impact of sample ratio and λ to the scalability. Figure 6(a) shows the impact of sample ratio on Twitch dataset. Figure 6(a) shows the running time of `OptAssign` and `GreedyAssign`. From Figure 6(a), one can observe that `OptAssign` takes much more running time than `GreedyAssign` does, i.e., around 100 times



(a) Twitch dataset ($\lambda = 0.6$) (b) Pokec dataset ($\lambda = 0.6$)

Fig. 7. Impact of sample ratio on the welfare increase of `OptAssign` and `GreedyAssign`.

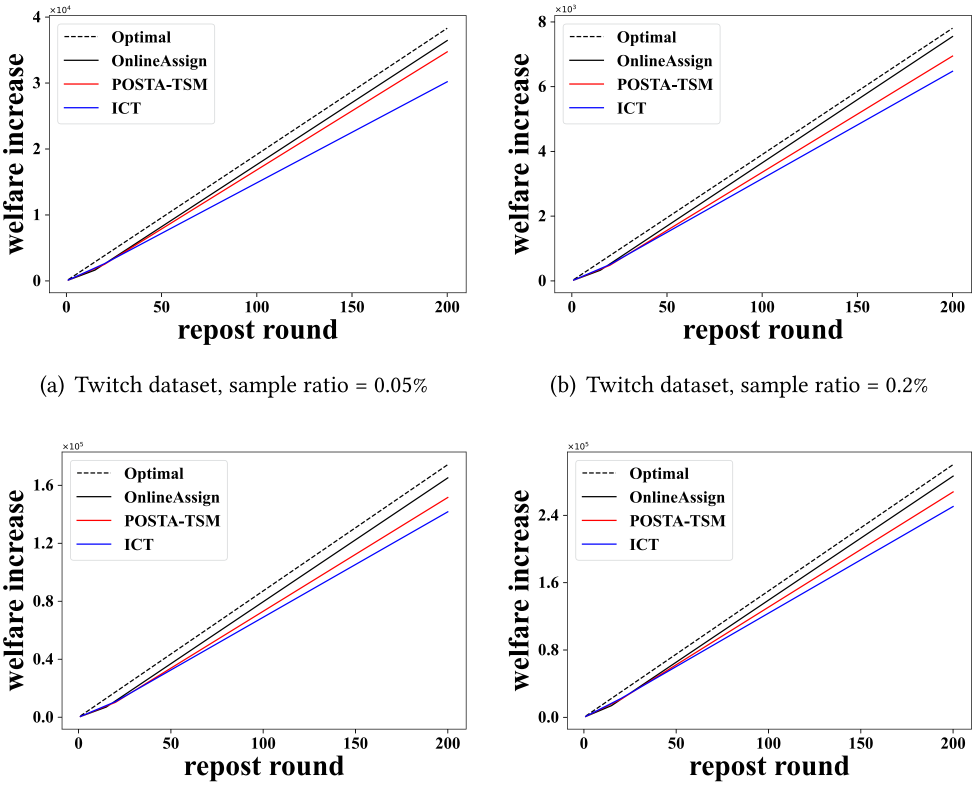
more. When the sample ratio increases from 0.05 to 0.2, the running time increases significantly, i.e., increased by around 100 times. One can observe that the impact of sample ratio has similar results on Pokec dataset as shown in Figure 6(b). Figure 6(c) shows the impact of λ on Twitch dataset. From Figure 6(c), one can observe that `OptAssign` takes much more running time than `GreedyAssign` does, i.e., around 180 times more. When the λ increases from 0.2 to 0.8, the running time has hardly changed. One can observe that the impact of sample ratio has similar results on Pokec dataset as shown in Figure 6(d).

6.3 Evaluating the Impact of Sample Ratio

We study the impact of sample ratio on the welfare increase of our proposed protocols on the Twitch dataset and Pokec dataset.

Impact of sample ratio on `OptAssign` and `GreedyAssign`. We set $\lambda = 0.6$ as default, and vary the value of sample ratio to evaluate the impact of sample ratio. From Figure 7(a), one can observe that under different sample ratios on Twitch dataset, the welfare increase achieved by `GreedyAssign` is only slightly less than that achieved by `OptAssign`, i.e., the difference is less than 1%. When the sample ratio increases from 0.05% to 0.2%, the welfare increase of both `GreedyAssign` and `OptAssign` increases significantly, i.e., approximately 5 times. From Figure 7(b), one can observe that under different sample ratios on Pokec dataset, the welfare increase achieved by `GreedyAssign` is only slightly less than that achieved by `OptAssign`, i.e., the different is less than 1%. When the sample ratio increases from 0.05% to 0.2%, the welfare increase of both `GreedyAssign` and `OptAssign` increases significantly, i.e., approximately 2 times more.

Impact of sample ratio on `OnlineAssign`. We now evaluate `OnlineAssign` (Protocol 3) which is designed for the user-centric aspect with unknown CTRs. We run $T = 200$ rounds. We assume that for requester r, the observed samples of CTRs across different rounds are independent and identically distributed Gaussian distribution with mean θ_r and variance 1. For comparison, we show the cumulative welfare increase of two baselines methods `POSTA-TSM` [23] and `ICT` [64]. We set sample ratio = 0.05% and 0.2% respectively. We use control variable method to conduct the experiments. Figures 8(a) and 8(b) show the cumulative welfare increase under different sample ratios on Twitch dataset. For the experiment results, one can observe that the slope of the `OnlineAssign` curve increases with time and converges to the slope of `Optimal`. Besides, under each sample ratio, after round 30 or so, the curve of `OnlineAssign` is always above `POSTA-TSM` [23] and `ICT` [64] and steeper than `POSTA-TSM` [23] and `ICT` [64], which shows `OnlineAssign` performs much better than the baseline methods when the time horizon is large.



(a) Twitch dataset, sample ratio = 0.05%

(b) Twitch dataset, sample ratio = 0.2%

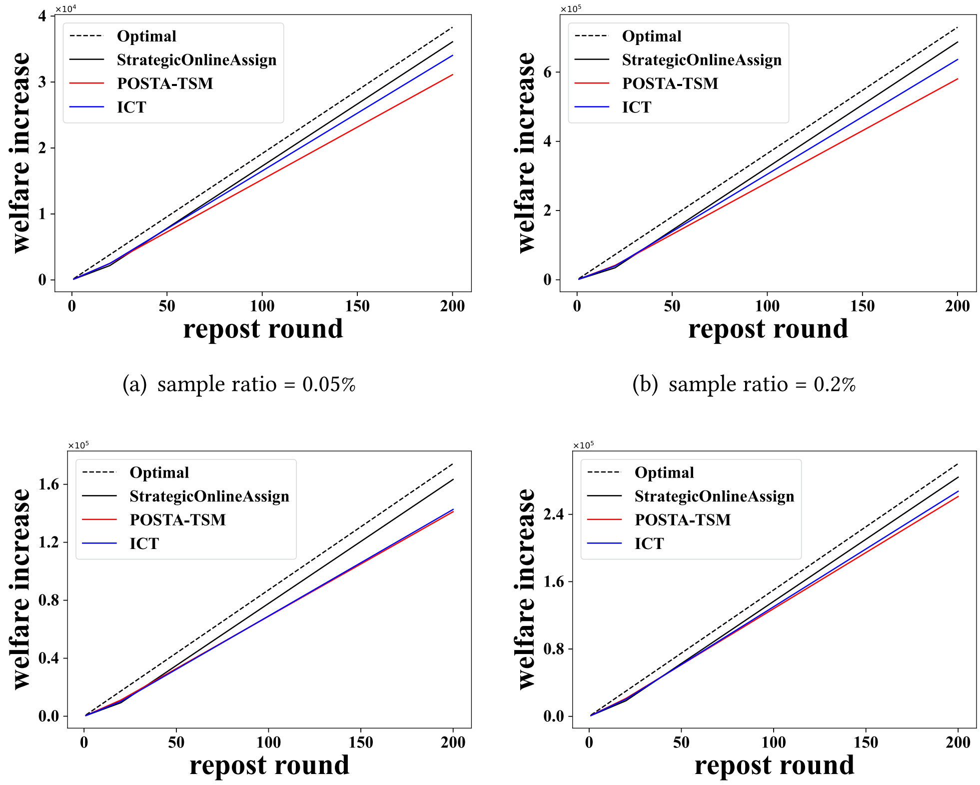
(c) Pokec dataset, sample ratio = 0.05%

(d) Pokec dataset, sample ratio = 0.2%

Fig. 8. Impact of sample ratio on the welfare increase of OnlineAssign.

Similar experiment results conducted on Pokec dataset are shown in Figures 8(c) and 8(d). Both experiment results verify the logarithmic growth of regret stated in Theorem 2. In other words, protocol OnlineAssign can achieve a near-optimal single-round welfare increase in the later rounds, even though the exact values of CTRs are unknown at the beginning.

Impact of sample ratio on StrategicOnlineAssign. We evaluate the protocol StrategicOnlin- eAssign proposed in Section 4.2 for the platform-centric aspect with unknown CTRs, i.e., the setting where the platform does not know the CTRs, unit valuations and unit costs. We apply protocol StrategicOnlineAssign on all the datasets for $T = 200$ rounds. We set $\gamma = 20$ as default. For comparison, we show the cumulative welfare increase of two baselines methods POSTA-TSM [23] and ICT [64]. We set sample ratio = 0.05% and 0.2%, respectively. We use control variable method to conduct the experiments. Figures 9(a) and 9(b) show the cumulative welfare increase under different sample ratios on Twitch dataset. For the experiment results, one can observe that the slope of the StrategicOnlineAssign curve increases with time and converges to the slope of Optimal. Besides, under each sample ratio, after round 20 or so, the curve of StrategicOnlineAssign is always above POSTA-TSM [23] and ICT [64] and steeper than POSTA-TSM [23] and ICT [64], which shows OnlineAssign performs much better than the baseline methods when the time horizon is large. Similar experiment results conducted on Pokec dataset are shown in Figures 9(c) and 9(d). Both experiment results verify the logarithmic growth of regret stated in Theorem 2. In other words, protocol StrategicOnlineAssign can achieve a near-optimal single-round welfare increase in the later rounds, even though the exact values of CTRs are unknown at the beginning.



(a) sample ratio = 0.05%
(b) sample ratio = 0.2%
(c) sample ratio = 0.05%
(d) sample ratio = 0.2%

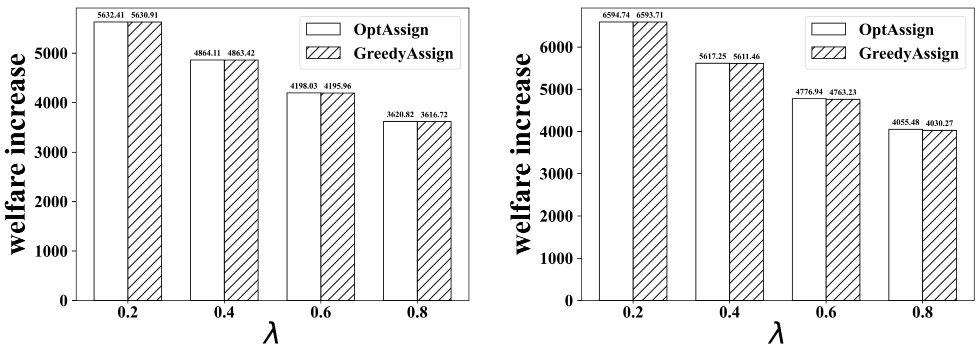
Fig. 9. Impact of sample ratio on the welfare increase of `StrategicOnlineAssign`.

6.4 Evaluating the Impact of λ

We study the impact of λ on the welfare increase of our proposed protocols on the Twitch dataset and Pokec dataset.

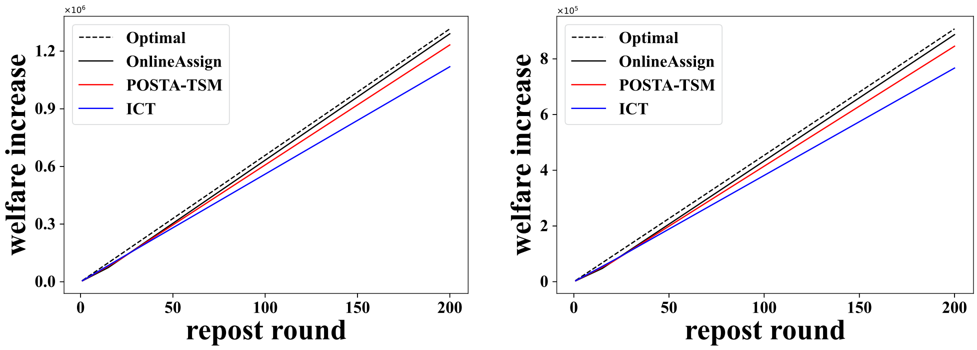
Impact of λ on `OptAssign` and `GreedyAssign`. We set sample ratio = 0.15% as default, and vary the value of λ to evaluate the impact of λ. From Figure 10(a), one can observe that under different λ, the welfare increase achieved by `GreedyAssign` is only slightly less than that achieved by `OptAssign`, i.e., the different is less than 1%. When the λ increases from 0.2 to 0.8, the welfare increase of both `GreedyAssign` and `OptAssign` decreases significantly, i.e., approximately 33%. Figure 10(b) shows the impact of λ on Pokec dataset. From Figure 10(b), one can observe that under different λ, the welfare increase achieved by `GreedyAssign` is only slightly less than that achieved by `OptAssign`, i.e., the difference is less than 1%. When the λ increases from 0.2 to 0.8, the welfare increase of both `GreedyAssign` and `OptAssign` decreases significantly, i.e., it is approximately halved.

Impact of λ on `OnlineAssign`. We now evaluate `OnlineAssign` (Protocol 3) which is designed for the user-centric aspect with unknown CTRs. We run $T = 200$ rounds. We assume that for requester r, the observed samples of CTRs across different rounds are independent and identically distributed Gaussian distribution with mean θ_r and variance 1. For comparison, we show the cumulative welfare increase of two baselines methods `POSTA-TSM` [23] and `ICT` [64]. We set $\lambda = 0.2$ and 0.8 respectively. We use the control variable method to conduct the experiments. Figures 11(a) and 11(b) show the cumulative welfare increase under different λ on Twitch dataset. For the experiment results, one can observe that the slope of the `OnlineAssign` curve increases

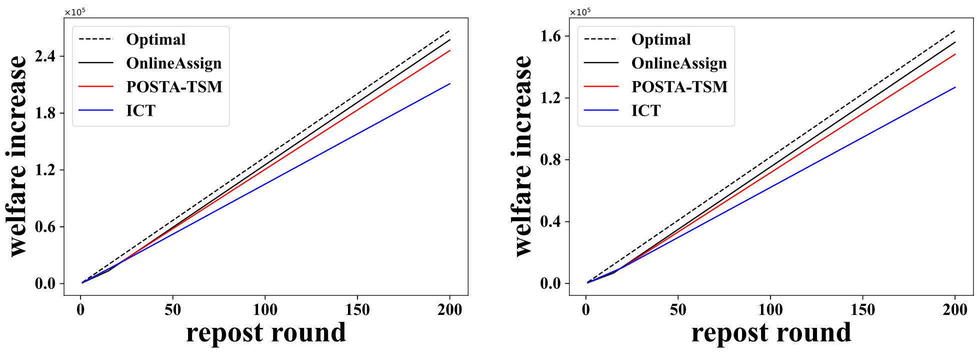


(a) Twitch dataset (sample ratio = 0.15%) (b) Pokec dataset (sample ratio = 0.15%)

Fig. 10. Impact of λ on the welfare increase of OptAssign and GreedyAssign.



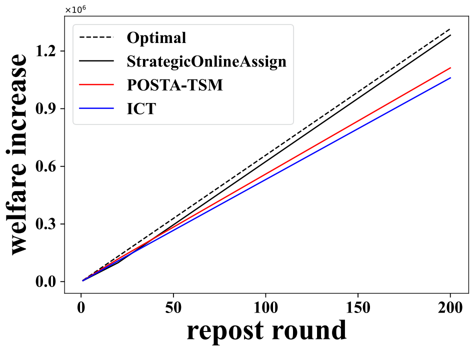
(a) $\lambda = 0.2$ (b) $\lambda = 0.8$



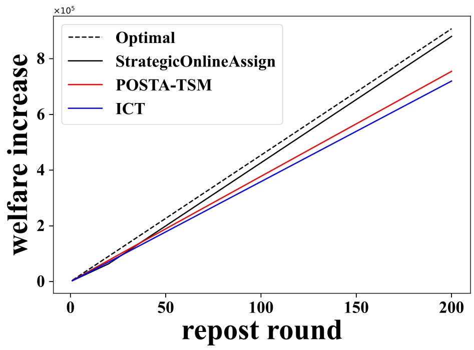
(c) $\lambda = 0.2$ (d) $\lambda = 0.8$

Fig. 11. Impact of λ on the welfare increase of OnlineAssign.

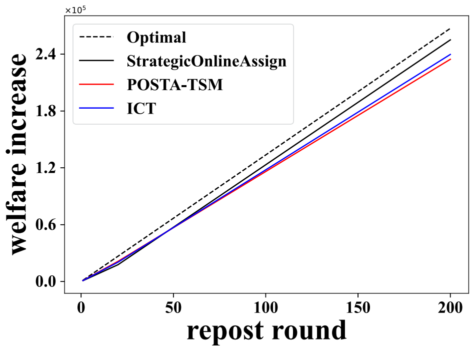
with time and converges to the slope of Optimal. Besides, under each λ, after round 40 or so, the curve of OnlineAssign is always above POSTA-TSM [23] and ICT [64] and steeper than POSTA-TSM [23] and ICT [64], which shows OnlineAssign performs much better than the baseline methods when the time horizon is large. Similar experiment results conducted on Pokec dataset are shown in Figures 11(c) and 11(d). Both experiment results verify the logarithmic growth of regret stated in Theorem 2. In other words, protocol OnlineAssign can achieve a near-optimal single-round welfare increase in the later rounds, even though the exact values of CTRs are unknown at the beginning.



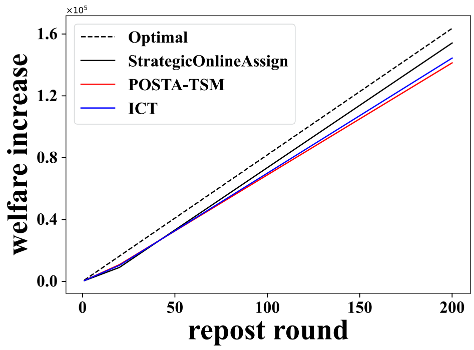
(a) $\lambda = 0.2$



(b) $\lambda = 0.8$



(c) $\lambda = 0.2$



(d) $\lambda = 0.8$

Fig. 12. Impact of λ on the welfare increase of `StrategicOnlineAssign`.

Impact of λ on `StrategicOnlineAssign`. We evaluate the protocol `StrategicOnlineAssign` proposed in Section 4.2 for the platform-centric aspect with unknown CTRs, i.e., the setting where the platform does not know the CTRs, unit valuations and unit costs. We apply protocol `StrategicOnlineAssign` on all the datasets for $T = 200$ rounds. We set $\gamma = 20$ as default. For comparison, we show the cumulative welfare increase of two baselines methods `POSTA-TSM` [23] and `ICT` [64]. We set $\lambda = 0.2$ and 0.8 respectively. We use control variable method to conduct the experiments. Figures 12(a) and 12(b) show the cumulative welfare increase under different λ on Twitch dataset. For the experiment results, one can observe that the slope of the `StrategicOnlineAssign` curve increases with time and converges to the slope of `Optimal`. Besides, under each λ, after round 25 or so, the curve of `StrategicOnlineAssign` is always above `POSTA-TSM` [23] and `ICT` [64] and steeper than `POSTA-TSM` [23] and `ICT` [64], which shows `OnlineAssign` performs much better than the baseline methods when the time horizon is large. Similar experiment results conducted on Pokec dataset are shown in Figures 12(c) and 12(d). Both experiment results verify the logarithmic growth of regret stated in Theorem 2. In other words, protocol `StrategicOnlineAssign` can achieve a near-optimal single-round welfare increase in the later rounds, even though the exact values of CTRs are unknown at the beginning.

7 Related Work

This section will discuss research works related to this paper from application perspective and methodology perspective.

7.1 Application Aspect

Social advertising. Numerous existing works [1, 14, 21, 43] focus on boosting the content visibility for users who are product sellers, whose target can be categorized into the field of social advertising [4, 31]. Social advertising models are typically employed by platforms such as X, Instagram, Weibo and Facebook through broadcasting [27, 49, 55, 60, 61]. The behavior of "reposting" content is the most typical type, which can be a video, an image, or simply a textual post containing an advertising message or idea to spread. The influence of reposting has been studied broadly [48, 58, 59, 62]. Some recent works [31, 57, 64] also study the rewarded reposting in OSN. Chouaki et al. [12] proposed a comprehensive approach using data donations to capture a realistic and detailed picture of news consumption. Swift et al. [50] addressed the fairness and efficiency in content spread across different demographic groups in social networks. Zheng et al. [66] studied maximizing revenue and welfare while addressing the challenges of fair reward distribution among suppliers. In short, social advertising as an emerging marketing model, is powerful to boost the visibility of contents (including ads), which can bring monetization opportunities not only to the advertisers but also to the platform and its users.

Assignment mechanism. We then introduce several applications related to our reposting service concerning the assignment aspect. The first application that is related to assignment is crowdsourcing. Crowdsourcing has attracted extensive attention from both academia and the industry [22, 36, 54]. Its core function is task assignment, i.e., assigning tasks to suitable workers, so that the total payoff from these assignments is maximized. To this end, maximum weight matching algorithms are often used to maximize the total payoff. There have been many successful crowdsourcing platforms such as Amazon **Mechanical Turk (MTurk)** and Upwork. Apart from the assignment aspect, crowdsourcing also needs to design proper incentive mechanisms to attract users to participate and judiciously leverage the supply and demand in the system. The second related application is **sponsored search auction (SSA)** [7] which is also a similar problem to ours due to the way of modeling factors such as CTRs, valuations of requesters, costs of suppliers and welfare. Liu et al. [33] proposed a method to ensure a balance between task diversity and assignment accuracy. Huang et al. [23] proposed a heuristic algorithm for maximizing task utility and stable online matching, incorporating both offline and online task assignment scenarios. However, solutions in SSA cannot be applied to our problem for the following major reasons: (1) SSA simply uses a greedy assignment rule while our assignment rule is matching-based; and (2) SSA involved one-side private information while our problem has two-sides private information (unit valuations and costs).

7.2 Methodology Aspect

Maximum weighted matching. The **maximum weighted matching (MWM)** problem serves as a building block of our proposed protocols. The MWM problem is to find a set of vertex-disjoint edges with maximum weight in given a weighted bipartite graph. In the 1950s, Kuhn [29] and Munkres [39] developed the Hungarian algorithm to solve the MWM problem. Later, Edmonds and Karp [16] observed that implementing the Hungarian algorithm for MWM amounted to computing single-source shortest paths n times on a non-negatively weighted graph. Thus, the running time of their algorithm depends on the implementation of Dijkstra's algorithm [17, 24, 52, 53]. Faster algorithms are proposed when the weights are bounded integers in $[-N, N]$, Duan and Su [15] present a new scaling algorithm that runs in $O(m\sqrt{n}\log N)$ time, with the constraint that weights are integers within the range of $[0, N]$. Angriman et al. [3] presented a batch-dynamic algorithm for maintaining a 1/2-approximation of maximum weighted matching in fully dynamic graphs, highlighting its practical efficiency in handling large-scale updates with time complexity

$O(m + n)$. Koana et al. [28] extended kernelization techniques to maximum-weighted matching, proposing efficient data reduction rules that significantly speeded up existing algorithms, and its time complexity is $O(mn \log n)$. Chaudhary and Zehavi [10] explored the parameterized complexity of acyclic matchings, providing significant insights into maximum weighted matching in various graph classes.

Online mechanism design. For another research line, our work is closely related to **combinatorial multi-armed bandit (CMAB)**, mechanism design and MAB mechanism design. Combinatorial multi-armed bandit [11, 18] is a variant of the classic MAB model [9], where multiple arms (a.k.a. a super arm) can be pulled in each round in contrast. Basu and Sankararaman [8] introduced a mechanism for double auctions where both buyers and sellers learned their valuations through bandit feedback. Min and Russo [38] and Li et al. [30] developed bandit algorithms with theoretical guarantees for nonstationary environments, focusing on adapting to changing reward distributions over time. Zheng et al. [65] proposed a neural network-based bandit algorithm that efficiently handled non-stationary environments and context-dependent reward structures for combinatorial decisions. Zhang and Luo [63] explored online learning strategies for predicting click-through rates in contextual PPC auctions to maximize revenue and minimize regret. For the user-centric social aspect, we customize the framework of CMAB to our setting by taking an assignment as a super arm. One difference is we have defined a valid assignment profile, which means not all the combinations are valid. Mechanism design [41, 42, 44] aims to incentivize players to act truthfully. **Vickrey-Clarke-Groves (VCG)** mechanism [13, 19, 56] is one of the most well-known auction mechanisms. Lyu et al. [34] investigated using offline reinforcement learning to design dynamic mechanisms in VCG auctions, emphasizing the balance between pessimistic and optimistic strategies. Kandasamy et al. [26] presented a bandit-based approach to VCG mechanism design, where agents' values were learned over time to ensure truthful reporting and efficient outcomes. Our algorithms in a platform-centric setting are variants of the VCG mechanism. MAB mechanism is in the intersection of the above two fields. The traditional MAB model simply assumes that all arms are static choices. However, in many applications [2, 35], the arms can represent rational and selfish individuals. Thus, mechanism design has been applied in the MAB context to deal with the interplay between online learning and the strategic players, leading to MAB mechanisms [32].

8 Conclusion and Future Work

In this study, we have developed and analyzed online incentive protocols for reposting services in online social networks (OSNs). Our work addresses the need for effective mechanisms to boost content visibility through reposting, focusing on both user-centric and platform-centric aspects. From a user-centric perspective, we formulated the reposting problem as a welfare maximization task, where the goal is to optimize the assignment of suppliers (users willing to repost) to requesters (users seeking reposts) based on their click-through rates (CTRs), valuations, and costs. Our proposed online learning protocol `OnlineAssign`, which employs a combinatorial multi-armed bandit approach, effectively balances the exploration-exploitation tradeoff. We demonstrated that this protocol achieves sub-linear regret, ensuring that the performance asymptotically approaches the optimal solution. For the platform-centric aspect, where users' valuations and costs remain private, we designed an "explore-then-commit" online protocol `StrategicOnlineAssign`. This protocol not only estimates the unknown parameters but also guarantees truthfulness through a carefully crafted charging and rewarding scheme by using the Vickrey-Clarke-Groves (VCG) mechanism. We proved that our protocol maintains a sub-linear regret, providing robust performance in realistic settings where user behaviors and preferences are not fully known. Extensive experiments on six public datasets validated the efficiency and effectiveness of our proposed protocols. The results highlighted the superiority of our approach in terms of welfare maximization and computational efficiency, offering fundamental insights into how social network structures influence the efficacy of reposting services. Future work can explore the integration of more sophisticated machine learning models to predict user behavior and further enhance the accuracy of CTRs estimate. Additionally, extending our protocols to handle dynamic and evolving user interactions in OSNs would be a valuable direction for subsequent research.

Acknowledgments

The work of John C.S. Lui was supported in part by the RGC GRF-14202923.

References

[1] Eitan Altman, Parmod Kumar, Srinivasan Venkatramanan, and Anurag Kumar. 2013. Competition over timeline in social networks. In *2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM'13)*. IEEE, 1352–1357.

[2] Luzi Anderegg and Stephan Eidenbenz. 2003. Ad hoc-VCG: A truthful and cost-efficient routing algorithm for mobile ad hoc networks with selfish agents. In *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*. 245–259.

[3] Eugenio Angriman, Michał Boroń, and Henning Meyerhenke. 2022. A batch-dynamic suitor algorithm for approximating maximum weighted matching. *ACM J. Exp. Algorithmics* 27, Article 1.6 (Jul. 2022), 41 pages.

[4] Cigdem Aslay, Francesco Bonchi, Laks V. S. Lakshmanan, and Wei Lu. 2016. Revenue maximization in incentivized social advertising. *arXiv preprint arXiv:1612.00531* (2016).

[5] Sitaram Asur, Bernardo A. Huberman, Gabor Szabo, and Chunyan Wang. 2011. Trends in social media: Persistence and decay. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 5. 434–437.

[6] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine Learning* 47, 2-3 (2002), 235–256.

[7] Moshe Babaioff, Yogeshwer Sharma, and Aleksandrs Slivkins. 2009. Characterizing truthful multi-armed bandit mechanisms: Extended abstract. In *Proceedings of the 10th ACM Conference on Electronic Commerce*. 79–88.

[8] Soumya Basu and Abishek Sankararaman. 2024. Double auctions with two-sided bandit feedback. In *Proceedings of the 37th International Conference on Neural Information Processing Systems (NIPS'23)*. Curran Associates Inc., Red Hook, NY, USA, Article 168, 12 pages.

[9] Sébastien Bubeck and Nicolo Cesa-Bianchi. 2012. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *arXiv preprint arXiv:1204.5721* (2012).

[10] Juhi Chaudhary and Meirav Zehavi. 2023. Parameterized results on acyclic matchings with implications for related problems. In *Graph-Theoretic Concepts in Computer Science*, Daniël Paulusma and Bernard Ries (Eds.). Springer Nature Switzerland, Cham, 201–216.

[11] Wei Chen, Yajun Wang, and Yang Yuan. 2013. Combinatorial multi-armed bandit: General framework and applications. In *International Conference on Machine Learning*. PMLR, 151–159.

[12] Salim Chouaki, Abhijnan Chakraborty, Oana Goga, and Savvas Zannettou. 2024. What news do people get on social media? Analyzing exposure and consumption of news through data donations. In *Proceedings of the ACM on Web Conference 2024 (WWW'24)*. Association for Computing Machinery, New York, NY, USA, 2371–2382.

[13] Edward H. Clarke. 1971. Multipart pricing of public goods. *Public Choice* (1971), 17–33.

[14] Ranbir Dhounchak, Veeraruna Kavitha, and Eitan Altman. 2017. A viral timeline branching process to study a social network. In *2017 29th International Teletraffic Congress (ITC'29)*, Vol. 3. IEEE, 19–24.

[15] Ran Duan and Hsin-Hao Su. 2012. A scaling algorithm for maximum weight matching in bipartite graphs. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 1413–1424.

[16] Jack Edmonds and Richard M. Karp. 1972. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM (JACM)* 19, 2 (1972), 248–264.

[17] Michael L. Fredman and Robert Endre Tarjan. 1987. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM (JACM)* 34, 3 (1987), 596–615.

[18] Yi Gai, Bhaskar Krishnamachari, and Rahul Jain. 2012. Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations. *IEEE/ACM Transactions on Networking* 20, 5 (2012), 1466–1478.

[19] Theodore Groves. 1973. Incentives in teams. *Econometrica: Journal of the Econometric Society* (1973), 617–631.

[20] Haoran Gu, Shiyuan Zheng, Xudong Liu, Hong Xie, and John C. S. Lui. 2024. Online Incentive Protocol Design for Reposting Service in Online Social Networks. https://www.dropbox.com/scl/fi/aa1g6f388a9v0knwglurq/_supplementary.pdf?rlkey=wa7mk7a1y8i6x7qasaga6xc51&st=bdu99qh4&dl=0. (2024). Online.

[21] Eduardo Hargreaves, Claudio Agosti, Daniel Menasché, Giovanni Neglia, Alexandre Reiffers-Masson, and Eitan Altman. 2019. Fairness in online social network timelines: Measurements, models and mechanism design. *Performance Evaluation* 129 (2019), 15–39.

[22] Danula Hettiachchi, Vassilis Kostakos, and Jorge Goncalves. 2022. A survey on task assignment in crowdsourcing. *ACM Computing Surveys (CSUR)* 55, 3 (2022), 1–35.

[23] Weiyi Huang, Peng Li, Bo Li, Lei Nie, and Haizhou Bao. 2023. Towards stable task assignment with preference lists and ties in spatial crowdsourcing. *Information Sciences* 620 (2023), 16–30.

[24] Donald B. Johnson. 1977. Efficient algorithms for shortest paths in sparse networks. *Journal of the ACM (JACM)* 24, 1 (1977), 1–13.

[25] Andreas Jungherr. 2016. Twitter use in election campaigns: A systematic literature review. *Journal of Information Technology & Politics* 13, 1 (2016), 72–91.

[26] Kirthevasan Kandasamy, Joseph E. Gonzalez, Michael I. Jordan, and Ion Stoica. 2024. VCG mechanism design with unknown agent values under stochastic bandit feedback. *J. Mach. Learn. Res.* 24, 1, Article 53 (Mar. 2024), 45 pages.

[27] Mohammad Reza Karimi, Erfan Tavakoli, Mehrdad Farajtabar, Le Song, and Manuel Gomez Rodriguez. 2016. Smart broadcasting: Do you want to be seen?. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1635–1644.

[28] Tomohiro Koana, Viatcheslav Korenwein, André Nichterlein, Rolf Niedermeier, and Philipp Zschoche. 2021. Data reduction for maximum matching on real-world graphs: Theory and experiments. *ACM J. Exp. Algorithmics* 26, Article 1.3 (Apr. 2021), 30 pages.

[29] Harold W. Kuhn. 1955. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly* 2, 1-2 (1955), 83–97.

[30] Li Li, Jiajie Shen, Bochun Wu, Yangfan Zhou, Xin Wang, and Keqin Li. 2023. Adaptive data placement in multi-cloud storage: A non-stationary combinatorial bandit approach. *IEEE Transactions on Parallel and Distributed Systems* 34, 11 (2023), 2843–2859.

[31] Yung-Ming Li and Ya-Lin Shiu. 2012. A diffusion mechanism for social advertising over microblogs. *Decision Support Systems* 54, 1 (2012), 9–22.

[32] Chang Liu, Qingpeng Cai, and Yukui Zhang. 2017. Multi-armed bandit mechanism with private histories. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*. 1607–1609.

[33] Xiuya Liu, Tianzhang Xing, Xianjia Meng, and Chase Q. Wu. 2024. TA-GAE: Crowdsourcing diverse task assignment based on graph autoencoder in AIoT. *IEEE Internet of Things Journal* 11, 8 (2024), 14508–14522.

[34] Boxiang Lyu, Zhaoran Wang, Mladen Kolar, and Zhuoran Yang. 2022. Pessimism meets VCG: Learning dynamic mechanism design via offline reinforcement learning. In *Proceedings of the 39th International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (Eds.), Vol. 162. PMLR, 14601–14638.

[35] Huadong Ma, Dong Zhao, and Peiyan Yuan. 2014. Opportunities in mobile crowd sensing. *IEEE Communications Magazine* 52, 8 (2014), 29–35.

[36] Ke Mao, Licia Capra, Mark Harman, and Yue Jia. 2017. A survey of the use of crowdsourcing in software engineering. *Journal of Systems and Software* 126 (2017), 57–84.

[37] Julian J. McAuley and Jure Leskovec. 2012. Learning to discover social circles in ego networks. In *NIPS*, Vol. 2012. Citeseer, 548–56.

[38] Seungki Min and Daniel Russo. 2023. An information-theoretic analysis of nonstationary bandit learning. In *Proceedings of the 40th International Conference on Machine Learning (ICML'23)*. JMLR.org, Article 1034, 19 pages.

[39] James Munkres. 1957. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics* 5, 1 (1957), 32–38.

[40] Roger B. Myerson. 1981. Optimal auction design. *Mathematics of Operations Research* 6, 1 (1981), 58–73.

[41] Yadati Narahari. 2014. *Game Theory and Mechanism Design*. Vol. 4. World Scientific.

[42] Noam Nisan and Amir Ronen. 2007. Computationally feasible VCG mechanisms. *Journal of Artificial Intelligence Research* 29 (2007), 19–47.

[43] Alexandre Reiffers-Masson, Eduardo Hargreaves, Eitan Altman, Wouter Caarls, and Daniel S. Menasché. 2017. Timelines are publisher-driven caches: Analyzing and shaping timeline networks. *ACM SIGMETRICS Performance Evaluation Review* 44, 3 (2017), 26–29.

[44] Tim Roughgarden. 2010. Algorithmic game theory. *Commun. ACM* 53, 7 (2010), 78–86.

[45] Benedek Rozemberczki, Carl Allen, and Rik Sarkar. 2019. Multi-scale Attributed Node Embedding. (2019). arXiv:cs.LG/1909.13021

[46] Benedek Rozemberczki and Rik Sarkar. 2020. Characteristic functions on graphs: Birds of a feather, from statistical descriptors to parametric models. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM'20)*. ACM, 1325–1334.

[47] Benedek Rozemberczki and Rik Sarkar. 2021. Twitch Gamers: A Dataset for Evaluating Proximity Preserving and Structural Role-based Node Embeddings. (2021). arXiv:cs.SI/2101.03091

[48] Anja Rudat and Jürgen Buder. 2015. Making retweeting social: The influence of content and context information on sharing news in Twitter. *Computers in Human Behavior* 46 (05 2015).

[49] Nemanja Spasojevic, Zhisheng Li, Adithya Rao, and Prantik Bhattacharyya. 2015. When-to-post on social networks. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2127–2136.

[50] Ian P. Swift, Sana Ebrahimi, Azade Nova, and Abolfazl Asudeh. 2022. Maximizing fair content spread via edge suggestion in social networks. *Proc. VLDB Endow.* 15, 11 (Jul. 2022), 2692–2705.

[51] Lubos Takac and Michal Zabovsky. 2012. Data analysis in public social networks. In *International Scientific Conference and International Workshop Present Day Trends of Innovations*, Vol. 1.

[52] Mikkel Thorup. 2003. Integer priority queues with decrease key in constant time and the single source shortest paths problem. In *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*. 149–158.

[53] Mikkel Thorup. 2007. Equivalence between priority queues and sorting. *Journal of the ACM (JACM)* 54, 6 (2007), 28–es.

[54] Yongxin Tong, Zimu Zhou, Yuxiang Zeng, Lei Chen, and Cyrus Shahabi. 2020. Spatial crowdsourcing: A survey. *The VLDB Journal* 29, 1 (2020), 217–250.

[55] Utkarsh Upadhyay, Abir De, and Manuel Gomez Rodriguez. 2018. Deep reinforcement learning of marked temporal point processes. *Advances in Neural Information Processing Systems* 31 (2018).

[56] William Vickrey. 1961. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of Finance* 16, 1 (1961), 8–37.

[57] Lei Wang, Da Qian, and Lin Zhu. 2018. The effect of system generated cues on microblog rewarding repost behavior–a source credibility perspective. *Journal of Electronic Commerce Research* 19, 1 (2018), 104–118.

[58] Yizhou Yan, Fujio Toriumi, and Toshiharu Sugawara. 2020. Influence of retweeting on the behaviors of social networking service users. In *International Conference on Complex Networks and Their Applications*. Springer, 671–682.

[59] Yizhou Yan, Fujio Toriumi, and Toshiharu Sugawara. 2021. Understanding how retweets influence the behaviors of social networking service users via agent-based simulation. *Computational Social Networks* 8, 1 (2021), 1–21.

[60] Ali Zarezade, Abir De, Utkarsh Upadhyay, Hamid R. Rabiee, and Manuel Gomez-Rodriguez. 2017. Steering social activity: A stochastic optimal control point of view. *J. Mach. Learn. Res.* 18 (2017), 205–1.

[61] Ali Zarezade, Utkarsh Upadhyay, Hamid R. Rabiee, and Manuel Gomez-Rodriguez. 2017. RedQueen: An online algorithm for smart broadcasting in social networks. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. 51–60.

[62] Jing Zhang, Biao Liu, Jie Tang, Ting Chen, and Juanzi Li. 2013. Social influence locality for modeling retweeting behaviors. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI'13)*. AAAI Press, 2761–2767.

[63] Mengxiao Zhang and Haipeng Luo. 2024. Online learning in contextual second-price pay-per-click auctions. In *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research)*, Sanjoy Dasgupta, Stephan Mandt, and Yingzhen Li (Eds.), Vol. 238. PMLR, 2395–2403.

[64] Zehao Zhao, Chaokun Zhang, Tie Qiu, and Keqiu Li. 2021. Recruiting MCS workers strategy with non-fixed reward in social network. In *2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD'21)*. IEEE, 800–805.

[65] J. Zheng, H. Gao, H. Dai, Z. Zheng, and F. Wu. 2023. Neural contextual combinatorial bandit under non-stationary environment. In *2023 IEEE International Conference on Data Mining (ICDM'23)*. IEEE Computer Society, Los Alamitos, CA, USA, 878–887.

[66] Shiyuan Zheng, Hong Xie, and John C. S. Lui. 2023. Pricing social visibility service in online social networks: Modeling and algorithms. *IEEE Transactions on Network Science and Engineering* 10, 2 (2023), 859–870.

Received 29 December 2023; revised 30 June 2024; accepted 6 September 2024