Optimizing Social Visibility in OSNs With Anonymity Guarantees: Efficient Algorithms and Applications

Shiyuan Zheng, Hong Xie, *Member, IEEE*, and John C.S. Lui, *Fellow, IEEE*

***Abstract*—Online social network (OSN) is an ideal venue to enhance one's visibility.This paper considers how a user (called requester) in an OSN selects a small number of users and invites them as new friends/followers so as to maximize her "*social visibility*". More importantly, the requester has to do this under the anonymity setting, which means she is not allowed to know the neighborhood information of other users in the OSN. In this paper, we first develop a mathematical model to quantify the social visibility and formulate the problem of visibility maximization with anonymity guarantee, abbreviated as "VisMAX-A". We prove that the VisMAX-A problem is NP-hard even with an "perfect query oracle" which knows users' neighborhood information. Then we design an algorithmic framework named as "AdaExp," which adaptively expands the requester's visible set in multiple rounds. In each round of the expansion, the algorithmic framework uses a query oracle with anonymity guarantee to select one user to increase the requester's visibility. We prove the theoretical guarantees of our algorithmic framework, which quantify how the query oracle may influence the solution guarantee. By using probabilistic data structures like the k-minimum values (KMV) sketch and online learning methods like multi-armed bandit (MAB), we design an efficient query oracle with anonymity guarantees to estimate the best candidate to increase requester's visibility. We also conduct experiments on real-world social networks and validate the effectiveness of our algorithms. Lastly, we demonstrate applications of our algorithm on real-world data and show that one can increase her social visibility significantly by only adding a few (e.g., one or two) friends/followers in the OSNs.**

***Index Terms*—Approximation algorithms, KMV sketch, multi-armed bandit, social network, social visibility.**

I. Introduction

Online social network (OSN) is a popular venue for one to share information and gain attention. For example, in social networking sites like Facebook, users share their opinions, status and likes/dislikes to their friends via the friendship network. In video sharing sites like YouTube, users share their videos to their subscribers via the subscriber network. In photo sharing sites like Instagram, users share photos to their followers via the follower network. In product review sites like Epinions, users share their opinions on products to others via the trust network.

In an OSN, users with more direct attention (e.g., friends, subscribers, followers, etc.), can make their contents (e.g., opinions, videos, photos, etc.) visible to more users in the network, and this may bring higher commercial benefit to these users. For example, consider the OSN in YouTube, generally a user having more subscribers can get a larger amount of views on the video she published. Note that the viewers not only can be the subscribers, but can also be those who have not subscribed but come across the video via other means, e.g., being recommended by subscribers of the video publisher. Moreover, more views may lead to more advertising exposures, which means a larger reward from YouTube. Informally, we say that the users who can attract more attention are more *socially visible*, in other words, they have a larger *social visibility*.

One way to increase a user's social visibility is to get more direct attention, for instance, attracting new friends, new subscribers, new followers, etc. For example, Dou+[1] is an official service provided by TicTok, where content creators (requesters) who want to increase visibility can pay to gain more followers. The requester can make an order to specify how many new followers she wants to grow (lager amount causes higher payment), and then the system will assign available users to the requester. Besides, there are also many non-official platforms provide similar services of helping requesters to boost their visibility. The motivation behind such needs is that, if the requester gets more followers, then her content can be visible to more users which in turn can bring her more commercial benefit. Motivated by services like Dou+ mentioned above, instead of considering natural connection establishment (e.g, one is interested in user r and follows r spontaneously), we consider the connections established by request and payment. We aim to design algorithms for such services to select available users judiciously.Note that not all the users in an OSN

Manuscript received July 14, 2021; revised February 8, 2022; accepted March 8, 2022. Date of publication March 16, 2022; date of current version June 27, 2022. The work of John C.S. Lui was supported by GRF under Grant 14200321. The work of Hong Xie was supported in part by Chongqing Talents: Exceptional Young Talents Project under Grant cstc2021ycjh-bgzxm0195 and in part by the Fundamental Research Funds for the Central Universities under Grant 2020CDJ-LHZZ-057. Recommended for acceptance by Prof. Yang Yang. *(Corresponding author: Hong Xie).*

Shiyuan Zheng and John C.S. Lui are with the Department of Computer Science and Engineering, Chinese University of Hong Kong, Shatin, Hong Kong (e-mail: syzheng@cse.cuhk.edu.hk; cslui@cse.cuhk.edu.hk).

Hong Xie is with the College of Computer Science, Chongqing University, Chongqing 400044, China (e-mail: xiehong2018@foxmail.com).

This article has supplementary downloadable material available at https://doi.org/10.1109/TNSE.2022.3159672, provided by the authors.

Digital Object Identifier 10.1109/TNSE.2022.3159672

[1] https://doujia.douyin.com/

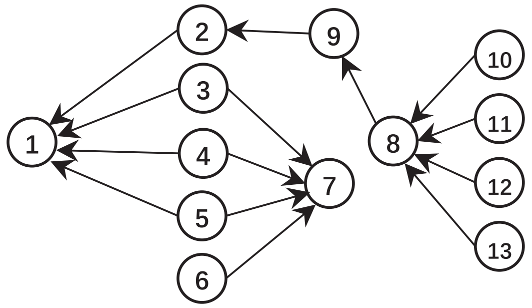


Fig. 1. An example of OSN with thirteen nodes and directed edges.

are willing to establish connections with others. For example, it is not always easy to make friends with someone in the Facebook OSN, and it is also difficult to ask someone to a subscriber in the YouTube OSN. Moreover, the establishment of a new connection comes with a certain cost on money, time or effort. Thus, there is a constraint on the scope of users a requester can select from and the number of users a requester can build new connections with. This implies that the requester needs to judiciously select the targets to establish new connections, so as to maximize the effect of visibility boosting. More interestingly, there are some networks in which users are not allowed to know the neighborhood information of any other users, and this creates a hurdle to measure the potential of other users to improve the requester's visibility. To illustrate, consider the following examples:

Example 1: Consider a simple social network with directed edge as illustrated in Fig. 1. We say user v is an incoming neighbor of user u if there is a directed edge from v to u, representing u gets a direct attention from v (e.g., v follows or subscribes u). For simplicity, a user is only socially visible to 1-hop and 2-hop incoming neighbors in this example. For example, in Fig. 1, user 1 is socially visible to users in the set $\{2, 3, 4, 5, 9\}$, which we call the "visible set" of user 1. Suppose only a subset of users are willing to establish new connection as requested, named as "available users". Let them be $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$. Then, suppose user 1 is the requester, who has a quota of adding only one new incoming neighbor to increase her social visibility, i.e., increase the cardinality of her social visible set. After trying all available users, i.e., $\{6, 7, 8, 9, 10, 11\}$, one can find that by adding user 8 as a new incoming neighbor, the requester can maximize the increase of her social visible set, that is 5 ($\{8, 10, 11, 12, 13\}$).

Example 1 illustrates that a requester can significantly increase her social visibility by simply adding just one new incoming neighbor.

In practice, a requester may have a quota of adding several new incoming neighbors, and the OSNs we consider in real life are usually very large in size, i.e., billions of nodes with trillions of edges. This implies a computational challenge of determining the optimal set of new incoming neighbors. We use the following example to illustrate the computational complexity.

Example 2: We consider the same setting as Example 1, but we change the quota of adding new incoming neighbors from 1 to 2. That is, the requester, i.e., user 1, aims to add two new incoming neighbors to maximize the cardinality of her social visible set. To achieve this, one can enumerate all the two-user pairs in the available user set $\{6, 7, 8, 9, 10, 11\}$ so as to find out the "optimal pair". In this case, the computational complexity is $\binom{6}{2}$. However, the number of available users in a real-world OSN can be billions, and a user may have a quota of adding new incoming neighbors, so the number of tests is $\binom{\#\text{availableusers}}{\#\text{quotas}}$. This makes it computationally expensive to find the optimal new incoming neighbor set.

What makes the problem more challenging and interesting is that, due to privacy constraints, the OSN network topology may not be available to the requester, and other users may not even want to disclose the information of members in their visible sets. To illustrate such anonymity setting, consider the following example.

Example 3: Consider the same setting as Example 2 of adding two new incoming neighbors to the requester, i.e., user 1. The anonymity setting is that: (1) Each user only knows her own social visible set, and she does not know the remaining part of the OSN; (2) The requester is not allowed to know the membership of any other user's social visible set. The objective of the requester is to select the optimal new incoming neighbor set and satisfy this anonymity requirement.

Examples 1–3 illustrate the problem of increasing a requester's social visibility by adding new incoming neighbors, and the underlying computational challenge and anonymity guarantee challenge in selecting the optimal set of new incoming neighbors which maximize the requester's social visibility. Note that in the above examples, we consider a very simple model of social visibility, i.e., 1-hop and 2-hop incoming neighbors. In practice, the social visibility is more complicated due to the psychological behavior and the heterogeneity of users, making the social visibility maximization problem very challenging.

To the best of our knowledge, no previous work has studied how to maximize one's visibility, especially with the query anonymity requirement. Note that the influence propagation model [1] is different from the social visibility model considered in this paper. More specifically, a user who is in the social visible set of a requester does not mean she is influenced by the requester, and a user is influenced by a requester does not mean that she is in the social visible set of a user. In addition, the approach of solving the influence maximization problem can not be directly applied in our problem, especially when we have to consider the anonymity guarantee. Thus, this is a new research problem in OSNs.

Note that the social visibility problem is relevant in the real-world and our solution can be used to provide customized visibility boosting service for OSN users. Specifically, it helps requesters who want to increase their visibility to target the most profitable candidates, and avoid wasting time or money on those who can only bring a marginal increase on their social visibility. Thus, our solution can be applied to many fields, such as self-marketing, advertising service and so on. Motivated by this, we aim to answer the following questions: *(1) How to formulate a mathematical model to quantify social visibility? (2) How to develop computationally efficient algorithms* *to maximize social visibility of a requester? (3) How to provide anonymity guarantee for our social visibility maximization algorithm?* We answer these questions. Our contributions are:

- We propose a mathematical model to quantify the social visibility. We formulate the problem of visibility maximization which only based on users' local information and at the same time, provide anonymity guarantee (VisMAX-A).
- We design an algorithmic framework (AdaExp), which adaptively expands the requester's visible set in multiple rounds. In each round of expansion, a *query oracle* returns an estimation of the "best" new incoming neighbor with a guaranteed accuracy, and at the same time, it also provides the anonymity guarantee. Our algorithmic framework sequentially selects the estimated "best" new incoming neighbor to build connection until the quota is used up. We prove theoretical guarantees of our algorithmic framework, which quantify how the query oracle influences the solution guarantee. These insights guide us to design the query oracle.
- We design a query oracle by a novel combination of the KMV sketch technique [2] and the multi-armed bandit (MAB) strategy [3]. Relying on the KMV sketch and the one-way hash function, we generate samples on the increase of visible set. Then we design a refined multi-armed bandit (MAB) strategy to estimate the "best" new incoming neighbor from these samples while optimizing estimation accuracy with a given budget on sampling. We also prove that our query oracle satisfies the desired accuracy properties.
- We conduct experiments on real-world social network datasets, and the results validate the effectiveness of our framework. We demonstrate applications of our algorithm on real-world data and demonstrate that one can increase her social visibility significantly by only adding a few (e.g., one or two) friends/followers in the OSNs.

The rest of this paper is organized as follows. Section II models the social visibility and formulates the visibility maximization problem with anonymity guarantee (VisMAX-A). In Section III, we present an algorithmic framework to solve the problem based on a *query oracle*. In Section IV, we design an algorithm to implement the query oracle. We present experimental validation in Section V and study two real-world applications in Section VI. Finally, we present related work in Section VII and conclude in Section VIII.

II. Model & Problem Formulation

In this section, we first develop a mathematical model to quantify the social visibility. Then we formulate the problem of visibility maximization with anonymity guarantee.

A. The Online Social Network Model

Consider an OSN which is characterized by a weighted directed graph $\mathcal{G} \triangleq (\mathcal{U}, \mathbf{W})$, where $\mathcal{U} \triangleq \{1, \ldots, U\}$ denotes a finite set of $U \in \mathbb{N}_+$ users and $\mathbf{W} \triangleq [w_{v \to u} : v, u \in \mathcal{U}] \in \mathbb{R}_+^{U \times U}$ represents a summarization of the weights of edges. We use $w_{v \to u} = 0$ to model that there is no directed edge from v to u. The graph $\mathcal{G}$ does not contain self-loop edges, i.e., $w_{u \to u} = 0, \forall u \in \mathcal{U}$. The weight $w_{v \to u}$ quantifies the *influence strength* of user u over user v. For example, in a Twitter-like OSN, a directed edge from v to u can be interpreted as v follows u, and the weight $w_{v \to u}$ corresponds to the frequency that v replies to, or likes or retweets the tweet posted by u. Fig. 2 shows a graph model for a Twitter-like OSN. One can see that user 1 is followed by users 2,3,5 and the weights $w_{5 \to 1} = 19, w_{2 \to 1} = 3, w_{3 \to 1} = 1$ indicate that user 5 pays more attention to the updates of user 1 than users 2 and 3 do. Facebook-like OSNs correspond to a special case of the graph $\mathcal{G}$ with a symmetric weight matrix $\mathbf{W}$, i.e., $w_{u \to v} = w_{v \to u}$ for all $v, u \in \mathcal{U}$. Furthermore, each edge indicates a friendship between two users and the associated weight can be interpreted as the frequency that two users interact with each other.

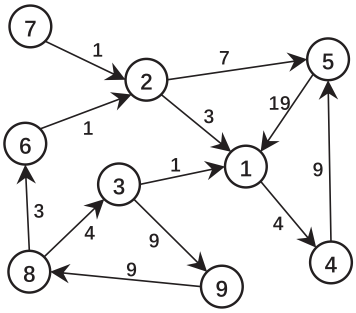


Fig. 2. An illustrating example of our OSN model with nine nodes.

B. The Social Visibility Model

We say a user v is an incoming neighbor of user u, if there is a directed edge from v to u. The incoming neighbor set of user u (denoted by $\mathcal{N}(u)$) is the set of all the incoming neighbors of user u defined as $\mathcal{N}(u) \triangleq \{v | w_{v \to u} > 0, v \in \mathcal{U}\}$.

For example, in Fig. 2, the incoming neighbor set of user 1 is $\mathcal{N}(1) = \{2, 3, 5\}$.

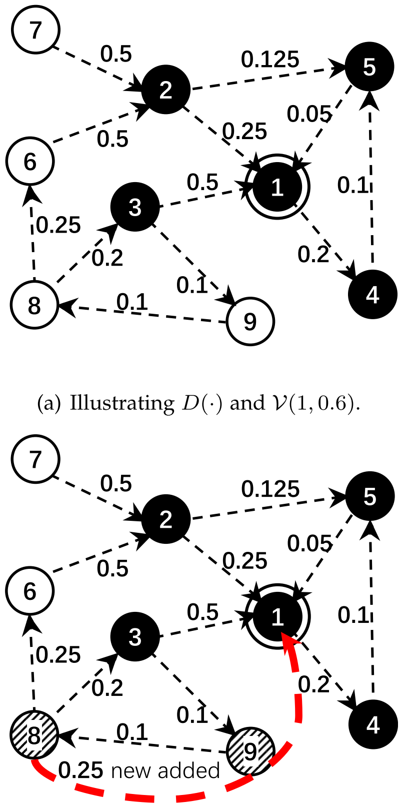
We say a *requester* is a user in the OSN who aims to increase her visibility by requesting others to be her new incoming neighbors. *Available users* are users in the OSN who are willing to be a new incoming neighbor of the requester if selected and requested. We say available users are available to requesters.

Definition 1 (visibility distance): Visibility distance (denoted by $d_{v \to u}$) is a measure to quantify the degree of difficulty in which a user influences her incoming neighbors, which is defined as:

$$d_{v \to u} = \begin{cases} \infty, & \text{if } v \notin \mathcal{N}(u), \\ D(w_{v \to u}), & \text{otherwise}, \end{cases}$$

where $D : \mathbb{R}_+ \to \mathbb{R}_+$ denotes a distance mapping function, from influence strength (i.e., weight of edge) to visibility distance.

One interpretation of the visibility distance $d_{v \to u}$ is the cost that user u must pay to get her contents spread to her incoming neighbor v. Another interpretation can be the time delay when spreading a content which values highly on timeliness. A larger



(a) Illustrating $D(\cdot)$ and $\mathcal{V}(1, 0.6)$.

(b) Illustrating new $\mathcal{V}(1, 0.6)$.

Fig. 3. The change of 0.6-visible set after adding new edge with weight $w_{8\to1} = 3$ and visibility distance $d_{1\to8} = 0.25$: user 8 and user 9 are newly included into the 0.6-visible set of user 1.

$d_{v\to u}$ implies a larger cost or delay. One possible form of D, for example, can be

$$D(w_{v\to u}) = \frac{1}{1 + w_{v\to u}}, v \in \mathcal{N}(u). \tag{1}$$

Applying (1) to Fig. 2, the visibility distance from user 2 to user 1 can be calculated as $1/(1 + 3) = 0.25$. Furthermore, Fig. 3(a) shows the visibility distances of user pairs in Fig. 2, where each real number associated with the dashed edge from v to u represents the visibility distance $d_{v\to u}$ calculated by $1/(1 + w_{v\to u})$. The following assumption states a family of visibility distance functions $D(\cdot)$ that we can have.

Assumption 1: The visibility distance $D(w_{v\to u})$ decreases in influence strength $w_{v\to u}$, where $v \in \mathcal{N}(u)$.

Assumption 1 captures that user u has a smaller visibility distance $D(w_{v\to u})$ for the incoming neighbor v, if her influence $w_{v\to u}$ to the incoming neighbor gets stronger, which also means she is more likely to be visible to v. For example, (1) satisfies Assumption 1.

Define a directed path in graph $\mathcal{G}$ as $\vec{p} \triangleq (x_0 \to x_1 \to \cdots \to x_n)$. Let $\mathcal{E}(\vec{p})$ denote the set of all directed edges on path $\vec{p}$, i.e., $\mathcal{E}(\vec{p}) = \{(x_0, x_1), (x_1, x_2), \ldots, (x_{n-1}, x_n)\}$. We define the visibility distance length $L(\vec{p})$ of the path $\vec{p}$ to be the sum of the visibility distances of all the edges on path $\vec{p}$, formally, $L(\vec{p}) \triangleq \sum_{(v,u)\in\mathcal{E}(\vec{p})} d_{v\to u}$. For example, consider a path $\vec{p} = (7 \to 2 \to 1)$ in Fig. 3(a). Its visibility distance length can be calculated as

$$L(\vec{p}) = d_{7\to2} + d_{2\to1} = \frac{1}{1 + w_{7\to2}} + \frac{1}{1 + w_{2\to1}} = 0.75.$$

Let $\mathcal{P}_{v\to u}$ denote the set of all directed paths (without circles) from user v to user u. Base on $\mathcal{P}_{v\to u}$, we define the concept of *τ-visible*, *τ-visible set* and *τ-visibility*, where $\tau \in \mathbb{R}_+$ denotes a positive real number.

Definition 2 (visible threshold, τ-visible set, τ-visibility): Define τ as the visible threshold, where $\tau \in \mathbb{R}_+$. A user u is τ-visible to user v, if v satisfies

$$\min\nolimits_{\vec{p}\in\mathcal{P}_{v\to u}} L(\vec{p}) \leq \tau.$$

The τ-visible set of user u is the set of all users to whom user u is τ-visible, denoted as

$$\mathcal{V}(u, \tau) \triangleq \left\{ v \middle| v \in \mathcal{U}, \min\nolimits_{\vec{p}\in\mathcal{P}_{v\to u}} L(\vec{p}) \leq \tau \right\}.$$

Lastly, the τ-visibility of user u is the cardinality of user u's τ-visible set.

Namely, a user u is τ-visible to user v, if there exists at least one directed path from v to u with the summation of the visibility distance of all edges along the path being less than or equal to τ. For example, Fig. 3(a) shows that user 1 is 0.6-visible to user 2, while user 1 is not 0.6-visible to user 6. Moreover, the 0.6-visible set of user 1 is $\{2, 3, 4, 5\}$, i.e., $\mathcal{V}(1, 0.6) = \{2, 3, 4, 5\}$ and thus her 0.6-visibility is 4.

Remark. (1) The notion of visibility defined in Definition 2 is motivated by the recency property of contents, i.e., users are interested in timely contents, in real-world social networks such as TicTok. Each link causes certain delay in spreading a content. The summation of the distance across links can be interpreted as the total amount of delay and the threshold τ can be interpreted as the amount of time that the content becomes out-of-date and no longer attractive to users. (2) Note that neither the independent cascade model nor the linear threshold model can capture the recency property of contents. In the linear threshold model, each user is associated with a threshold, quantifying how difficult the user can be convinced by her neighbors. In our model, the threshold τ captures the recency property of contents, i.e., the amount of time that the content becomes out-of-date.

C. The Visibility Maximization Problem

Consider one requester denoted by $r \in \mathcal{U}$. To increase the visibility of r, one can add some new incoming neighbors (new followers, new subscribers, etc.) to r. Formally, we assume that the requester r is given a quota of adding $m \in \mathbb{N}_+$ new incoming neighbors to increase her visibility and each one of these m incoming edges to r has a default weight $\bar{w} \in \mathbb{R}_+$ representing the default influence strength of a new added link. Let $\mathcal{U}' \subseteq \mathcal{U}$ denote the set of available users who are willing to be a new incoming neighbor of requester r. Let $\mathcal{M} \subseteq \mathcal{U}'$ denote a set of users with cardinality $|\mathcal{M}| \leq m$ and no user in

$\mathcal{M}$ is an incoming neighbor of the requester r, i.e., $\mathcal{M} \cap \mathcal{N}(r) = \emptyset$. Let $\Delta_{\boldsymbol{\theta}}(\mathcal{M})$ denote the visibility improvement, i.e., the increase of the requester r's τ-visibility after adding directed edges with default weight $\bar{w} \in \mathbb{R}_+$ from users in $\mathcal{M}$ to r, where $\boldsymbol{\theta}$, for the purpose of simplifying presentation, is defined as a vector of given model parameters (the requester, default weight and visible threshold), i.e., $\boldsymbol{\theta} \triangleq [r, \bar{w}, \tau]$.

Now we illustrate the change of visible set after adding a new incoming neighbor with Fig. 3. When $\mathcal{M} = \{8\}$, $\bar{w} = 3$, there is a new directed edge from user 8 to the requester, i.e., user 1, with weight $w_{8\to1} = \bar{w} = 3$. It can be interpreted as user 8 starts to follow user 1. The visibility distance of the added edge can be calculated as $d_{8\to1} = D(w_{8\to1}) = 1/(1 + w_{8\to1}) = 0.25$. One can find that the expansion of user 1's 0.6-visible set is $\{8, 9\}$ and the improvement of her 0.6-visibility is $\Delta_{\boldsymbol{\theta}}(\mathcal{M}) = |\{8, 9\}| = 2$.

The objective is to judiciously select the set $\mathcal{M} \subseteq \mathcal{U}'$ so as to maximize the improvement in the requester's τ-visibility under her *local information* (will be defined in Definition 3) while preserving *query anonymity* (will be defined in Definition 4). Note that the requester does not know the graph $\mathcal{G}$, but instead, she only has her local information, which is defined as follows.

Definition 3 (local information): The local information to a user u is defined as:

1) the identities and the visibility distance lengths of members in user u's τ-visible set;
2) the edges started from or ended at u;
3) upon establishing a link with a new incoming neighbor, the updated identities and visibility distance lengths of members in user u's new τ-visible set.

For example, in a social network like Facebook or Twitter, each user knows her local information, if the visible set is defined as the set of all the 1-hop neighbors. For more general cases of visible set such as multiple hop neighbors, a user can use the APIs provided by the OSN to do some simple local search to get the local information. In practice, due to privacy concerns, a user may not want to disclose the information (e.g., IDs, names, etc.) about members in her τ-visible set. For example, one may hide the viewers of their blogs. To this end, we define the notion of query anonymity.

Definition 4 (query anonymity): Query anonymity is satisfied when queries about members of any user's τ-visible set at any visible threshold τ is *not allowed*. An anonymity preserving query is a query that satisfies the query anonymity requirement.

With the notions of local information and query anonymity defined above.

Problem 1 (visibility maximization (VisMAX-A)): Suppose each user only has her local information defined in Definition 3. Given the graph $\mathcal{G}$, the requester r, the set $\mathcal{U}'$ of available users, the visible threshold τ, the default weight of the new edge $\bar{w}$, and the new incoming neighbor quota m, the objective is to select a set of users $\mathcal{M}$ as new incoming neighbors of r so to maximize the increase of visibility of r with query anonymity guarantees:

$$\begin{aligned} \underset{\mathcal{M}}{\text{maximize}} \quad & \Delta_{\boldsymbol{\theta}}(\mathcal{M}) \\ s.t. \quad & |\mathcal{M}| \leq m; \mathcal{M} \cap \mathcal{N}(r) = \emptyset; \mathcal{M} \subseteq \mathcal{U},' \\ & \text{Query anonymity is preserved,} \end{aligned}$$

where $\boldsymbol{\theta} = [r, \bar{w}, \tau]$ is a vector of given model parameters.

Remark: (1) There are two challenges in solving Problem 1. The first one is the computational challenge caused by the combinatorial structure of the problem. The second one is the query anonymity guarantee, which makes it difficult to evaluate the visibility improvement of adding one or several new available users as new incoming neighbors. (2) A closely related problem is the Influence Maximization (IM) problem. Even without the query anonymity requirement, conventional algorithms for the IM problem can not be directly applied to our problem due to the following reasons. First, the the social visibility model is the foundation of our work and it is different from diffusion models such as linear threshold model and independent cascade model, which are the foundation of the IM Problem. In particular, in these diffusion models, the influence is propagated in a recursive manner, while the social visibility is determined by the visibility distance. Second, our objective function is different from that of the IM problem. The objective function of the IM problem is the number of influenced nodes and it is #P-hard to compute the objective function for each given seed set under both linear threshold model and independent cascade model. However, the objective function in our problem is the improvement of visibility, which can be efficiently computed by a series of set operations. (3) Our model can be straightforwardly extended to capture important real-world factors such as preferences and dynamics of OSN. First, the available users form a subset of the user population. For different requesters, one can use different available user sets to capture preference, i.e., users can have varying preferences for different requesters, where the availability of a user models the preference to a requester. Second, when the OSN is evolving dynamically, each requester is served under a snapshot of the OSN. Our model can be directly applied to handle this snapshot of the OSN.

Idea of our approach. Our key idea is to trade the accuracy of evaluating the visibility improvement (by adding one or several new available users as new incoming neighbors) for anonymity. Thus, we will first address the computational challenge of Problem 1, assuming that an anonymity preserving query oracle (characterized by visibility improvement evaluating accuracy) is given. We quantify impact of the visibility improvement evaluating accuracy on the solution quality of Problem 1. This give us insights to design algorithms to implement the anonymity preserving query oracle attaining a good balance between visibility improvement evaluating accuracy and solution quality of Problem 1.

III. Algorithmic Framework

In this section, we design an algorithmic framework to solve Problem 1, assuming that an anonymity preserving query oracle (characterized by visibility improvement evaluating accuracy) is given. We also quantify impact of the visibility improvement evaluating accuracy on the solution quality of Problem 1. These insights will guide us to design algorithms to implement the anonymity preserving query oracle in Section IV.

A. Design of the Algorithmic Framework

NP-hardness Analysis. We first define a "perfect query oracle," which will be useful for us to study the fundamental computational challenge of Problem 1.

Definition 5 (Perfect query oracle): A perfect query oracle can return the exact value of $\Delta_{\boldsymbol{\theta}}(\mathcal{M})$ for each query of $\mathcal{M}$ with query anonymity guarantee.

Suppose that a perfect query oracle is given, the following theorem states the fundamental computational challenge of the VisMAX-A problem.

Theorem 1: Suppose that a perfect query oracle is given. The VisMAX-A problem, i.e., Problem 1, is NP-hard.

Remark: All technical proofs are presented in our supplementary file. Theorem 1 states that even when a perfect query oracle is given, it is computationally expensive to find the exact optimal solution for Problem 1. Problem 1 becomes more challenging when such a perfect query oracle is not given. Furthermore, designing such a perfect query oracle can usually be very difficult. Hence, we next design an algorithm to solve Problem 1 without assuming such a perfect query oracle.

The AdaExp Algorithmic Framework. To simplify the presentation, we define candidate and candidate set. Given a set of users $\mathcal{S}$ who have been newly added as incoming neighbors of the requester r, a user x is a candidate if he is (1) an available user (i.e., $x \in \mathcal{U}'$), (2) not in the latest incoming neighbor set of r (i.e., $x \notin \mathcal{N}(r) \cup \mathcal{S}$) and, (3) not trivial (i.e., $x \neq r$). Thus we define the set of all the potential new incoming neighbors as candidate set, denoted as

$$\mathcal{P}(\mathcal{S}) \triangleq \mathcal{U}' \setminus (\{r\} \cup \mathcal{N}(r) \cup \mathcal{S}).$$

To facilitate the design of the AdaExp algorithmic framework, in the following, we define, analyze and characterize the *marginal gain* of adding a new incoming neighbor.

Definition 6 (marginal gain): Given a set $\mathcal{S}$ of users who have been newly added as incoming neighbors of the requester r. The marginal gain of adding a candidate $x \in \mathcal{P}(\mathcal{S})$ is defined as $\delta(x, \mathcal{S}) \triangleq \Delta_{\boldsymbol{\theta}}(\mathcal{S} \cup \{x\}) - \Delta_{\boldsymbol{\theta}}(\mathcal{S})$.

The following lemma states a closed-form formula for the objective function $\Delta_{\boldsymbol{\theta}}(\mathcal{M})$.

Lemma 1: The objective function $\Delta_{\boldsymbol{\theta}}(\mathcal{M})$ can be derived as

$$\Delta_{\boldsymbol{\theta}}(\mathcal{M}) = |\cup_{v \in \mathcal{M}} \mathcal{V}(v, \tau - D(\bar{w})) \setminus \mathcal{V}(r, \tau)|,$$

where $|\mathcal{M}| \leq m$, $\mathcal{M} \cap \mathcal{N}(r) = \emptyset$ and $\mathcal{M} \subseteq \mathcal{U}'$.

Remark: Lemma 1 states that the objective function can be evaluated from the users' local information. Based on Lemma 1, we derive $\delta(x, \mathcal{S})$ in the following lemma.

Lemma 2: Given a set $\mathcal{S}$ of users who have been newly added as incoming neighbors of the requester r. The marginal gain of adding a candidate $x \in \mathcal{P}(\mathcal{S})$ can be derived as

$$\delta(x, \mathcal{S}) = |[\mathcal{V}(x, \tau - D(\bar{w})) \setminus \mathcal{V}(r, \tau)] \setminus [\cup_{v \in \mathcal{S}} \mathcal{V}(v, \tau - D(\bar{w}))]|.$$

Algorithm 1: Adaptive Expansion for VisMAX-A (AdaExp).

```
Input: the requester $r$, new incoming neighbor quota $m$
Output: new incoming neighbor set $\mathcal{M}_{\text{AdaExp}}$
$\mathcal{S} \leftarrow \emptyset$
while $|\mathcal{S}| < m$ do
    $\hat{v}^* \leftarrow \texttt{QueryOracle}(\epsilon, \mathcal{S})$
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{\hat{v}^*\}$
    $\mathcal{N}(r) \leftarrow \mathcal{N}(r) \cup \{\hat{v}^*\}$
end while
return $\mathcal{M}_{\text{AdaExp}} \leftarrow \mathcal{S}$
```

Remark: From Lemma 2, we can observe that the marginal gain of adding a candidate x can be computed from the users' local information defined in Definition 3. Based on the definition of marginal gain, we define the best candidate. Given a set $\mathcal{S}$ of users who have been added, we define the best candidate for current $\mathcal{S}$ as $v^*(\mathcal{S}) \in \arg\max_{x \in \mathcal{P}(\mathcal{S})} \delta(x, \mathcal{S})$. We denote the marginal gain associated with the best candidate $v^*(\mathcal{S})$ by $\delta^*(\mathcal{S}) \triangleq \max_{x \in \mathcal{P}(\mathcal{S})} \delta(x, \mathcal{S})$. We define an oracle to query about the best candidate.

Definition 7 (query oracle): A query oracle denoted by $\texttt{QueryOracle}(\epsilon, \mathcal{S})$ is a function which outputs a randomized estimation of the best candidate which satisfies $\mathbb{E}[\delta^*(\mathcal{S}) - \delta(\hat{v}^*, \mathcal{S})] \leq \epsilon$, where $\hat{v}^* = QueryOracle(\epsilon, \mathcal{S})$ denotes the output of the oracle.

We defer the detail of designing the query oracle to Section IV. In this section, we mainly focus on how to use the query oracle to design an algorithm to address the VisMAX-A problem. We also analyze how the performance of query oracle will influence the theoretical guarantee.

Algorithm 1 outlines our framework. The key idea is doing adaptive expansion on visible set, i.e., sequentially selecting and connecting with m users. In each round, only one candidate is added as a new incoming neighbor, and this candidate is directly decided by query oracle $\texttt{QueryOracle}(\epsilon, \mathcal{S})$. Combined with Definition 7, we know that this candidate is the best candidate estimated by the query oracle $\texttt{QueryOracle}(\epsilon, \mathcal{S})$. The computational complexity of Algorithm 1 is m times the complexity of the query oracle $QueryOracle(\epsilon, \mathcal{S})$.

B. Theoretical Guarantee

The "performance gap" of Algorithm 1 (AdaExp) arises from two parts. One is from the adaptive expansion and the other one is from the randomness caused by the query oracle. The following theorem presents the theoretical guarantee for Algorithm 1 (AdaExp).

Theorem 2: Let $\mathcal{M}_{\text{AdaExp}}$ denote the output of the Algorithm 1. Then we have

$$\mathbb{E}[\Delta_{\boldsymbol{\theta}}(\mathcal{M}_{\text{AdaExp}})] \geq (1 - 1/e)\Delta_{\boldsymbol{\theta}}(\mathcal{M}^*) - m\epsilon,$$

where $\mathcal{M}^*$ denotes the optimal solution of VisMAX-A via exhaustive search.

Remark: Theorem 2 states that the approximation ratio decreases as the bound of expected error ϵ of the query oracle increases. Our solution can achieve a high theoretical guarantee

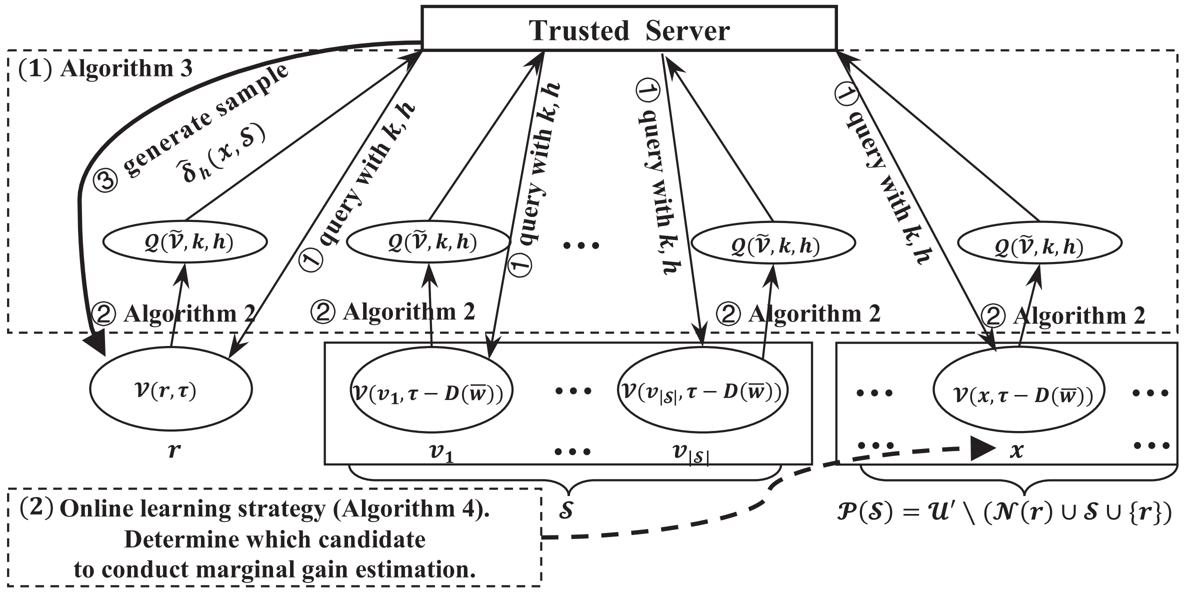

Trusted Server
(1) Algorithm 3
③ generate sample $\hat{\delta}_h(x,\mathcal{S})$
① query with k,h
① query with k,h
① query with k,h
① query with k,h
$\mathcal{Q}(\tilde{\mathcal{V}},k,h)$
$\mathcal{Q}(\tilde{\mathcal{V}},k,h)$
$\mathcal{Q}(\tilde{\mathcal{V}},k,h)$
$\mathcal{Q}(\tilde{\mathcal{V}},k,h)$
② Algorithm 2
② Algorithm 2
② Algorithm 2
② Algorithm 2
$\mathcal{V}(r,\tau)$
r
$\mathcal{V}(v_1,\tau-D(\bar{w}))$
v_1
$\mathcal{V}(v_{|\mathcal{S}|},\tau-D(\bar{w}))$
$v_{|\mathcal{S}|}$
$\mathcal{V}(x,\tau-D(\bar{w}))$
x
$\mathcal{S}$
$\mathcal{P}(\mathcal{S})=\mathcal{U}'\setminus(\mathcal{N}(r)\cup\mathcal{S}\cup\{r\})$
(2) Online learning strategy (Algorithm 4).
Determine which candidate
to conduct marginal gain estimation.

Fig. 4. The framework to estimate the best candidate $v^*(\mathcal{S})$ with anonymity guarantee.

when ϵ is small. To analyze the "performance gap" caused by the adaptive selection, we prove the monotonicity and submodularity of the objective function $\Delta_{\boldsymbol{\theta}}(\mathcal{M})$ in our supplementary file. We next design a framework to implement the query oracle.

IV. Query Oracle Design

In this section, we design an algorithm to implement the query oracle satisfying conditions in Definition 7. It is a novel combination of KMV sketch [2] and MAB online learning method (best arm identification version) [3]. We also show that the anonymity guarantee is achieved at a very small extra computational cost and a small reduction on the solution quality.

A. Key Idea of Query Oracle Design

The key idea is to design a framework to *estimate* the best candidate $v^*(\mathcal{S})$ with query anonymity guarantee and error bound guarantee, so that we can implement the query oracle and make the AdaExp algorithm applicable to the VisMAX-A problem. Fig. 4 outlines the framework of generating one *sample* of marginal gain $\delta(x,\mathcal{S})$ of adding user x as a new incoming neighbor of requester r. The logic of the estimation framework is as follows:

1) We consider a *trusted server*, who uses a probabilistic data structure like KMV sketch to generate anonymity preserving queries about users' visible sets. At each round, the trusted server uses these *query outcomes* to produce one *unbiased sample* $\hat{\delta}(x,\mathcal{S})$ on the marginal gain $\delta(x,\mathcal{S})$. We also use queries to produce unbiased samples on the marginal gain function $\delta(\cdot,\mathcal{S})$ of other candidates in $\mathcal{P}(\mathcal{S})$. The procedure of generating one sample is illustrated by the dashed rectangle (1) in Fig. 4.
2) To speed up the search process by reducing the rounds of samplings for estimating the best candidate $v^*(\mathcal{S})$, we design an online learning strategy to sequentially determine which user to conduct marginal gain sampling. This is denoted by the dashed rectangle (2) in Fig. 4.

In the following subsections, we proceed to introduce the details of each component.

B. Generate One Sample via KMV Sketch

Query one user with anonymity guarantee via KMV sketch [2]. For simplicity of presentation, let $\tilde{\mathcal{V}}$ denote the visible set that the trusted server queries for. For example, $\tilde{\mathcal{V}}=\mathcal{V}(v,\tau-D(\bar{w}))$ when the trusted server queries user v about her $(\tau-D(\bar{w}))$-visible set and $\tilde{\mathcal{V}}=\mathcal{V}(r,\tau)$ when the trusted server queries the requester r about her τ-visible set. The query outcome generated and returned by the queried user is denoted by $\mathcal{Q}(\tilde{\mathcal{V}},k,h)$, whose details will be made clear later.

As illustrated in Fig. 4, the trusted server only needs to send a one-way hash function h and sketch size k to the queried users, and each queried user applies Algorithm 2 with h and k to generate a KMV sketch of $\tilde{\mathcal{V}}$ (i.e., the query outcome $\mathcal{Q}(\tilde{\mathcal{V}},k,h)$) and sends it back to the trusted server.

Algorithm 2 outlines how the queried user generates the outcome $\mathcal{Q}(\tilde{\mathcal{V}},k,h)$ with the given parameters and her local information to response the trusted server, with the requirement of query anonymity satisfied. The one-way hash function h is used to map the ID of each element $v\in\tilde{\mathcal{V}}$ (the ID of the user lies in $[U]$, where $U=|\mathcal{U}|$) into a value in [0,1). Note that there are a number of ways to construct such one-way hash functions. In this work, we will not go into details on the construction. But rather we focus on the case that we are given a family of hash functions $\mathcal{H}$, such that $\{[h(x),x\in\mathcal{U}],h\in\mathcal{H}\}=[0,1]^U$.

Lemma 3: The computational complexity of Algorithm 2 is $O(\log k\cdot|\tilde{\mathcal{V}}|)$.

Remark: Lemma 3 states that the computational complexity of Algorithm 2 is linear to the cardinality of the queried set and is logarithmic to the sketch size.

Algorithm 2: Response Query With KMV Sketch.

```
Input: hash function $h$, sketch size $k$, visible set $\tilde{\mathcal{V}}$.
Output: the KMV sketch $\mathcal{Q}(\tilde{\mathcal{V}}, k, h)$.
Notation: $max(\mathcal{K})$ // return the largest value in $\mathcal{K}$
$\mathcal{K} \leftarrow \emptyset$
for each $v \in \tilde{\mathcal{V}}$ do
    $val = h(v)$
    if $|\mathcal{K}| < k$ then
        insert $val$ into $\mathcal{K}$
    end if
    if $|\mathcal{K}| \geq k$ and $val < max(\mathcal{K})$ then
        remove $max(\mathcal{K})$
        insert $val$ into $\mathcal{K}$
    end if
end for
return $\mathcal{Q}(\tilde{\mathcal{V}}, k, h) \leftarrow \mathcal{K}$
```

Algorithm 3: Generate a Sample of the Marginal Gain $\delta(x, \mathcal{S})$.

```
Input: hash function $h$, sketch size $k$, users $\{x, r\} \cup \mathcal{S}$.
Output: estimated marginal gain $\hat{\delta}_h(x, \mathcal{S})$.
Apply Algorithm 2 to generate the following query outcomes
    $\mathcal{Q}(\mathcal{V}(v, \tau - D(\bar{w})), k, h), \forall v \in \{x\} \cup \mathcal{S},$
    $\mathcal{Q}(\mathcal{V}(r, \tau), k, h).$
$K_{\oplus} \leftarrow |[\mathcal{Q}(\mathcal{V}(x, \tau - D(\bar{w})), k, h) \setminus \mathcal{Q}(\mathcal{V}(r, \tau), k, h)]$
        $\setminus \cup_{v \in \mathcal{S}} \mathcal{Q}(\mathcal{V}(v, \tau - D(\bar{w})), k, h)|$
$q \leftarrow$ the $k$-th smallest value in the following set
    $\left[\cup_{v \in \mathcal{S} \cup \{x\}} \mathcal{Q}(\mathcal{V}(v, \tau - D(\bar{w})), k, h)\right] \cup \mathcal{Q}(\mathcal{V}(r, \tau), k, h).$
return $\hat{\delta}_h(x, \mathcal{S}) \leftarrow \frac{K_{\oplus}}{k} \frac{k-1}{q}$
```

The query outcome $\mathcal{Q}(\tilde{\mathcal{V}}, k, h)$ consists of k minimum hashed values (that's why it is named as KMV sketch) and has the following nice properties. First, the query outcome $\mathcal{Q}(\tilde{\mathcal{V}}, k, h)$ preserves the anonymity of the members in the visible set $\tilde{\mathcal{V}}$ because the one-way hash function h is computationally infeasible to invert. Second, the query outcome $\mathcal{Q}(\tilde{\mathcal{V}}, k, h)$ can be applied to estimate the *cardinality* of the visible set $\tilde{\mathcal{V}}$ as follows. Suppose we have selected a one-way hash function h from $\mathcal{H}$, then as the set cardinality gets larger, the k-th smallest hashed value becomes smaller. Therefore, the k-th smallest hashed value carries information of the cardinality of a set. Formally, we have the following lemma to estimate the cardinality of $\tilde{\mathcal{V}}$.

Lemma 4: The query outcome satisfies

$$\mathbb{E}_{h \sim Uniform(\mathcal{H})}\left[\frac{k-1}{\max \mathcal{Q}(\tilde{\mathcal{V}}, k, h)}\right] = |\tilde{\mathcal{V}}|,$$

where $\max \mathcal{Q}(\tilde{\mathcal{V}}, k, h)$ is the maximum value in $\mathcal{Q}(\tilde{\mathcal{V}}, k, h)$.

Remark: Lemma 4 states that the KMV sketch can be applied to produce an unbiased estimator on the cardinality.

Unbiased samples of $\delta(x, \mathcal{S})$ via multiple queries. As an extension of Lemma 4, we obtain the following theorem, which guides us to use the query outcomes returned by multiple involved users to generate unbiased samples on the marginal gain function $\delta(x, \mathcal{S})$.

Theorem 3 (Restatement from [2]): Suppose the trusted server has query outcomes (i.e., the KMV sketches) of n users with visible sets $\tilde{\mathcal{V}}_i, \forall i \in [n]$, denoted by $\mathcal{Q}(\tilde{\mathcal{V}}_i, k, h), \forall i \in [n]$. Then we have

$$\mathbb{E}_{h \sim Uniform(\mathcal{H})}\left[\frac{|\oplus(\mathcal{Q}(\tilde{\mathcal{V}}_1, k, h), \ldots, \mathcal{Q}(\tilde{\mathcal{V}}_n, k, h))|}{k} \frac{k-1}{q}\right] = |\oplus(\tilde{\mathcal{V}}_1, \ldots, \tilde{\mathcal{V}}_n)|, \tag{2}$$

where the function $\oplus(\cdot, \ldots, \cdot)$ denotes the result of a sequence of set operations over its parameters, and q is the k-th smallest value in the union of queries outcomes $\cup_{i \in [n]} \mathcal{Q}(\tilde{\mathcal{V}}_i, k, h)$.

Remark: Theorem 3 states that the KMV sketch can be applied to produce an unbiased estimator on the cardinality of the operations (i.e., union, intersection, etc.) of multiple sets. Recall that Lemma 2 provides a formula for the marginal gain function $\delta(x, \mathcal{S})$, which is the cardinality of the set obtained by a number of set operations over several sets, i.e., $\mathcal{V}(v, \tau - D(\bar{w}))$ for $\forall v \in \{x\} \cup \mathcal{S}$ and $\mathcal{V}(r, \tau)$ for the requester r. With this observation and Theorem 3, we design Algorithm 3, which shows how the trusted server generates an unbiased sample $\hat{\delta}_h(x, \mathcal{S})$ of the marginal gain $\delta(x, \mathcal{S})$ using the query outcomes $\mathcal{Q}(\tilde{\mathcal{V}}, k, h)$ returned by involved users. Finally, the trusted server sends the sample $\hat{\delta}_h(x, \mathcal{S})$ to the requester r, and the requester r selects a candidate x for the next round of sampling. Note that the sample $\hat{\delta}_h(x, \mathcal{S})$ preserves the anonymity as it is only a real number.

Lemma 5: The computational complexity of Algorithm 3 is $O(\log k \cdot m \cdot U)$.

Remark: Lemma 5 states that the computational complexity of Algorithm 3 is $O(\log k \cdot m \cdot U)$ in the worst case. The following corollary states the unbiasedness of the sample.

Corollary 1: The sample $\hat{\delta}_h(x, \mathcal{S})$ generated by Algorithm 3 is unbiased, i.e, $\mathbb{E}_{h \sim Uniform(\mathcal{H})}[\hat{\delta}_h(x, \mathcal{S})] = \delta(x, \mathcal{S})$.

Remark: Corollary 1 implies that one can estimate $\delta(x, \mathcal{S})$ accurately by generating a sufficiently large number of IID unbiased samples $\hat{\delta}_h(x, \mathcal{S})$ with different hash functions h. There would not be information leakage for the following reasons. First, the responses to the queries are processed through hash function which guarantees the IDs of users are encrypted. Second, only samples $\hat{\delta}_h(x, \mathcal{S})$ would be revealed to the users, which are just real numbers. Third, the responses are submitted to the trusted server for targeting available users and not accessible from other users.

C. Best Candidate Estimation via MAB

Best candidate estimation algorithm. Given the set $\mathcal{S}$ of users who have been adaptively added as new incoming neighbors of the requester r. We consider a sampling budget T, i.e., the number of rounds the requester r can do sampling using Algorithm 3 to find out the best candidate. In round $t \in [T]$, the requester r selects a user $x_t \in \mathcal{P}(\mathcal{S})$. Then, the trusted server first selects a hash function h_t from $\mathcal{H}$ uniformly at random, and generates a sample $\hat{\delta}_{h_t}(x_t, \mathcal{S})$ via querying users in

Algorithm 4: QueryOracle($\epsilon, \mathcal{S}$).

```
Input: the set of users who have been added $\mathcal{S}$, the set of candidates $\mathcal{P}(\mathcal{S})$ with $N = |\mathcal{P}(\mathcal{S})|$, the number of rounds $T$, exploration parameter $a > 0$.
Output: the estimated best candidate $\hat{v}_T^*(\mathcal{S})$.
for each round $t = 1, \ldots, N$ do
    Select hash function $h_t$ from $\mathcal{H}$ uniformly at random
    Select a user $v_t \in \mathcal{P}(\mathcal{S}) \setminus \cup_{i=1}^{t-1}\{v_i\}$
    Apply Algorithm 3 to generate a sample $\hat{\delta}_{h_t}(v_t, \mathcal{S})$
    Initialize the sample average: $\bar{r}_N(v_t) \leftarrow \hat{\delta}_{h_t}(v_t, \mathcal{S})/U$
end for
Initialize the number of queries $n_{x,N} \leftarrow 1, \forall x \in \mathcal{P}(\mathcal{S})$
for each round $t = N+1, \ldots, T$ do
    Select one candidate to query
        $x \in \arg\max_{v\in\mathcal{P}(\mathcal{S})} \bar{r}_{t-1}(v) + \sqrt{\frac{a}{n_{v,t-1}}}$
    Select hash function $h_t$ from $\mathcal{H}$ uniformly at random
    Apply Algorithm 3 to generate a sample $\hat{\delta}_{h_t}(x, \mathcal{S})$
    Update the number of queries
        $n_{x,t} \leftarrow n_{x,t-1} + 1,$
        $n_{v,t} \leftarrow n_{v,t-1}, \forall v \in \mathcal{P}(\mathcal{S}) \setminus \{x\}.$
    Update the sample average:
        $\bar{r}_t(x) \leftarrow \left(1 - \frac{1}{n_{x,t}}\right)\bar{r}_{t-1}(x) + \frac{\hat{\delta}_{h_t}(x, \mathcal{S})/U}{n_{x,t}}.$
        $\bar{r}_t(v) \leftarrow \bar{r}_{t-1}(v), \forall v \in \mathcal{P}(\mathcal{S}) \setminus \{x\}.$
end for
Return: $\hat{v}_T^*(\mathcal{S}) \in \arg\max_{x\in\mathcal{P}(\mathcal{S})}\bar{r}_T(x)$
```

$\{x_t, r\} \cup \mathcal{S}$ as illustrated in Algorithm 3 with h_t and sends $\hat{\delta}_{h_t}(x_t, \mathcal{S})$ to r. Based on the results of sampling history, the requester r then selects the next user x_{t+1} from $\mathcal{P}(\mathcal{S})$ for $(t+1)$-th round and gets sample $\hat{\delta}_{h_{t+1}}(x_{t+1}, \mathcal{S})$. The objective is to estimate the best candidate with high accuracy. One challenge to achieve the above goal is the efficiency and coverage trade-off. On the one hand, the requester r should use more rounds to generate samples of users who are very likely to be the best candidate $v^*(\mathcal{S})$, so as to differentiate it from other "good" ones with high confidence. On the other hand, r should generate sufficient samples for every candidate to avoid being trapped among some sub-optimal candidates.

We design an online learning algorithm to address this challenge, which is outlined in Algorithm 4. The key idea of Algorithm 4 is that in each round t, we select the user $v \in \mathcal{P}(\mathcal{S})$ who has the maximum value of certain *index* to do the sampling in the next round. The *index* for each candidate $v \in \mathcal{P}(\mathcal{S})$ is the empirical average of the *normalized samples* for user v plus a *penalty*. The penalty decreases in the number of rounds that user v has been sampled, hence giving certain opportunities to do sampling on other users who have been rarely sampled. Finally, after T rounds of samplings, the Algorithm 4 outputs the user with highest empirical average of the normalized samples as the estimated best candidate, denoted by $\hat{v}_T^*(\mathcal{S})$.

Lemma 6: The computational complexity of Algorithm 4 is $O(\log k \cdot m \cdot U \cdot T)$. Taking Algorithm 4 as the query oracle, Algorithm 1 (AdaExp) has a computational complexity of $O(\log k \cdot m^2 \cdot U \cdot T)$.

Remark: Lemma 6 states that the computational complexity of Algorithm 1 (AdaExp) is quadratic in the new incoming neighbor quota m, logarithmic in sketch size k and linear in the sampling budget T and network size U in the worst case.

Theoretical guarantees for the estimation algorithm. To present the theoretical guarantee of Algorithm 4, we define the following notation to quantify the hardness of estimating the best candidate

$$H \triangleq \sum\nolimits_{x\in\mathcal{P}(\mathcal{S})} \frac{1}{\Delta_x^2},$$

where we define $\Delta_x = \frac{\delta^*(\mathcal{S})}{U} - \frac{\delta(x,\mathcal{S})}{U}, \forall x \in \mathcal{P}(\mathcal{S}) \setminus \{v^*(\mathcal{S})\}$ and $\Delta_{v^*(\mathcal{S})} = \min_{x\in\mathcal{P}(\mathcal{S})\setminus\{v^*(\mathcal{S})\}}\Delta_x$. The following theorem states the theoretical guarantee of Algorithm 4.

Theorem 4: If Algorithm 4 is run with parameter

$$0 < a \leqslant \frac{25}{36}\frac{T - |\mathcal{P}(\mathcal{S})|}{H},$$

then the output $\hat{v}_T^*(\mathcal{S})$ of Algorithm 4 satisfies

$$\mathbb{E}[\delta^*(\mathcal{S}) - \delta(\hat{v}_T^*, \mathcal{S})] \leqslant 2\, TU|\mathcal{P}(\mathcal{S})|\exp\left(-\frac{2a}{25}\right).$$

Remark: (1) Theorem 4 states that the expected error, i.e., the difference between marginal gain brought by the ground truth best candidate $v^*(\mathcal{S})$ and the estimated best candidate $\hat{v}_T^*(\mathcal{S})$ decreases in round T with an exponential rate. Finally, combining Definition 7 (query oracle) with Theorem 4, we can observe that, Algorithm 4 is a query oracle with ϵ. Namely, Algorithm 4 implements a query oracle that satisfies Definition 7. In practice, the exact value of H is unknown, which makes it difficult to determine the key parameter a. One way to address this challenge is via the following upper bound of H:

$$\begin{aligned} H &\triangleq \sum_{x\in\mathcal{P}(\mathcal{S})} \frac{1}{\Delta_x^2} \leq \sum_{x\in\mathcal{P}(\mathcal{S})} \frac{1}{\Delta_{\min}^2} \\ &\leq \sum_{x\in\mathcal{P}(\mathcal{S})} \frac{1}{(1/U)^2} = \frac{|\mathcal{P}(\mathcal{S})|}{(1/U)^2} = |\mathcal{P}(\mathcal{S})|U^2, \end{aligned}$$

where $\Delta_{\min} = \min_{x\in\mathcal{P}(\mathcal{S})}\Delta_x$.

V. Performance Evaluation

In this section, we conduct experiments on public datasets to evaluate the performance of Algorithm 1 (AdaExp). Experimental results show that our algorithm has a high computational efficiency and high accuracy.

A. Experimental Settings

Datasets. We use four public datasets to conduct experiments. Table I shows overall statistics of these four datasets.

TABLE I
STATISTICS OF FOUR DATASETS AND THE ASSOCIATED DEFAULT PARAMETERS

datasets	#nodes	#links	directed	weighted	τ	$D(\bar{w})$
Residence	217	2,672	yes	yes	0.9	0.6
Blogs	1,224	19,025	yes	no	3	1
Facebook	5,908	41,729	no	no	2	1
DBLP	10,000	55,734	no	yes	0.7	0.3
GitHub	37,700	289,003	no	no	2	1
Epinions	75,888	508,837	yes	no	2	1

• *Residence [4]:* it contains friendship ratings between 217 residents living at a residence hall located on the Australian National University campus. Nodes correspond to residents and edges correspond to friendship ties which is directed and weighted. The weights of edges indicate the strength of friendship, which vary from the strongest to the weakest: 5 (best friend), 4 (close friend), 3 (friend), 2, 1.

• *Blogs [4]:* it contains front-page hyperlinks between blogs in the context of 2004 US election. Nodes correspond to blogs and edges correspond to hyperlinks between blogs.

• *Facebook [5]:* it is a page-page network of verified Facebook sites where nodes correspond to official Facebook pages and edges to mutual likes between sites.

• *DBLP [6]:* it is a co-author network extracted from the DBLP Bibliography where nodes correspond scholars who have published papers in major conferences and edges to co-author relationships between two scholars. The weights of edges indicate the number of cooperations between two scholars. • *GitHub [5]:* it is a social network where nodes correspond to developers who have starred at least 10 repositories and edges to mutual follower relationships. • *Epinions [4]:* it is a "trust" network from Epinions.com, a consumer review site where users can post reviews for products. Nodes correspond to users and edges correspond to the trust relationship between users. For example, a directed edge from user a to b represents a trusting b.

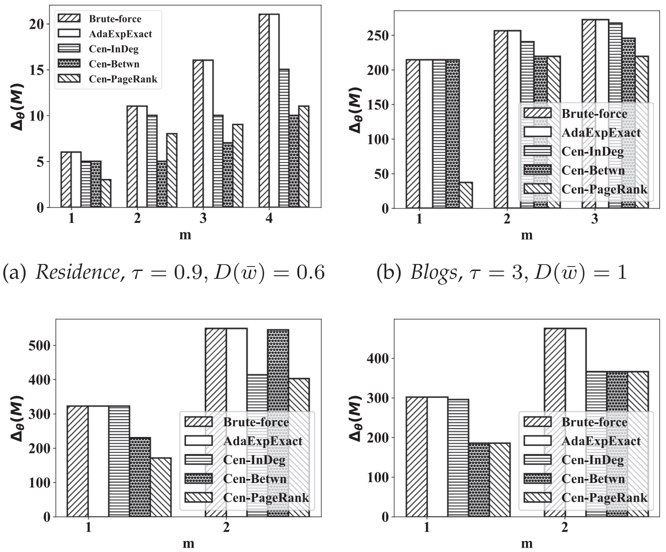
One may observe that these datasets summarized in Table I are not in a large scale. One reason for selecting them is that some comparison baselines (will be introduced later) such as the brute-force algorithm are time consuming on large datasets, due to high computational complexity.

Parameter setting. Note that the raw datasets in Table I do not contain any information on visibility distance function, visible threshold τ, and the default visibility distance $D(\bar{w})$ of a new added link. We choose values of parameters τ and $D(\bar{w})$ is as follows. For unweighted datasets, we assign value 1 to weights and visibility distances of all the edges, as well as the default weight $\bar{w}$ and its corresponding default visibility distance $D(\bar{w})$ of a new added link. We set the visible threshold as a value which results in a reasonable size of visible set for most users in the network. For unweighted datasets, we set the visible threshold of datasets from Facebook, GitHub and Epinions as $\tau = 2$ and set the visible threshold of dataset Blogs as $\tau = 3$. For weighted datasets, we select some typical forms of visibility distance function as follows. For dataset Residence, as the weight of edges lie in a small range, i.e., $\{1, 2, 3, 4, 5\}$, we consider a linear visibility distance function $D(w) = 1 - 0.2(w - 1)$, where $D(w)$ varies in the range of $D(5) = 0.2$ to $D(1) = 1$. We set the weight of newly added links as $\bar{w}$ satisfying $D(\bar{w}) = 0.6$ which can be a suitable visibility distance for new established connection. We then set the visible threshold for dataset Residence as $\tau = 0.9$, which results in a reasonable size of visible set for most users. For dataset DBLP, we find that the weight of edges has a high variance, i.e., typically between 1 to 60 and some even larger, so we consider the nonlinear visibility distance function $D(w) = 1/(1 + w)$. With this visibility distance function, we set the weight of newly added links as $\bar{w}$ satisfying $D(\bar{w}) = 0.3$ and set the visible threshold for the DBLP dateset as $\tau = 0.7$. We select a typical requester r as follows. We first use above parameters to compute the τ-visible set of each node and then select a requester r with a median size visible set in the dataset. Unless we vary them explicitly, we consider the following default parameters: the sample budget $T = 4\ U$ (U is the number of the nodes in the network), and the default size of the KMV sketch $k = 50$. Note that we will also vary T and k respectively to study their impact on the running time and the performance of our algorithm. Unless we state explicitly, we consider the most computationally challenging case where all users are available to the requester, i.e., $\mathcal{U}' = \mathcal{U}$. We also study the case where only a subset of users are available to the requester.

Metrics & baselines. We use $\Delta_{\boldsymbol{\theta}}(\mathcal{M})$, the improvement in visibility as the evaluation metric. Note that the "error" of the AdaExp algorithm arises from two parts. One is from the adaptive expansion and the other one is from the query oracle. Thus, we compare our algorithm with the following baselines. (1) *Brute-force:* selects the optimal set of new incoming neighbors via exhaustive search. (2) *AdaExpExact:* is a variant of AdaExp algorithm, where a perfect query oracle (i.e., $\epsilon = 0$ in Definition 7) is given. (3) *Cen-InDeg:* selects m candidates with the largest in-degree centrality [7]. (4) *Cen-Betwn:* selects m candidates with the largest betweenness centrality [8]. (5) *Cen-PageRank:* selects m candidates with the largest PageRank score [9]. Although the social visibility maximization problem has certain connections to link prediction problem, influence maximization problem and friend recommendation problem, we do not compare with algorithms of these problems due to the following reasons. First, algorithms for these problems do not provide anonymity guarantee, while our algorithm does. Second, the objectives of link prediction problem and friend recommendation problem are not to maximize a user's visibility, and the influence maximization problem considers influence diffusion models which are different from our social visibility model. One can refer to Section VII for more details on the differences. Furthermore, to the best of our knowledge, we are the first to study the visibility maximization problem with anonymity guarantees in OSNs. Thus, comparing to the above five baselines is sufficient.

B. The Impact of Adaptive Expansion

We aim to evaluate the effectiveness and efficiency of the adaptive expansion method developed in the AdaExp algorithm.



(a) *Residence*, $\tau = 0.9, D(\bar{w}) = 0.6$ (b) *Blogs*, $\tau = 3, D(\bar{w}) = 1$

(c) *Facebook*, $\tau = 2, D(\bar{w}) = 1$ (d) *DBLP*, $\tau = 0.7, D(\bar{w}) = 0.3$

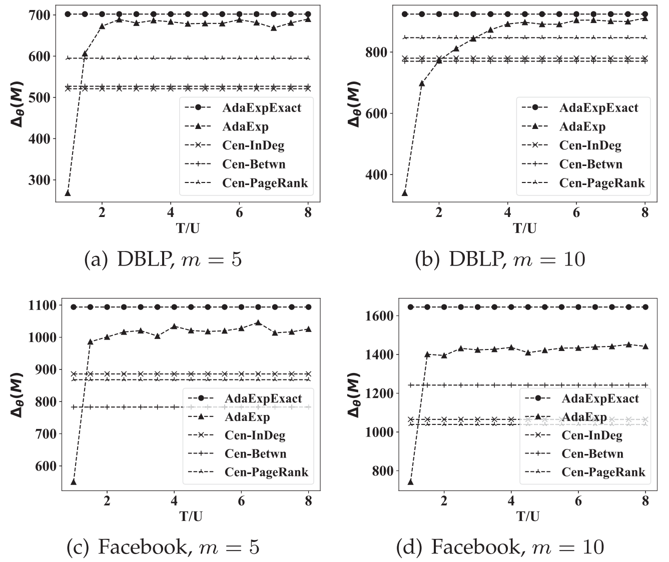
Fig. 5. Comparing AdaExpExact with brute-force method and heuristic methods in terms of visibility improvement.

To eliminate the potential bias caused by the query oracle, we focus on AdaExpExact. We compare AdaExpExact with Brute-force and three simple baseline methods, i.e., Cen-InDeg, Cen-Betwn and Cen-PageRank. Through this, we also show the problem is non-trivial even without the anonymity guarantee. Note that the brute-force method we use to get the optimal solution comes with a high computational cost, so one can only apply it to relatively small datasets and with relatively small quota m. Fig. 5 shows that AdaExpExact achieves nearly the same visibility improvement as the brute-force method for all the datasets and for all values of m we tried. This implies that the error arises from the "adaptive" part of AdaExp is very small. In addition, AdaExpExact significantly outperforms Cen-InDeg, Cen-Betwn and Cen-PageRank. This implies that our algorithm is non-trivial even there is no requirement to guarantee anonymity.

C. The Impact of Anonymity Guarantee

We aim to evaluate the efficiency and effectiveness of the proposed query oracle. Recall that the proposed query oracle (i.e., Algorithm 4) has two key parameters: the sampling budget T and the size of KMV sketch k. Note that the new incoming neighbor quota m also influences the performance of AdaExp. In the following, we study their impact respectively. We also compare AdaExp with centrality-based heuristic methods, i.e., Cen-InDeg, Cen-Betwn and Cen-PageRank. In this subsection, for brevity, we only present the result on two largest datasets in Table I, i.e., Facebook and DBLP. Note that we have similar results on the other two smaller datasets.

Impact of sampling budget T. We set the size of the KMV sketch as $k = 50$ and we will study the impact of k in the next experiment. We consider two different selections of the quota on the new incoming neighbor, i.e., $m = 5$ and $m = 10$, respectively. Fig. 6 shows visibility improvement $\Delta_\theta(\mathcal{M})$ achieved by AdaExp, AdaExpExact and three simple baseline methods on datasets DBLP and Facebook. Note that the number of the nodes in dataset DBLP is $U = 10\,000$, we thus vary the sampling budget T from U to 8 U with step 0.5 U. Fig. 6(a) shows that as the sampling budget T increases, AdaExp achieves a larger visibility improvement. When the sampling budget is around 3 U, AdaExp achieves a visibility improvement very close to AdaExpExact. AdaExp almost always outperforms heuristic methods unless the sampling budget T is very small. Thus, each candidate will be sampled around 3 times on average. This implies that AdaExp can utilize the sample budget efficiently. Fig. 6(b) further validates this observation when we vary the new incoming neighbor quota from 5 to 10. However, AdaExp achieves a visibility improvement very close to AdaExpExact when the sampling budget is around 4 U, which is slightly larger than $m = 5$. This is probably because the algorithm performs better when the quota m is smaller and less sampling budget is needed to achieve a certain accuracy. Fig. 6(c) and (d) show the results on dataset Facebook. For both $m = 5$ and $m = 10$, as the sampling budget increases from 1.5 U to 8 U,the visibility improvement goes up mildly. The sampling budget T has a smaller impact on dataset Facebook than dataset DBLP. Furthermore, AdaExp outperforms centrality-based heuristic method when $T \geq 1.5$ U.

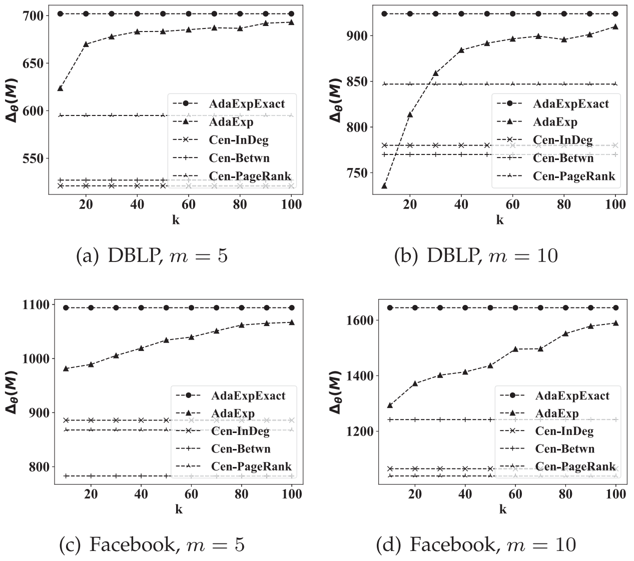


(a) DBLP, $m = 5$ (b) DBLP, $m = 10$

(c) Facebook, $m = 5$ (d) Facebook, $m = 10$

Fig. 6. The impact of T on visibility improvement of AdaExp ($k = 50$).

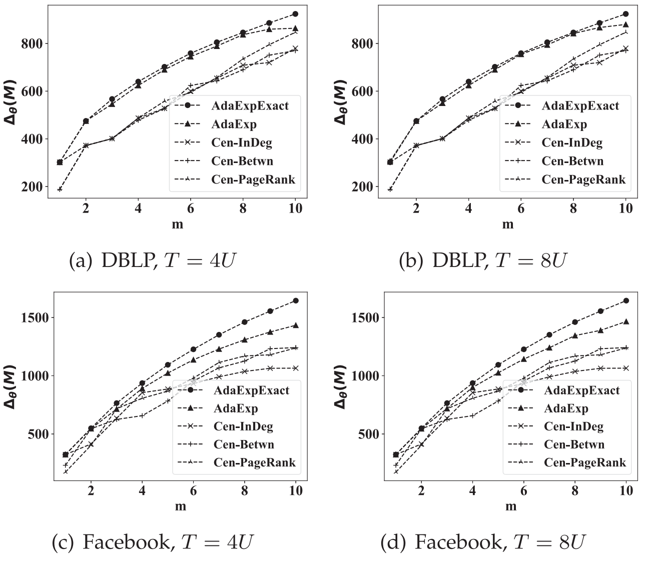
Impact of sketch size k. Now, we fix the sampling budget $T = 4$ U. We also consider $m = 5$ and $m = 10$, respectively. We vary the sketch size k from 10 to 100 with step 10 to study its impact on the performance of AdaExp. Fig. 7 shows the visibility improvement achieved by AdaExp, AdaExpExact and heuristic methods based on dataset DBLP and dataset Facebook. Fig. 7(a) shows that as the sketch size k increases, AdaExp achieves a larger improvement in the visible set. When the sketch size is around 50, AdaExp achieves an visibility improvement close to AdaExpExact. This is important because it implies that AdaExp only requires a small sketch size to generate a good estimation. Namely, the computational cost of the trusted server and queried users is small. AdaExp almost always outperforms heuristic methods unless the size of KMV sketch k is very small, indicating that small sketch size



(a) DBLP, $m = 5$ (b) DBLP, $m = 10$

(c) Facebook, $m = 5$ (d) Facebook, $m = 10$

Fig. 7. The impact of k on the visibility improvement of AdaExp (T=4 U).



(a) DBLP, $T = 4U$ (b) DBLP, $T = 8U$

(c) Facebook, $T = 4U$ (d) Facebook, $T = 8U$

Fig. 8. The impact of m on the visibility improvement of AdaExp ($k = 50$).

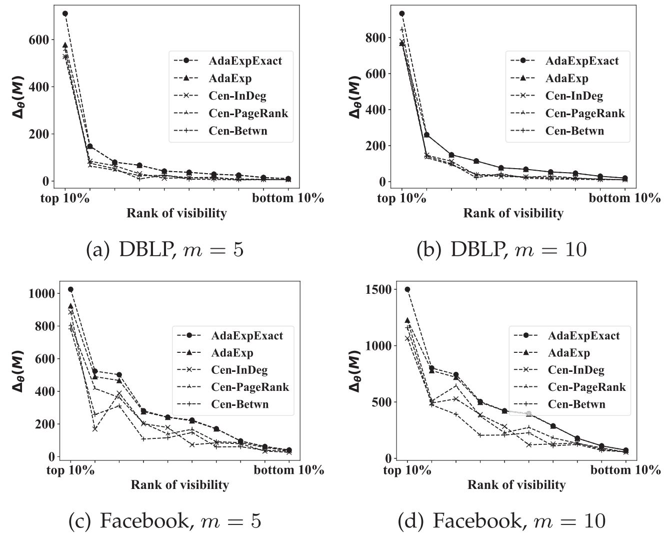
can cause inaccurate estimation. Fig. 7(b) further validates this observation when we vary m from 5 to 10. Fig. 7(c) and (d) show the results on dataset Facebook. Fig. 7(c) shows that as the sketch size k increases, AdaExp achieves a larger improvement in the visible set. Compared to dataset DBLP, dataset Facebook requires larger sketch size to achieve a visibility improvement close to AdaExpExact Fig. 7(d) further validates this observation when we vary m from 5 to 10.

Impact of the quota on the new incoming neighbor m. Now, we fix the sketch size $k = 50$ and consider two different selections of sampling budget T, i.e., $T = 4\ U$ and $T = 8\ U$ respectively. Fig. 8 shows the visibility improvement achieved by AdaExp, AdaExpExact and heuristic methods with different quota m on the datasets DBLP and Facebook. We vary the quota m on the new incoming neighbor from 1 to 10 to study its impact on the performance of AdaExp. Fig. 8(a) shows that as the new incoming neighbor quota m increases, the visibility improvement achieved by all these three methods on dataset DBLP when sketch size $k = 50$ the sampling budget $T = 4\ U$. Besides, there is a diminishing increase as m increases. Also, one can observe that AdaExp and AdaExpExact always achieve a better result than heuristic method. Furthermore, AdaExp achieves a similar visibility improvement as AdaExpExact, but the gap between them increases as new incoming neighbor quota m increases. Fig. 8(b) shows the result of dataset DBLP when we vary the sampling budget from $T = 4\ U$ to $T = 8\ U$. The difference between AdaExpExact and AdaExp is slightly smaller. These imply that AdaExp has a high accuracy under different new incoming neighbor quota m, especially when m is small. Fig. 8 (c) and (d) show a similar result on dataset Facebook.

Impact of available users. In this subsection, we explore the visibility improvement achieved under different settings of available users. We consider that only ten percent of users are available and evaluate AdaExp under 10 settings of available users: ranked top 10% by visibility, ranked 10%-20% by visibility, ..., ranked in 80%-90% by visibility, ranked in bottom 10% by visibility. Intuitively, one can guess that for the same requester, the setting where available users are from top-ranked users can achieve larger visibility improvement than the setting where available users are from bottom-ranked users. However, it is not obvious how much the difference would be. We study the visibility improvement achieved by AdaExpExact and AdaExp on the datasets DBLP and Facebook under 10 different settings stated as above. The parameters of networks we used are still the same as Table I. When we experiment on AdaExp, we set the sampling budget as $T = 0.4\ U$ (since the number of choice is $0.1\ U$ now) and the sketch size $k = 50$. We consider two different selection of the quota, $m = 5$ and $m = 10$. Fig. 9 shows the visibility improvement achieved by AdaExp and AdaExpExact on these two datasets under different settings of available users. Fig. 9 (a) and (b) show that for both $m = 5$ and $m = 10$, the visibility improvement achieved by AdaExp and AdaExpExact decrease as available users' visibility ranking get lower, and there is a sharp decrease when available users degrade from top 10% to 10%-20%. Besides, from Fig. 9(b), one can observe that heuristic methods can achieve a larger visibility improvement than AdaExp when the available users are top 10% and $m = 10$. Furthermore, except for the setting of top 10%, AdaExp almost achieves the same visibility improvement as AdaExpExact, indicating a high accuracy of AdaExp when available users are not top-ranked by visibility. Fig. 9(c) and (d) show a similar result on dataset Facebook.

D. Running Time Analysis

To understand the scalability of our method, we study the impact of U, T, m and k on the running time respectively. The running time is evaluated on a single MacBook PC with an Apple M1 Pro processor and 16 GB of memory. We randomly select $0.2\ U$ users from $\mathcal{U}$ as available users. To study the impact of network size U on the running time, we experiment on three relatively large datasets DBLP, GitHub and Epinions. For all these three datasets, we set $k = 50$, $T = 4\ U$ and



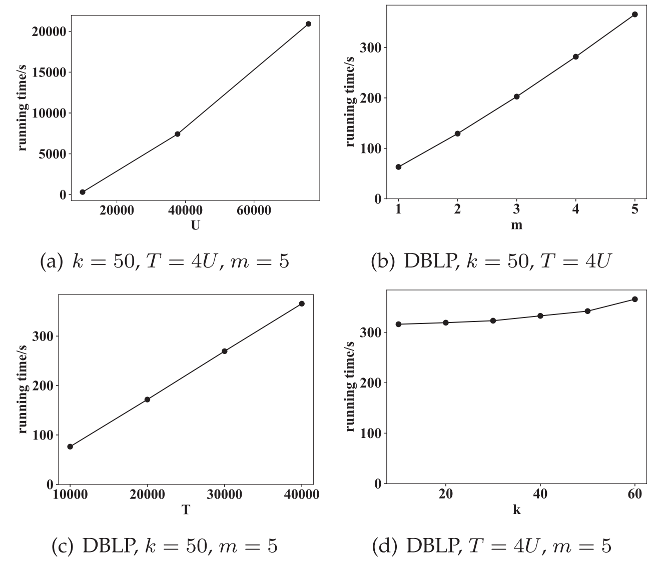
(a) DBLP, $m = 5$

(b) DBLP, $m = 10$

(c) Facebook, $m = 5$

(d) Facebook, $m = 10$

Fig. 9. The impact of available users settings on the visibility improvement of AdaExp.



(a) $k = 50, T = 4U, m = 5$

(b) DBLP, $k = 50, T = 4U$

(c) DBLP, $k = 50, m = 5$

(d) DBLP, $T = 4U, m = 5$

Fig. 10. The impact of U, m, T and k on the running time.

$m = 5$. Fig. 10(a) shows that the running time is almost linear with the network size U. This shows that our algorithm is scale in network size. Fig. 10(b) shows the impact of quota m on the running time on dataset DBLP, where we set $T = 4\ U$, $k = 50$ and vary m among $1, 2, 3, 4 \text{and} 5$. The results show that the running time is almost linear to m, which indicates our algorithm is scale in the quota m. Note that the observed relationship between m and running time is more satisfactory than the quadratic relationship stated in Lemma 6. This is because Lemma 6 states the time complexity in the worst case, and in practice the average cardinality of queried sets decreases largely as m increases. Fig. 10(c) shows the impact of sampling budget T on the running time on dataset DBLP, where we set $k = 50$, $m = 5$ and very T among $1\ U, 2\ U, 3\ U \text{and} 4\ U$. We can observe that the running time is linear to T. This shows that our algorithm is scale in the sample size. Fig. 10(d) shows the impact of sketch size k on the running time on dataset DBLP, where we set $T = 4\ U$, $m = 5$ and very k among $30, 50 \text{and} 70$. We can see that the running time increases slightly as k increases. Thus, it is efficient to improve the accuracy by increasing the sketch size with small increase of time cost. These results about running times also verify the time complexity analysis.

VI. Application of Our Framework

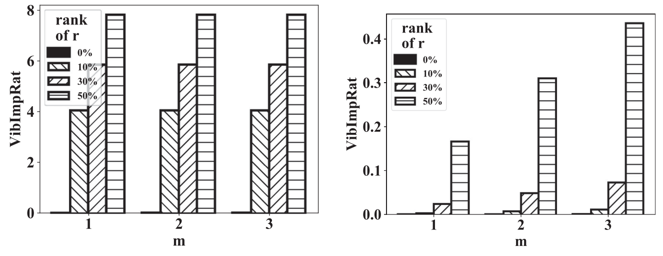
In this section, we apply our framework to two real-world applications (Epinions and DBLP). We aim to answer which users in Epinions or DBLP would have significant improvement on their social visibility by adding only a few new incoming neighbors. We study two settings, (1) the setting that all users in the network are available users who are willing to be a new incoming neighbor of the requester and, (2) the setting that only a subset of the users are available users.

A. Application to Epinions

Epinions.com is a consumer review site where users can write reviews for products. Each user of the site can decide whether to "trust" other users by reading their reviews, checking out their profile pages or evaluating their Webs of Trust. Such "Web of Trust" helps the Epinions.com system to predict how helpful and believable a review will be to a user so that she can find what she is looking for easily. Also, Epinions.com system rewards the users with high trust. Thus, we consider the problem of providing consulting services to the users in Epinions.com who want to increase their social visibility. In particular, we aim to find those users who would get significant improvement in their social visibility by adding only a few new incoming neighbors (i.e., newly getting "trusted" by some users) since they are very likely to become the customers of such consulting services.

We consider a public data set from Epinion [4]. Nodes correspond to users and edges correspond to the trust relationship between users. Each user of the site can decide whether to "trust" other users. The Epinions OSN can be modeled as a directed unweighted graph with 75 879 nodes and 508 837 edges, For example, a directed edge from user a to b represents a trusting b. Since the Epinions OSN is unweighted, we set the visibility distance associated with all the edges as 1. and the visible threshold as $\tau = 2$.

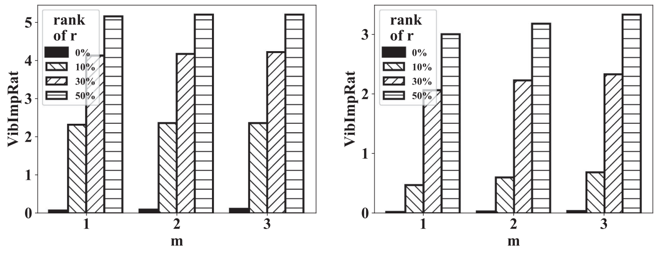
For each user, we compute her τ-visibility and rank all the users based on the their τ-visibility in descending order. We select users with different level of τ-visibility to play the role of a requester r. For each of these users, we apply the AdaExp algorithm to generate the approximate optimal set $\mathcal{M}_{\text{AdaExp}}$ under different new incoming neighbor quota m and different sets $\mathcal{U}'$ of available users, respectively. We vary m from 1 to 3. Also, we use two different settings of available users $\mathcal{U}'$. One is $\mathcal{U}' = \mathcal{U}$, where all users in the OSN are available users, i.e., willing to include r into their trust list when requested by r. This can happened when sufficient incentives are given. The second one is $\mathcal{U}' = \{v \in \mathcal{U} | \text{in} - \text{degree of } v \text{ ranked in the bottom } 50\%\}$, where available users are those whose in-degree is ranked in the bottom 50% in the OSN. In fact, this setting is realistic in real world since influential users with large in-degree are usually less willing to establish connection and become an incoming



(a) $|\mathcal{U}'| = |\mathcal{U}|$.

(b) $|\mathcal{U}'| = 0.5|\mathcal{U}|$.

Fig. 11. VibImpRat of requesters from different tiers in Epinions.



(a) $|\mathcal{U}'| = |\mathcal{U}|$.

(b) $|\mathcal{U}'| = 0.5|\mathcal{U}|$.

Fig. 12. VibImpRat of requesters from different tiers in DBLP.

neighbor of the requester, e.g., in the Twitter OSN, it is unusual for a Hollywood superstar to follow a requester who is just a nobody only because the requester requests him to do so.

To better quantify the effect of improvement, we define *visibility improvement ratio* as the ratio between the visibility improvement of requester r's τ-visibility and r's original τ-visibility, i.e.,

$$\mathrm{VibImpRat} \triangleq \frac{\Delta_{\boldsymbol{\theta}}(\mathcal{M}_{\mathrm{AdaExp}})}{\mathcal{V}(r,\tau)}.$$

Fig. 11 shows the log of improvement ratio. Fig. 11(a) shows that when $\mathcal{U}' = \mathcal{U}$, for the user with the largest τ-visibility, adding incoming neighbors only expands her τ-visible set marginally. For the users whose τ-visibility ranks behind 10%, adding incoming neighbors significantly increases the size of their visible set. Fig. 11(b) shows that when users whose in-degree is ranked in the bottom 50% are available, the visibility improvement reduces compared with Fig. 11(a). For users whose original visible set has a median size (i.e., its size ranked around 50%), adding three incoming neighbor can increase her visibility by around $e^{0.4} - 1 \approx 50\%$. This still implies a significant improvement on the visibility for users with a median size visible set. Again, the improvement ratio of the social visibility decreases in user's original visibility. In summary, for users whose original visible set is small, adding a few incoming neighbors can significantly improve their visibility. For users whose visibility is high, adding a few incoming neighbors can improve their visibility significantly only when users with a large in-degree are available.

B. Application to DBLP

We also consider the problem of providing consulting service to scholars in DBLP (introduced in Section V) to make their work more influential. We aim to answer which users in DBLP would have significant improvement in their visibility (i.e., the impact of scholars' researching work) by adding some new neighbors (i.e., establish academic cooperation with new partners), which ones would only have small increase.

We use the same modeling and parameter setting for DBLP in Section V. First, we compute her original τ-visible set for each user with the parameters $\tau = 0.7$ and rank them by the size of 0.7-visible set in descending order. Then we do the same as we do for Epinions, and the results of DBLP are shown in Fig. 12. When $\mathcal{U}' = \mathcal{U}$, Fig. 12(a) shows a similar observation to the Epinions: for users who do not have a very large original τ-visibility, our framework can achieve a significant improvement. Fig. 12(b) shows that in DBLP, when the selection of new incoming neighbors are restricted to available users $\mathcal{U}' = \{v \in \mathcal{U} | \mathrm{in} - \mathrm{degreeof}v\mathrm{rankedinthebottom50\%}\}$, the reduce of visibility improvement is relatively small compared with Epinions. Namely, for DBLP adding a small number of incoming neighbors can improve a user's social visibility significantly, unless her original social visibility is very high.

VII. Related Work

Application perspective. From an application perspective, our work is related to friendship recommendation [10]–[13], link prediction [14]–[16] and influence maximization [1], [17], [18]. Friendship recommendation [10]–[13] aim to building new connections (e.g., friendship, follower relationship and so on). The friendship recommendation framework suggests new connections based on some similarity metrics, with the assumption that users may want to be friends with users who have similar interest, etc. Different from the friendship recommendation framework, our framework is about adding incoming neighbors to a user to maximize her social visibility, where similarity metrics do not work. Link prediction [14]–[16] aims to predict future (or missing) links among users based on currently observed connections. This is usually achieved by inferring the link formation process. Different from link prediction, our social visibility maximization framework adds links but the added links aims to maximize the social visibility which may not be predicted by the link formation process). Also, our framework provides anonymity guarantee, while the link prediction does not. Influence Maximization (IM) [1], [17], [18] aims to find a small subset of nodes (seed nodes) in a social network that could maximize the spread of information. The influence maximization framework provides a "viral marketing"-like alternative to improve one's influence via information propagation. Our framework has three main differences from the IM framework: (1) our work is built on a social visibility model, which is different from the influence diffusion model; (2) our objective function is different from that of the IM problem; and (3) our work provides anonymity guarantee.

Methodology perspective. Our framework maximizes social visibility with anonymity guarantee via a novel combination of adaptive expansion strategy, the KMV sketch and the MAB algorithm. The KMV sketch technique has been widely used to estimate the cardinality of record size [19], [20]. Cohen *et al.*

[21] introduce a new estimator for the size of sets intersection based on the MinHash sketch technique. A general unbiased estimation over a sequence of set operations is proposed in [2]. Using KMV sketch, we develop a framework based on a trusted server to conduct anonymity preserving queries whose outcomes are KMV sketches, and prove the unbiasedness of the samples (generated by query outcomes). Multi-armed bandit (MAB) framework in best arm identification version [3] concerns the problem of estimating the arm with the highest reward from a set of arms. Each round one arm is pulled and a noise sample on the associated reward will be revealed. We develop a framework to estimate the best new incoming neighbor with anonymity guarantee, and we apply techniques in the MAB problems [3] to prove theoretical guarantees on our estimation framework. We incorporate this estimation framework into our adaptive expansion algorithm, and prove how the estimation accuracy influences the solution quality.

VIII. Conclusion

This paper presents a framework to maximize social visibility with anonymity guarantee. We develop a mathematical model to quantify social visibility and formulate the social visibility maximization (VisMAX-A) problem. Then we prove that this problem is NP-hard. And based on a query oracle, we develop a computationally efficient algorithm AdaExp to address the VisMAX-A problem with an approximation ratio slightly lower than $(1-1/e)$. By a novel combination of the KMV sketch technique and MAB online learning method, we propose a query oracle to efficiently estimate the best candidate in each iteration of AdaExp, while satisfying the query anonymity requirement. Finally, we conduct experiments on real-world social network datasets to validate the effectiveness and applicability of our framework.

References

[1] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proc. 9th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2003, pp. 137–146.

[2] K. Beyer, P. J. Haas, B. Reinwald, Y. Sismanis, and R. Gemulla, "On synopses for distinct-value estimation under multiset operations," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2007, pp. 199–210.

[3] J.-Y. Audibert, S. Bubeck, and R. Munos, "Best arm identification in multi-armed bandits," in *Proc. COLT*, 2010, pp. 41–53.

[4] J. Kunegis, "KONECT - The koblenz network collection," in *Proc. Int. Conf. World Wide Web Companion*, 2013, pp. 1343–1350.

[5] J. Leskovec and A. Krevl, "SNAP datasets: Stanford large network dataset collection," Jun. 2014. [Online]. Available: http://snap.stanford.edu/data

[6] W. Nawaz, K.-U. Khan, Y.-K. Lee, and S. Lee, "Intra graph clustering using collaborative similarity measure," *Distributed Parallel Databases*, vol. 33, no. 4, pp. 583–603, 2015.

[7] S. P. Borgatti, "Centrality and network flow," *Social Netw.*, vol. 27, no. 1, pp. 55–71, 2005.

[8] L. C. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, vol. 40, no. 1, pp. 35–41, Mar. 1977.

[9] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: Bringing order to the web," Stanford InfoLab, Tech. Rep. 1999-66, Nov. 1999.

[10] P. Resnick and H. R. Varian, "Recommender systems," *Commun. ACM*, vol. 40, no. 3, pp. 56–59, 1997.

[11] H. Ma, "On measuring social friend interest similarities in recommender systems," in *Proc. 37th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2014, pp. 465–474.

[12] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in *Proc. 4th ACM Int. Conf. Web Search Data Mining*, 2011, pp. 287–296.

[13] X. Xie, "Potential friend recommendation in online social network," in *Proc. IEEE/ACM Int. Conf. Green Comput. Commun.*, 2010, pp. 831–835.

[14] L. Lü and T. Zhou, "Link prediction in complex networks: A survey," *Physica A, Stat. Mechanics Appl.*, vol. 390, no. 6, pp. 1150–1170, Mar. 2011.

[15] V. Martínez, F. Berzal, and J.-C. Cubero, "A survey of link prediction in complex networks," *ACM Comput. Surv.*, vol. 49, no. 4, 2017, Art. no. 69.

[16] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 58, no. 7, pp. 1019–1031, 2007.

[17] P. Domingos and M. Richardson, "Mining the network value of customers," in *Proc. 7th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2001, pp. 57–66.

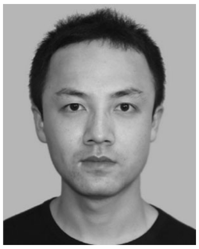
[18] M. Richardson and P. Domingos, "Mining knowledge-sharing sites for viral marketing," in *Proc. 8th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2002, pp. 61–70.

[19] G. Cormode *et al.*, "Synopses for massive data: Samples, histograms, wavelets, sketches," *Found. Trends Databases*, vol. 4, no. 1/3, pp. 1–294, 2011.

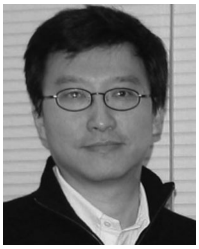
[20] X. Wang, Y. Zhang, W. Zhang, X. Lin, and W. Wang, "Selectivity estimation on streaming spatio-textual data using local correlations," *Proc. VLDB Endowment*, vol. 8, pp. 101–112, 2014.

[21] R. Cohen, L. Katzir, and A. Yehezkel, "A minimal variance estimator for the cardinality of Big Data set intersection," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2017, pp. 95–103.

Shiyuan Zheng received the B.E. degree in software engineering from Shandong University, Jinan, China, in 2018. She is currently working toward the Ph.D. degree with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong, under the supervision of Prof. John C.S. Lui. Her current research interests include social network analysis, game theory, and online learning algorithms.



Hong Xie (Member, IEEE) received the B.Eng. degree from the School of Computer Science and Technology, The University of Science and Technology of China, Hefei, China, in 2010, and and the Ph.D. degree from the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong, in 2015, proudly under the supervision of Prof. John C.S. Lui. He is currently a Research Professor with the College of Computer Science, Chongqing University, Chongqing, China. He was a Postdoctoral Research Fellow with The Chinese University of Hong Kong, Hong Kong, and National University of Singapore, Singapore.



John C.S. Lui (Fellow, IEEE) received the Ph.D degree in computer science from the University of California, Los Angeles, Los Angeles, CA, USA. From 2005 to 2011, he was a Chairman with CSE Department. He is currently the Choh-Ming Li Chair Professor with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong. His current research interests include communication networks, system security (such as, cloud security and mobile security), network economics, network sciences, large-scale distributed systems, and performance evaluation theory. He is an Elected Member of IFIP WG 7.3 and a Croucher Senior Research Fellow. He was the recipient of the various departmental teaching awards and the CUHK Vice-Chancellors Exemplary Teaching Award. He was also the co-recipient of the IFIP WG 7.3 Performance 2005 and IEEE IFIP NOMS 2006 best student paper awards. He serves in the Editorial Board of the IEEE/ACM Transactions on Networking, IEEE Transactions on Computers, IEEE Transactions on Parallel and Distributed Systems, *Journal of Performance Evaluation*, and *International Journal of Network Security*.