Adaptive Diffusion Of Sensitive Information In Online Social Networks

Xudong Wu, Luoyi Fu, Huan Long, Dali Yang, Yucheng Lu, Xinbing Wang and Guihai Chen
Shanghai Jiao Tong University
{xudongwu, yiluofu, longhuan, yangdali, eugene_lu, xwang8}@sjtu.edu.cn, gchen@cs.sjtu.edu.cn

Abstract—The cascading of sensitive information such as private contents and rumors is a severe issue in online social networks. One approach for limiting the cascading of sensitive information is constraining the diffusion among social network users. However, the diffusion constraining measures limit the diffusion of non-sensitive information diffusion as well, resulting in the bad user experiences. To tackle this issue, in this paper, we study the problem of how to minimize the sensitive information diffusion while preserve the diffusion of non-sensitive information, and formulate it as a constrained minimization problem where we characterize the intention of preserving non-sensitive information diffusion as the constraint. We study the problem of interest over the fully-known network with known diffusion abilities of all users and the semi-known network where diffusion abilities of partial users remain unknown in advance. By modeling the sensitive information diffusion size as the reward of a bandit, we utilize the bandit framework to jointly design the solutions with polynomial complexity in the both scenarios. Moreover, the unknown diffusion abilities over the semi-known network induce it difficult to quantify the information diffusion size in algorithm design. For this issue, we propose to learn the unknown diffusion abilities from the diffusion process in real time and then adaptively conduct the diffusion constraining measures based on the learned diffusion abilities, relying on the bandit framework. Extensive experiments on real and synthetic datasets demonstrate that our solutions can effectively constrain the sensitive information diffusion, and enjoy a 40% less diffusion loss of non-sensitive information comparing with four baseline algorithms.

Index Terms—Information diffusion, Online social Networks, Constraining sensitive Information diffusion, Multi-arm bandit

1 Introduction

The prevalence of online social networks such as Facebook, Twitter and Wechat facilitates the information diffusion among users, and thus enables the efficient promotion of positive informations, e.g., products, news, innovations [1]- [8]. Although such efficient diffusion can easily lead to large-scale diffusion called information cascading, the unconstrained cascading behavior could meanwhile cause the sensitive information to be incautiously diffused over the network [9]- [20]. Here the sensitive information refers to any kind of information that needs to be prohibited from cascading such as rumors, personal contents, and trade secrets. The cascading of such sensitive information may cause the risk of leaking users' privacies or arising panics among publics [9]- [20]. With this concern, several social network medias (e.g., Facebook, Twitter) have claimed authorities to block accounts of users and delete some posts or tweets when they violate relevant rules about privacies or securities [9] [21] [22]. Thus network managers are able to take measures to prohibit the cascading of sensitive information.

The existing attempts that share the closest correlation with prohibiting sensitive information diffusion belong to the rumor influence minimization [9]- [20], whose current strategies can mainly be classified into two aspects. The first is diffusing the truths over network to counteract rumors [12]- [14]. However, diffusing truths is only suitable for constraining the rumors, while is not suitable for constraining the diffusion of the other kinds of sensitive informations, including personal informations, trade secrets, and etc. The second is temporarily blocking a number of users with high diffusion abilities [9] [10] [15] [16] or blocking a number of social links among users [17]- [20] in hope of minimizing the diffusion of a rumor. Although such strategy is effective for preventing rumors about some significant events like earthquakes, terrorist attacks and political elections, it is unrealistic for network managers to adopt this strategy on constraining the diffusion of sensitive informations with various contents that widely exist in our daily lives. If network managers take such measure, it is required to block a much larger size of users or links. Then two critical problems arise. Firstly, blocking too many users or social links will degrade user experiences and may arouse complaints for the right violation. Secondly, blocking users or social links for restraining rumors also brings the loss of the diffusion of positive informations, say information loss, which is not beneficial to the viral marketers that utilize information cascading to promote products [1]- [6], [23] [24].

Regarding the limitations of existing solutions, in this paper, we take the first look into limiting the cascading of sensitive informations while preserving the diffusion of non-sensitive ones to lower the information loss. Considering the randomness of the users accepting informations diffused from their social neighbors, we adopt the widely used random diffusion model that each user diffuses information to his social neighbor successfully with a diffusion probability via the social link between them. Then our technical objective is adjusting the diffusion probabilities via social links to minimize the diffusion size of sensitive informations, under the constraint of keeping the value of the sum of diffusion probabilities via all social links. Corresponding to the reality, we consider a case where

some advertisements in viral marketing and some rumors simultaneously diffuse over an online social network. In this case, decreasing diffusion probabilities models the measures such as deleting partial posts or fanpages reposted by users [25] [26], while the measures for increasing diffusion probabilities include sticking and adding pushes or deliveries of the posts reposted by given users [16] [27]. Then, if network managers decrease the diffusion probability from a user holding rumors, the advertisements diffused from the user will inevitably be constrained as well. Thus, for lowering the diffusion loss of the advertisements and preserving the global diffusion ability of the whole network on diffusing non-sensitive informations, a natural approach is increasing the diffusion probabilities from one or more other users which hold the advertisements.

We study the problem of interest on both fully-known and semi-known networks which are the two main scenarios considered in current studies on information diffusion [1]- [16]. Over the fully-known network, we assume network managers know the diffusion abilities of all users. The examples for the fully-known network lie on the social networks for enterprises (e.g., Skype) or special interest groups (SIGs) (e.g., Douban[1]). As the full topology of a local social network, which consists of the staff of a same enterprise or the members in a same SIG, is available to network managers, it is feasible to quantify the diffusion abilities of all users. On the contrast, the semi-known network here refers to the case that diffusion abilities of partial users remain unknown in advance. For example, the data of Facebook was reported to be utilized to influence the 2016 election in the US, which then led to a severe trust crisis for Facebook. Thus, due to the privacy concern and potential side effect, even for network managers, it is difficult to obtain the full topology of some global large scale social networks like Facebook, Wechat. Unless the full network topology is known, we cannot evaluate the diffusion abilities of all users.

Over the fully-known network, although we can determine the diffusion probability variations via social links through solving a constrained minimization problem, the huge size of social links in current large scale networks leads to the high complexity of the problem. Moreover, the unknown diffusion abilities of partial users over the semi-known network induce it infeasible to directly solve the constrained minimization problem for minimizing the diffusion size of sensitive informations.

To tackle the above challenges, we utilize the constrained combinatorial multi-arm bandit framework to jointly design our solutions over the fully-known and semi-known networks, where we take the diffusion size of sensitive informations as the reward of a bandit and model the probability variations as the arms in bandit. With this mapping, we determine the probability variations through a constrained arms picking process with the aim of minimizing the obtained rewards. Through incorporating the constraint of diffusion probability variations into the construction of the arms of bandit, we relax the problem of interest into an unconstrained minimization problem when determining the diffusion probability variations based on the arms. This enables us to determine the probability variations via social links with high efficiency. Furthermore, for coping with the unknown diffusion abilities over the semi-known network, we propose to iteratively learn the unknown diffusion abilities through learning the reward distributions of the arms based on the rewards obtained from previously picked arms, and then determine the diffusion probability variations based on the learned reward distributions of arms.

Our main contributions are summarized as follows:

(1) We take the first look into minimizing the diffusion size of sensitive informations while preserving the diffusion of non-sensitive ones. We formulate the problem of interest into a constrained minimization problem where we characterize the intention of preserving non-sensitive information diffusions as the constraint.

(2) We propose an efficient bandit based framework to jointly explore the solutions over the fully-known and semi-known networks within polynomial running time. Moreover, we design the distributed implementation scheme of our solutions for the further improvement of time efficiency.

(3) We further extend our bandit based solution into a "learning- determining" manner for addressing the challenge of unknown diffusion abilities in semi-known networks. We theoretically prove that the regret bound of our solution is sub-linear to the diffusion time, indicating that the probability variations returned by our solution approximates to the optimal one with the increase of diffusion time.

(4) We perform extensive experiments on both real and synthetic social network datasets. The results demonstrate that the proposed algorithms can effectively constrain the diffusion of sensitive informations, and more importantly, enjoy a superiority over four baselines in terms of 40% less information diffusion loss.

The rest of this paper is organized as follows. We formulate the problem in Section 2. Then we present the solution in fully-known network in Section 3 and the solution in semi-known network in Section 4. We report the experimental results in Section 5. At last, we review the related works in Section 6 and conclude the paper in Section 7.

2 Preliminaries

2.1 Network Model

We model the online social network as a directed graph $G = (V, E)$, where each node in $V(|V| = n)$ represents a user in the network and each directed edge in $E(|E| = m)$ represents a social link between a pair of users. We say the node v is a neighbor of node u if there is an edge in E with the source node being u and the destination node being v. Each node is classified as either a **sensitive** node or a **non-sensitive** one. In correspondence to social networks, sensitive nodes refer to the individuals who hold sensitive informations (e.g., rumors or private informations of users). Moreover, each edge $i \in E$ has a weight w_i representing the probability that the source it connects can successfully diffuse information to the destination node via it. That is, we assume that the diffusion results via each edge are independent, and the diffusion result via an edge i follows the Bernoulli distribution $\mathcal{B}(w_i)$. We assume that the weight on each edge follows a uniform distribution of $U(0, w_{max})(w_{max} \in (0, 1))$.

1. https://www.douban.com/

As mentioned earlier, we study the adaptive diffusion of sensitive informations over both fully-known and semi-known networks. In the sequel, we respectively present the definitions of the two scenarios. Specifically, in our setting, the semi-known network consists of the *informed* nodes and the *uninformed* nodes, and the nodes in the fully-known network are all *informed*.

Definition 1. ***(Informed and uninformed nodes.)*** *We define a node as an informed node if the network manager knows all the neighbor nodes of the node in network, and define the node as an uninformed node otherwise.*

Definition 2. ***(Fully-known network.)*** *In the fully-known network, we assume the network manager knows the neighbor nodes of all the nodes in network. In other words, all the nodes in the fully-known network are informed.*

On the other hand, in the semi-known network, we assume the network managers just have the partial knowledge of the topology of a given social network.

Definition 3. ***(Semi-known network.)*** *The semi-known network is a social network where uninformed nodes coexist along with informed nodes.*

2.2 Diffusion Model

Over both the fully-known and semi-known networks, the sensitive informations can only be diffused from the sensitive nodes. We assume there are T time rounds. A non-sensitive node will turn to sensitive once it receives sensitive informations, and from then on until the end of the T rounds, as long as the sensitive informations it holds are not out of date, it will have chance to diffuse sensitive informations to its neighbors. The t-th round refers to the time from time stamp t to $t+1$. We use V^t to denote the set of the sensitive nodes at time stamp t. We denote the edges whose source nodes are in sensitive as the target edges.

Definition 4. ***(Target edge.)*** *If the source node of an edge is in sensitive, we denote the edge as the target edge. We use $\mathfrak{E}^t$ to denote the set all target edges at the beginning of the t-th round.*

We study the problem of adaptively adjusting diffusion probabilities via edges at the beginning of each round, for taking measures in real time to minimize the sensitive information diffusion. For this end, we define the duration of each round as the time for two-hop diffusion. That is, the informations diffuse two hops during a round. In particular, during each round, the source node of an edge i, say S_i, first diffuses informations I_i to the destination node D_i successfully with probability w_i, and D_i then diffuses the received informations to its neighbors. Moreover, in the first hop in each round, we define each sensitive source node having a single chance to diffuse sensitive informations to each of its neighbors. If a node receives sensitive informations during the first hop, it further has one chance to diffuse the received sensitive informations to its own neighbors in the second hop. Notably, we define that each sensitive node has the chance to diffuse sensitive informations to their neighbors in each round as long as the sensitive informations it holds are not out of date. Our insight for such definition comes from the real behavior that users of social medias are probably to repeatedly review the Moments or Tweets, which are posted by their friends several days or even several months ago.

At the same time, the diffusion of non-sensitive informations also occurs in the network, and is in a same manner with the diffusion of sensitive informations. A node will simultaneously hold and diffuse non-sensitive and sensitive informations, if the node receives the both kinds of informations during diffusion.

With the above network and diffusion models, Lemma 1 scales the size of sensitive nodes after T rounds, if there is no measure on constraining sensitive information diffusion.

Lemma 1. *Given the sensitive node set V^1 and the target edge set $\mathfrak{E}^1$ at the beginning of the 1-st round, the expected size of sensitive nodes until the end of the T rounds is upper bounded by $M_T = |V^1| + |\mathfrak{E}^1| w_{max} 2T + \frac{n-|V^1|}{4} w_{max}(2T+1)T$.*

The proof for the Lemma 1 is in Appendix A (in supplemental material). In Lemma 1, we assume the social network follows the power-law degree distribution, which is a popular property of social network structure [28] [29], and $w_{max} \ll 1$.

2.3 Problem Formulation

Lemma 1 suggests that there is a potential large scale cascading of sensitive informations over social networks if the network managers do not take any measure, and further motivates the study in this paper.

The problem of interest is adaptively adjusting the diffusing probabilities via target edges in $\mathfrak{E}^t$ $(1 \leq t \leq T)$ at the beginning of each round, for minimizing the diffusion size of sensitive informations. In the t-th round, we denote the destination nodes of the target edges in $\mathfrak{E}^t$ as *target nodes*. The $|\mathfrak{E}^t|$-dimensional vector $\overrightarrow{\beta_0^t}$ denotes the original diffusion probabilities via the target edges, with $\overrightarrow{\beta_0^t}(i)$ representing the original diffusion probability via edge i. With the diffusion model given above, in the t-th round, the expected diffusion size of sensitive informations from the destination node of a target edge i can be quantified as $\overrightarrow{\beta_0^t}(i) \cdot \sum_{j \in E(S,i)} w_j$. Here, $E(i)$ is the set of edges whose sources are the destination node of edge i, and $|E(i)|$ equals d_i which represents the out-degree of the destination node of edge i. Then, we let the $|\mathfrak{E}^t|$-dimensional vector $\overrightarrow{D}^t$ denote the diffusion abilities of the $|\mathfrak{E}^t|$ target nodes, where $\overrightarrow{D}^t(i) = \sum_{j \in E(S,i)} w_j$ quantifies the diffusion ability of the destination node of edge i in a round.

Furthermore, we adopt $\overrightarrow{\Delta\beta}^t$ to represent the vector of the probability variations on the target edges in the t-th round, with $\overrightarrow{\Delta\beta}^t(i)$ representing the diffusion probability variation on edge i. Here, $\overrightarrow{\Delta\beta}^t(i) < 0$ means constraining the diffusion via edge i, while $\overrightarrow{\Delta\beta}^t(i) > 0$ means promoting the diffusion via edge i. Then, with the aim of minimizing the diffusion size of sensitive informations, our technical objective is exploring an optimal variation vector $\overrightarrow{\Delta\beta}^{t*}$ in each round to minimize the value of $\sum_{i=1}^{T} \overrightarrow{D^t} \cdot (\overrightarrow{\beta_0^t} + \overrightarrow{\Delta\beta}^{t*})$. In addition, as we illustrated before, we intend to minimize the diffusion size of sensitive information while preserving the diffusion of non-sensitive information in order to reduce the information loss. Such intention is technically conducted by maintaining the sum of the diffusion probabilities via target edges, i.e., $\sum_{i \in \mathfrak{E}^t} (\overrightarrow{\beta_0^t}(i) + \overrightarrow{\Delta\beta^t}(i)) = \sum_{i \in \mathfrak{E}^t} \overrightarrow{\beta_0^t}(i)$ and

$\sum_{i \in \mathfrak{E}^t} \overrightarrow{\Delta\beta^t}(i) = 0$. By this, we formally give the adaptive diffusion problem as follows.

Problem statement. During the T diffusion rounds, we try to minimize the value of

$$\sum_{t=1}^{T} \overrightarrow{D}^t \cdot (\overrightarrow{\beta_0}^t + \overrightarrow{\Delta\beta}^t) \tag{1}$$

by determining the value of $\overrightarrow{\Delta\beta}^t$ $(1 \leq t \leq T)$ in each round. The constraint for the variation vector $\overrightarrow{\Delta\beta}^t$ $(1 \leq t \leq T)$ is $\sum_{i \in \mathfrak{E}^t} \overrightarrow{\Delta\beta}^t(i) = 0\,(1 \leq t \leq T)$.

Next, we present the solution to the adaptive diffusion problem over fully-known network in Section 3, and will extend our study to the semi-known network in Section 4.

3 Adaptive Diffusion in Fully-known Networks

In the fully-known network, the problem of interest in Eqn. (1) is a classical Linear Programming (LP) problem. However, the classical solutions (e.g., Simplex Algorithm [30], Ellipsoid Algorithm [31] and Karmarkar Algorithm [32]) for the LP problem cannot be efficiently applied to problem (1) in adaptive diffusion, due to the high dimension of the variable vector $\overrightarrow{\Delta\beta}^t$. With the dimension of $\overrightarrow{\Delta\beta}^t$ being $|\mathfrak{E}^t|$, the complexities for the Simplex, Ellipsoid and Karmarkar algorithms are respectively scaled as $O(2^{|\mathfrak{E}^t|})$, $O(|\mathfrak{E}^t|^6)$, $O(|\mathfrak{E}^t|^{3.5})$ [30] [31] [32].

Solution overview. For the issue of the high complexity of classical solutions, we seek the solution for the adaptive diffusion based on the *bandit* framework. In particular, we model the probability variation vector $\overrightarrow{\Delta\beta}^t$ in each round as the *arm* of a bandit, and model the diffusion size of sensitive information as the *reward* obtained from the bandit after picking such arm. By this, we explore the efficient solution for the adaptive diffusion through exploring efficient arm picking algorithm under the objective of minimizing obtained rewards. In addition, we adopt the bandit framework here also under the consideration that the bandit model will enable us to deal with the partial unknown diffusion abilities in the semi-known network (in Section 4). That is, we utilize the bandit framework to jointly design the solutions in both the fully-known and semi-known networks.

In the following, in Section 3.1, we will first give a brief introduction to the bandit framework. Then we will present the mappings between the components in bandit and the key elements in the adaptive diffusion problem, along with our ideas of the efficient solution for adaptive diffusion. With this mapping, in Section 3.2, we will give our algorithm for efficiently determining the probability variation vector $\overrightarrow{\Delta\beta}^t$ in each round based on the bandit framework.

3.1 Mapping Adaptive Diffusion in Fully-known Network into Bandit

3.1.1 Introduction of the Bandit Framework

We seek the solutions for the adaptive diffusion problem by mapping it into a **Constrained Combinatorial Multi-Arm Bandit (CCMAB)** framework, which is a variation of the Multi-Arm Bandit (MAB) for coping with the combinatorial optimization problem in online manner. In the general formation of the MAB, there are multiple independent arms that provide stochastic rewards with certain distributions. The objective of the MAB-based optimizing problems is, through sequentially picking an arm in each round based on the reward distributions of arms, maximizing the sum of the rewards obtained from all picked arms. Furthermore, the major feature of the CCMAB framework that we adopt in this paper lies on the dependencies between arms, different from the independences in general MAB model. In each round, for maximizing the obtained rewards, the CCMAB picks a *super-arm* which consists of a set of *base-arms* under given constraints. Such feature enables the CCMAB to be adopted in the online combinatorial optimization problems.

Corresponding to the adaptive diffusion problem studied in this paper, we take the sensitive information diffusion size whose expectation is quantified by $\sum_{t=1}^{T} \overrightarrow{D}^t \cdot (\overrightarrow{\beta_0}^t + \overrightarrow{\Delta\beta}^t)$ as the reward of bandit, and take the variation vector $\overrightarrow{\Delta\beta}^t$ as the *super-arm* that we need to determine in each round. Then our objective becomes to determining the picked super-arm in each round to minimize the overall rewards obtained during the T rounds. Notably, since our objective is minimizing the sensitive information diffusion, different from the common objective in bandit that maximizing rewards [33] [34], we aim at *minimizing* the rewards.

3.1.2 Mapping Adaptive Diffusion into CCMAB

To be more precise, we present below the mappings between the components in the adaptive diffusion over fully-known network and the key elements (i.e., base-arm, super-arm, and rewards) in CCMAB as below.

Base-arm: In the CCMAB model, the base-arms are the constitutes of the super-arm that we need to determine in each round. As each super-arm here is the vector $\overrightarrow{\Delta\beta}^t$ which specifies the variation of diffusion probability over each target edge (Definition 4) and satisfies the constraint $\sum_{i \in \mathfrak{E}^t} \overrightarrow{\Delta\beta}^t(i) = 0$, we set the base-arms as the $|\mathfrak{E}^t|$-dimensional vectors with *pair-wise* non-zero elements. Specifically, each base-arm is set to only have two non-zero elements, with the negative probability variation on one target edge and the positive probability variation of the same amount on the other target edge. For example, the vectors $\overrightarrow{\beta_1}$ and $\overrightarrow{\beta_2}$ in Fig. 1 are two base-arms with the dimension being 4. Thus, each base-arm satisfies the constraint $\sum_{i \in \mathfrak{E}^t} \overrightarrow{\Delta\beta}^t(i) = 0$, and as a result, the super-arm which is the sum of a set of base-arms must also satisfy the constraint. The definition of the base-arm is as follows.

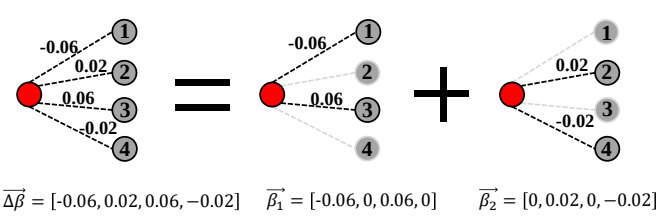


Fig. 1. A toy example of combining base-arms $\vec{\beta}_1$ and $\vec{\beta}_2$ into a super-arm $\vec{\Delta\beta}$. Here, $\vec{\Delta\beta} = \vec{\beta}_1 + \vec{\beta}_2$.

Definition 5. ***(Base-arm.)*** *We use vector $\overrightarrow{\beta_r}$ to denote a base-arm, and each base-arm has the following three characteristics:*

- $\overrightarrow{\beta_r}$ *only has two non-zero elements;*
- $\sum_{j=1}^{|\mathfrak{E}^t|} \overrightarrow{\beta_r}(j) = 0$*;*
- $\overrightarrow{\beta_r}(j) \in \{n\Delta_p | n \in \mathbf{Z}, -w_j \leq n\Delta_p \leq 1 - w_j\}$, $1 \leq j \leq |\mathfrak{E}^t|$, $\mathbf{Z}$ *is the set of integers.*

Here, $\overrightarrow{\beta_r}(j)$ *represents the probability variation on edge* j *determined by the base-arm* $\overrightarrow{\beta_r}$*,* w_j *is the original diffusion probability via edge* j*, and* $\Delta_p (\Delta_p > 0, \frac{w_j}{\Delta_p} = O(1))$ *quantifies the granularity when adjusting diffusion probabilities.*

For each base-arm $\overrightarrow{\beta_r}$, we define a vector $\overrightarrow{I_r}$, where $\overrightarrow{I_r}(j) = \begin{cases} 0, if\, \overrightarrow{\beta_r}(j) = 0 \\ 1, if\, \overrightarrow{\beta_r}(j) \neq 0 \end{cases}$, that will be used later. By the third characteristic above, we constrain the optional probability variations on each target edge into the finite set $\{n\Delta_p | n \in \mathbf{Z}, -w_j \leq n\Delta_p \leq 1 - w_j\}$ for having a reasonable size of base-arms. Thus, the size of the base-arms in the t-th round is $\Theta\left(|\mathfrak{E}^t|^2\right)$, since there are $\binom{|\mathfrak{E}^t|}{2}$ pair-wise combinations of the target edges in $\mathfrak{E}^t$ and each combination has a $O(1)$ size of optional probability variations. In addition, the target edge set $\mathfrak{E}^t$, as well as the size and dimension of base-arms, change over time during diffusion. This is because that once a non-sensitive node becomes sensitive, we will add the edges connecting with such source node into the set $\mathfrak{E}^t$, while once the sensitive informations a sensitive node holds are outdated, we will remove the edges connecting with such source node from the set $\mathfrak{E}^t$.

Super-arm: The super-arm in the t-th round refers to the diffusion probability variation vector $\overrightarrow{\Delta\beta^t}$, which is the sum of a set of base-arm vectors and specifies the variation of diffusion probability over each target edge. A toy example of the combination of two base-arms is presented in Fig. 1. Moreover, since the super-arm which consists of a set of base-arms given in Definition 5 must satisfy the constraint $\sum_{i \in \mathfrak{E}^t} \overrightarrow{\Delta\beta^t}(i) = 0$, we relax the problem of interest in Eqn. (1) into an unconstrained optimization problem when determining the value of the super-arm from base-arms.

Reward: For selecting a set of base-arms to consistitute the picked super-arm under the objective of minimizing the reward of bandit, we elaborate below how to evaluate the rewards of super-arm and base-arm in adaptive diffusion.

The **reward of super-arm** during each round refers to the diffusion size of sensitive informations from the destination nodes of target edges. Specifically, we quantify the mean reward of the super-arm in the t-th round by the objective function $\overrightarrow{D^t} \cdot (\overrightarrow{\beta_0^t} + \overrightarrow{\Delta\beta^t})$. Then, considering that the diffusion size of sensitive informations can be quantified as the sum of a number of diffusion results on edges, which are independent and follow the Bernoulli distributions, we assume the diffusion size follows the Gaussian distribution $\mathcal{N}(\overrightarrow{D^t} \cdot (\overrightarrow{\beta_0^t} + \overrightarrow{\Delta\beta^t}), \sigma)$. Here, σ specifies the variance of the reward distribution. In reality, network managers can observe the actual diffusion size of sensitive informations, as well as the reward of the super-arm, from the numbers of the visits, shares, likes or replies of the sensitive informations.

Reward of base-arm. Upon giving the reward of the super-arm above, we move to the rewards of base-arms.

We first introduce the definition of *diffusion feedback*. For an edge a, we count the diffusion size of informations I_a from the destination node D_a as the diffusion feedback of edge a in a round. In this paper, we assume that the diffusion feedbacks of different edges can be independently observed by network managers. The insight for such assumption is that the source users of the shares are available in some social medias (e.g., Facebook and Weibo). Specially, if multiple target edges share a same destination node, we can also count the diffusion feedbacks of different target edges respectively under the assumption of independent feedback observation. Furthermore, taking a base-arm whose two non-zero elements are on the target edges a and b as an example, we take the sum of the diffusion feedbacks of edges a and b as the diffusion feedback of the base-arm. Such diffusion feedbacks can be observed from the numbers of the visits, shares, likes or replies of the sensitive informations diffused from D_a and D_b.

In addition, for a base-arm $\overrightarrow{\beta_r}$, we extract the constant $\overrightarrow{D^t} \cdot (\overrightarrow{\beta_0^t} \odot \overrightarrow{I_r})$ from $\overrightarrow{D^t} \cdot \overrightarrow{\beta_0^t}$. Then, the **reward of a base-arm** $\overrightarrow{\beta_r}$ is quantified by the diffusion feedback of $\overrightarrow{\beta_r}$ minus the constant $\overrightarrow{D^t} \cdot (\overrightarrow{\beta_0^t} \odot \overrightarrow{I_r})$, with the mean reward being $\overrightarrow{D^t} \cdot \overrightarrow{\beta_r}$.

Relation between the rewards of super-arm and base-arms. In this paper, we let the reward of each base-arm incorporated into the super-arm can be independently observed by network managers. To this end, we give below the rule for judging whether the combinations of a set of base-arms are *valid*.

Definition 6. ***(Valid rule.)***

Rule: *Combination of* $\overrightarrow{\beta_1}, \overrightarrow{\beta_2}, \ldots, \overrightarrow{\beta_r}$ *is valid if* $\forall x, y (x \neq y), 1 \leq x, y \leq r, \overrightarrow{\beta_x}(i) \cdot \overrightarrow{\beta_y}(i) = 0\, (\forall i, 1 \leq i \leq |\mathfrak{E}^t|)$.

The valid rule above controls that, in each round, the non-zero probability variation on each target edge can only appear in one base-arm, and enables the rewards of base-arms to be independently evaluated and observed. Such independent reward evaluation facilitates efficiently determining which base-arms to constitute the super-arm, since we do not need to consider the overlap among the rewards of base-arms. Also, as we will present in Section 4, the independent reward observation will enable us to cope with the partial unknown diffusion abilities in semi-known networks. Under the valid rule, a super-arm can only be combined by a set of valid base-arms. Then, the reward of the super-arm in each round is equal to the sum of the rewards of all picked base-arms adding a constant $\overrightarrow{D^t} \cdot \overrightarrow{\beta_0^t}$.

Summary of the mappings. Upon mapping the adaptive diffusion problem (Eqn. (1)) into the CCMAB model, we transfer determining the variation vector $\overrightarrow{\Delta\beta^t}$ for minimizing the diffusion size of sensitive information in the T rounds to determining each super-arm $\overrightarrow{\Delta\beta^t}$, which consists of a set of base-arms, for minimizing the rewards obtained in the T rounds. With this transformation, we give our algorithm for determining the probability variation vector as follows.

3.2 Algorithm in Fully-known Network

3.2.1 Algorithm Design

Based on the above mappings, the aim of our algorithm for **A**daptive **D**iffusion in **F**ully-known **N**etwork, named **ADFN**, is selecting a combination of base-arms with the minimum sum of mean rewards. We present the

pseudo code of **ADFN** in Algorithm 1, where we use the $combination$ to denote the set of the selected base-arms. In **ADFN**, we iteratively select the base-arm v with the minimum mean reward. Then, if the base-arm v is not conflicted with all the base-arms in the current $combination$ and has negative mean reward, we add it into the $combination$. Furthermore, the super-arm in each round is determined by the sum of the probability variation vectors represented by the base-arms in the $combination$.

Now, we present the complexity of the algorithm **ADFN**. In each round, **ADFN** needs to traverse all the base-arms for obtaining a valid $combination$ with the minimum reward, and costs a complexity of $O(|\mathfrak{E}^t|^2)$. Thus, **ADFN** costs a polynomial time complexity in terms of the network size, when determining the super-arm in each round. Moreover, since the size of base-arms and the complexity of **ADFN** is the twice order of network size, we then propose the distributed implementation scheme of **ADFN** for the further improvement of time efficiency.

3.2.2 Design for Distributed Implementation

The complexity of **ADFN** is mainly on the traverse of the base-arms. By this, we propose to distributedly implement **ADFN** by storing the base-arms in a distributed manner, and then conduct **ADFN** over the distributedly stored base-arms. The idea of the distributed implementation is presented as follows.

For distributedly implementing **ADFN**, we chop the base-arms into blocks and traverse the blocks in a parallel manner by multitasking. Technically, we store the base-arms into N storage units. For each given base-arm, we store the IDs of all the other base-arms that can be validly combined with it along with its ID. We call the main procedures in **ADFN** as **master**, and the distributed storage units as **slaves**. Each slave keeps a local lookup table recording the mean reward of each local base-arm. With such distributed storage units, **ADFN** is then distributedly implemented as below.

In each round, every slave first selects a valid local $combination$ with the minimum reward from all the local base-arms, and returns the local $combination$ to the master. The master then emerges all the base-arms in the N local $combinations$ from slaves into the $ActionPool$. Upon generating the $ActionPool$, the master determines the value of $\overrightarrow{\Delta\beta}^t$ as Algorithm 1. We algorithmically present the procedures for the master and the slaves in Appendix C in the supplemental material.

Algorithm 1: ADFN in the t-th diffusion round

Input: All the base-arms in the t-th round
Output: Variation Probability vector $\overrightarrow{\Delta\beta}^t$
$ActionPool \leftarrow$ All the base-arms, $combination \leftarrow \varnothing$;
while $ActionPool \neq \varnothing$ **do**
  $v =$MIN($ActionPool$);
    `/* MIN(S) returns the item with the smallest reward in set S. */`;
  if $\overrightarrow{D}^t \cdot \overrightarrow{\beta}_v > 0$ **then**
    End While ;
  end
  $ActionPool \leftarrow ActionPool \backslash \{v\}$;
  if *VALID(combination, v)* **then**
    $combination \leftarrow combination \cup \{v\}$;
  end
end
for $\overrightarrow{\beta_i} \in combination$ **do**
  $\overrightarrow{\Delta\beta}^t = \overrightarrow{\Delta\beta}^t + \overrightarrow{\beta_i}$;
end
return $\overrightarrow{\Delta\beta}^t$

4 Adaptive Diffusion in Semi-known Networks

Now, we proceed to explore the solutions for the adaptive diffusion over the semi-known network where the diffusion abilities of partial users remain unknown in advance. Over the semi-known network, besides the complexity issue, another major difficulty for solving the adaptive diffusion problem comes from the lack of exact diffusion abilities of partial target nodes. That is, if without an exact diffusion ability vector $\overrightarrow{D}^t$, we are unable to directly solve the optimization problem given in Eqn. (1).

4.1 Solution Overview

In Section 3, we have mapped the probability variation vector $\overrightarrow{\Delta\beta}^t$ into the super-arm in CCMAB, and have associated the diffusion abilities with the rewards of base-arms. With this mapping, for coping with with the unknown diffusion abilities, we propose to iteratively *learn* the unknown diffusion abilities via learning the reward distributions of base-arms from the rewards obtained in previous rounds, and then *determine* the super-arm $\overrightarrow{\Delta\beta}^t$ based on the learned reward distributions. Similar to our solution in the fully-known network, we conduct such "learning- determining" process also relying on the CCMAB. Before we elaborating our solution, let us introduce an additional element in CCMAB (i.e., regret) that we need under the initially unknown reward distributions, besides the base-arm, super-arm and reward that we have introduced in Section 3.

Regret: The regret, which is a metric for evaluating the performances of bandit-based solutions on coping with unknown reward distributions, quantifies the gap between the reward obtained from the adopted super-arm and the reward of the optimal super-arm given exact reward distributions. Corresponding to the adaptive diffusion problem, the regret in the t-th round quantifies the difference between the diffusion size of sensitive information under the super-arm $\overrightarrow{\Delta\beta}^t$ and that under the optimal probability variation vector $\overrightarrow{\Delta\beta}^{t*}$. Thus, in the t-th round, the expected regret in adaptive diffusion problem refers to the value of

$$\overrightarrow{D^t} \cdot (\overrightarrow{\beta_0^t} + \overrightarrow{\Delta\beta}^t) - \overrightarrow{D^t} \cdot (\overrightarrow{\beta_0^t} + \overrightarrow{\Delta\beta}^{t*}) = \overrightarrow{D^t}\overrightarrow{\Delta\beta}^t - \overrightarrow{D^t}\overrightarrow{\Delta\beta}^{t*}.$$

Since a lower regret demonstrates better performance of the adopted super-arm in each round, the aim of minimizing the sensitive information diffusion over the T rounds is equivalent to minimizing the cumulative regret over the T rounds whose expectation is $\sum_{t=1}^{T} \overrightarrow{D^t}\overrightarrow{\Delta\beta}^t - \overrightarrow{D^t}\overrightarrow{\Delta\beta}^{t*}$.

Summary of the mappings. Together with the mappings of the base-arm, super-arm and reward given in Section 3, the objective of the bandit based solution in the semi-known network is also selecting a set of base-arms with

the minimum sum of rewards to form the super-arm. As the diffusion ability vector $\overrightarrow{D^t}$ remains unknown in advance, the mean rewards of base-arms (i.e., $\overrightarrow{D^t} \cdot \overrightarrow{\beta_i}$) are also unknown. For this issue, we propose to iteratively learn the mean rewards of base-arms from the diffusion feedback in each round and determine the super-arm based on the currently learned mean rewards of base-arms. With this idea, we proceed to give the algorithm for the adaptive diffusion over the semi-known network.

4.2 Algorithm over Semi-known Network

4.2.1 General Idea of the Proposed Algorithm

We design the algorithm for minimizing the overall reward under the unknown mean rewards of base-arms as the "determining-learning" process. Initially, for the base-arms whose two non-zero elements are both on the target edges with informed destination nodes, we set the mean reward as $\overrightarrow{D^t} \cdot \overrightarrow{\beta_i}$. Besides, for the base-arms associated with the uninformed destination nodes and without the exact mean rewards, we attach each of such base-arms an initial estimated mean reward which, obviously, is an unreliable estimated value. Then in each round, the algorithm mainly consists of two phases: (1) Determining the probability variation vector at the beginning of the round by determining the picked super-arm which consists of a set of base-arms; (2) At the end of the round, refining the estimated mean reward of each picked base-arm based on the rewards obtained from the picked base-arms in the current round. We provide the main ideas of the two phases as below.

In the determining phase, there are two complementary selections, i.e., *Exploitation* and *Exploration*, when determining the super-arm.

- **Exploitation:** The objective of Exploitation is to get the minimum reward in the current round. That is, it picks the super-arm which consists of a set of valid base-arms with the minimum sum of current estimated mean rewards.
- **Exploration:** The objective of Exploration is picking a super-arm which consists of as many base-arms as possible. Since the learning phase only refines the mean rewards of the base-arms combined into the picked super-arm, the Exploration enables us to learn the mean rewards of as many base-arms as possible.

The ideas of the two selections indicate that the Exploitation aims at obtaining the minimum reward in the current round. On the contrast, the Exploration aims at learning the mean rewards of as many base-arms as possible. Since the reliable estimation of the mean rewards of more base-arms are helpful to determine the better super-arm in future rounds, Exploration essentially benefits obtaining less reward in future rounds. Then, with the aim of minimizing the reward in a long run (over T rounds), how to balance the trade-off between the two selections? To cope with such dilemma, we design the procedures in the determining phase based on the $\varepsilon - greedy$ process, which is one of the two most widely adopted solutions for balancing the trade off between Exploitation and Exploration [33] [34] [35]. The other one is the UCB (Upper Confidence Bound) approach [36]. The idea of $\varepsilon - greedy$ is picking the super-arm as Exploration with probability ε, and as Exploitation with probability $(1-\varepsilon)$, where ε decreases with time. As we will experimentally demonstrate in Section 5, our proposed $\varepsilon - greedy$ based algorithm performs better than UCB-based algorithm on the adaptive information diffusion problem.

In the learning phase, the main task is updating the estimated mean reward of each picked base-arm based on the reward obtained from each base-arm.

In summary, we give in Algorithm 2 the framework for determining the probability variation vector over semi-known network. In the determining phase, we determine the super-arm as Exploration with probability $\varepsilon_t = \frac{\varepsilon_0}{\sqrt{t}}$, which decreases over time. Upon observing the diffusion size of sensitive informations during the current round, in the learning phase, we update the estimated mean reward of each base-arm combined into the picked super-arm.

Algorithm 2: Algorithm over semi-known network

```
for t = 1 to T do
    // Determining phase
    ε_t ← ε_0/√t ;
    if ε_t then
        Super-arm ←Exploration;
    else
        Super-arm ←Exploitation;
    end
    Picking the super-arm;
    Observing the diffusion size of sensitive informations
      in current round;
    // Learning phase
    Updating the estimated mean reward of each
      base-arm in super-arm;
end
```

With the above general framework in Algorithm 2, we respectively present the details in the two phases as follows.

4.2.2 Bandit Based Algorithm for Adaptive Diffusion

1. Initialization. At the beginning of each round, for the base-arms which emerge in the previous rounds, we set their mean rewards as the last estimated values. For the base-arms which are built upon the uninformed nodes that newly become sensitive in last round, we uniformly attach such base-arms an initial estimated mean reward.

2. Procedures in the determining phase. The main task of the determining phase in each round is determining a super-arm which consists of a set of base-arms selected as Exploration or Exploitation.

Algorithm for Exploration. As noted earlier, the objective in the Exploration is selecting a valid combination of as many base-arms as possible. We give the pseudo code of the algorithm in Exploration in Algorithm 3. We use $combination$ to denote the set of the selected base-arms. In Algorithm 3, we first randomly choose a base-arm and then iterate all the other base-arms. During the iterations, if the base-arm can be validly combined with all the other base-arms in the current $combination$, we add it into the $combination$.

Algorithm for Exploitation. Different from the Exploration, the Exploitation aims at selecting a combination of base-arms with the minimum sum of estimated mean rewards. For this, we propose a greedy strategy in Algorithm 4 to select the combination of base-arms in Exploitation. Specifically, we iteratively select the base-arm v with the minimum

Algorithm 3: Exploration

Input: All the base-arms
Output: A set of selected base-arms
$ActionPool \leftarrow$ All the base-arms;
$u \leftarrow$RANDOM($ActionPool, 1$);
/* RANDOM(S, n) returns n random items in set S. */;
$combination \leftarrow \{u\}$;
for v *in* $ActionPool \backslash u$ **do**
  /* VALID(a_1, a_2) returns a boolean value of whether the combination of vectors a_1 and a_2 is valid or not. */;
  if *VALID(combination, v)* **then**
    $combination \leftarrow combination \cup \{v\}$;
  end
end
return $combination$

estimated mean reward $\mu_{v,t}$. Then, if the base-arm v is not conflicted with all the base-arms in the current $combination$ and has negative estimated mean reward, we add it into the $combination$.

Algorithm 4: Exploitation

Input: All the base-arms
Output: A set of selected base-arms
$ActionPool \leftarrow$ All the base-arms;
$combination \leftarrow \varnothing$;
while $ActionPool \neq \varnothing$ **do**
  $v =$MIN($ActionPool$);
  /* MIN(S) returns the item with the minimum reward in set S. */;
  if $\mu_{v,t} > 0$ **then**
    End While;
  end
  $ActionPool \leftarrow ActionPool \backslash \{v\}$;
  if *VALID(combination, v)* **then**
    $combination \leftarrow combination \cup \{v\}$;
  end
end
return $combination$;

Then the super-arm in each round is determined by the sum of the probability variation vectors represented by the base-arms in the $combination$.

3. Procedures in the learning phase. After picking the super-arm, the reward of each base-arm is then observed after the diffusion during current round. The task in the learning phase is updating the estimated mean reward of each base-arm in the $combination$. We use $reward(i)$ to denote the reward obtained from base-arm $\overrightarrow{\beta_i}$. Specifically, given a base-arm $\overrightarrow{\beta_i}$ whose two non-zero elements are on the target edges a and b, we take the diffusion size of sensitive information from the destination nodes of the edges a and b as the diffusion feedback of the base-arm. In reality, such diffusion size can be counted from the increase of the numbers of the visits, share, likes or replies of the sensitive informations diffused from the destination nodes of edges a and b. In some popular social medias like Twitter and Weibo, such numbers are available from the home pages of the destination nodes, without knowing who are the neighbors of the destination nodes.

As illustrated in Section 3.1, the $reward(i)$ is quantified by the diffusion feedback minus the constant $\overrightarrow{D^t} \cdot (\overrightarrow{\beta_0^t} \odot \overrightarrow{I_i})$. Moreover, we use $T_{i,t}$ to denote the times that $\overrightarrow{\beta_i}$ has been combined into the super-arm until the t-th round. Then the estimated mean reward of $\overrightarrow{\beta_i}$ is updated as

$$\mu_{i,t} \leftarrow [(T_{i,t} - 1) * \mu_{i,t-1} + reward(i)]/T_{i,t},$$

which is the average of the rewards obtained from $\overrightarrow{\beta_i}$ in the $T_{i,t}$ rounds. Notably, for the base-arms whose two non-zero elements are both on the target edges with informed destination nodes, we have the exact mean rewards of them. Then we keep the mean rewards of such base-arms as their exact values during the T rounds, and only update the estimated mean rewards of the base-arms with unknown reward distributions.

3. Summary. We summarize the procedures in the determining and learning phases in Algorithm 5, named **ADSN** (**A**daptive **D**iffusion in **S**emi-known **N**etwork), which we propose to determine the probability variation vector for minimizing the sensitive information diffusion over the semi-known network. **ADSN** globally follows the $\varepsilon-greedy$ approach and consists of procedures of the determining phase, including Exploration and Exploitation, and the learning phase. In each round, with probability $\frac{\varepsilon_0}{\sqrt{t}}$, **ADSN** determines the value of super-arm as Exploration, and with probability $(1 - \frac{\varepsilon_0}{\sqrt{t}})$ as Exploitation. After the diffusion during the current round, **ADSN** then updates the estimated mean reward of each base-arm in the $combination$.

Algorithm 5: ADSN

Input: All the base-arms, and ε_0, T, $T_{i,t-1} = 0$ for all base-arms
Output: A sequence of super-arms
for $t = 1$ *to* T **do**
  $\varepsilon_t \leftarrow \frac{\varepsilon_0}{\sqrt{t}}$, $\overrightarrow{\Delta\beta^t} = \vec{0}$;
  if ε_t **then**
    $combination \leftarrow$Exploration;
  else
    $combination \leftarrow$Exploitation;
  end
  for $\overrightarrow{\beta_i} \in combination$ **do**
    $\overrightarrow{\Delta\beta^t} = \overrightarrow{\Delta\beta^t} + \overrightarrow{\beta_i}$;
  end
  Picking the super-arm $\overrightarrow{\Delta\beta^t}$;
  Observing diffusion feedback;
  for $\overrightarrow{\beta_i}$ *in combination* **do**
    if $\overrightarrow{\beta_i}$ *without exact mean reward* **then**
      $T_{i,t} \leftarrow T_{i,t-1} + 1$;
      $\mu_{i,t} \leftarrow [(T_{i,t} - 1) * \mu_{i,t-1} + reward(i)]/T_{i,t}$;
    end
  end
end

4.3 Performance Analysis of ADSN on Regret

As illustrated in Section 4.1, the regret is a metric which evaluates the performance of the solutions to the bandit-based problem. The regret refers to the reward gap between the picked super-arms and the optimal super-arms during the T rounds. Corresponding to the adaptive diffusion problem, the regret in the t-th round quantifies the difference between the diffusion size of sensitive informations under the variation vector $\overrightarrow{\Delta\beta^t}$ returned by **ADSN** and that under the optimal probability variation vector $\overrightarrow{\Delta\beta^{t*}}$. We present in Theorem 1 the upper bound of the expected regret of **ADSN** over the T diffusion rounds.

Theorem 1. *The expected regret of **ADSN** over T diffusion rounds, i.e., $\mathbb{E}[R_{\textbf{BLAG}}]$, is upper bounded by*

$$\mathbb{E}[R_{\textbf{ADSN}}] \leq O\left(M'\sqrt{T}\right),$$

where M' denotes the size of base-arms during the T-th round.

Proof. (Sketch.) Since **ADSN** is designed as the $\varepsilon - greedy$ approach which conducts Exploration with probability ε_t and conducts Exploitation with probability $(1-\varepsilon_t)$, the expected reward in the t-th round is equal to

$$\mathbb{E}[\overrightarrow{D^t} \cdot \overrightarrow{\Delta\beta}^t] = \varepsilon_t \mathbb{E}[\overrightarrow{D^t} \cdot \overrightarrow{\Delta\beta}^t_{ep}] + (1-\varepsilon_t)\mathbb{E}[\overrightarrow{D^t} \cdot \overrightarrow{\Delta\beta}^t_{et}],$$

where $\Delta\beta^t_{ep}$ is the super-arm determined by Exploration and $\Delta\beta^t_{et}$ is the super-arm determined by Exploitation. Moreover, since the regret in the t-th round is defined as $\overrightarrow{D^t} \cdot \overrightarrow{\Delta\beta}^t - \overrightarrow{D^t} \cdot \overrightarrow{\Delta\beta}^{t*}$, we give the expected regret in the t-th round as

$$\begin{aligned}&\mathbb{E}[\overrightarrow{D^t} \cdot \overrightarrow{\Delta\beta}^t - \overrightarrow{D^t}\overrightarrow{\Delta\beta}^{t*}]\\ =&\varepsilon_t \mathbb{E}[\overrightarrow{D^t} \cdot \overrightarrow{\Delta\beta}^t_{ep} - \overrightarrow{D^t}\overrightarrow{\Delta\beta}^{t*}] + (1-\varepsilon_t)\mathbb{E}[\overrightarrow{D^t} \cdot \overrightarrow{\Delta\beta}^t_{et} - \overrightarrow{D^t}\overrightarrow{\Delta\beta}^{t*}].\end{aligned}$$

We prove in Appendix B (in supplemental material) that, for the regret in Exploitation over T rounds, we have

$$\sum_{t=1}^{T}(1-\varepsilon_t)\mathbb{E}[\overrightarrow{D^t} \cdot \overrightarrow{\Delta\beta}^t_{et} - \overrightarrow{D^t}\overrightarrow{\Delta\beta}^{t*}] \leq O\left(M'\sqrt{T}\right). \quad (2)$$

Then for the regret in Exploration over T rounds, we have

$$\sum_{t=1}^{T}\varepsilon_t \mathbb{E}[\overrightarrow{D^t} \cdot \overrightarrow{\Delta\beta}^t_{ep} - \overrightarrow{D^t}\overrightarrow{\Delta\beta}^{t*}] \leq O\left(M'\sqrt{T}\right). \quad (3)$$

Together with Eqns. (2) and (3), we obtain the conclusion in Theorem 1 that $\mathbb{E}[R_{\textbf{BLAG}}] \leq O\left(M'\sqrt{T}\right)$. □

Theorem 1 presents that the maximum gap between the diffusion size of sensitive informations under the probability variations returned by our solution and that under the optimal probability variations is $O\left(M'\sqrt{T}\right)$. Moreover, since the solution in the fully-known network has a better performance under the known reward distributions comparing the solution in semi-known network, the $O\left(M'\sqrt{T}\right)$ also upper bounds the regret of the solution returned by Algorithm 1 in the fully-known network. In addition, Theorem 1 presents that the regret bound of **ADSN** sub-linearly grows with the number of rounds, indicating that the super-arm returned by **ADSN** approximates to the optimal super-arm when T is sufficiently large. In Appendix B (in supplemental material), we further provably show that **ADSN** exhibits less expected regret comparing with the UCB-based approaches to justify our selection of $\varepsilon - greedy$.

Remark. We give the lower bound of the regret of **ADSN** based on the Lai and Robbins' Theorem [37]. The Lai and Robbins' theorem presents that, for a $\varepsilon - greedy$ based solution with initially unknown reward distributions, the lower regret bound of such solution is $O(\log T \cdot \sum_{a|\Delta_a>0} \frac{\Delta_a}{KL(r_a||r^*)})$ where Δ_a denotes the gap between the rewards of an arm a and the optimal arm and $KL(r_a||r^*)$ is the KL-divergence between the reward distribution r_a of an arm a and the reward distribution r^* of the optimal arm [37] [38]. Thus, given the initially unknown reward distributions, the lower bound the regret of **ADSN** is $O(\log T)$.

4.4 Complexity and Distributed Implementation of ADSN

4.4.1 Complexity

Now, we present the complexity of **ADSN** in each round. In the Exploration, **ADSN** needs to traverse all the base-arms, and costs a complexity of $O(|\mathfrak{E}^t|^2)$, where $|\mathfrak{E}^t|$ is the size of the target edges. In the Exploitation, **ADSN** also needs to traverse the base-arms for obtaining a valid $combination$ with the minimum reward, and costs a complexity of $O(|\mathfrak{E}^t|^2)$. In the learning phase, **ADSN** iterates the picked base-arms to update their estimated mean rewards based on the diffusion feedback and costs a complexity of $O(|\mathfrak{E}^t|^2)$. Together with the complexities in the above procedures, the overall complexity of **ADSN** is $O(|\mathfrak{E}^t|^2)$.

Moreover, for further improvement of time efficiency, we also propose the distributed implementation scheme of **ADSN** as follows.

4.4.2 Design for Distributed Implementation

The core of the distributed implementation of **ADSN** is also the distributed storage of base-arms. We store the base-arms into N storage units, and through which, we can parallelly conduct the traverse process in both Exploration and Exploitation. We take the main procedure of **ADSN** as the **master**, and the N storage units as **slaves**. We explicate below the distributed implementation of Exploration, Exploitation and the learning phase in **ADSN**.

Exploration: The master first randomly selects a base-arm from a randomly chosen slave. Then the master sends the 'Exploration' order and the selected base-arm to each slave. Once receiving the selected base-arm, each slave initiates an empty local $combination$ and traverses the base-arms stored in its memory. If a slave finds a base-arm that can be validly combined with the selected base-arm and is valid with the existing local $combination$, the slave adds the base-arm into local $combination$. By the end, each slave sends the local $combination$ back to the master. The master then, starting from the selected super-arm, traverses all the local $combinations$ to check the validity of the base-arms belonging to them in sequence, and if valid, the master adds the base-arm into the final $combination$.

Exploitation: Every slave has a local lookup table that contains the current estimated mean rewards of all the local base-arms. Upon receiving the 'Exploitation' order from the master, every slave finds a local $combination$ with the minimum sum of rewards and then sends the local $combination$ back to the master. The master then emerges all the base-arms returned by the slaves into an $ActionPool$. Over the $ActionPool$, the master generates the valid $combination$ via the greedy manner as shown in Algorithm 4.

Learning phase: This part is mainly accomplished by the salves. After the information diffusion during a round, each slave updates the estimated mean rewards of the local base-arms which are picked in the current round, and then updates the local lookup table.

We also algorithmically present the above procedures in Appendix C in the supplemental material.

5 Experimental Results

In this section, we evaluate the performance of the proposed solutions for the adaptive diffusion problem from the following four aspects, i.e., information diffusion loss, regret, cascading scale of sensitive informations, and the time efficiency of distributed implementation. (1) The information diffusion loss quantifies the loss of non-sensitive information diffusion when constraining sensitive information diffusion. The lower information diffusion loss means better user experience in reality. (2) We compare the reward obtained from **ADSN** with that of **CUCB** to justify the effectiveness of **ADSN** on minimizing the overall reward. (3) We present the cascading scale of sensitive information over time to show that our solution can effectively constrain the diffusion of sensitive information. (4) We report the running time of **ADFN** and **ADSN** as the original procedures and the distributed implementation schemes.

5.1 Datasets and Settings

Datasets. Our experiments are over the datasets of three real social networks and three synthetic networks[2], whose basic descriptions and statistics are summarized as follows:

- **Twitter:** This dataset consists of partial users and social links in Twitter, and includes 81, 306 nodes and 1,768,149 edges.
- **Livejournal:** Livejournal is an online community with almost 10 million members. This datasets contains 4,847,571 nodes and 68,993,773 edges.
- **Pokec:** Pokec is the most popular online social network in Slovakia. This dataset contains a network with 1,632,803 nodes and 30,622,564 edges.
- **B.A. graph** (short for Barabasi Albert graph) [39]: A synthetic graph that forms as newly added nodes preferentially attaching to existing nodes with higher degrees. The **B.A. graph** well captures the power-law degree distribution in real social networks. Each graph has two parameters: n (total number of nodes), p (number of edges that each new node connects with existing nodes). We generate three networks as the B.A. model with the parameters respectively being ($n = 1,000$, $p = 3$), ($n = 5,000$, $p = 4$) and ($n = 10,000$, $p = 5$).

Settings. We set the original diffusion probability via each social link as 5×10^{-3}. In the experiments over semi-known networks, we set the diffusion abilities of half of the users as unknown. We initially set the diffusion abilities of uninformed nodes as the average of the diffusion abilities of all the informed nodes, and set the initial mean rewards of base-arms built upon uninformed node based on such initial diffusion abilities. All the algorithms over a computer with Ubuntu 16.04 LTS, 40 cores 2.30 GHz and 128 GB memory.

5.2 Information Diffusion Loss

As illustrated earlier, we aim at constraining the diffusion of sensitive informations while lowering the diffusion loss of non-sensitive informations, for the aim of preserving the global diffusion ability of the whole network on diffusing non-sensitive informations. In experiments, for quantifying the diffusion loss of non-sensitive informations, we run the diffusion process with and without sensitive information constraining measures respectively. In the cases without such measures, we tag the size of users receiving sensitive informations in each round as N^s_{wo} and tag the size of users receiving non-sensitive informations as N^n_{wo}. While in the cases with such measures, we tag the diffusion sizes of sensitive information and non-sensitive information respectively as N^s_w and N^n_w. With the above denotations, we quantify the information diffusion loss in each round as $info\text{-}loss = \frac{N^n_{wo} - N^n_w}{N^s_{wo} - N^s_w}$. Then, the lower value of $info\text{-}loss$ means less information diffusion loss, when constraining the diffusion of sensitive informations.

We compare the $info\text{-}loss$ of **ADFN** on fully-known network and **ADSN** on semi-known network with the following four baselines:

(1) DRIMUX [9]: As an online rumor blocking algorithm, DRIMUX periodically blocks a given fraction α of the most influential users to constrain rumor diffusion. For user experience, DRIMUX sets a threshold of the blocking time of each user. Referring to the settings in [9], when conducting DRIMUX, we set the fraction of blocked users in each round as $\alpha = 10^{-4}, 5 \times 10^{-4}, 10^{-3}$. Also, we set the threshold of blocking time of each user as 100 rounds.

(2) TIBS [15]: For minimizing the diffusion of rumors, TIBS proposes to block a given number K of the most influential users during the diffusion. Then, when conducting TIBS, we set $K = 10, 25, 50$ according to [15].

(3) RIPOSTE [16]: Typically, RIPOSTE forwards an item with a probability larger than a given amount if a user likes the item, and with a probability smaller than the given amount otherwise. In our scenario, we implement **RIPOSTE** by randomly decreasing or increasing the diffusion probabilities via some edges by a constant value.

(4) Monotone [27]: This strategy constrains the rumor diffusion through decreasing the diffusion probabilities with time. We fit this strategy into our scenario also by decreasing the diffusion probabilities via edges with time.

Before we conduct the diffusion constraining algorithms, over each network, we start the diffusion of sensitive information from the node with the highest out-degree and start the diffusion of non-sensitive information from a randomly chosen node both for 50 rounds. Then, the users which receive sensitive information during the 50 rounds are taken as the initial sensitive nodes in V^1. With such initialization, Fig. 2 plots the results of $info\text{-}loss$ over fully-known networks during the following 1000 rounds and Fig. 3 plots the results of $info\text{-}loss$ over semi-known networks. Corresponding to the reality, if users in social medias visit the Facebook, Moments, Tweets, Weibo, and etc once per hour, then the diffusion via each hop corresponds to one hour and each round corresponds to two hours. Thus, we set the timescale of 1000 rounds under the consideration that the timeliness of a given information lasts for three months.

From Figs. 2 and 3, we can see that both **ADFN** and **ADSN** yield the best performance on lowering information diffusion loss, and have the 40% less $info\text{-}loss$ comparing with the four baseline algorithms. The $info\text{-}loss$ of

2. The three real datasets are downloaded from http://snap.stanford.edu/data/index.html

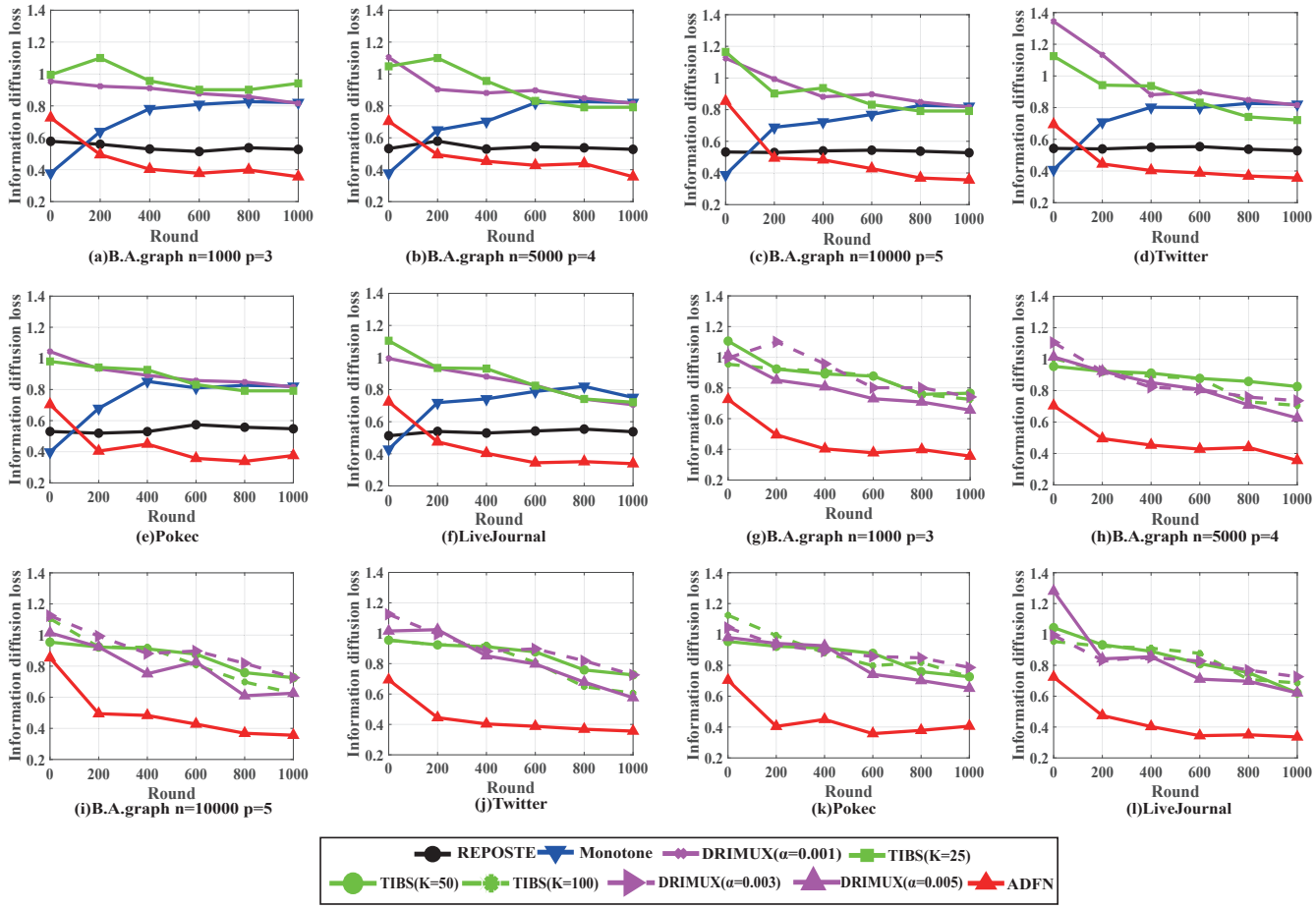


Fig. 2. Information loss incurred by different algorithms over fully-known networks.

RIPOSTE is always around 0.5 because it takes the same measure in each round that randomly increase or decrease diffusion probabilities via social links. Since the sensitive users are more than the users holding non-sensitive information in initialization, the $info\text{-}loss$ of **RIPOSTE** is less than 1. The $info\text{-}loss$ of **Monotone** grows from 0.4 to 1 during the 1000 rounds since the diffusion probabilities decrease with time. For **DRIMUX** and **TIBS**, the values of $info\text{-}loss$ are around 1 during the 1000 rounds. The reason behind is that **DRIMUX** and **TIBS** block the most influential users in the network and have the comparable effect on both the sensitive and non-sensitive information diffusion.

For **ADFN** and **ADSN**, we can see from Figs. 2 and 3 that **ADFN** and **ADSN** do not exhibit their effectiveness in the first few rounds and have the comparable performance with **DRIMUX** and **TIBS**. This is because that there are just a fraction of users holding sensitive informations at the beginning, and **ADFN** and **ADSN** prefer to largely decrease the diffusion probabilities starting from high-degree nodes. Notably, **ADSN** has bad performance in more rounds comparing with **ADFN**. The reason behind is that **ADSN** needs to explore the base-arms with unknown mean rewards in the early stage. Furthermore, Fig. 3 reveals that, with the increase of network size, **ADSN** undergoes more rounds with unsatisfiable $info\text{-}loss$. This is because that the increase of network size leads to the larger size of base-arms, and consequently, results in the consumption of more rounds on learning the exact mean rewards.

5.3 Reward of Bandit

Now, we proceed to evaluate the performance of **ADSN** on minimizing reward, comparing with the **CUCB** (**C**ombinatorial **U**pper **C**onfidence **B**ound) which is the extension of the UCB-based bandit solution to the combinatorial problem [35]. **CUCB** selects the super-arm based on the estimated means and the variances of the reward of base-arms. Corresponding to the adaptive diffusion problem, **CUCB** selects a combination of base-arms with the minimum sum of $\mu_{i,t} - \frac{c\sigma}{\sqrt{T_{i,t}}}$ (c is a preset constant).

As illustrated before, minimizing the sensitive information diffusion size is equivalent to minimizing the reward obtained from the bandit. We present in Table 1 the rewards of **ADSN** and **CUCB** over the B.A. graph with $n = 10k$ and $p = 5$. In this comparison, we set the number of diffusion rounds as $1000, 3000$ and set the size of base-arms as $200, 5k, 20k, 2M$ to reveal the effect of the number of diffusion rounds and the size of base-arms. We set $\varepsilon_0 = 1$.

From Table 1, we can see that **ADSN** significantly outperforms **CUCB** on minimizing reward. This is because that **ADSN** conducts Exploration with the decreasing probability ε_t, and determines the value of super-arms as Exploitation with high probability in the later stage. Such preference for Exploitation then enables **ADSN** to achieve stable performance. However, in **CUCB**, the base-arms with the smaller value of $T_{i,t}$ tends to have the smaller value of $\mu_{i,t} - \frac{c\sigma}{\sqrt{T_{i,t}}}$. With the aim of obtaining the minimum sum of

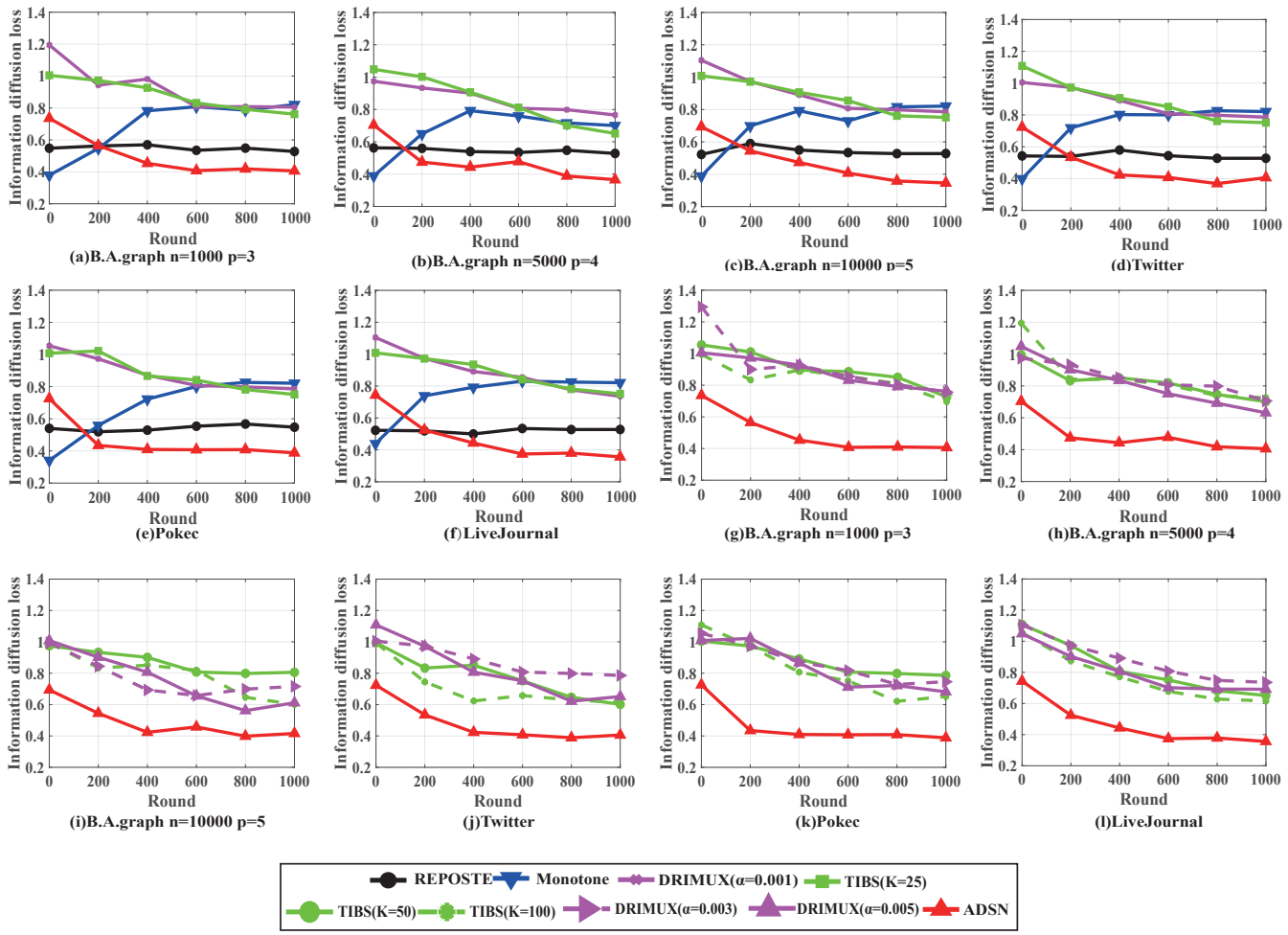


Fig. 3. Information loss incurred by different algorithms over semi-known networks.

TABLE 1
Comparison of cumulative reward between **CUCB** and **ADSN** under different sizes of base-arms.

Rounds	Base-arms	CUCB	ADSN
1000	200	-0.6	-6.7
	5K	-0.6	-32
	20K	-0.4e-1	-4.8e-1
	2M	-0.2e-3	-12.3e-2
3000	200	-1.4	-8.4
	5K	-0.8e-1	-22.3e-1
	20K	-0.5e-1	-109.9e-1
	2M	-1.6e-2	-14.8e-2

$\mu_{i,t} - \frac{c\sigma}{\sqrt{T_{i,t}}}$, **CUCB** tends to pick the base-arms which have never been picked or have just been picked in a few rounds. Due to the large scale of the base-arm set, most of the base-arms are picked only a few times during the limited rounds. As a result, **CUCB** spends most of the diffusion rounds on Exploration and leads to the unstable performance.

Furthermore, with the number of base-arms increasing from 200 to $2M$, **CUCB** yields more deteriorating performance. This is also because that **CUCB** tends to pick the base-arms with smaller $T_{i,t}$. With the increase of the number of base-arms, **CUCB** costs more rounds on Exploration and results in the worse performance.

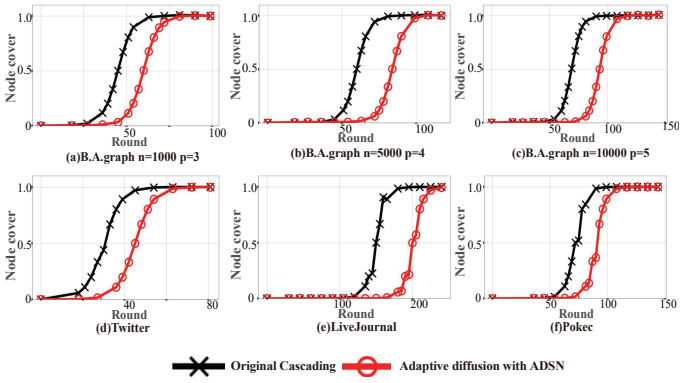
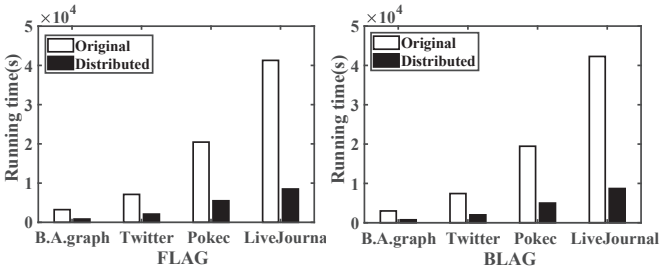


Fig. 4. Diffusion size of sensitive information under Original cascading and **ADSN**

5.4 Diffusion Size of Sensitive Information

We now take a look into the effectiveness of the proposed solutions on constraining the cascading of sensitive information. We present in Fig. 4 the ratio of sensitive users, say "Node Cover", over time in the three synthetic networks and three real social networks. The black line in Fig. 4 represents the increase of "Node Cover" over time under original diffusion (i.e., without diffusion constraining measures), and the red line represents the "Node Cover" under **ADSN**. A common observation from Figs. 4 (a)-(f) is



(a) On Fully-known network (b) On Semi-known network

Fig. 5. Running time of **ADFN** and **ADSN**.

that the diffusion size of sensitive information undergoes a transition, i.e., the diffusion size increases explosively after a certain time. Notably, **ADSN** can effectively postpone such transition of sensitive information diffusion size. Considering the timeliness of sensitive information in reality, if the transition is postponed to after the timeliness of sensitive information, no sensitive information will be widely spread.

5.5 Distributed Implementation

Last but not least, we report the running time of both **ADFN** and **ADSN** in one round to validate the efficiency of the proposed distributed implementation schemes. In Fig. 5, Fig. 5(a) presents the running time of **ADFN** as the original procedures in Algorithm 1 and as the distributed implementation scheme, and Fig. 5(b) presents the running time of **ADSN**. In the distributed implementation, we set the number of slaves as 32. That is, we distributedly store the base-arms into 32 units, and parallelly conduct the traverse process in both Exploration and Exploitation over the 32 units. Since the traverse of base-arms is the most time-consuming task when determining the super-arm, we can see from Fig. 5 that the running time of the distributed implementation is much less than the running time of the original procedures.

6 Related Works

For characterizing the information diffusion process in online social networks, Kempe *et al.* [23] first propose two classic diffusion models: Independent Cascading (IC) model and linear threshold (LT) model. In the IC model, each user has a single chance to successfully diffuse the information to his neighbors with a given probability after this user having received the information. While in the LT model, a user would get the information if a certain fraction of his neighbors have received the information. Since then, a great deal of works study the Influence Maximization (IM) problem, which focuses on efficiently selecting the optimal seed users to trigger a diffusion process in hope of maximizing the final information diffusion size [1]. Recently, due to the high cost of seeding influential users, Shi *et al.* [3] propose to let influential users repost the required information while seed the ordinary users for lowering the cost of IM campaign. Similar to the multi-round setting in this paper, the seed selection for maximizing the information diffusion in multiple time rounds is considered in [2] [40]. Moreover, considering the widespread interactions between the cyber (online) and physical (offline) worlds, offline events are utilized in [7] to further improve the performance of IM.

On the contrast of the IM problem, there are also abundant researches focusing on minimizing the influence of rumors. One strategy for rumor influence minimization is diffusing the truths over network to counteract rumors [12]- [14]. Specifically, the competitive linear threshold (CLT) model that characterizes the competing diffusion of truth and rumor is introduced in [12]. Then He *et al.* [12] and Chen *et al.* [14] propose to select a set of seed users to maximize the diffusion of truths under the CLT model. Chen *et al.* [13] extends the IC model to describe the diffusion of positive informations under the effect of negative information, and studies how to maximize the positive information diffusion. However, such clarifying measure cannot be used to constrain the diffusion of private sensitive informations such as personal informations, trade secrets.

Another class of rumor blocking measures focuses on blocking a certain number of influential users [9] [15] or social links [17]- [20]. On one hand, Song *et al.* [15] propose to temporarily block a number of users with high diffusion abilities to reduce the diffusion of rumors before a deadline. With the consideration of user experiences, Wang *et al.* [9] study the online rumor blocking problem that periodically blocking a fraction of users during the rumor diffusion, and set a threshold to controls the blocking time of each user. Further, for coping with the unforeseen events in rumor diffusion, the adaptive blocking strategy is proposed in [10]. On the other hand, considering that straightforwardly blocking users is not desirable, [17]- [20] propose to block a given number of social links for minimizing the diffusion of rumors. However, as we illustrated before, this kind of measures may incur much information diffusion loss, if being adopted to constrain the diffusion of the sensitive informations considered in this paper. In addition, taking measures to constrain or promote information diffusion is also related to the studies about the effect of human behaviors on diffusion [29] [41] [42].

Besides the information diffusion, our work is also related to the combinatorial multi-arm bandit model. [36] [43] introduce the general multi-arm bandit model where only one arm is picked in each round. Recent studies [40] [44] [45] utilize the combinatorial bandit in the IM problem over unknown or dynamic networks, where the diffusion probabilities in IC model are assumed to be unknown in advance. In each round, the solutions proposed in [40] [44] [45] first take the diffusion results in previous rounds as the feedback to learn the diffusion probability via each edge, and then conduct the seed selection based on the learned diffusion probabilities.

7 Conclusion and Future Work

In this paper, we study the problem of constraining the diffusion of sensitive informations in social networks while preserving the diffusion of non-sensitive informations. We model the diffusion constraining measures as the variations of diffusion probabilities via social links, and model the problem of interest as adaptively determining the probability variations through a constrained minimization problem in multiple rounds. We utilize the CCMAB framework to

jointly design our solutions in the fully-known and semi-known networks. Over the fully-known network, we propose the CCMAB based algorithm **ADFN** to efficiently determine the probability variations via social links. Over the semi-known network, for tackling the challenge of unknown diffusion abilities of partial users, we propose the algorithm **ADSN** to iteratively learn the unknown diffusion abilities and determine the probability variations based on the learned diffusion abilities in each round. The analysis of regret bound and extensive experiments have been conducted to justify the superiority of our solutions.

In addition, in the current work, we define the constraint of maintaining the sum of diffusion probabilities via edges in the objective problem, for the aim of preserving the global diffusion ability of the whole network on diffusing non-sensitive informations. In the future work, we will explore other relevant solutions such as simultaneously minimizing the sensitive information diffusion and maximizing the non-sensitive information diffusion.

Acknowledgement

This work was supported by National Key R&D Program of China 2018YFB1004705, 2018YFB2100302, and NSF China under Grant (No. 61822206, 61832013, 61960206002, 61532012).

References

[1] Y. Li, J. Fan, Y. Wang, and K. L. Tan, "Influence maximization on social graphs: A survey", in *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, vol. 30, no. 10, pp. 1852-1872, 2018.

[2] L. Sun, W. Huang, P. S. Yu, and W. Chen, " Multi-round influence maximization", in *Proc. ACM SIGKDD*, 2018.

[3] Q. Shi, C. Wang, J. Chen, Y. Feng, and C. Chen, "Post and repost: A holistic view of budgeted influence maximization", in *Neurocomputing*, vol. 338, pp. 92-100, 2019.

[4] X. Wu, L. Fu, Y. Yao, X. Fu, X. Wang, and G. Chen, "GLP: a novel framework for group-level location promotion in Geo-social networks", in *IEEE/ACM Transactions on Networking (TON)*, vol. 26, no. 6, pp. 1-14, 2018.

[5] Y. Lin, W. Chen, and J. C. Lui, "Boosting information spread: An algorithmic approach", in *Proc. IEEE ICDE*, 2017.

[6] Y. Zhang, and B. A Prakash, "Data-aware vaccine allocation over large networks", in *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 10, no. 2, article 20, 2015.

[7] Q. Shi, C. Wang, J. Chen, Y. Feng, and C. Chen, "Location driven influence maximization: Online spread via offline deployment", in *Knowledge-Based Systems*, vol. 166, pp. 30-41, 2019.

[8] H. T. Nguyen, T. P. Nguyen, T. N. Vu, and T. N. Dinh, "Outward influence and cascade size estimation in billion-scale networks, in *Proc. ACM SIGMETRICS*, 2017.

[9] B. Wang, G. Chen, L. Fu, L. Song, and X. Wang, "Drimux: Dynamic rumor influence minimization with user experience in social networks", in *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, vol. 29, no. 10, pp. 2168-2181, 2017.

[10] Q. Shi, C. Wang, D. Ye, J. Chen, Y. Feng, and C. Chen, "Adaptive Influence Blocking: Minimizing the Negative Spread by Observation-based Policies", in *Proc. IEEE ICDE*, 2019.

[11] S. Wen, J. Jiang, Y. Xiang, S. Yu, W. Zhou, and W. Jia, "To shut them up or to clarify: Restraining the spread of rumors in online social networks", in *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 12, pp. 3306-3316, 2014.

[12] X. He, G. Song, W. Chen, and Q. Jiang, "Influence blocking maximization in social networks under the competitive linear threshold model", in *Proc. SIAM SDM*, 2012.

[13] W. Chen, A. Collins, R. Cummings, T. Ke, Z. Liu, D. Rincon, X. Sun, Y. Wang, W. Wei, and Y. Yuan, "Influence Maximization in Social Networks When Negative Opinions May Emerge and Propagate", in *Proc. SIAM SDM*, 2011.

[14] C. Budak, D. Agrawal, and A. El Abbadi, "Limiting the spread of misinformation in social networks", in *Proc. ACM WWW*, 2011.

[15] C. Song, W. Hsu, and M. L. Lee, "Temporal influence blocking: minimizing the effect of misinformation in social networks", in *Proc. IEEE ICDE*, 2017.

[16] G. Giakkoupis, R. Guerraoui, A. Jégou, A.M. Kermarrec, and N. Mittal, "Privacy-conscious information diffusion in social networks", in *International Symposium on Distributed Computing*, pp. 480–496, Springer, 2015.

[17] Q. Yao, C. Zhou, L. Xiang, Y. Cao, and L. Guo, "Minimizing the negative influence by blocking links in social networks", in *International conference on trustworthy computing and services*, pp. 65–73, 2014.

[18] H. Tong, B. A. Prakash, T. Eliassi-Rad, M. Faloutsos, and C. Faloutsos, "Gelling, and melting, large graphs by edge manipulation", in *Proc. ACM CIKM*, 2012.

[19] M. Kimura, K. Saito, and H. Motoda, "Minimizing the Spread of Contamination by Blocking Links in a Network", in *Proc. AAAI*, 2008.

[20] E. B. Khalil, B. Dilkina, and L. Song, "Scalable diffusion-aware optimization of network topology", in *Proc. ACM SIGKDD*, 2014.

[21] M. M. Kircher, "A woman named Isis claims shes been blocked from signing into Facebook", https://www.businessinsider.com/woman-named-isis-blocked-facebook-2015-11, 2015.

[22] S. Fiegerman, "Facebook, google, twitter accused of enabling isis", https://money.cnn.com/2016/12/20/technology/twitter-facebook-google-lawsuit-isis/index.html, 2016.

[23] D. Kempe, J. Kleinberg and É. Tardos, "Maximizing the spread of influence through a social network", in *Proc. ACM SIGKDD*, pp. 137–146, 2003.

[24] S. Feng, G. Cong, A. Khan, X. Li, Y. Liu, and Y. M. Chee, "Inf2vec: Latent Representation Model for Social Influence Embedding", in *Proc. IEEE ICDE*, 2018.

[25] A. Guille and H. Hacid, "A predictive model for the temporal dynamics of information diffusion in online social networks", in *Proc. ACM WWW*, 2012.

[26] Y. Yang, J. Tang, C.W. Leung, Y. Sun, Q. Chen, J. Li and Q. Yang, "RAIN: Social Role-Aware Information Diffusion", in *Proc. AAAI*, 2015.

[27] X. Xu, X. Chen, "Modeling time-sensitive information diffusion in online social networks", in *Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 408–413, IEEE, 2015.

[28] A.L. Barabási and R.Albert, "Emergence of scaling in random networks", in *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[29] J. Iribarren and E. Moro, "Impact of human activity patterns on the dynamics of information diffusion", in *Physical review letters*, vol. 103, no. 3, pp. 038702, APS, 2009.

[30] V. Klee and G.J. Minty, "How good is the simplex algorithm, in *WASHINGTON UNIV SEATTLE DEPT OF MATHEMATICS*, 1970.

[31] L. G. Khachiyan, "A polynomial Algorithm in Linear Programming", in *Soviet Mathematics Doklady*, vol. 20, no. 1, pp. 191–194, 1979.

[32] N. Karmarkar, "A new polynomial-time algorithm for linear programming", in *Proc. ACM symposium on Theory of computing*, pp. 302-311, ACM, 1984.

[33] Y. Gai, B. Krishnamachari, and R. Jain, "Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations", in *IEEE/ACM Trans. on Networking (TON)*, vol. 20, pp. 1466–1478, 2012.

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

[35] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem", in *Machine learning*, vol. 47, pp. 235–256, Springer, 2002.

[36] P. Auer, "Using confidence bounds for exploitation-exploration trade-offs", in *Journal of Machine Learning Research*, vol. 3, pp. 397-422, 2002

[37] T. L. Lai, and H. Robbins, "Asymptotically efficient adaptive allocation rules", in *Advances in applied mathematics*, vol. 6, no. 1, pp. 422, 1985.

[38] S. Wang, and L. Huang, "Multi-armed Bandits with Compensation", in *Proc. NeurIPS*, 2018.

[39] Barabási–Albert model, https://en.wikipedia.org/wiki/Barab%C3%A1si%E2%80%93Albert_model, Wikipedia, 2018.

[40] S. Lei, S. Maniu, L. Mo, R. Cheng and P. Senellart, "Online influence maximization", in *Proc. ACM SIGKDD*, 2015.

[41] M. Karsai, M. Kivelä, R. K. Pan, K. Kaski, J. Kertész, A. Barabási and J. Saramäki, "Small but slow world: How network topology and burstiness slow down spreading", in *Physical Review E*, vol. 83, no. 2, pp. 025102, APS, 2011.

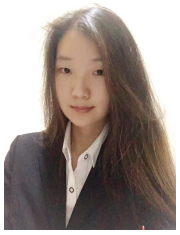
[42] S. Myers, C. Zhu and J. Leskovec, "Information diffusion and external influence in networks", in *Proc. ACM SIGKDD*, 2012.

[43] S. Mannor and O. Shamir, "From bandits to experts: On the value of side-observations", in *Proc. NeurIPS*, 2011.

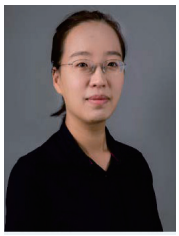
[44] X. Wu, L. Fu, J. Meng, and X. Wang, "Evolving Influence Maximization", in *arXiv preprint arXiv:1804.00802*, 2018.

[45] Z. Wen, B. Kveton, M. Valko, and S. Vaswani, "Online influence maximization under independent cascade model with semi-bandit feedback", in *Proc. NeurIPS*, 2017.

Xudong Wu received his B. E. degree in Information and Communication Engineering from Nanjing Institute of Technology, China, in 2015. He is currently pursuing his Ph.D. degree in Department of Computer Science and Engineering in Shanghai Jiao Tong University. His research of interests are in the area of social networking and big data, machine learning and combinatorial optimization.



Luoyi Fu received her B. E. degree in Electronic Engineering from Shanghai Jiao Tong University, China, in 2009 and Ph.D. degree in Computer Science and Engineering in the same university in 2015. She is currently an Assistant Professor in Department of Computer Science and Engineering in Shanghai Jiao Tong University. Her research of interests are in the area of social networking and big data, scaling laws analysis in wireless networks, connectivity analysis and random graphs. She has been a member of the Technical Program Committees of several conferences including ACM MobiHoc 2018-2020, IEEE INFOCOM 2018-2020.



Huan Long is currently an associate professor in the department of computer science and engineering in Shanghai Jiao Tong University. She is also a member of the John Hopcroft Center. Her main research interests lie in theoretical computer science, especially concurrency theory and machine learning theory.

Dali Yang is a B.E. undergraduate at Department of Aotomation in Shanghai Jiao Tong University, China. He is working as a research intern supervised by Dr. Luoyi Fu. His research interests include social networking and big data.

Yucheng Lu received his B. E. degree in Department of Computer Science and Engineering at Shanghai Jiao Tong University, China, 2018. During his undergraduate study, he was working as an research intern supervised by Dr. Luoyi Fu. His research interests include combinatorial optimization, machine learning and big data. He is pursuing Ph. D. degree in the Cornell University, New York, USA.

Xinbing Wang received the B.S. degree (with hons.) from the Department of Automation, Shanghai Jiaotong University, Shanghai, China, in 1998, and the M.S. degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China, in 2001. He received the Ph.D. degree, major in the Department of electrical and Computer Engineering, minor in the Department of Mathematics, North Carolina State University, Raleigh, in 2006. Currently, he is a professor in the Department of Electronic Engineering, Shanghai Jiaotong University, Shanghai, China. Dr. Wang has been an associate editor for IEEE/ACM Transactions on Networking and IEEE Transactions on Mobile Computing, and the member of the Technical Program Committees of several conferences including ACM MobiCom 2012, 2018-2019, ACM MobiHoc 2012-2014, IEEE INFOCOM 2009-2017.

Guihai Chen received the B.S. degree from Nanjing University, the M.E. degree from Southeast University, and the Ph.D. degree from The University of Hong Kong. He visited the Kyushu Institute of Technology, Japan, in 1998, as a Research Fellow, and the University of Queensland, Australia, in 2000, as a Visiting Professor. From 2001 to 2003, he was a Visiting Professor with Wayne State University. He is currently a Distinguished Professor and a Deputy Chair with the Department of Computer Science, Shanghai Jiao Tong University. He has published over 200 papers in peer-reviewed journals and refereed conference proceedings in the areas of wireless sensor networks, high-performance computer architecture, peer-to-peer computing, and performance evaluation. He is a member of the IEEE Computer Society. He has served on technical program committees of numerous international conferences.

Supplemental Material for "Adaptive Diffusion Of Sensitive Information In Online Social Networks"

Xudong Wu, Luoyi Fu, Huan Long, Dali Yang, Yucheng Lu, Xinbing Wang and Guihai Chen
Shanghai Jiao Tong University
{xudongwu, yiluofu, longhuan, yangdali, eugene_lu, xwang8}@sjtu.edu.cn, gchen@cs.sjtu.edu.cn

Appendix A
Proof for Lemma 1

Lemma 1. Given the sensitive node set V^1 and the target edge set $\mathfrak{E}^1$ at the beginning of the 1-st round, the expected size of sensitive nodes until the end of the T rounds is upper bounded by $M_T = |V^1| + |\mathfrak{E}^1| w_{max} 2T + \frac{n-|V^1|}{4} w_{max}(2T+1)T$.

Proof. Throughout the proof, we characterize the size of sensitive nodes during diffusion via hops. Specifically, there are two hops during each round and totally $2T$ hops during the T rounds. We use $S(t)$ to denote the set of the sensitive nodes after the diffusion in the t-th hop, and use $N(S(t))$ to denote the set of non-sensitive neighbors of the nodes in $S(t)$. Then, we use $\partial S(t)$ to denote the set of edges connecting the nodes in $S(t)$ and $N(S(t))$, and use $\partial(S(t), j)$ to denote the set of edges that are in the set $\partial S(t)$ and connect the destination node v_j.

With the symbols defined above, assuming that all the edges whose destination nodes are a same node v_j share the same original diffusion probability w_j, for the size of the set $\partial S(t)$, we have

$$\begin{aligned}&\mathbb{E}[|\partial S(t+1)| - |\partial S(t)|]\\ \leq &\sum_{v_j \in N(S(t))} [1-(1-w_j)^{|\partial(S(t),j)|}](d_j - |(\partial(S(t),j)|),\end{aligned}$$

where d_j is the degree of node v_j. Throughout the proof, we assume that the out-degree of a node is equal to its in-degree. Given $w_j \leq w_{max} \ll 1$, with approximation, we have

$$\begin{aligned}&1-(1-w_j)^{|(\mathfrak{E}^t,j)|}\\ =&1-[1-w_j|(\mathfrak{E}^t,j)| + o(w_j)] \approx w_j|(\mathfrak{E}^t,j)|\end{aligned}$$

Then

$$\begin{aligned}&\mathbb{E}[|\partial S(t+1)| - |\partial S(t)|]\\ \approx &\sum_{v_j \in N(S(t))} w_j|(\partial(S(t),j)|(d_j - |(\partial(S(t),j)|).\end{aligned}$$

Suppose that the social network G has a power-law degree distribution. Then, $|(\partial(S(t),j)|$ can be rewritten as

$$|(\partial(S(t),j)| = \sum_{v_i \in S(t)} \frac{d_i}{\sum_{v_m \in V} d_m} \cdot d_j.$$

At last, we have

$$\begin{aligned}&\mathbb{E}[|\partial S(t+1)| - |\partial S(t)|]\\ \leq &\sum_{v_j \in N(S(t))} w_j {d_j}^2 \left[\sum_{v_i \in S(t)} \frac{d_i}{\sum_{v_m \in V} d_m}\left(1 - \sum_{v_i \in S(t)} \frac{d_i}{\sum_{v_m \in V} d_m}\right)\right]\\ < &\left(\frac{\sum_{v_i \in S(t)} \frac{d_i}{\sum_{v_m \in V} d_m} + (1 - \sum_{v_i \in S(t)} \frac{d_i}{\sum_{v_m \in V} d_m})}{2}\right)^2 \cdot\\ &\sum_{v_j \in N(S(t))} {d_j}^2 w_{max}\\ <&\frac{1}{4}|N(S(t))|,\end{aligned}$$

when w_{max} is sufficiently small. Provided that $S(t) \cup N(S(t)) \subseteq V$ and $S(t) \cap N(S(t)) = \emptyset$, we further have

$$|\partial S(t+1)| - |\partial S(t)| < \frac{1}{4}(N - |S(t)|)$$

Furthermore, the expected size of nodes that become sensitive in each hop is also upper bouned by the diffusion probability w_{max} and the value of $|\partial S(t)|$, then

$$\sum_{t=1}^{2T} |S(t+1)| - |S(t)| < \sum_{t=1}^{2T} |\partial S(t)| w_{max}.$$

For the initial value of $|S(t)|$ and $|\partial S(t)|$, we have $|S(0)| = |V^1|$ and $|\partial S(0)| \leq |\mathfrak{E}^1|$. Consequently,

$$\begin{aligned}|S(2T)| &< |V^1| + w_{max} \sum_{t=1}^{2T} |\partial S(t)|\\ &< |V^1| + w_{max} \sum_{t=1}^{2T} \left[|\mathfrak{E}^1| + \frac{(N-|V^1|)}{4} \cdot t\right]\\ &= |V^1| + |\mathfrak{E}^1| w_{max} 2T + \frac{N-|V^1|}{4} w_{max} T(2T+1).\end{aligned}$$

Thus we obtain the conclusion in Lemma 1. □

Appendix B
Proofs in Regret Analysis in Section 4.3

B.1 Proof for Theorem 1

According to Section 4.3, the expected regret of **ADSN** in the t-th round is given by

$$\mathbb{E}[\overrightarrow{D^t} \cdot \overrightarrow{\Delta\beta}^t - \overrightarrow{D^t}\overrightarrow{\Delta\beta}^{t*}]$$

$$=\varepsilon_t\mathbb{E}[\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^t_{ep}-\overrightarrow{D^t}\overrightarrow{\Delta\beta}^{t*}]+(1-\varepsilon_t)\mathbb{E}[\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^t_{et}-\overrightarrow{D^t}\overrightarrow{\Delta\beta}^{t*}].$$

Before we proceed to analyze the expected regret of **ADSN** over the T rounds, let us give a few properties that we will use later. (1): Given $\mu_{i,t}$ being the estimated mean reward of base-arm $\overrightarrow{\beta_i}$ at round t, according to the Chernoff-Hoeffding Bound [1], we have

$$\mathcal{P}\left(|\overrightarrow{D^t}\cdot\overrightarrow{\beta}_i-\mu_{i,t}|>\delta\right)\leq 2e^{-\frac{T_{i,t}\delta^2}{2\sigma^2}},$$

where $T_{i,t}$ is the times that base-arm $\overrightarrow{\beta}_i$ has been picked until the t-th round. For the base-arms with exact mean rewards, we have $\overrightarrow{D^t}\cdot\overrightarrow{\beta}_i-\mu_{i,t}=0$. (2): The base-arms chosen in exploitation satisfy: $\mathbb{E}\left[\sum_{\overrightarrow{\beta_i}\in\mathcal{S}_t}\mu_{i,t}\right]\leq\mathbb{E}\left[\sum_{\overrightarrow{\beta_i}\in\mathcal{S}^*_t}\mu_{i,t}\right]$, where $\mathcal{S}_t$ denotes the output *combination* of Algorithm 4 in the main text, and $\mathcal{S}^*_t$ is the set of base-arms in the optimal *combination*. Without loss of generality, we sort the elements in vector $\overrightarrow{D^t}$ and let $B^t_0\triangleq\sum_{j=1}^{|\mathfrak{E}^t|}\overrightarrow{\beta}^t_0(j)$, $B^t_*\triangleq\sum_{j=1}^{B^t_0}\overrightarrow{D^t}(j)-\overrightarrow{D^t}\cdot\overrightarrow{\beta}^t_0$ and $B^t_\times\triangleq\sum_{j=|\mathfrak{E}^t|-B^t_0+1}^{|\mathfrak{E}^t|}\overrightarrow{D^t}(j)-\sum_{j=1}^{B^t_0}\overrightarrow{D^t}(j)$.

Lemma B.1. *In the t-th round, for any valid combination $\overrightarrow{\Delta\beta^t}$, we have $\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta^t}-\overrightarrow{D^t}\cdot\overrightarrow{\beta}^t_0\geq B^t_*$.*

Proof. In the t-th round, let vector $\overrightarrow{y}^*$ be

$$\overrightarrow{y}^*(j)\triangleq\begin{cases}1, & 1\leq j\leq B^t_0\\ 0, & B^t_0<j\leq|\mathfrak{E}^t|\end{cases}$$

then

$$\sum_{j=1}^{|\mathfrak{E}^t|}(\overrightarrow{\Delta\beta^t}(j)-\overrightarrow{y}^*(j))=0$$

For $1\leq j\leq B^t_0$, $\overrightarrow{\Delta\beta^t}(j)-\overrightarrow{y}^*(j)\leq 0$. Similarly, for $B^t_0<j\leq|\mathfrak{E}^t|$, $\overrightarrow{\Delta\beta^t}(j)-\overrightarrow{y}^*(j)\geq 0$. Then

$$\begin{aligned}&\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta^t}-\overrightarrow{D^t}\cdot\overrightarrow{y}^*\\ =&\sum_{j=1}^{B^t_0}\overrightarrow{D^t}(j)(\overrightarrow{\Delta\beta^t}(j)-\overrightarrow{y}^*(j))+\sum_{j=B^t_0+1}^{|\mathfrak{E}^t|}\overrightarrow{D^t}(j)(\overrightarrow{\Delta\beta}(j)-\overrightarrow{y}^*(j))\\ \geq&\overrightarrow{D^t}(B_0)\sum_{j=1}^{B^t_0}(\overrightarrow{\Delta\beta}(j)-\overrightarrow{y}^*(j))\\ +&\overrightarrow{D^t}(B^t_0+1)\sum_{j=B^t_0+1}^{|\mathfrak{E}^t|}(\overrightarrow{\beta^t}(j)-\overrightarrow{y}^*(j))\\ \geq&\left[\overrightarrow{D^t}(B^t_0+1)-\overrightarrow{D^t}(B^t_0)\right]\sum_{j=B^t_0+1}^{|\mathfrak{E}^t|}(\overrightarrow{\Delta\beta^t}(j)-\overrightarrow{y}^*(j))\geq 0\end{aligned}$$

Finally, we have

$$\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta^t}-\overrightarrow{D^t}\cdot\overrightarrow{\beta^t}_0\geq\overrightarrow{D^t}\cdot\overrightarrow{y}^*-\overrightarrow{D^t}\cdot\overrightarrow{\beta^t}_0=B^t_*$$

□

Lemma B.2. *In the t-th round, for any two valid combinations $\overrightarrow{\Delta\beta^t}_1,\overrightarrow{\Delta\beta^t}_2$, we have $\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta^t}_1-\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta^t}_2\leq B^t_\times$.*

Proof. Similar to Lemma B.1, let

$$\overrightarrow{z}^*(j)\triangleq\begin{cases}1, & 1\leq j\leq|\mathfrak{E}^t|-B^t_0\\ 0, & |\mathfrak{E}^t|-B^t_0<j\leq|\mathfrak{E}^t|\end{cases}$$

and, since $\sum_{j=1}^{|\mathfrak{E}^t|}\Delta\beta^t(j)=\sum_{j=1}^{|\mathfrak{E}^t|}\beta^t_0(j)$, then

$$\forall\overrightarrow{\Delta\beta^t}:\sum_{j=1}^{|\mathfrak{E}^t|}\Delta\beta^t(j)=B^t_0,\quad\overrightarrow{D^t}\overrightarrow{\Delta\beta^t}\geq\sum_{j=|\mathfrak{E}^t|-B^t_0+1}^{|\mathfrak{E}^t|}\overrightarrow{D^t}(j)$$

we have

$$\overrightarrow{D}^t\cdot\overrightarrow{\Delta\beta^t_1}-\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta^t_2}\leq\sum_{j=1}^{B^t_0}\overrightarrow{D^t}(j)-\sum_{j=|\mathfrak{E}^t|-B^t_0+1}^{|\mathfrak{E}^t|}\overrightarrow{D^t}(j)=B^t_\times$$

□

Lemma B.1 gives the bound of the value of $\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta^t}$, which denotes the mean value of the reward of any super-arm. At the same time, Lemma B.2 gives the bound of the gap between the values of $\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta^t}$ of any two super-arms.

Now, we move to derive the bound of the expected regret in the t-th round. We first explore the bound of the regret in the Exploitation, i.e., $\mathbb{E}[\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^t_{et}-\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^{t*}]$. Let $\mathcal{F}_t$ denote the event $\left\{|\overrightarrow{D^t}\cdot\overrightarrow{\beta}_i-\mu_{i,t}|\leq\frac{c\sigma}{\sqrt{T_{i,t}}},\quad\forall\overrightarrow{\beta^t}_i\in\mathcal{V}^t\right\}$, where $\mathcal{V}^t$ is the base-arm set in the t-th round and c is a constant. We let a constant $M^t=|\mathcal{V}^t|$. Based on Lemmas B.1 and B.2, Lemma B.3 gives the expected regret in the Exploitation in the t-round.

Lemma B.3. *For the regret in the the Exploitation in the t-th round, i.e., $\mathbb{E}[\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^t_{et}-\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^{t*}]$, we have*

$$\mathbb{E}[\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^t_{et}-\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^{t*}]\leq c\sigma\sum_{\overrightarrow{\beta_i}\in\mathcal{S}_t}\frac{1}{\sqrt{T_{i,t}}}+B^t_\times\mathcal{P}(\overline{\mathcal{F}_t}).\quad(1)$$

Proof. Decomposing the regret $\mathbb{E}[\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^t_{et}-\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^{t*}]$ by event $\mathcal{F}_t$, we have

$$\begin{aligned}\mathbb{E}[\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^t_{et}-\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^{t*}]=&\mathbb{E}[\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^t_{et}-\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^{t*}|\mathcal{F}_t]\mathcal{P}(\mathcal{F}_t)\\ &+\mathbb{E}[\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^t_{et}-\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^{t*}|\overline{\mathcal{F}_t}]\mathcal{P}(\overline{\mathcal{F}_t})\end{aligned}\quad(2)$$

For the term $\mathbb{E}[\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^t_{et}-\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^{t*}|\mathcal{F}_t]\mathcal{P}(\mathcal{F}_t)$, we have

$$\begin{aligned}&\mathbb{E}[\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^t_{et}-\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^{t*}|\mathcal{F}_t]\\ \leq&\mathbb{E}\left[\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^t_{et}-\sum_{\overrightarrow{\beta_i}\in\mathcal{S}_t}\mu_{i,t}|\mathcal{F}_t\right]+\mathbb{E}\left[\sum_{\overrightarrow{\beta_i}\in\mathcal{S}_t}\mu_{i,t}|\mathcal{F}_t\right]-\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^{t*}\\ \overset{(i)}{\leq}&\mathbb{E}\left[\sum_{\overrightarrow{\beta_i}\in\mathcal{S}_t}\left|\overrightarrow{D^t}\cdot\overrightarrow{\beta}_i-\mu_{i,t}\right|\middle|\mathcal{F}_t\right]+\left(\mathbb{E}\left[\sum_{i\in\mathcal{S}^*_t}\mu_{i,t}\middle|\mathcal{F}_t\right]-\overrightarrow{D^t}\overrightarrow{\Delta\beta}^{t*}\right)\\ \overset{(ii)}{\leq}&c\sigma\sum_{i\in\mathcal{S}_t}\frac{1}{\sqrt{T_{i,t}}}\end{aligned}\quad(3)$$

The rationale of the Inequality (i) above is that as Exploitation selects the set $\mathcal{S}_t$ of base-arms instead of the set $\mathcal{S}^*_t$ for minimizing the reward, there must be $\mathbb{E}\left[\sum_{i\in\mathcal{S}^*_t}\mu_{i,t}\middle|\mathcal{F}_t\right]\geq\mathbb{E}\left[\sum_{i\in\mathcal{S}_t}\mu_{i,t}|\mathcal{F}_t\right]$. In addition, the Inequality (ii) holds since that under the event $\mathcal{F}_t$, we have $\left|\overrightarrow{D^t}\cdot\overrightarrow{\beta}_i-\mu_{i,t}\right|\leq$

$\frac{c\sigma}{\sqrt{T_{i,t}}}$ and $\mathbb{E}\left[\sum_{i\in\mathcal{S}_t}\left|\overrightarrow{D^t}\cdot\overrightarrow{\beta}_i-\mu_{i,t}\right|\middle|\mathcal{F}_t\right]\leq c\sigma\sum_{i\in\mathcal{S}_t}\frac{1}{\sqrt{T_{i,t}}}$. For the term $\left(\mathbb{E}\left[\sum_{i\in\mathcal{S}_t^*}\mu_{i,t}\middle|\mathcal{F}_t\right]-\overrightarrow{D^t}\overrightarrow{\Delta\beta}^{t*}\right)$, since $\overrightarrow{D^t}\overrightarrow{\Delta\beta}^{t*}$ represents the minimum reward obtained from the optimal super-arm, then $\mathbb{E}\left[\sum_{i\in\mathcal{S}_t^*}\mu_{i,t}\middle|\mathcal{F}_t\right]\geq\overrightarrow{D^t}\overrightarrow{\Delta\beta}^{t*}$ and $\left(\mathbb{E}\left[\sum_{i\in\mathcal{S}_t^*}\mu_{i,t}\middle|\mathcal{F}_t\right]-\overrightarrow{D^t}\overrightarrow{\Delta\beta}^{t*}\right)\geq 0$.

Furthermore, we move to the term $\mathbb{E}[\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^t_{et}-\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^{t*}|\overline{\mathcal{F}_t}]\mathcal{P}(\overline{\mathcal{F}_t})$ in Eqn (2). According to Lemma B.2 that $\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^t_{et}-\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^{t*}|\overline{\mathcal{F}_t}\leq B^t_\times$ always holds, we have

$$\mathbb{E}[\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^t_{et}-\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^{t*}|\overline{\mathcal{F}_t}]\mathcal{P}(\overline{\mathcal{F}_t})\leq B^t_\times\mathcal{P}(\overline{\mathcal{F}_t}). \quad (4)$$

Together with Inequalities (3) and (4), we finish the proof for Lemma B.3. □

Upon Lemma B.3 giving the expected regret in the Exploitation in the t-th round, we continue to explore the sum of the regret in the Exploitation overall all the T rounds.

Lemma B.4. *For the regret in the the Exploitation in all the T rounds, we have*

$$\mathbb{E}\left[\sum_{t=1}^T\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^t_{et}-\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^{t*}\right]\leq 2c\sigma M'\sqrt{T}+1, \quad (5)$$

where $M'=\max\{M^1,\ldots,M^T\}$ and .

Proof. With the regret in the Exploitation in the t-th round given in Eqn. (1), for the regret over T rounds, we have

$$\begin{aligned}&\mathbb{E}\left[\sum_{t=1}^T\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^t_{et}-\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^{t*}\right]\\ \leq& c\sigma\sum_{t=1}^T\sum_{i\in\mathcal{S}_t}\frac{1}{\sqrt{T_{i,t}}}+B'_\times\sum_{t=1}\mathcal{P}(\overline{\mathcal{F}_t})\\ \leq& c\sigma\sum_{\beta_i\in\mathcal{V}^t}\sum_{l=1}^{T_{i,T}-1}\frac{1}{\sqrt{l}}+B'_\times\sum_{t=1}^T\mathcal{P}(\overline{\mathcal{F}_t})\\ \overset{(i)}{\leq}& 2c\sigma M'\sqrt{T}+B'_\times\sum_{t=1}^T\mathcal{P}(\overline{\mathcal{F}_t}),\end{aligned} \quad (6)$$

where $B'_\times=\max\{B^1_\times,\ldots,B^T_\times\}$. The Inequality (i) above holds because $T_{i,T}-1\leq T$ and $\sum_{l=1}^T\frac{1}{\sqrt{l}}\leq 2\sqrt{T}$.

For the term $B'_\times\sum_{t=1}^T\mathcal{P}(\overline{\mathcal{F}_t})$, let $c\geq\sqrt{2\ln 2B'_\times M'T}$, we have

$$B'_\times\sum_{t=1}^T\mathcal{P}(\overline{\mathcal{F}_t})\leq B'_\times M'T\cdot 2e^{-\frac{c^2}{2}}\leq 1. \quad (7)$$

Together with Inequalities (6) and (7), we obtain the bound of the regret in Exploitation in Lemma B.4. □

On the other hand, the regret in Exploration is given by $\sum_{t=1}^T\varepsilon_t\left(\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^t_{ep}-\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^{t*}\right)$.

Combining the expected regret in the Exploitation and Exploration, we can obtain the upper bound of the regret of **ADSN**, say $\mathbb{E}[R_{\mathbf{ADSN}}]$ as

$$\begin{aligned}&\mathbb{E}[R_{\mathbf{ADSN}}]\leq\\ &\sum_{t=1}^T\left(\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^t_{et}-\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^{t*}\right)+\sum_{t=1}^T\varepsilon_t\left(\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^t_{ep}-\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^{t*}\right)\\ &\overset{(i)}{\leq} 2c\sigma M'\sqrt{T}+1+B'_\times\sum_{t=1}^T\varepsilon_t\\ &\overset{(ii)}{\leq} 2c\sigma M'\sqrt{T}+1+2B'_\times\sqrt{T}.\end{aligned} \quad (8)$$

The Inequality (i) holds with Lemma B.4, and the Inequality (ii) holds with $\varepsilon_t=\frac{1}{\sqrt{t}}$ and $\sum_{t=1}^T\frac{1}{\sqrt{t}}\leq 2\sqrt{T}$. Since $M'=O(|\mathfrak{E}^T|^2)\geq B'_\times$, we have

$$\mathbb{E}[R_{\mathbf{ADSN}}]\leq O(M'\sqrt{T}). \quad (9)$$

Thus we finish the proof for Theorem 1.

B.2 Superiority of ADSN on Regret

We further prove **ADSN** has superior performance on the regret comparing with the alternative algorithm **CUCB** **CUCB** selects the super-arm based on the estimated means and variances of base-arm rewards. Corresponding to the adaptive diffusion problem which aims at minimizing the reward, **CUCB** selects the *combination* with the minimum sum of $\mu_{i,t}-\frac{c\sigma}{\sqrt{T_{i,t}}}$. Next, we are going to present that **CUCB** has twice regret bound comparing with **ADSN**.

For the regret in the t-th round, we have

$$\begin{aligned}&\mathbb{E}[\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^t-\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^{t*}]=\mathbb{E}[\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^t-\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^{t*}|\mathcal{F}_t]P(\mathcal{F}_t)\\ &+\mathbb{E}[\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^t-\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^{t*}|\overline{\mathcal{F}_t}]P(\overline{\mathcal{F}_t})\end{aligned} \quad (10)$$

For $\mathbb{E}[\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^t-\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^{t*}|\mathcal{F}_t]$, we have

$$\begin{aligned}&\mathbb{E}[\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^t-\overrightarrow{D^t}\overrightarrow{\Delta\beta}^{t*}|\mathcal{F}_t]\\ \leq&\mathbb{E}\left[\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^t-\sum_{\overrightarrow{\beta_i}\in\mathcal{S}_t}\left(\mu_{i,t}-\frac{c\sigma}{\sqrt{T_{i,t}}}\right)|\mathcal{F}_t\right]\\ +&\mathbb{E}\left[\sum_{\overrightarrow{\beta_i}\in\mathcal{S}_t}\left(\mu_{i,t}-\frac{c\sigma}{\sqrt{T_{i,t}}}\right)\middle|\mathcal{F}_t\right]-\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^{t*}\\ \leq&\mathbb{E}\left[\sum_{\overrightarrow{\beta_i}\in\mathcal{S}_t}\left|\overrightarrow{D^t}\cdot\overrightarrow{\beta}_i-\left(\mu_{i,t}-\frac{c\sigma}{\sqrt{T_{i,t}}}\right)\right|\middle|\mathcal{F}_t\right]\\ +&\left(\mathbb{E}\left[\sum_{\overrightarrow{\beta_i}\in\mathcal{S}_t^*}\left(\mu_{i,t}-\frac{c\sigma}{\sqrt{T_{i,t}}}\right)\middle|\mathcal{F}_t\right]-\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^*\right)\leq 2c\sigma\sum_{\overrightarrow{\beta_i}\in\mathcal{S}_t}\frac{1}{\sqrt{T_{i,t}}}\end{aligned} \quad (11)$$

The above inequalities hold because, under event $\mathcal{F}_t$, we have

$$\mu_{i,t}-\frac{c\sigma}{\sqrt{T_{i,t}}}\leq\overrightarrow{D^t}\cdot\overrightarrow{\beta}_i.$$

Here c and $\mathcal{F}_t$ follow the same definitions as those in **ADSN**. Then the expected regret of **CUCB** in the t-round is equal to

$$\mathbb{E}[\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^t-\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^{t*}]\leq 2c\sigma\sum_{\overrightarrow{\beta_i}\in\mathcal{S}_t}\frac{1}{\sqrt{T_{i,t}}}+B^t_\times\mathcal{P}(\overline{\mathcal{F}_t})$$

Then the regret bound $\mathbb{E}[R_{\mathbf{CUCB}}]$ of **CUCB** is given by

$$\begin{aligned}&\mathbb{E}[R_{\mathbf{CUCB}}]=\mathbb{E}\left[\sum_{t=1}^T\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^t-\overrightarrow{D^t}\cdot\overrightarrow{\Delta\beta}^{t*}\right]\\ \leq& 2c\sigma\sum_{t=1}^T\sum_{i\in\mathcal{S}_t}\frac{1}{\sqrt{T_{i,t}}}+B^t_\times\sum_{t=1}\mathcal{P}(\overline{\mathcal{F}_t})\leq 4c\sigma M'\sqrt{T}+1.\end{aligned} \quad (12)$$

The parameter M' in the Eqns. (8) and (12) refers the size of the base-arm set in the T-th round. Then we have $M' = \Theta(|\mathfrak{E}^T|^2) = \Theta(m^2)$. On the other hand, $B'_\times \leq \sum_{j=1}^{|\mathfrak{E}^T|} \overrightarrow{D^T}(j) = \Theta(m)$. Then $\mathbb{E}[R_{\textbf{CUCB}}] = 4c\sigma M'\sqrt{T} + 1 \approx 4c\sigma M'\sqrt{T}$ and $\mathbb{E}[R_{\textbf{ADSN}}] = 2c\sigma M'\sqrt{T} + 2B'_\times\sqrt{T} + 1 \approx 2c\sigma M'\sqrt{T}$. Comparing to $\mathbb{E}[R_{\textbf{ADSN}}]$, we can see that **ADSN** has half of the regret bound of **CUCB** when T is limited. The reason of the larger regret lies on the rationale of **CUCB** that estimated the reward of base-arms as $\mu_{i,t} - \frac{c\sigma}{\sqrt{T_{i,t}}}$ and then determining the value of super-arm by a valid *combination* with the minimum sum of $\mu_{i,t} - \frac{c\sigma}{\sqrt{T_{i,t}}}$. As the base-arms, which have never been picked before or just been picked for few times, have small value of $T_{i,t}$ and thus have the smaller value of $\mu_{i,t} - \frac{c\sigma}{\sqrt{T_{i,t}}}$, **CUCB** actually tends to pick the base-arms with unstable estimated rewards. That is **CUCB** prefers Exploration during all the T rounds in the adaptive diffusion problem and thus obtains unstable rewards. On the contrast, **ADSN** limits Exploration rounds through the decreasing probability ε_t, and has a comparatively more stable performance on getting smaller reward.

Appendix C
Procedures In Distributed Implementation of ADFN and ADSN

Here, we present the procedures for the master and slaves in the distributed implementation of **ADFN** in Algorithms 1 and 2, and present the procedures for the master and salves of **ADSN** in Algorithms 3 and 4.

Algorithm 1: Distributed ADFN–Master

```
Input: All the base-arms, number of rounds T
Output: Combination
for t=1 to T do
    Combination ← ∅;
    Conducting Distributed ADFN–Slave for all salves;
    for local_combination of each slave do
        insert_Buffer(local_combination);
    end
    ActionPool ← Buffer;
    while ActionPool ≠ ∅ do
        v =MIN(ActionPool);
        if μ_{v,t} > 0 then
            End While ;
        end
        ActionPool ← ActionPool\{v};
        if VALID(Combination, v) then
            Combination ← Combination ∪ {v};
        end
    end
end
return Combination
```

Algorithm 2: Distributed ADFN–Slave

```
ActionPool ← base-arms in local_memory;
local_combination ← ∅;
while ActionPool ≠ ∅ do
    v =MIN(ActionPool);
    if μ_{v,t} > 0 then
        End While ;
    end
    ActionPool ← ActionPool\{v};
    if VALID(local_combination, v) then
        local_combination ← local_combination∪{v};
    end
end
return local_combination
```

Algorithm 3: Distributed ADSN–Master

```
Input: All the base-arms, ε_0, number of rounds T
Output: Combination
for t=1 to T do
    ε_t ← ε_0/√t;
    if ε_t then
        send(master, selected slave, any base-arm);
        recv(selected slave, master, selected base-arm);
        u ← selected base-arm;
        Combination ← u;
        send(master, slaves, Exploration, u);
        Conducting Distributed BLAG–Slave for all
          salves;
        for local_combination of each slave do
            insert_Buffer(local_combination);
        end
        ActionPool = Buffer;
        for v in ActionPool do
            if VALID(Combination, v) then
                Combination ← Combination ∪ {v};
            end
        end
    else
        send(master, slaves, Exploitation);
        Combination ← ∅;
        Conducting Distributed BLAG–Slave for all
          salves;
        for local_combination of each slave do
            insert_Buffer(local_combination);
        end
        ActionPool = Buffer;
        while ActionPool ≠ ∅ do
            v =MIN(ActionPool);
            ActionPool ← ActionPool\{v};
            if VALID(Combination, v) then
                Combination ← Combination ∪ {v};
            end
        end
    end
end
return combination
```

References

[1] D. Pollard, " Convergence of stochastic processes", pringer Science & Business Media, 2012.

Algorithm 4: Distributed ADSN–Slave

```
recv(slave, master, order);
if order == Exploration then
    local_combination ← u;
    ActionPool ← base-arms in local_memory;
    for v in ActionPool do
        if VALID(Combination, v) then
            local_combination ←
              local_combination ∪ {v};
        end
    end
else
    ActionPool ← base-arms in local_memory;
    local_combination ← ∅;
    while ActionPool ≠ ∅ do
        v =MIN(ActionPool);
        if μ_{v,t} > 0 then
            End While ;
        end
        ActionPool ← ActionPool\{v};
        if VALID(local_combination, v) then
            local_combination ←
              local_combination ∪ {v};
        end
    end
end
return local_combination
//learning phase
for β⃗_i in Combination do
    if β⃗_i ∈ local_memory then
        T_{i,t} ← T_{i,t−1} + 1;
        μ_{i,t} ← [(t − 1) × μ_{i,t−1} + reward(i)]/T_{i,t};
        Update μ_{i,t} in look_up_table;
    end
end
```