Online Contextual Influence Maximization With Costly Observations

Anıl Ömer Sarıtaç, Altuğ Karakurt, and Cem Tekin, *Member, IEEE*

***Abstract*—In the *online contextual influence maximization problem with costly observations*, the learner faces a series of epochs in each of which a different *influence spread process* takes place over a network. At the beginning of each epoch, the learner exogenously influences (activates) a set of seed nodes in the network. Then, the influence spread process takes place over the network, through which other nodes get influenced. The learner has the option to observe the spread of influence by paying an *observation cost*. The goal of the learner is to maximize its cumulative reward, which is defined as the expected total number of influenced nodes over all epochs minus the observation costs. We depart from the prior work in three aspects: 1) the learner does not know how the influence spreads over the network, i.e., it is unaware of the influence probabilities; 2) influence probabilities depend on the context; and 3) observing influence is costly. We consider two different influence observation settings: costly edge-level feedback, in which the learner freely observes the set of influenced nodes, but pays to observe the influence outcomes on the edges of the network; and costly node-level feedback, in which the learner pays to observe whether a node is influenced or not. Since the offline influence maximization problem itself is NP-hard, for these settings, we develop online learning algorithms that use an approximation algorithm as a subroutine to obtain the set of seed nodes in each epoch. When the influence probabilities are Hölder continuous functions of the context, we prove that these algorithms achieve sublinear regret (for any sequence of contexts) with respect to an approximation oracle that knows the influence probabilities for all contexts. Our numerical results on several networks illustrate that the proposed algorithms perform on par with the state-of-the-art methods even when the observations are cost free.**

***Index Terms*—Influence maximization, combinatorial bandits, social networks, approximation algorithms, costly observations, regret bounds.**

I. Introduction

In recent years, there has been growing interest in understanding how influence spreads in a social network [2]–[7]. This interest is motivated by the proliferation of viral marketing in social networks. For instance, nowadays many companies promote their products on social networks by giving free samples of certain products to a set of seed nodes/users, expecting them to influence people in their social circles into purchasing these products. The objective of these companies is to find out the set of nodes that can collectively influence the greatest number of other nodes in the social network. This problem is called the *influence maximization* (IM) problem.

In the IM problem, the spread of influence is modeled by an *influence graph*, where directed edges between nodes represent the paths that the influence can propagate through and the weights on the directed edges represent the likelihood of the influence propagation, i.e., the influence probability. Numerous models are proposed to model the spread of influence, with the most popular ones being *independent cascade* (IC) and *linear threshold* (LT) models [8]. In the IC model, the influence propagates on each edge independently from the other edges of the network, and an influenced node has only a single chance to influence its neighbors. Hence, only recently influenced nodes can propagate the influence. Thus, the influence stops to spread when the recently influenced nodes fail to influence their neighbors. On the other hand, in the LT model, a node's chance to get influenced depends on whether the sum of weights of its active neighbors exceeds a threshold or not.

Most of the prior work in IM assume that the influence probabilities of the influence graph are known and that the influence spread process is observed [9]–[14], and focus on designing computationally efficient algorithms to maximize the influence spread. However, in many practical settings, it is impossible to know the exact influence probabilities beforehand. For instance, a firm that wants to introduce a new product or to advertise its existing products in a new social network may not know the influence probabilities on the edges of the network. In contrast to the prior works mentioned above, our focus is to design an optimal learning strategy when the influence probabilities are unknown.

In the marketing example given above, influence depends on the product that is being advertised as well as the identities of the users. Hence, the characteristic (context) of the product affects the influence probabilities. The strand of literature that is closest to the problem we consider in this paper in terms of the dependence of the influence probabilities on the context is called *topic-aware IM* [4]–[7]. To the best of our knowledge, none of the prior works in topic-aware IM develop learning algorithms with provable performance guarantees for the case when the

Manuscript received January 9, 2018; revised April 29, 2018 and August 8, 2018; accepted August 8, 2018. Date of publication August 19, 2018; date of current version May 8, 2019. This paper was presented in part at Allerton 2016 [1]. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. M. Rabbat. *(Corresponding author: Cem Tekin.)*

A. Ömer Sarıtaç is with the Department of Industrial Engineering, Bilkent University, Ankara 06800, Turkey (e-mail: omer.saritac@bilkent.edu.tr).

A. Karakurt was the Department of Electrical and Electronics Engineering, Bilkent University, Ankara 06800, Turkey. He is now with the Department of Electrical and Computer Engineering, The Ohio State University, Columbus, OH 43210 USA (e-mail: karakurt.1@osu.edu).

C. Tekin is with the Department of Electrical and Electronics Engineering, Bilkent University, Ankara 06800, Turkey (e-mail: cemtekin@ee.bilkent.edu.tr).

Digital Object Identifier 10.1109/TSIPN.2018.2866334

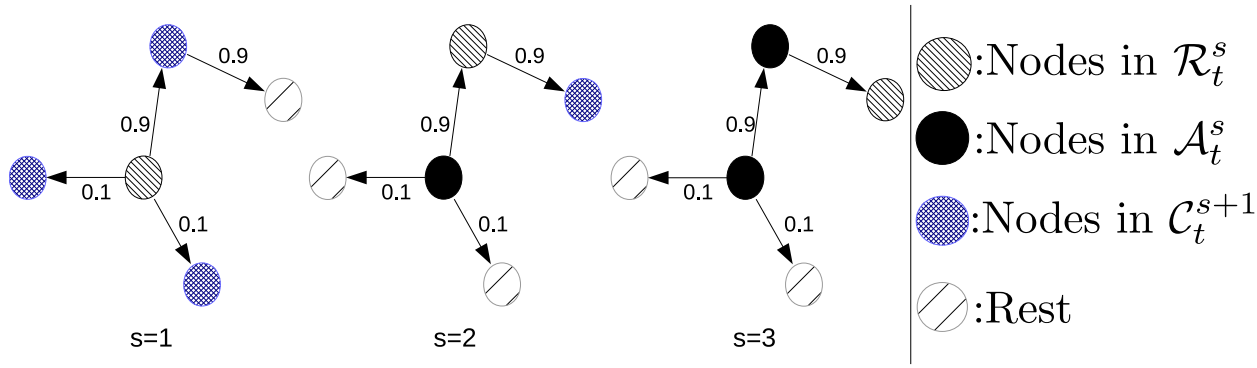

Fig. 1. An illustration that shows the influence spread process for $k = 1$ and time slots $s = 1, 2, 3$ in epoch t. Numbers on the edges denote the influence probabilities. For this example, the influence spread in epoch t is 2, and the expected influence spread is $0.1 + 0.1 + 0.9 + 0.9 * 0.9 = 1.91$. $\mathcal{R}_t^s$ denotes the set of nodes influenced in time slot s of epoch t. $\mathcal{R}_t^1$ denotes the seed node. $\mathcal{A}_t^s$ denotes the set of nodes influenced prior to time slot s of epoch t. $\mathcal{C}_t^{s+1}$ denotes the set of nodes that might be influenced in time slot $s + 1$ of epoch t.

influence probabilities are unknown. In addition, prior works in IM that consider unknown influence probabilities do not consider the cost of feedback (cost of observation of the influence spread process) [1], [15]–[18]. However, this cost exists in most of the real-world applications of IM. For instance, finding out who influenced a specific person into buying a product might require conducting a costly investigation (e.g., a survey).

Motivated by such real-world applications, in this paper, we define a new learning model for IM, called the *Online Contextual Influence Maximization Problem with Costly Observations* (OCIMP-CO). In contrast to IM, which is a single-shot problem, OCIMP-CO is a sequential decision making problem. In OCIMP-CO, the learner (e.g., the firm in the above example), faces a series of epochs in each of which a different influence campaign is run. At the beginning of each epoch, the learner observes the context of that epoch. For instance, the context can be the type of the influence campaign (e.g., one influence campaign might promote a sports equipment, while another influence campaign might promote a mobile data plan). After observing the context, the learner chooses a set of k seed nodes to influence. We call these nodes *exogenously influenced nodes*. Then, the influence spreads according to the IC model, which is explained in detail in Section III-A. The nodes that are influenced as a result of this process are called *endogenously influenced nodes*. An illustration of the influence spread process is given in Fig. 1. At the end of each epoch, the learner obtains as its reward the number of endogenously influenced nodes. The goal of the learner is to maximize its cumulative expected influence spread minus the observation costs over epochs.

In this paper, we consider two different influence observation settings: costly edge-level feedback, in which the learner freely observes the set of influenced nodes, but pays to observe the influence outcomes on the edges of the network; and costly node-level feedback, in which the learner pays to observe whether a node is influenced or not. For the costly edge-level feedback setting we propose a learning algorithm called *Contextual Online INfluence maximization COstly Edge-Level feedback* (COIN-CO-EL) to maximize the learner's reward for any given number of epochs. COIN-CO-EL can use any approximation algorithm for the offline IM problem as a subroutine to obtain the set of seed nodes in each epoch. When the influence probabilities are Hölder continuous functions of the context, we prove that COIN-CO-EL achieves $O(c^{1/3}T^{(2\theta+d)/(3\theta+d)})$ regret (for any sequence of contexts) with respect to an approximation oracle that knows the influence probabilities for all contexts. Here, c represents the observation cost, θ is the exponent of Hölder condition for the influence probabilities, and d represents the dimension of the context. Then, we also propose an algorithm for the costly node-level feedback setting, called *Contextual Online INfluence maximization COstly Node-Level feedback* (COIN-CO-NL), which learns the influence probabilities by performing smart explorations over the influence graph. We also show that COIN-CO-NL enjoys $O(c^{1/3}T^{(2\theta+d)/(3\theta+d)})$ regret. In addition, we prove that for the special case when the influence probabilities do not depend on the context, i.e., the context-free online IM problem with costly observations, our algorithms achieve $O(c^{1/3}T^{2/3})$ regret. We conclude that this bound is tight in terms of the observation cost and the time order by proving that the regret lower bound for this case is $\Omega(c^{1/3}T^{2/3})$.

The contributions are summarized as follows:

- We propose OCIMP-CO, where the influence probabilities depend on the context and are unknown a priori.
- We propose online learning algorithms for both costly edge-level and costly node-level feedback settings, and prove that the proposed algorithms achieve $O\left(c^{1/3}T^{(2\theta+d)/(3\theta+d)}\right)$ regret for any sequence of contexts when the influence probabilities are Hölder continuous functions of the context.
- We show that our algorithms achieve $O(c^{1/3}T^{2/3})$ regret for the context-free online IM problem with costly observations, which is optimal.
- We empirically evaluate performance of our algorithms on several real-world networks, and show that they perform on par with the state-of-the-art methods even when the observations are cost-free.

The rest of this paper is organized as follows. Related work is given in Section II. Problem description and regret definition are given in Section III. The approximation guarantee that an approximation algorithm can provide given a set of estimated

influence probabilities is described in Section IV. The learning algorithms and their regret analyses for the costly edge-level and costly node-level feedback settings are considered in Sections V and VI respectively. Then, several extensions are proposed in Section VII. Detailed experiments on the proposed algorithms and their extensions are carried out in Section VIII. Concluding remarks are given in Section IX.

II. Related Work

A. Influence Maximization

The IM problem was first proposed in [8], where it is proven to be NP-Hard and an approximately optimal solution is given. However, the solution given in [8] does not scale well because it often requires thousands of Monte Carlo samples to estimate the expected influence spread of each seed set. This motivated the development of many heuristic methods with lower computational complexity [10], [14], [19], [20].

In numerous other works, algorithms with approximation guarantees are developed for the IM problem, such as CELF [11], CELF++ [12] and NewGreedy [14]. In addition to these works, in [21], an approximation algorithm based on reverse influence sampling is proposed and its run-time optimality is proven. In [9], the authors improved the scalability of this algorithm by proposing two new algorithms TIM and TIM+. More recently, [13] developed IMM which is an improvement on TIM in terms of efficiency while preserving its theoretical guarantees. None of the works mentioned above consider the context information. IM based on context information is studied in several other works such as [4], [6], [7]. However, in contrast to our work which solves a more general problem, these works assume that the influence probabilities are known and topics/contexts are discrete. Moreover, in OCIMP-CO, context is represented by a collection of continuous features (which can be discretized if necessary). It is also worth mentioning that, to the best of our knowledge, there exists no work that solves the online version of the IM problem where observing the influence spread process is costly.

B. Multi-Armed-Bandits (MAB)

Several recent works use MAB-based methods to solve the IM problem when the influence probabilities are unknown. In these works, as in ours, the set of arms chosen at each epoch corresponds to the seed set of nodes.

For instance, [17] presents a combinatorial MAB problem where multiple arms are chosen at each epoch, and these arms probabilistically trigger the other arms. In our terminology, multiple arms chosen at each epoch correspond to the set of seed nodes and probabilistically triggered arms correspond to nodes other than the set of seed nodes. For this problem, a logarithmic gap-dependent regret bound is proven with respect to an approximation oracle. In a subsequent work, the dependence of the regret on the inverse of the minimum positive arm triggering probability is removed under more stringent assumptions on the reward function [22]. However, the problem in [17] and [22] does not involve any contexts.

Another general MAB model that uses greedy algorithms to solve the IM problem with unknown graph structure and influence probabilities is proposed in [18]. In addition, [23] considers a non-stationary IM problem, in which the influence probabilities are unknown and time varying. OCIMP-CO is more general than this, since the context can also be used to model the time-varying nature of the influence probabilities (for instance, one dimension of the context can be the time).

An online method for the IM problem that uses an *upper confidence bound* (UCB) based and an ϵ-greedy based algorithm is proposed in [24], but theoretical analysis of this method is not carried out. In another related work [15], the IM problem is defined on an undirected graph where the influence probabilities are assumed to be linear functions of the unknown parameters, and a linear UCB-based algorithm is proposed to solve it. The prior works described above assume that the influence outcomes on each edge in the network are observed by the learner. Recently, another observation model, called node-level feedback, is proposed in [16]. This model assumes that only the influenced nodes are observable while the spread of influence over the edges is not. However, no regret analysis is provided for this model.

There also exists another strand of literature that studies contextual MAB and its combinatorial variants under the linear realizability assumption [25]–[27]. This assumption enforces the relation between the expected rewards (also known as scores in combinatorial MAB literature) of the arms and the contexts to take a linear form, which boils down learning to estimating an unknown parameter vector. This enables the development of learning algorithms that can achieve $\tilde{O}(\sqrt{T})$ regret.

While [25] directly models the expected reward of an arm as a linear function of the context, [26] and [27] consider the combinatorial MAB problem where the expected reward of an action is a monotone and Lipschitz continuous function of the expected scores of the arms associated with the action. This model is more restrictive than ours since it forces the arm scores (i.e., the influence probabilities in our setting) to be linear in the context. In contrast, in our work, we only assume that the influence probabilities are Hölder continuous functions of the context (see Assumption 1).

In conclusion, our work differentiates itself by considering context as well as the cost of observation in the online IM problem. The differences between our work and the prior works are summarized in Table I.

III. Problem Description

A. Definition of the Influence

Consider a learner (e.g., a viral marketing engine) operating on a social network with n nodes/users and m edges. The set of nodes is denoted by V and the set of edges is denoted by E. The *network graph* is denoted by $G(V, E)$. The set of children of node i is given by $\mathcal{N}_i := \{j \in V : (i, j) \in E\}$, and the set of parents of node i is given by $\mathcal{V}_i := \{j \in V : (j, i) \in E\}$.

Ads arrive to the learner sequentially over time in discrete epochs, indexed by $t \in \{1, 2, \ldots\}$. Without loss of generality, context of the ad at tth epoch comes from a d-dimensional

TABLE I
COMPARISON OF OUR WORK WITH PRIOR WORKS

	Our Work	[17], [18], [23]	[4]–[7]	[8]–[12], [14], [19]–[21]	[24]
Context	Yes	No	Yes	No	No
Online Learning	Yes	Yes	No	No	Yes
Regret Bound	Yes	Yes	No	No	No
Costly Observation	Yes	No	No	No	No

context set $\mathcal{X} := [0,1]^d$, and is denoted by x_t. The *influence graph* at epoch t is denoted by $G(V, E, p^{x_t})$, where $p^{x_t} := \{p^{x_t}_{i,j}\}_{(i,j)\in E}$ is the set of influence probabilities and $p^{x_t}_{i,j} \in [0,1]$ denotes the probability that node i influences node j when the context is x_t. These influence probabilities are unknown to the learner a priori.

At the beginning of epoch t, the learner exogenously influences $k < n$ nodes in the network. The set of these nodes is denoted by S_t, which is also called the *action* at epoch t. An action is an element of the set of k-element subsets of V, which is denoted by $\mathcal{M}$. Nodes in S_t disperse the ad in their social circles according to the IC model. A node that is a neighbor of an influenced node probabilistically gets influenced if it shares the ad in its social circle. A node that has not been influenced yet is called an *inactive* node, whereas a node that has been influenced is called an *active* node. In the IC model, each epoch consists of a sequence of time slots indexed by $s \in \{1, 2, \ldots\}$. Let $\mathcal{A}^s_t$ denote the set of nodes that are already active at the beginning of time slot s of epoch t, $\mathcal{R}^s_t$ denote the set of nodes that are activated for the first time at time slot s of epoch t, and $\mathcal{C}^s_t$ denote the set of nodes that might be activated at time slot s of epoch t. In the IC model, we have $\mathcal{A}^1_t = \emptyset$, $\mathcal{R}^1_t = S_t$, $\mathcal{A}^{s+1}_t = \mathcal{A}^s_t \cup \mathcal{R}^s_t$ and $\mathcal{C}^{s+1}_t = \{j \in \{\cup_{i\in\mathcal{R}^s_t} \mathcal{N}_i\} - \mathcal{A}^{s+1}_t\}$. For $j \in \mathcal{C}^{s+1}_t$, let $\tilde{\mathcal{V}}^{s+1}_t(j) = \{i \in \mathcal{V}_j \cap \mathcal{R}^s_t\}$ denote the set of nodes in $\mathcal{R}^s_t$ that can influence j. In the IC model, we have

$$\Pr\left(j \in \mathcal{R}^{s+1}_t | j \in \mathcal{C}^{s+1}_t\right) = 1 - \prod_{i\in\tilde{\mathcal{V}}^{s+1}_t(j)} (1 - p^{x_t}_{i,j}). \quad (1)$$

Suppose that the influence spread process started from a seed set S of nodes. We denote the expected number of endogenously influenced nodes (also called the *expected influence spread*) given context $x \in \mathcal{X}$ and action S as $\sigma(x, S)$, where the expectation is taken over the randomness of the influence spread given S.

We assume that similar contexts have similar effects on the influence probabilities. This similarity is formalized in the following assumption.

Assumption 1: There exists $L > 0$, $\theta > 0$ such that for all $(i,j) \in E$ and $x \in \mathcal{X}$, $|p^{x'}_{i,j} - p^{x}_{i,j}| \leq L\|x' - x\|^\theta$, where $\|.\|$ denotes the Euclidean norm in $\mathbb{R}^d$.

Note that when $\theta > 1$, the influence probabilities that satisfy Assumption 1 are constants. Thus, in this degenerate case, the problem reduces to the context-free online IM problem.

B. Definition of the Reward and the Regret

For a given network graph $G(V, E)$, let $\hat{\boldsymbol{p}} = \{\hat{p}^x\}_{x\in\mathcal{X}}$ denote the set of estimated and $\boldsymbol{p} = \{p^x\}_{x\in\mathcal{X}}$ denote the set of true influence probabilities. We define $\hat{\sigma}(x, S)$ as the expected influence spread of action S on $G(V, E, \hat{p}^x)$. For the influence spread process that results from action S, we call an edge $(i,j) \in E$ *activated* if node i influenced node j in this process.

We assume that the learner can (partially) observe the influence spread process by paying an observation cost. In particular, we propose two different influence observation settings:

1) Costly Edge-Level Feedback: In this setting, at the end of each epoch, the learner freely observes the set of influenced nodes, but pays to observe the influence outcomes on the edges of the network. The cost of each observation is fixed and known.

2) Costly Node-Level Feedback: In this setting, at the end of a time slot of an epoch, the learner may pay to observe whether a node is activated or not. The cost of each observation is fixed and known. The set of influenced nodes is not freely revealed to the learner at the end of an epoch.[1]

We will compare the performance of the learner with the performance of an oracle that knows the influence probabilities perfectly. For this, we define below the *omnipotent oracle*.

Definition 1: The *omnipotent oracle* knows the influence probabilities $p^x_{i,j}$ $\forall(i,j) \in E$ and $\forall x \in \mathcal{X}$. Given context x, it chooses $S^*(x) \in \arg\max_{S\in\mathcal{M}} \sigma(x, S)$ as the seed set.

The expected total reward of the omnipotent oracle by epoch T given a sequence of contexts $\{x_t\}^T_{t=1}$ is given by

$$\text{Rew}^*(T) := \sum_{t=1}^{T} \sigma(x_t, S^*(x_t)).$$

Since finding $S^*(x_t)$ is computationally intractable [8], we propose another (weaker) oracle that only has an approximation guarantee, which is called the (α, β)-approximation oracle $(0 < \alpha, \beta < 1)$.

Definition 2: The (α, β)-approximation oracle knows the influence probabilities $p^x_{i,j}$, $\forall(i,j) \in E$ and $\forall x \in \mathcal{X}$. Given x, it generates an α-approximate solution with probability at least β, i.e., it chooses the seed set $S^{(\alpha,\beta)}(x)$ from the set of actions $\mathcal{M}$ such that $\sigma(x, S^{(\alpha,\beta)}(x)) \geq \alpha \times \sigma(x, S^*(x))$ with probability at least β.

Note that the expected total reward of the (α, β)-approximation oracle by epoch T is at least $\alpha\beta\,\text{Rew}^*(T)$. Next, we define the approximation algorithm that is used by the learner, which takes the set of estimated influence probabilities as input. Examples of approximation algorithms for the IM problem can be found in [8] and [9].

Definition 3: The (α, β)-approximation algorithm takes as input the estimated influence probabilities $\hat{p}^x_{i,j}$, $\forall(i,j) \in E$ and $\forall x \in \mathcal{X}$. Given x, it chooses $\hat{S}^{(\alpha,\beta)}(x)$ from the set of actions $\mathcal{M}$

[1]Note that this does not hinder the learner's capability to obtain the reward as in the MAB problem with paid observations [28].

such that $\hat{\sigma}(x, \hat{S}^{(\alpha,\beta)}(x)) \geq \alpha \times \hat{\sigma}(x, \hat{S}^*(x))$ with probability at least β, where $\hat{S}^*(x) \in \arg\max_{S \in \mathcal{M}} \hat{\sigma}(x, S)$.

Similar to the related works in online learning that deal with computationally intractable problems, including the works on combinatorial MAB [22], [26], [27], we compare the learner with an (α, β)-approximation oracle. When doing this, as usual in prior work, we set the benchmark cumulative reward as $\alpha\beta$ fraction of the optimal reward. Hence, for a sequence of context arrivals $\{x_t\}_{t=1}^T$ the (α, β)-regret of the learner that uses learning algorithm π, which chooses the sequence of actions $\{S_t\}_{t=1}^T$, with respect to the (α, β)-approximation oracle by epoch T is defined as

$$R_\pi^{(\alpha,\beta)}(T) := \alpha\beta \text{Rew}^*(T) - \sum_{t=1}^{T} \sigma(x_t, S_t) + c \sum_{t=1}^{T} B_t \quad (2)$$

where B_t represents the number of observations in epoch t and c represents the cost per observation.

Our goal in this work is to design online learning algorithms that can work together with any approximation algorithm designed for the offline IM problem, whose expected (α, β)-regrets, i.e., $\mathbb{E}[R_\pi^{(\alpha,\beta)}(T)]$, grow slowly in time and in the cardinality of the action set, without making any statistical assumptions on the context arrival process.

IV. Approximation Guarantee

The maximum difference between the true and estimated influence probabilities given context x is defined as $\Delta_x(\boldsymbol{p}, \hat{\boldsymbol{p}}) := \max_{(i,j)\in E} |p_{i,j}^x - \hat{p}_{i,j}^x|$, and the maximum difference over all contexts is defined as $\Delta(\boldsymbol{p}, \hat{\boldsymbol{p}}) = \sup_{x\in\mathcal{X}} \Delta_x(\boldsymbol{p}, \hat{\boldsymbol{p}})$. The following theorem, originally given in [17, Lemma 6] provides a relation between the influence spread of action S for $G(V, E, \hat{p}^x)$ and $G(V, E, p^x)$.

Theorem 1: ([17, Lemma 6]) If $\Delta_x(\boldsymbol{p}, \hat{\boldsymbol{p}}) = \Delta_x$, then $|\hat{\sigma}(x, S) - \sigma(x, S)| \leq mn\Delta_x$, for all $S \in \mathcal{M}$.

The next theorem provides an approximation guarantee for the (α, β)-approximation algorithm with respect to the omnipotent oracle, when it runs using $\hat{\boldsymbol{p}}$ instead of $\boldsymbol{p}$.

Theorem 2: If $\Delta(\boldsymbol{p}, \hat{\boldsymbol{p}}) = \Delta$, then

$$\mathbb{E}\left[\sigma(x, \hat{S}^{(\alpha,\beta)}(x))\right] \geq \alpha\beta \times \sigma(x, S^*(x)) - \beta(1+\alpha)mn\Delta$$

for all $x \in \mathcal{X}$.

Proof: See Appendix A. ■

V. Contextual Online Influence Maximization With Costly Edge-Level Feedback (COIN-CO-EL)

In this section, we propose the *Contextual Online INfluence maximization COstly Edge-Level feedback* (COIN-CO-EL) algorithm. The pseudocode of COIN-CO-EL is given in Algorithm 1. COIN-CO-EL is an online algorithm that can use any offline IM algorithm as a subroutine. In order to exploit the context information efficiently, COIN-CO-EL aggregates information gained from past epochs with similar contexts together while forming the influence probability estimates. This aggregation is performed by creating a partition $\mathcal{Q}$ of the context set $\mathcal{X}$ based on the similarity information given in Assumption 1.

Algorithm 1: COIN-CO-EL.

Require: $T, q_T, G = (V, E), D(t), t = 1, \ldots, T$
Initialize sets: Create the partition $\mathcal{Q}$ of $\mathcal{X}$ such that $\mathcal{X}$ is divided into q_T^d identical hypercubes with edge lengths $1/q_T$
Initialize counters: $f_{i,j}^Q = s_{i,j}^Q = 0, \forall (i,j) \in E$, $\forall Q \in \mathcal{Q}, t = 1$
Initialize estimates: $\hat{p}_{i,j}^Q = 0, \forall (i,j) \in E, \forall Q \in \mathcal{Q}$
while $t \leq T$ **do**
  Find the partition $Q_t \in \mathcal{Q}$ that x_t belongs to
  Compute the set of under-explored edges $Y_{Q_t}(t)$ given in (3) and the set of under-explored nodes $U_{Q_t}(t)$ given in (4)
  if $|U_{Q_t}(t)| \geq k$ **then** {Explore}
    Select S_t randomly from $U_{Q_t}(t)$ such that $|S_t| = k$
  else if $U_{Q_t}(t) \neq \emptyset$ and $|U_{Q_t}(t)| < k$ **then**
    Select the $|U_{Q_t}(t)|$ many elements of S_t as $U_{Q_t}(t)$ and the remaining $k - |U_{Q_t}(t)|$ elements of S_t by using an (α, β)-approximation algorithm on $G(V, E, \hat{\boldsymbol{p}}_t)$
  else {Exploit}
    Select S_t by using an (α, β)-approximation algorithm on $G(V, E, \hat{\boldsymbol{p}}_t)$
  end if
  Observe the set of edges in $Y_{Q_t}(t) \cap F_t$, incur cost $c \times |Y_{Q_t}(t) \cap F_t|$
  Update the successes and failures $\forall (i,j) \in Y_{Q_t}(t) \cap F_t$:
  for $(i,j) \in Y_{Q_t}(t) \cap F_t$ **do**
    if $a_{i,j} = 1$ **then**
      $s_{i,j}^{Q_t} + +$
    else if $a_{i,j} = 0$ **then**
      $f_{i,j}^{Q_t} + +$
    end if
    $\hat{p}_{i,j}^{Q_t} = \frac{s_{i,j}^{Q_t}}{s_{i,j}^{Q_t} + f_{i,j}^{Q_t}}$
  end for
  $t = t + 1$
end while

Each set in the partition has a size (i.e., the maximum distance between any two contexts in the set) that is less than a time-horizon dependent threshold. This implies that the influence probability estimates formed by observations in a certain set of the partition do not deviate too much from the actual influence probabilities that correspond to the contexts that are in the same set.

Recall from (1) that in the IC model, at each time slot $s + 1$ of epoch t, nodes in $\mathcal{R}_t^s$ attempt to influence their children by activating the edges connecting them to their children. We call such an attempt in any time slot of epoch t an *activation attempt*. Let F_t be the set of edges with activation attempts at epoch t. F_t is simply the collection of outgoing edges from the active nodes at the end of epoch t, and hence, is known by the learner in the costly edge-level feedback setting. For $(i,j) \in F_t$, we call

$a_{i,j}$ the *influence outcome* on edge (i,j): $a_{i,j}=1$ implies that node j is influenced by node i while $a_{i,j}=0$ implies that node j is not influenced by node i. The learner does not have access to $a_{i,j}$'s beforehand, but can observe them by paying a cost c for each observation.[2] COIN-CO-EL keeps two counters $f^Q_{i,j}(t)$ and $s^Q_{i,j}(t)$ for each $(i,j)\in E$ and each $Q\in\mathcal{Q}$. The former denotes the number of observed failed activation attempts on edge (i,j) at epochs prior to epoch t when the context was in Q, while the latter denotes the number of observed successful activation attempts on edge (i,j) in epochs prior to epoch t when the context was in Q.

At the beginning of epoch t, COIN-CO-EL observes x_t and finds the set $Q\in\mathcal{Q}$ that contains x_t, which is denoted by Q_t.[3] For each $Q\in\mathcal{Q}$, COIN-CO-EL keeps sample mean estimates of the influence probabilities. For any $x\in Q$ and $(i,j)\in E$, the estimate of $p^x_{i,j}$ at epoch t is denoted by $\hat{p}^Q_{i,j}(t)$.[4] This estimate is updated whenever the influence on edge (i,j) is observed by COIN-CO-EL for some context $x\in Q$.

COIN-CO-EL decides on which seed set of nodes S_t to choose based on $\hat{\boldsymbol{p}}_t:=\{\hat{p}^{Q_t}_{i,j}(t)\}_{(i,j)\in E}$. Since these values are noisy estimates of the true influence probabilities, two factors play a role in the accuracy of these estimates: estimation error and approximation error. Estimation error is due to the noise introduced by the randomness of the influence samples, and decreases with the number of samples that are used to estimate the influence probabilities. On the other hand, approximation error is due to the noise introduced by quantization of the context set, and increases with the size of Q_t. There is an inherent tradeoff between these errors. In order to decrease the approximation error, partition $\mathcal{Q}$ must be refined. This will create more sets in $\mathcal{Q}$, and hence, will result in smaller number of samples in each set, which will cause the estimation error to increase. In order to optimally balance these errors, size of the sets in $\mathcal{Q}$ and the number of observations that fall into each of these sets must be adjusted carefully. COIN-CO-EL achieves this by using a time-horizon dependent partitioning parameter q_T, which is used to partition $\mathcal{X}$ into q_T^d identical hypercubes with border lengths $1/q_T$.[5] When $\hat{\boldsymbol{p}}_t$ is far away from $\boldsymbol{p}$, the estimation accuracy is low. Hence, in order to achieve sublinear regret, the estimate $\hat{\boldsymbol{p}}_t$ should improve over epochs for all edges $(i,j)\in E$ and for all $Q\in\mathcal{Q}$. This is achieved by alternating between two phases of operation: *exploration* and *exploitation*.

In order to define when COIN-CO-EL explores and exploits, we first define the set of under-explored edges at epoch t for $Q\in\mathcal{Q}$, which is given as

$$Y_Q(t):=\{(i,j)\in E|f^Q_{i,j}(t)+s^Q_{i,j}(t)<D(t)\} \tag{3}$$

where $D(t)$ is a positive, increasing function called the *control function*.[6] Based on this, the set of under-explored nodes at epoch t for $Q\in\mathcal{Q}$ is defined as

$$U_Q(t):=\{i\in V|\exists j\in\mathcal{N}_i: f^Q_{i,j}(t)+s^Q_{i,j}(t)<D(t)\}. \tag{4}$$

COIN-CO-EL ensures that the influence probability estimates are accurate when $U_{Q_t}(t)=\emptyset$. In this case, it *exploits* by running an (α,β)-approximation algorithm on $G(V,E,\hat{\boldsymbol{p}}_t)$. Since, $\hat{\boldsymbol{p}}_t$ is accurate, it chooses not to pay to observe the influence outcomes on the edges in this phase.

On the other hand, COIN-CO-EL assumes that the influence probability estimates are inaccurate when $U_{Q_t}(t)\neq\emptyset$. In this case, it *explores* by selecting the seed set of nodes according to the following rule: (i) When $|U_{Q_t}(t)|<k$, it selects all of the nodes in $U_{Q_t}(t)$ and the remaining $k-|U_{Q_t}(t)|$ nodes are selected by the (α,β)-approximation algorithm; (ii) When $|U_{Q_t}(t)|\geq k$, k nodes are randomly selected from $U_{Q_t}(t)$. When it explores, COIN-CO-EL also observes the influence outcomes to improve its estimates. For this, it pays and observes the influence outcomes on the edges in the set $Y_{Q_t}(t)\cap F_t$, which denotes the set of under-explored edges with activation attempts.

A. Upper Bounds on the Regret

The following theorem shows that the expected (α,β)-regret of COIN-CO-EL is sublinear in time for any sequence of context arrivals $x_1,\ldots,x_T$. Specifically, when an (α,β)-approximation algorithm is used as the offline IM algorithm in COIN-CO-EL, then the expectation of the regret of COIN-CO-EL given in (2) is bounded by a sublinear function of the number of epochs. This implies that the expected regret of COIN-CO-EL averaged over epochs converges to zero as the number of epochs increases, and hence, the average reward of COIN-CO-EL becomes at least $\alpha\beta$ fraction of the average reward of the omnipotent oracle.

Theorem 3: When COIN-CO-EL uses an (α,β)-approximation algorithm as the subroutine, and when $q_T=\lceil T^{1/(3\theta+d)}\rceil$ and $D(t)=(c+1)^{-2/3}t^{2\theta/(3\theta+d)}$, we have

$$\begin{aligned}
&\mathbb{E}\left[R^{(\alpha,\beta)}_{\text{COIN-CO-EL}}(T)\right]\\
&\leq\left(\frac{m}{k}+1\right)\alpha\beta(n-k)\left\lceil T^{\frac{1}{3\theta+d}}\right\rceil^d\left\lceil(c+1)^{-2/3}T^{\frac{2\theta}{3\theta+d}}\right\rceil\\
&+mc\left\lceil(c+1)^{-2/3}T^{\frac{2\theta}{3\theta+d}}\right\rceil\left\lceil T^{\frac{1}{3\theta+d}}\right\rceil^d\\
&+\beta(1+\alpha)mnLd^{\theta/2}T^{\frac{2\theta+d}{3\theta+d}}\\
&+\frac{\beta(1+\alpha)\pi m^2n(c+1)^{1/3}}{\sqrt{2}}\times\frac{T^{\frac{2\theta+d}{3\theta+d}}}{\frac{2\theta+d}{3\theta+d}}\\
&=O\left(c^{1/3}T^{\frac{2\theta+d}{3\theta+d}}\right)
\end{aligned}$$

for an arbitrary sequence of contexts $\{x_t\}_{t=1}^T$.

Proof: See Appendix B. ■

[2]For example, in viral marketing the marketer can freely observe a person who bought a product [16]. It can also observe the people who influenced that person to buy the product by performing a costly investigation (e.g., conducting a survey).

[3]If there are multiple such sets, then one of them is randomly selected.

[4]We will drop the epoch index when it is clear from the context.

[5]The value of q_T given in Theorem 3 achieves the balance between estimation and approximation errors. When the time horizon T is not known in advance, the same regret bound can be achieved by COIN-CO-EL by using the standard doubling trick [29].

[6]$D(t)$ is a sublinear function of t and is also inversely proportional to c.

Remark 1: As observed from Theorem 3, COIN-CO-EL explores less when the cost of observation is large. In addition, when the cost is 0, the regret bound is equivalent to the regret bound in [1, Th. 3], which does not consider the observation costs and assumes that the influence outcomes are always observed. This result shows that sublinear number of observations is sufficient to achieve the same order of regret as in [1].

The next corollary gives an upper bound on the expected (α, β)-regret of COIN-CO-EL when $\theta > 1$, which corresponds to the context-free online IM problem with costly observations.

Corollary 1: When $\theta > 1$ and COIN-CO-EL uses an (α, β)-approximation algorithm as the subroutine with $q_T = 1$ and $D(t) = (c+1)^{-2/3} t^{2/3}$, we have

$$\mathbb{E}\left[R^{(\alpha,\beta)}_{\text{COIN-CO-EL}}(T)\right] = O\left(c^{1/3} T^{2/3}\right).$$

Proof: The proof follows directly from the proof of Theorem 3. Since $q_T = 1$, sum of the regrets incurred over exploration epochs and due to observing the influence outcomes is proportional to $cD(T) = O(c^{1/3}T^{2/3})$. Moreover, the regret incurred over an exploitation epoch only depends on the estimation error since there is no approximation error. Essentially, from Lemma 4, it is observed that if the learner exploits at epoch t, then it incurs at most $O(c^{1/3}t^{-1/3})$ regret in that epoch. Summing this from 1 to T gives $O(c^{1/3}T^{2/3})$ regret due to exploitation epochs. ■

The regret bounds given in Theorem 3 and Corollary 1 are gap-independent. For the cost-free online IM problem it is shown in [17] that there exists a learning algorithm with $\tilde{O}(T^{1/2})$ gap-independent regret. In the following subsection, we show that learning in the online IM problem with costly observations is inherently more difficult than the cost-free version of the same problem by proving $\Omega(c^{1/3}T^{2/3})$ lower bound on the regret.

B. Lower Bound on the Regret for the Context-Free Online IM Problem

In this section, we consider the special case of OCIMP-CO when $\theta > 1$, and show that the regret lower bound is $\Omega(c^{1/3}T^{2/3})$.

Theorem 4: Consider OCIMP-CO with $\theta > 1$. Assume that both edge-level and node-level feedbacks are costly, where the cost of each observation is $c > 0$. For this problem, there exists a problem instance (influence graph) for which any learning algorithm π run with an exact solver, i.e., $(\alpha, \beta) = (1, 1)$, which makes OT observations by epoch T, will incur regret

$$\mathbb{E}\left[R^{(1,1)}_{\pi}(T)\right] \geq \max\left\{1.88 \times \left(\frac{\sqrt{6}}{16}\right)^{2/3} k_0^{2/3} c^{1/3} m^{1/3} T^{2/3}, \frac{\sqrt{6}}{16} k_0 \sqrt{T}\right\}$$

for $T \geq \frac{3}{8}\frac{m}{O}k_0$, where $k_0 := \left(1 - \frac{k}{m}\right)^2$. Here, O is the average number of observations made by the learning algorithm in an epoch, which is a positive real number such that OT is an integer. Since we have at most m observations in each epoch, $0 < O \leq m$.

Proof: See Appendix C. ■

Proof of Theorem 4 is built on the lower bound proofs developed for prediction with expert advice and MAB problems [28], [30]–[32]. The differences lie in the formulation of the problem instance we show our worst-case regret lower bound on and the way we handle actions, which do not correspond to individual arms but a combination of the arms. In particular, we use the fact that we can decouple actions from the arms as long as observations of the arms determine the actions taken by the learner.

VI. Contextual Online Influence Maximization With Costly Node-level Feedback (COIN-CO-NL)

The node-level feedback setting is proposed in [16]. In this setting, at the end of an epoch, the learner observes the set of activated nodes, but not the influence outcomes (i.e., edge-level feedback). In this section, we consider an extension to the node-level feedback setting, where at the end of a time slot of an epoch, the learner may choose to observe whether a node is activated or not by paying c for each observation, which implies that the learner can observe the influence spread process at node-level by costly observations. This is a plausible alternative to the original node-level feedback setting when monitoring status of the nodes in the network is costly. Moreover, obtaining temporal information about when a node gets activated is also plausible in many applications. For instance, in Twitter, a node gets activated when it re-tweets the content of another node that it is following. Similarly, in viral marketing, a node gets activated when it purchases the marketed product. Similar to the previous setting, the goal of the learner is to minimize expectation of the regret given in (2). For this purpose, we propose a variant of COIN-CO-EL called COIN-CO-NL, which is able to achieve sublinear regret when only costly node-level feedback is available.

The only difference of COIN-CO-NL from COIN-CO-EL is in the exploration phases. In exploration phases COIN-CO-NL selects the seed set S_t and the nodes to observe $Z_{t,s}$ in time slot s of epoch t in a way that allows perfect inference of influence outcomes on certain edges of the network. We introduce more flexibility to COIN-CO-NL and allow $|S_t| \leq k$. We use the fact that the learner is able to perfectly obtain edge-level feedback from node-level feedback when the children nodes of the seed nodes are distinct. In this case, by observing the children nodes of the seed nodes at $s = 2$ (seed nodes are activated at $s = 1$), the learner can perfectly infer (observe) the influence outcome on the edges between the children nodes and the seed nodes. In order to ensure that the children nodes of the seed nodes are distinct, in the worst-case, the learner can choose a single seed node in exploration phases.

As in COIN-CO-EL, COIN-CO-NL keeps counters $f^Q_{i,j}(t)$ and $s^Q_{i,j}(t)$ for the failed and successful activation attempts perfectly inferred from node-level feedback. These are used at each epoch to calculate $Y_Q(t)$ in (3) and $U_Q(t)$ in (4). When $U_{Q_t}(t) = \emptyset$, COIN-CO-NL operates in the same way as COIN-CO-EL. When $U_{Q_t}(t) \neq \emptyset$, COIN-CO-NL explores in the following way: When $|U_{Q_t}(t)| \geq k$, instead of choosing

k nodes randomly from $|U_{Q_t}(t)|$, it randomly chooses as many nodes as possible from $|U_{Q_t}(t)|$ with distinct children such that $\left(|\bigcup_{i\in S_t}\mathcal{N}_i| = \sum_{i\in S_t}|\mathcal{N}_i|\right)$. Similarly, when $|U_{Q_t}(t)| < k$, it randomly chooses as many nodes as possible from $U_{Q_t}(t)$ as long as children nodes of the chosen nodes are distinct, and chooses the remaining nodes from $V - U_{Q_t}(t)$ as long as $|S_t| \leq k$ and $|\bigcup_{i\in S_t}\mathcal{N}_i| = \sum_{i\in S_t}|\mathcal{N}_i|$. Then, after the seed nodes are chosen it observes all of the nodes j such that $j \in \bigcup_{i\in S_t}\mathcal{N}_i$ and $(i,j) \in Y_{Q_t}(t)$ for some $i \in S_t$ at $s = 2$. This way, $f^Q_{i,j}(t)$ and $s^Q_{i,j}(t)$ values are updated for a subset of the nodes in $Y_{Q_t}(t)$.

The (α,β)-regret of COIN-CO-NL is bounded in the following theorem.

Theorem 5: When COIN-CO-NL uses an (α,β)-approximation algorithm as the subroutine, and when $q_T = \lceil T^{1/(3\theta+d)}\rceil$ and $D(t) = (c+1)^{-2/3}t^{2\theta/(3\theta+d)}$, we have

$$\begin{aligned}
&\mathbb{E}\left[R^{(\alpha,\beta)}_{\text{COIN-CO-NL}}(T)\right] \\
&\leq \alpha\beta m(n-k)\left\lceil T^{\frac{1}{3\theta+d}}\right\rceil^d\left\lceil (c+1)^{-2/3}T^{\frac{2\theta}{3\theta+d}}\right\rceil \\
&\quad + mc\left\lceil (c+1)^{-2/3}T^{\frac{2\theta}{3\theta+d}}\right\rceil\left\lceil T^{\frac{1}{3\theta+d}}\right\rceil^d \\
&\quad + \beta(1+\alpha)mnLd^{\theta/2}T^{\frac{2\theta+d}{3\theta+d}} \\
&\quad + \frac{\beta(1+\alpha)\pi m^2 n(c+1)^{1/3}}{\sqrt{2}}\times\frac{T^{\frac{2\theta+d}{3\theta+d}}}{\frac{2\theta+d}{3\theta+d}} \\
&= O\left(c^{1/3}T^{\frac{2\theta+d}{3\theta+d}}\right)
\end{aligned}$$

for an arbitrary sequence of contexts $\{x_t\}_{t=1}^T$.

Proof: See Appendix B. ■

Like COIN-CO-EL, COIN-CO-NL also ensures that when it is in an exploitation epoch t, each edge is observed at least $D(t)$ many times. However, because exploration phases last longer under node-level feedback, the regret incurred due to explorations is greater. In the worst-case, COIN-CO-NL can only update one edge during one exploration epoch.

The next corollary gives an upper bound on the expected (α,β)-regret of COIN-CO-NL when $\theta > 1$, which corresponds to the context-free online IM problem with costly node-level feedback.

Corollary 2: When $\theta > 1$ and COIN-CO-NL uses an (α,β)-approximation algorithm as the subroutine with $q_T = 1$ and $D(t) = (c+1)^{-2/3}t^{2/3}$, we have

$$\mathbb{E}\left[R^{(\alpha,\beta)}_{\text{COIN-CO-NL}}(T)\right] = O\left(c^{1/3}T^{2/3}\right).$$

Proof: The proof follows directly from the proof of Theorem 5 and the proof of Corollary 1. ■

Finally, we note that Theorem 4 also provides a matching lower bound for the node-level feedback setting.

VII. Extensions

A. Improved Exploration Phase

To improve the performance of COIN-CO-EL in exploration phases, we consider two additional exploration strategies. In the first variant, which is called COIN-CO-EL+, instead of choosing nodes in S_t randomly from $U_{Q_t}(t)$, a modified version of TIM+ [9], which is restricted to choose S_t only from $U_{Q_t}(t)$, is used to select S_t. The motivation behind this choice is that since TIM+ is an (α,β)-approximation algorithm, it may provide a larger influence spread, even when the influence probability estimates are not completely accurate. In the second variant, which is called COIN-CO-EL-HD, S_t is chosen using the High-Degree heuristic [14]. High-Degree chooses nodes who have the highest out-degree values as its seed set. The motivation for using High-Degree in the exploration phases is twofold. Firstly, since the influence probability estimates are highly inaccurate in the initial epochs, an (α,β)-approximation algorithm whose performance depends on the accuracy of the influence probability estimates may not work well. Therefore, High-Degree, which does not use these estimates but uses the graph structure can work better. Secondly, High-Degree is much faster than an (α,β)-approximation algorithm, since its node selection strategy is very simple. Moreover, both COIN-CO-EL+ and COIN-CO-EL-HD have the same theoretical performance guarantees as COIN-CO-EL, since the regret analysis carried out for COIN-CO-EL is agnostic to the type of algorithm used to select from the set of under-explored nodes. We perform experiments on COIN-CO-EL+ and COIN-CO-EL-HD in Section VIII.

B. A Randomized Algorithm for Costly Edge-Level Feedback: ϵ_t-Greedy-CO-EL

In this section, we propose an algorithm that is inspired by the ϵ_t-greedy strategy for the MAB problem proposed in [30]. This algorithm, called ϵ_t-Greedy-CO-EL (pseudocode given in Algorithm 2), is similar to the online IM algorithm presented in [24]. While it does not take the context into account, in our experiments we run a context-aware version of this algorithm by using the procedure described in Algorithm 3.

As its name suggests, in each epoch ϵ_t-Greedy-CO-EL explores with probability ϵ_t and exploits with probability $1-\epsilon_t$, where ϵ_t is a positive decreasing function of t. For an edge (i,j), ϵ_t-Greedy-CO-EL keeps the parameters $s_{i,j}$ and $f_{i,j}$, which are the counters for the observed successes and failures, respectively. When it exploits, it uses an (α,β)-approximation algorithm with the sample mean estimates of the influence probabilities as input to select the set of seed nodes. On the other hand, when it explores it uses an inflated version of the influence probability estimates, which is given as the sample mean plus the sample standard deviation. Then, it provides these as input to the (α,β)-approximation algorithm to select the seed set of nodes. Using the additional standard deviation term allows it to explore influence outcomes of the edges that are sampled relatively less, working similarly to the inflation factor used in UCB algorithms. When ϵ_t-Greedy-CO-EL explores, it observes the influence outcomes on all edges with activation attempts.

Algorithm 2: ϵ_t-Greedy-CO-EL.

Require: $T, G = (V, E), \epsilon_t, t = 1, \ldots, T$
Initialize counters: $f_{i,j} = 0, s_{i,j} = 0, \forall (i,j) \in E$, $t = 1$
while $t \leq T$ **do**
  Sample $z \sim$ Bernoulli(ϵ_t)
  if $z = 1$ **then** {Explore}
    $\hat{p}_{i,j} = \min\left\{\frac{s_{i,j}}{s_{i,j}+f_{i,j}} + \sqrt{\frac{s_{i,j}f_{i,j}}{(s_{i,j}+f_{i,j})^2(s_{i,j}+f_{i,j}+1)}}, 1\right\}$, $\forall (i,j) \in E$ (if $f_{i,j} = s_{i,j} = 0$, then $\hat{p}_{i,j} = 0$)
    Select S_t by using an (α, β)-approximation algorithm for the IM problem on $G(V, E, \{\hat{p}_{i,j}\}_{(i,j)\in E})$
    Observe the set of edges in F_t, incur cost $c \times |F_t|$
    Update the successes and failures $\forall (i,j) \in F_t$:
    for $(i,j) \in F_t$ **do**
      if $a_{i,j} = 1$ **then**
        $s_{i,j}++$
      else if $a_{i,j} = 0$ **then**
        $f_{i,j}++$
      end if
    end for
  else {Exploit}
    $\hat{p}_{i,j} = \frac{s_{i,j}}{s_{i,j}+f_{i,j}}$, $\forall (i,j) \in E$ (if $f_{i,j} = s_{i,j} = 0$, then $\hat{p}_{i,j} = 0$)
    Select S_t by using an (α, β)-approximation algorithm for the IM problem on $G(V, E, \{\hat{p}_{i,j}\}_{(i,j)\in E})$
  end if
  $t = t + 1$
end while

VIII. Experiments

In this section, we carry out numerical experiments to compare the performance of our algorithms with existing ones in numerous different settings.

A. Feedback Mechanisms

We consider three different feedback mechanisms in our experiments.

1) Cost-free Edge-Level Feedback: In this setting, the influence outcomes on all edges with activation attempts are observed at the end of each epoch with no cost ($c = 0$). This is the setting that is considered in our prior work [1].

2) Cost-Free Node-Level Feedback: In this setting, all activated nodes are observed at the end of each epoch without paying any observation cost [16]. Besides this, there is no other feedback available.

3) Costly Edge-Level Feedback: This setting is explained in Section III-B.

B. Setup

We use a real-world and a synthetic network: NetHEPT and NetHEPT-whose properties are listed in Table II. NetHEPT is extensively used in IM literature [9], [16], [24], and NetHEPT- is a random subgraph of NetHEPT where all of the nodes have a positive in-degree. In NetHEPT, roughly a third of the nodes have an in-degree value of 0, which means that they cannot be activated endogenously whereas in NetHEPT-, all of the nodes can be activated by a choice of seed set. In our experiments, we set $T = 5000$ unless noted otherwise.

TABLE II
Properties of the Networks Used in the Experiments

Dataset	$\|V\|$	$\|E\|$	Average In-degree
NetHEPT	15K	59K	3.86
NetHEPT-	4K	10.5K	2.63

We consider one dimensional contexts, i.e., $d = 1$ and assume that $k = 50$, which is a typical choice in the online IM literature [16], [24]. For our algorithms, we set $q_T = 2$ and initialize all influence probability estimates as 0. In order to make exploration phases of the algorithms scalable, for experiments with cost-free edge-level and cost-free node-level feedbacks, we set $D(t) = t^{2/5}/100$, and for experiments with costly edge-level feedback, we set $D(t) = c^{-2/3}t^{2/5}/200$ and $c = 0.1$.

We report both the time averaged regret and ℓ_2-error of the influence probability estimates, where ℓ_2-error at epoch t is given by $\ell_2^t := \sqrt{\sum_{(i,j)\in E}(p_{i,j}^{x_t} - \hat{p}_{i,j}^{Q_t})^2}$. The (α, β)-approximation algorithm used by our learning algorithms is chosen as TIM+ [9].

C. Defining the Influence Probabilities

The context x_t in any epoch t is sampled uniformly at random from $[0, 1]$. The influence probabilities are generated according to a Hölder-continuous surface over $[0, 1]$ defined by the following equations:

$$\frac{0.89}{1 + e^{(-1000\times(x_t - 0.5))}} + 0.01, \tag{5}$$

$$\frac{0.89}{1 + e^{(-1000\times((1-x_t) - 0.5))}} + 0.01. \tag{6}$$

In our simulations, we consider a network composed of two groups of nodes with conflicting opinions or interests. For this, we randomly partition the nodes in the network into two groups. The influence probabilities of the outgoing edges of the nodes in the two groups are calculated using (5) and (6) such that the influence probabilities are roughly between 0.01 and 0.9. Hence, when edges in one group have high influence probabilities, edges in the other group have low influence probabilities.

D. Algorithms

We compare performance of the proposed algorithms with various algorithms that we adapt to our problem. Inspired by the structure of COIN-CO-EL, we use the approach presented in Algorithm 3 to create a contextual version of any context-free MAB algorithm π. Formally, we let q_T^d independent instances

Algorithm 3: Adapting MAB Algorithms to OCIMP.

Require: $T, q_T, G = (V, E)$
1: Create the partition $\mathcal{Q}$ of $\mathcal{X}$ such that $\mathcal{X}$ is divided into q_T^d identical hypercubes with edge lengths $1/q_T$
2: Initialize ith instance of the algorithm $\pi_i, \forall i \in \{1, ..., q_T^d\}$
3: $t = 1$.
4: **while** $t \leq T$ **do**
5: Find the partition $Q_t \in \mathcal{Q}$ that x_t belongs to
6: Get arm indices $\hat{\boldsymbol{p}}_t = \{p_{i,j}^{Q_t}\}_{(i,j)\in E}$ from π_{Q_t}
7: Select S_t by using an (α, β)-approximation algorithm for the IM problem on $G(V, E, \hat{\boldsymbol{p}}_t)$
8: Obtain observations using the appropriate feedback setting
9: Update π_{Q_t} using the observations
10: $t = t + 1$
11: **end while**

of the algorithm denoted by $\{\pi_i\}_{i=1}^{q_T^d}$ to be run on separate sets in the context partition in all our simulations.

Algorithms for the cost-free edge-level feedback setting:

1) COIN+: A contextual learning algorithm proposed in our preliminary work [1], which is a variant of COIN-CO-EL+ that works for the cost-free edge-level feedback setting. COIN+ utilizes TIM+ to choose the seed set of nodes in exploration phases.

2) COIN-HD: A variant of COIN+, which utilizes the High-Degree heuristic to choose the seed set of nodes in exploration phases instead of using TIM+.

3) Thompson: An algorithm that draws the estimated influence probability of each edge from a Beta distribution, where the parameters of the Beta distribution for edge (i, j) are $s_{i,j}$ and $f_{i,j}$, which are the counters for the observed successful and failed activation attempts on edge (i, j) respectively (both are initialized as 1). Thompson updates these parameters in each epoch based on the influence outcomes.

4) ThompsonG: A variant of Thompson, where the parameters of the Beta distribution from which the estimated influence probabilities are drawn are calculated using global priors 1 and 19 for α and β parameters, respectively, as explained in [24]. These global priors are updated in each epoch based on the influence outcomes as in [24].

5) CB+MLE: A UCB-based algorithm proposed for the online IM problem that is explained in [24].

6) ϵ_t-Greedy: A variant of ϵ_t-Greedy-CO-EL, which works under $c = 0$ and uses $\epsilon_t = 1/\sqrt{t}$.

7) Pure Exploitation: A variant of COIN+, which always exploits at each epoch in the same way as COIN+ exploits.

8) CUCB: A UCB-based algorithm proposed in [17] for the combinatorial MAB problem.

Algorithms for the cost-free node-level feedback setting:

For this setting, it is known that for each endogenously influenced node $i \in V$, at least one of its parent nodes $j \in \mathcal{V}_i$ should be active. Thus, we use the frequentist credit assignment method proposed in [16] to adapt an algorithm designed for cost-free edge-level feedback to work under cost-free node-level feedback.

For this, let $\mathcal{V}_i^{'} \subset \mathcal{V}_i$ denote the set of active parents of i. We assume that the probability with which a node $i \in V$ is influenced by a node $j \in \mathcal{V}_i^{'}$ is $1/|\mathcal{V}_i^{'}|$. Then, we sample from this distribution one of the nodes l in $\mathcal{V}_i^{'}$ as the influencer of node i, set $a_{l,i} = 1$, and $a_{j,i} = 0$ for all other j in $\mathcal{V}_i^{'}$. Finally, the assigned influence outcomes are used to update the influence probability estimates.

All algorithms proposed for the cost-free edge-level feedback setting are adapted using the above procedure to work under the cost-free node-level feedback setting.

Algorithms for the costly edge-level feedback setting:

For this setting, we use COIN-CO-EL+, COIN-CO-EL-HD and ϵ_t-Greedy-CO-EL described in Sections VII-A and VII-B. As its learning parameter, ϵ_t-Greedy-CO-EL uses $\epsilon_t = 1/\sqrt{t}$. Since other algorithms do not explicitly separate exploration and exploitation phases, their extension to this setting is not straightforward, and hence, they are not considered for this setting.

E. Results for Cost-Free Edge-Level Feedback

Regret Comparison: Results in Fig. 2(a) and Fig. 2(c) show that most of the algorithms used in the cost-free edge-level feedback setting are outperformed by COIN+ and COIN-HD in the long run, and only CUCB and Thompson are able to achieve competitive average regret. We also observe that COIN+ and COIN-HD suffer from high exploration regret in the beginning due to performing large number of explorations. This issue arises especially in NetHEPT, which is a larger graph than NetHEPT-. However, after the initial exploration phase, COIN+ and COIN-HD learn the influence probabilities of the graph well enough to achieve much lower regret in exploitation epochs, which results in a quick reduction of their average regret.

ℓ_2-error Comparison: Since the ℓ_2-error measures the accuracy of the influence probability estimates, rate of decrease of the ℓ_2-error over epochs explains how well the influence probabilities are learned. Results in Fig. 2(b) and Fig. 2(d) show that COIN-HD and COIN+ achieve lower ℓ_2-errors than all other algorithms due to their explicit exploration phases. However, as seen from the average regret results, having accurate influence probability estimates does not always translate into achieving small average regret. This is due to the fact that knowledge about the influence probabilities associated with nodes that are not influential in the network does not have much impact on the seed set selection process. However, we argue that in more challenging networks of larger size, the disparity between ℓ_2-errors of these algorithms would be more indicative of their performance as the algorithms that do not conduct extensive exploration phases would be more likely to miss out on some of the more influential nodes in the network, and hence, underperform.

F. Results for Cost-Free Node-Level Feedback

Regret Comparison: Results in Fig. 2(e) and Fig. 2(f) show that COIN+ and COIN-HD outperform most of the benchmark algorithms and perform on par with CUCB and Thompson in the long run. Similar to the case for the cost-free edge-level feedback, initially both COIN+ and COIN-HD suffer high average

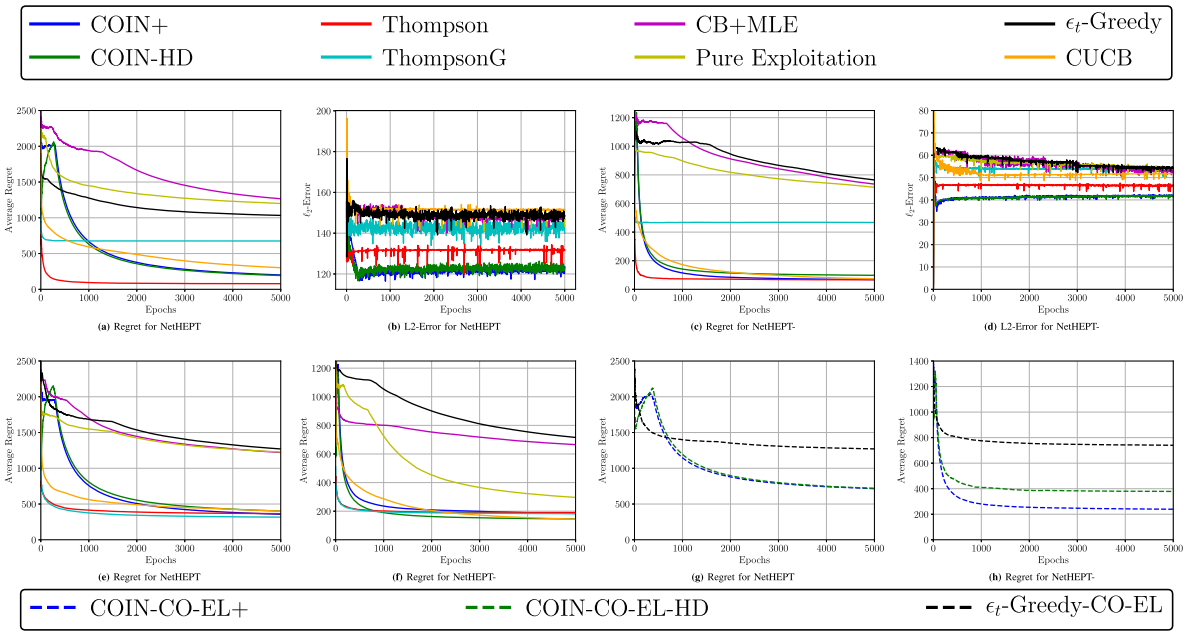

Fig. 2. Results for context-aware algorithms under (a)–(d) cost-free edge-level feedback, (e) and (f) cost-free node-level feedback, and (g) and (h) costly edge-level feedback.

regrets due to their extensive explorations, but benefit from it in the long run by achieving low exploitation regret and catching up with the best performing algorithms.

One interesting observation that highlights the importance of extensive explorations is the performance of Pure Exploitation in this setting. The performance of this greedy algorithm is very poor for NetHEPT, but significantly better for NetHEPT-. The smaller scale of the latter network benefits the algorithm as implicit exploration due to the influence spread process is enough to learn most of the network. However, in the case of NetHEPT, due to the larger scale of the network and the existence of nodes with zero in-degree, the greediness of the algorithm hurts it as it can not estimate many important influence probabilities accurately.

This observation supports the discussion about the ℓ_2-error results. The bigger and more complex the network is, the more valuable thorough exploration seems to become for the learning algorithms. We would expect to see the impact of the forced exploration phases of COIN+ and COIN-HD in more challenging settings, where implicit exploration of CUCB and Thompson might fail to identify all of the influential nodes.

Since the ℓ_2-error trend in this and the following experiments are similar to the results presented for the cost-free edge-level feedback setting, for the sake of brevity we show only the results that are related to the regret hereafter.

G. Results for the Costly Edge-Level Feedback

Regret Comparison: Results in Fig. 2(g) and Fig. 2(h) show that COIN-CO-EL-HD and COIN-CO-EL+ perform significantly better than ϵ_t-greedy-CO-EL on both networks. In addition, in order to separately observe how well COIN-CO-EL+ and COIN-CO-EL-HD performs in exploration and exploitation phases, the average regrets calculated separately over exploration and exploitation epochs are shown in Fig. 3.

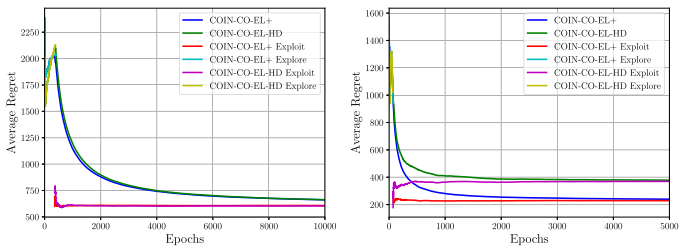

Fig. 3. Average regrets of COIN-CO-EL+ and COIN-CO-EL-HD calculated over exploration and exploitation epochs. Simulation for NetHEPT is carried over 10000 epochs to observe the convergence better.

We observe that immediately after their high regret exploration epochs, COIN-CO-EL+ and COIN-CO-EL-HD start to perform very close to the (α, β)-approximation oracle, suffering substantially lower regrets. The impact of this decrease is reflected on the average regret in the long run.

IX. Conclusion

In this paper, we propose a new online influence maximization problem where the influence probabilities depend on the context and influence outcomes are costly to observe. We develop computationally efficient learning algorithms for this problem, for both edge-level and node-level feedback settings, and prove that they achieve sublinear regret. We also show that these algorithms perform on par with their competitors on real-world networks. Since the online influence maximization problem is a special case of the combinatorial MAB problem with probabilistically

triggered arms, our model also generalizes the latter problem by introducing context dependent rewards and costly observations.

APPENDIX A
PROOF OF THEOREM 2

By definition of the (α,β)-approximation oracle, we have $\hat{\sigma}(x,\hat{S}^{(\alpha,\beta)}(x)) \geq \alpha \times \hat{\sigma}(x,\hat{S}^{*}(x))$ with probability at least β. Theorem 1 implies that for any seed set S, $|\hat{\sigma}(x,S) - \sigma(x,S)| \leq mn\Delta$. Using the results above, we obtain

$$\begin{aligned}\sigma(x,\hat{S}^{(\alpha,\beta)}(x)) &\geq \hat{\sigma}(x,\hat{S}^{(\alpha,\beta)}(x)) - mn\Delta \\ &\geq \alpha\hat{\sigma}(x,\hat{S}^{*}(x)) - mn\Delta \\ &\geq \alpha\hat{\sigma}(x,S^{*}(x)) - mn\Delta \\ &\geq \alpha\left(\sigma(x,S^{*}(x)) - mn\Delta\right) - mn\Delta \\ &= \alpha\sigma(x,S^{*}(x)) - (1+\alpha)mn\Delta\end{aligned}$$

with probability at least β. Since $\sigma(x,\hat{S}^{(\alpha,\beta)}(x))$ is non-negative, we obtain the following bound by taking the expectation:

$$\mathbb{E}[\sigma(x,\hat{S}^{(\alpha,\beta)}(x))] \geq \alpha\beta \times \sigma(x,S^{*}(x)) - \beta(1+\alpha)mn\Delta.$$

APPENDIX B
PROOF OF THEOREMS 3 AND 5

For $Q \in \mathcal{Q}$ let $\overline{p}^{Q}_{i,j} := \sup_{x\in Q} p^{x}_{i,j}$ and $\underline{p}^{Q}_{i,j} := \inf_{x\in Q} p^{x}_{i,j}$. Consider an algorithm π (COIN-CO-EL or COIN-CO-NL) with partitioning parameter $q_T = \lceil T^z \rceil$ and control function $D(t) = (c+1)^{\eta}t^{\gamma}$, where $0 < \gamma, z < 1$ and $\eta < 0$. For any sequence of context arrivals $\{x_t\}_{t=1}^{T}$, let $\mathcal{T}_T^{s,\pi}$ be the set of epochs by epoch T in which algorithm π exploits and $\mathcal{T}_T^{o,\pi}$ be the set of epochs by epoch T in which π explores. Since the activation attempts are random variables, $\mathcal{T}_T^{s,\pi}$ and $\mathcal{T}_T^{o,\pi}$ are random sets for which $\mathcal{T}_T^{s,\pi} \cup \mathcal{T}_T^{o,\pi} = \{1,\ldots,T\}$ with probability 1. By the definition of the exploration and exploitation phases of COIN-CO-EL and COIN-CO-NL, we have for any $t \in \mathcal{T}_T^{s,\pi}$

$$f^{Q_t}_{i,j}(t) + s^{Q_t}_{i,j}(t) \geq D(t)\ \forall (i,j) \in E. \tag{7}$$

The *simple* (α,β)-regret of algorithm π for epoch t is defined as

$$r_{\pi}^{(\alpha,\beta)}(t) := \alpha\beta \times \sigma(x_t,S^{*}(x_t)) - \sigma(x_t,S_t).$$

Let

$$R^{s}_{\pi}(T) := \sum_{t\in\mathcal{T}_T^{s,\pi}} r_{\pi}^{(\alpha,\beta)}(t)$$

be the regret incurred over epochs in which algorithm π exploits,

$$R^{o}_{\pi}(T) := \sum_{t\in\mathcal{T}_T^{o,\pi}} r_{\pi}^{(\alpha,\beta)}(t)$$

be the regret (except the cost of observing the influence outcomes or activated nodes) incurred over epochs in which algorithm π explores, and

$$R^{c}_{\pi}(T) = c \times \sum_{t=1}^{T} B_t$$

be the regret due to observing the influence outcomes (or activated nodes) over epochs in which algorithm π explores. Based on the above definitions, the regret can be decomposed as follows:

$$\mathbb{E}[R_{\pi}^{(\alpha,\beta)}(T)] = \mathbb{E}[R^{s}_{\pi}(T)] + \mathbb{E}[R^{o}_{\pi}(T)] + \mathbb{E}[R^{c}_{\pi}(T)].$$

The proof proceeds with bounding each of the terms given above. First, we bound $R^{o}_{\pi}(T)$ for COIN-CO-EL and COIN-CO-NL.

Lemma 1: When π = COIN-CO-EL runs with control function $D(t) = (c+1)^{\eta}t^{\gamma}$ and partitioning parameter $q_T = \lceil T^z \rceil$, where $0 < \gamma, z < 1$ and $\eta < 0$, we have

$$R^{o}_{\pi}(T) \leq \alpha\beta(n-k)\left(\frac{m}{k}+1\right)\lceil T^z\rceil^{d}\lceil (c+1)^{\eta}T^{\gamma}\rceil$$

with probability 1.

Proof: At each epoch when COIN-CO-EL explores, in the worst case, COIN-CO-EL will fail to influence the remaining $(n-k)$ nodes and the omnipotent oracle will influence all the remaining $(n-k)$ nodes. Hence, we have $r_{\pi}^{(\alpha,\beta)}(t) \leq \alpha\beta(n-k)$ for each $t \in \mathcal{T}_T^{o,\pi}$, which implies that

$$R^{o}_{\pi}(T) \leq \alpha\beta(n-k)|\mathcal{T}_T^{o,\pi}| \text{ with probability 1.} \tag{8}$$

Next, we will bound $|\mathcal{T}_T^{o,\pi}|$. Let $\mathcal{T}_T^{L}$ denote the set of exploration epochs of COIN-CO-EL where $|Y_{Q_t}(t)| < k$ and $\mathcal{T}_T^{H}$ denote the set of exploration epochs where $|Y_{Q_t}(t)| \geq k$. We have $\mathcal{T}_T^{o,\pi} = \mathcal{T}_T^{H} \cup \mathcal{T}_T^{L}$.

Firstly, we bound $|\mathcal{T}_T^{L}|$. Note that for each $Q \in \mathcal{Q}$, there will be at most $\lceil D(T)\rceil$ many epochs where $Q_t = Q$ and $|Y_{Q_t}(t)| < k$. Therefore,

$$|\mathcal{T}_T^{L}| < q_T^{d}\lceil D(T)\rceil \text{ with probability 1.} \tag{9}$$

Secondly, we bound $|\mathcal{T}_T^{H}|$. Let $u(t) := |F_t \cap Y_{Q_t}(t)|$. Note that given T, we have a total of mq_T^{d} many context set-edge pairs. Hence, the total number of explorations can be at most $mq_T^{d}\lceil D(T)\rceil$. Thus

$$\sum_{t\in\mathcal{T}_T^{H}} u(t) \leq mq_T^{d}\lceil D(T)\rceil \text{ with probability 1.} \tag{10}$$

From the definition of $\mathcal{T}_T^{H}$, we know that $u(t) \geq k$ for all $t \in \mathcal{T}_T^{H}$. Using this together with (10), we obtain $k|\mathcal{T}_T^{H}| \leq mq_T^{d}\lceil D(T)\rceil$, and hence,

$$|\mathcal{T}_T^{H}| \leq \frac{mq_T^{d}\lceil D(T)\rceil}{k} \text{ with probability 1.} \tag{11}$$

Hence, by summing (9) and (11), we obtain

$$|\mathcal{T}_T^{o,\pi}| \leq \left(\frac{m}{k}+1\right)\lceil T^z\rceil^{d}\lceil (c+1)^{\eta}T^{\gamma}\rceil \text{ with probability 1.} \tag{12}$$

The result follows by plugging the bound in (12) into (8). ■

Lemma 2: When π = COIN-CO-NL runs with control function $D(t) = (c+1)^{\eta}t^{\gamma}$ and partitioning parameter $q_T = \lceil T^z \rceil$, where $0 < \gamma, z < 1$ and $\eta < 0$, we have

$$R^{o}_{\pi}(T) \leq \alpha\beta(n-k)m\lceil T^z\rceil^{d}\lceil (c+1)^{\eta}T^{\gamma}\rceil$$

with probability 1.

Proof: At each epoch when COIN-CO-NL explores, in the worst case, COIN-CO-NL will fail to influence the remaining $(n-k)$ nodes and the omnipotent oracle will influence all the remaining $(n-k)$ nodes. Hence, we have $r_\pi^{(\alpha,\beta)}(t) \leq \alpha\beta(n-k)$ for each $t \in \mathcal{T}_T^{o,\pi}$, which implies that

$$R_\pi^o(T) \leq \alpha\beta(n-k)|\mathcal{T}_T^{o,\pi}| \text{ with probability 1.} \tag{13}$$

Next, we will bound $|\mathcal{T}_T^{o,\pi}|$. Note that COIN-CO-NL explores (infers the influence outcome on) at least one edge in each exploration epoch, and each context partition-edge pair will be explored at most $q_T^d\lceil D(T)\rceil$ times. Hence, we have

$$|\mathcal{T}_T^{o,\pi}| \leq mq_T^d\lceil D(T)\rceil \text{ with probability 1}$$

which gives the desired result when used together with (13). ■

Next, we bound the regret due to costly influence outcome (node activation) observations for COIN-CO-EL (COIN-CO-NL).

Lemma 3: When COIN-CO-EL or COIN-CO-NL runs with control function $D(t) = (c+1)^\eta t^\gamma$ and partitioning parameter $q_T = \lceil T^z\rceil$, where $0 < \gamma, z < 1$ and $\eta < 0$, we have

$$R_\pi^c(T) \leq cm\lceil T^z\rceil^d\lceil (c+1)^\eta T^\gamma\rceil \text{ with probability 1.}$$

Proof: Since COIN-CO-EL and COIN-CO-NL both keep an influence probability estimate for each edge $(i,j) \in E$ for each $Q \in \mathcal{Q}$, they keep mq_T^d parameters to represent the influence probability estimates. At each exploration epoch t, they observe the influence outcomes on the edges that correspond to a subset of the edges which are explored less than $D(t)$ times (COIN-CO-EL) or node activations for a subset of nodes that are adjacent to a subset of the under-explored edges (COIN-CO-NL). Thus, by the end of epoch T, the influence outcome on an edge (or a node) is observed at most $\lceil D(T)\rceil q_T^d$ times. Therefore, the total number of costly observations is at most $m\lceil D(T)\rceil q_T^d$ and each of these observations costs c.

Next, we bound the exploitation regret. For this, we first propose the following lemma, which bounds $r_\pi^{(\alpha,\beta)}(t)$ when COIN-CO-EL or COIN-CO-NL exploits at epoch t.

Lemma 4: When $\pi =$ COIN-CO-EL or $\pi =$ COIN-CO-NL runs with control function $D(t) = (c+1)^\eta t^\gamma$ and partitioning parameter $q_T = \lceil T^z\rceil$, where $0 < \gamma, z < 1$ and $\eta < 0$, we have

$$\mathbb{E}[r_\pi^{(\alpha,\beta)}(t)|t \in \mathcal{T}_T^{s,\pi}] \leq \beta(1+\alpha)mnLd^{\theta/2}q_T^{-\theta} + \frac{\beta(1+\alpha)\pi m^2 n t^{-\gamma/2}(c+1)^{-\eta/2}}{\sqrt{2}}.$$

Proof: In the analysis below, we consider $t \in \mathcal{T}_T^{s,\pi}$, hence all the expectations are conditioned on this event. Let $\Delta_t := \Delta_{x_t}(\boldsymbol{p}, \hat{\boldsymbol{p}}_t)$. By Theorem 2, we have

$$\begin{aligned}\mathbb{E}[r_\pi^{(\alpha,\beta)}(t)] &= \alpha\beta \times \sigma(x_t, S^*(x_t)) - \mathbb{E}[\sigma(x_t, S_t)] \\ &\leq \beta(1+\alpha)mn\mathbb{E}[\Delta_t].\end{aligned}$$

Since $\Delta_t \in [0,1]$, we have

$$\mathbb{E}[r_\pi^{(\alpha,\beta)}(t)] \leq \beta(1+\alpha)mn\int_0^1 \Pr(\Delta_t \geq y)dy. \tag{14}$$

Note that

$$\begin{aligned}\{\Delta_t \geq y\} &= \left\{\max_{(i,j)\in E}(|\hat{p}_{i,j}^{Q_t}(t) - p_{i,j}^{x_t}| \geq y\right\} \\ &= \bigcup_{(i,j)\in E}\left\{|\hat{p}_{i,j}^{Q_t}(t) - p_{i,j}^{x_t}| \geq y\right\} \\ &= \bigcup_{(i,j)\in E}\{\hat{p}_{i,j}^{Q_t}(t) - p_{i,j}^{x_t} \leq -y\} \cup \bigcup_{(i,j)\in E}\{\hat{p}_{i,j}^{Q_t}(t) - p_{i,j}^{x_t} \geq y\} \\ &\subset \bigcup_{(i,j)\in E}\{\hat{p}_{i,j}^{Q_t}(t) - \overline{p}_{i,j}^{Q_t} \leq -y\} \cup \bigcup_{(i,j)\in E}\{\hat{p}_{i,j}^{Q_t}(t) - \underline{p}_{i,j}^{Q_t} \geq y\}.\end{aligned}$$

Hence, by the union bound, we get

$$\begin{aligned}\Pr(\Delta_t \geq y) \leq &\sum_{(i,j)\in E}\Pr(\hat{p}_{i,j}^{Q_t}(t) - \overline{p}_{i,j}^{Q_t} \leq -y) \\ &+ \sum_{(i,j)\in E}\Pr(\hat{p}_{i,j}^{Q_t}(t) - \underline{p}_{i,j}^{Q_t} \geq y).\end{aligned}$$

By Assumption 1, we have $\overline{p}_{i,j}^{Q_t} - \underline{p}_{i,j}^{Q_t} \leq Ld^{\theta/2}q_T^{-\theta}$ for all $(i,j) \in E$ and $Q_t \in \mathcal{Q}$. Hence, we have $\overline{p}_{i,j}^{Q_t} \leq \mathbb{E}[\hat{p}_{i,j}^{Q_t}(t)] + Ld^{\theta/2}q_T^{-\theta}$ and $\underline{p}_{i,j}^{Q_t} \geq \mathbb{E}[\hat{p}_{i,j}^{Q_t}(t)] - Ld^{\theta/2}q_T^{-\theta}$ for all $(i,j) \in E$ and $Q_t \in \mathcal{Q}$. Using the fact above, we obtain

$$\begin{aligned}&\Pr(\hat{p}_{i,j}^{Q_t}(t) - \overline{p}_{i,j}^{Q_t} \leq -y) \\ &\leq \Pr(\hat{p}_{i,j}^{Q_t}(t) - \mathbb{E}[\hat{p}_{i,j}^{Q_t}(t)] \leq Ld^{\theta/2}q_T^{-\theta} - y)\end{aligned}$$

and

$$\begin{aligned}&\Pr(\hat{p}_{i,j}^{Q_t}(t) - \underline{p}_{i,j}^{Q_t} \geq y) \\ &\leq \Pr(\hat{p}_{i,j}^{Q_t}(t) - \mathbb{E}[\hat{p}_{i,j}^{Q_t}(t)] \geq y - Ld^{\theta/2}q_T^{-\theta}).\end{aligned}$$

Since (7) holds for $t \in \mathcal{T}_T^{s,\pi}$, by using the above inequalities together with Hoeffding's inequality, we obtain the following for $y \geq Ld^{\theta/2}q_T^{-\theta}$:

$$\sum_{(i,j)\in E}\Pr(\hat{p}_{i,j}^{Q_t}(t) - \underline{p}_{i,j}^{Q_t} \geq y) \leq me^{-2(y-Ld^{\theta/2}q_T^{-\theta})^2(c+1)^\eta t^\gamma} \tag{15}$$

$$\sum_{(i,j)\in E}\Pr(\hat{p}_{i,j}^{Q_t}(t) - \overline{p}_{i,j}^{Q_t} \leq -y) \leq me^{-2(y-Ld^{\theta/2}q_T^{-\theta})^2(c+1)^\eta t^\gamma}. \tag{16}$$

In order to bound (14), we will separate the integral into two parts. For $0 \leq y < Ld^{\theta/2}q_T^{-\theta}$, we have $\Pr(\Delta_t \geq y) \leq 1$. For $Ld^{\theta/2}q_T^{-\theta} \leq y \leq 1$ by (15) and (16), we have $\Pr(\Delta_t \geq y) \leq$

$2me^{-2(y-Ld^{\theta/2}q_T^{-\theta})^2(c+1)^\eta t^\gamma}$. Hence,

$$\begin{aligned}
&\int_0^1 \Pr(\Delta_t \geq y)dy \\
&= \int_0^{Ld^{\theta/2}q_T^{-\theta}} 1dy + \int_{Ld^{\theta/2}q_T^{-\theta}}^1 2me^{-2(y-Ld^{\theta/2}q_T^{-\theta})^2(c+1)^\eta t^\gamma} dy \\
&\leq Ld^{\theta/2}q_T^{-\theta} \\
&\quad + 2m\int_{Ld^{\theta/2}q_T^{-\theta}}^1 \frac{dy}{1+2(y-Ld^{\theta/2}q_T^{-\theta})^2(c+1)^\eta t^\gamma} \\
&= Ld^{\theta/2}q_T^{-\theta} + 2m\frac{(c+1)^{-\eta/2}t^{-\gamma/2}}{\sqrt{2}} \\
&\quad \times (\arctan(\sqrt{2}t^{\gamma/2}(c+1)^{\eta/2}(1-Ld^{\theta/2}q_T^{-\theta})) \\
&\leq Ld^{\theta/2}q_T^{-\theta} + \frac{m(c+1)^{-\eta/2}t^{-\gamma/2}\pi}{\sqrt{2}} \qquad (17)
\end{aligned}$$

since $e^{-y} \leq \frac{1}{1+y}$ for all $y \geq 0$ and that $\arctan(z) \leq \frac{\pi}{2}$ for all $z \in \mathbb{R}$. The result is obtained by substituting (17) in (14). ■

The next lemma uses Lemma 4 to bound $\mathbb{E}[R_\pi^s(T)]$.

Lemma 5: When $\pi =$ COIN-CO-EL or $\pi =$ COIN-CO-NL runs with control function $D(t) = (c+1)^\eta t^\gamma$ and partitioning parameter $q_T = \lceil T^z \rceil$, where $0 < \gamma, z < 1$, $\eta < 0$, we have

$$\begin{aligned}
\mathbb{E}[R_\pi^s(T)] &\leq \beta(1+\alpha)mnLd^{\theta/2}T^{1-\theta z} \\
&\quad + \frac{\beta(1+\alpha)\pi m^2 n(c+1)^{-\eta/2}}{\sqrt{2}} \times \frac{T^{1-\gamma/2}-\gamma/2}{(1-\gamma/2)}.
\end{aligned}$$

Proof: We utilize the following inequalities in the proof: $|\mathcal{T}_T^{s,\pi}| \leq T$ with probability 1 and $\sum_{t=1}^T t^{-x} \leq \frac{T^{1-x}-x}{(1-x)}$ $\forall x \in (0,1)$. For any realization of $\mathcal{T}_T^{s,\pi}$ denoted by $\mathcal{T} \subset \{1,\ldots,T\}$ we have

$$\begin{aligned}
&\mathbb{E}[R_\pi^s(T)|\mathcal{T}_T^{s,\pi} = \mathcal{T}] = \sum_{t\in\mathcal{T}} \mathbb{E}[r_\pi^{(\alpha,\beta)}(t)|t \in \mathcal{T}] \\
&\leq \beta(1+\alpha) \\
&\times \sum_{t\in\mathcal{T}} \left(mnLd^{\theta/2}\lceil T^z \rceil^{-\theta} + \frac{\pi m^2 n t^{-\gamma/2}(c+1)^{-\eta/2}}{\sqrt{2}} \right) \\
&\leq \beta(1+\alpha) \\
&\times \sum_{t=1}^T \left(mnLd^{\theta/2}\lceil T^z \rceil^{-\theta} + \frac{\pi m^2 n t^{-\gamma/2}(c+1)^{-\eta/2}}{\sqrt{2}} \right) \\
&\leq \beta(1+\alpha) \\
&\times \sum_{t=1}^T \left(mnLd^{\theta/2}T^{-z\theta} + \frac{\pi m^2 n t^{-\gamma/2}(c+1)^{-\eta/2}}{\sqrt{2}} \right) \\
&\leq \beta(1+\alpha)mnLd^{\theta/2}T^{1-\theta z} \\
&+ \frac{\beta(1+\alpha)\pi m^2 n(c+1)^{-\eta/2}}{\sqrt{2}} \times \frac{T^{1-\gamma/2}-\gamma/2}{(1-\gamma/2)}.
\end{aligned}$$

■

By summing the results of Lemmas 1, 3 and 5, we obtain for $\pi =$ COIN-CO-EL:

$$\begin{aligned}
\mathbb{E}[R_\pi^{(\alpha,\beta)}(T)] &\leq \alpha\beta(n-k)\left(\frac{m}{k}+1\right)\lceil T^z \rceil^d \lceil (c+1)^\eta T^\gamma \rceil \\
&\quad + \beta(1+\alpha)mnLd^{\theta/2}T^{1-\theta z} \\
&\quad + cm\lceil (c+1)^\eta T^\gamma \rceil \lceil T^z \rceil^d \\
&\quad + \frac{\beta(1+\alpha)\pi m^2 n(c+1)^{-\eta/2}}{\sqrt{2}} \times \frac{T^{1-\gamma/2}-\gamma/2}{(1-\gamma/2)}. \qquad (18)
\end{aligned}$$

Similarly, by summing the results of Lemmas 2, 3 and 5, we obtain for $\pi =$ COIN-CO-NL:

$$\begin{aligned}
\mathbb{E}[R_\pi^{(\alpha,\beta)}(T)] &\leq \alpha\beta(n-k)m\lceil T^z \rceil^d \lceil (c+1)^\eta T^\gamma \rceil \\
&\quad + \beta(1+\alpha)mnLd^{\theta/2}T^{1-\theta z} \\
&\quad + cm\lceil (c+1)^\eta T^\gamma \rceil \lceil T^z \rceil^d \\
&\quad + \frac{\beta(1+\alpha)\pi m^2 n(c+1)^{-\eta/2}}{\sqrt{2}} \times \frac{T^{1-\gamma/2}-\gamma/2}{(1-\gamma/2)}. \qquad (19)
\end{aligned}$$

Finally, we calculate the optimal values of the parameters, which minimize the regrets given in (18) and (19) . We observe that the terms in the regret bounds with the highest time orders are $O(T^{zd+\gamma})$, $O(T^{1-\gamma/2})$ and $O(T^{1-\theta z})$. Hence, the optimal z and γ should minimize $\max\{zd+\gamma, 1-\gamma/2, 1-\theta z\}$. This is achieved by setting $z = 1/(3\theta+d)$ and $\gamma = 2\theta/(3\theta+d)$. In addition, we also minimize the order of the cost in the regret bound, given the optimal z and γ values for the time order of the regret. For this, we look at the regret terms whose time orders are $T^{(2\theta+d)/(3\theta+d)}$. The cost order of these terms are $O((c+1)^{1+\eta})$ and $O((c+1)^{-\eta/2})$. Hence, to balance these terms, we set $\eta = -2/3$.

Appendix C
Proof of Theorem 4

Our proof is built on [28, proof of Th. 4]. In the proof, we assume that the learner is deterministic and makes a fixed number of observations, denoted by OT, by epoch T. It is well known that lower bound results for deterministic learners apply for stochastic learners as well [31].

First Step: Influence Graph and a Regret Lower Bound

We define a specific directed graph $\bar{G}(\bar{V},\bar{E})$ with the following properties. Assume that n is even and $m = n/2$. Let $\bar{V}_0 := \{v_0^1,\ldots,v_0^{n/2}\}$, $\bar{V}_1 := \{v_1^1,\ldots,v_1^{n/2}\}$, $\bar{E} := \{(v_0^1,v_1^1),(v_0^2,v_1^2),\ldots,(v_0^m,v_1^m)\}$ and $\bar{V} := \bar{V}_0 \cup \bar{V}_1$. Since the nodes in $\bar{V}_1$ cannot influence any other node, any sensible policy will only select nodes from $\bar{V}_0$ as the seed set. Let $A = \binom{m}{k}$ denote the cardinality of the action set $\mathcal{M}$. We index the actions in a way that V_i denotes the ith action, and hence, $\mathcal{M} := \{V_1,\ldots,V_A\}$. Due to the fact that each node in $\bar{V}_1$ has a single parent, in this setting edge-level and node-level feedbacks are equivalent. Thus, in the rest of the proof, we focus on only edge-level feedback. We assume that the influence probabilities are independent of context, we simplify the notation and use $\sigma(S)$ and S^* to denote the expected influence spread of action S and an optimal action, respectively.

We define $m+1$ problem instances on $\bar{G}(\bar{V},\bar{E})$, indexed by $\{0,1,\ldots,m\}$. For $h>0$, in the hth problem instance, we set the influence probabilities on the edges as $p_{v_0^h,v_1^h}=\frac{1+\epsilon}{2}$, and $p_i=\frac{1-\epsilon}{2}$ for $i\in\bar{E}-(v_0^h,v_1^h)$. Moreover, in the 0th problem instance, we set $p_i=\frac{1-\epsilon}{2}$ for all $i\in\bar{E}$. Note that with the above construction, for any problem instance $h>0$, there exists $\binom{m-1}{k-1}$ actions with expected influence spread $(k-1)\frac{1-\epsilon}{2}+\frac{1+\epsilon}{2}$, and $\binom{m}{k}-\binom{m-1}{k-1}=\binom{m-1}{k}$ actions with expected influence spread $k\frac{1-\epsilon}{2}$.

Let $\hat{q}=\{\hat{q}_1,\ldots,\hat{q}_A\}$ be the marginal distribution over the actions selected by the learner over time horizon T, where $\hat{q}_i=\frac{1}{T}\sum_{t=1}^{T}\mathbf{1}_{\{S_t=V_i\}}$ and $\mathbf{1}_{\{\cdot\}}$ denotes the indicator variable which is equal to one when the condition inside is satisfied and zero otherwise. Let J be a random variable distributed according to $\hat{q}$. Let $\mathbb{P}_h$ be the law of J when the problem instance is h and $\mathbb{E}_h$ denote expectations in problem instance h. Note that we have $\mathbb{P}_h(J=V_h)=\mathbb{E}_h[\frac{1}{T}\sum_{t=1}^{T}\mathbf{1}_{\{S_t=V_h\}}]$. In addition, for any problem instance $h>0$, we denote the set of optimal actions by $\mathcal{S}_h^*:=\{V_i:\mathbb{E}_h[\sigma(V_i)]=(k-1)\frac{1-\epsilon}{2}+\frac{1+\epsilon}{2}\}$. Hence, for the hth problem instance, we have

$$R_h(T):=\mathbb{E}_h[R^{(1,1)}(T)]-c\mathbb{E}_h\left[\sum_{t=1}^{T}B_t\right] \tag{20}$$

$$=\mathbb{E}_h\left[\sum_{t=1}^{T}\mathbf{1}_{\{S_t\notin\mathcal{S}_h^*\}}\epsilon\right]=\epsilon T\mathbb{E}_h\left[\sum_{t=1}^{T}\mathbf{1}_{\{S_t\notin\mathcal{S}_h^*\}}/T\right]$$

$$=\epsilon T\mathbb{P}_h(J\notin\mathcal{S}_h^*) \tag{21}$$

$$=\epsilon T\left(1-\mathbb{P}_h(J\in\mathcal{S}_h^*)\right). \tag{22}$$

Therefore, there exist at least one problem instance h' for which $R_{h'}(T)$ is greater than or equal to the mean of (22) over all problem instances $h>0$:

$$\sup_{h>0}R_h(T)\geq\epsilon T\left(1-\frac{1}{m}\sum_{h=1}^{m}\mathbb{P}_h(J\in\mathcal{S}_h^*)\right). \tag{23}$$

Second Step: Pinsker's Inequality

By Pinsker's inequality and union bound, we have

$$\mathbb{P}_h(J\in\mathcal{S}_h^*)\leq\mathbb{P}_0(J\in\mathcal{S}_h^*)+\sqrt{\frac{1}{2}\mathrm{KL}(\mathbb{P}_0||\mathbb{P}_h)}$$

$$\leq\sum_{V_i\in\mathcal{S}_h^*}\mathbb{P}_0(J=V_i)+\sqrt{\frac{1}{2}\mathrm{KL}(\mathbb{P}_0||\mathbb{P}_h)}.$$

Finally, by taking the average over all problem instances $h>0$, and then, applying Jensen's inequality using concavity of the square root, we obtain

$$\frac{1}{m}\sum_{h=1}^{m}\mathbb{P}_h(J\in\mathcal{S}_h^*)\leq\frac{k}{m}+\frac{1}{m}\sum_{h=1}^{m}\sqrt{\frac{1}{2}\mathrm{KL}(\mathbb{P}_0||\mathbb{P}_h)} \tag{24}$$

$$\leq\frac{k}{m}+\sqrt{\frac{1}{2m}\sum_{h=1}^{m}\mathrm{KL}(\mathbb{P}_0||\mathbb{P}_h)} \tag{25}$$

where (24) is due to the fact that $\sum_h\sum_{V_i\in\mathcal{S}_h^*}\mathbb{P}_0(J=V_i)=k$ since each action V_i is optimal exactly for k problem instances out of m problem instances with $h>0$.

Third Step: Computation of $KL(\mathbb{P}_0||\mathbb{P}_h)$

Let Z_t be the set of observed edges and O_t be the set of influence outcomes on the observed edges in epoch t, $O_{1:t}:=\{O_1,\ldots,O_t\}$, and let $\mathbb{P}_h^t$ be the law of $O_{1:t}$ in problem instance h. Note that the sequence of observations $O_{1:T}$ deterministically determines the sequence of actions $\{S_1,\ldots,S_T\}$ taken by the learner, which in turn determines the law of J. Similarly, $O_{1:t-1}$ deterministically determines Z_t due to the structure of the influence graph. In order to proceed from (25), first by the data processing inequality, we obtain $\mathrm{KL}(\mathbb{P}_0||\mathbb{P}_h)\leq\mathrm{KL}(\mathbb{P}_0^T||\mathbb{P}_h^T)$. Then, we apply the chain rule for KL-divergence to get

$$\mathrm{KL}(\mathbb{P}_0||\mathbb{P}_h)\leq\mathrm{KL}(\mathbb{P}_0^T||\mathbb{P}_h^T)$$

$$=\mathrm{KL}(\mathbb{P}_0^1||\mathbb{P}_h^1)+\sum_{t=2}^{T}\mathbb{E}_{o_{1:t-1}}[\mathrm{KL}(\mathbb{P}_0^t(.|o_{1:t-1})||\mathbb{P}_h^t(.|o_{1:t-1}))]$$

$$=\mathrm{KL}(\mathbb{P}_0^1||\mathbb{P}_h^1)$$

$$+\sum_{t=2}^{T}\sum_{o_{1:t-1}}\mathbb{P}_0^{t-1}(o_{1:t-1})\mathrm{KL}(\mathbb{P}_0^t(.|o_{1:t-1})||\mathbb{P}_h^t(.|o_{1:t-1}))$$

$$=\mathrm{KL}(\mathbb{P}_0^1||\mathbb{P}_h^1)+\sum_{t=2}^{T}\sum_{o_{1:t-1}}\mathbb{P}_0^{t-1}(o_{1:t-1})\mathbf{1}_{\{(v_0^h,v_1^h)\in Z_t|o_{1:t-1}\}}$$

$$\mathrm{KL}\left(\frac{1-\epsilon}{2}\middle\|\frac{1+\epsilon}{2}\right) \tag{26}$$

$$=\mathrm{KL}\left(\frac{1-\epsilon}{2}\middle\|\frac{1+\epsilon}{2}\right)\mathbb{E}_0\left[\sum_{t=1}^{T}\mathbf{1}_{\{(v_0^h,v_1^h)\in Z_t\}}\right] \tag{27}$$

where (26) is due to the fact that KL-divergence is non-zero only when the edge (v_0^h,v_1^h) is observed since the influence probabilities corresponding to the other edges are the same for problem instances 0 and h.

Using the inequality, $\mathrm{KL}(p||q)\leq\frac{(p-q)^2}{q(1-q)}$ we obtain

$$\sum_{h=1}^{m}\mathrm{KL}(\mathbb{P}_0^T||\mathbb{P}_h^T)$$

$$=\mathrm{KL}\left(\frac{1-\epsilon}{2}\middle\|\frac{1+\epsilon}{2}\right)\sum_{h=1}^{m}\mathbb{E}_0\left[\sum_{t=1}^{T}\mathbf{1}_{\{(v_0^h,v_1^h)\in Z_t\}}\right]$$

$$=\mathrm{KL}\left(\frac{1-\epsilon}{2}\middle\|\frac{1+\epsilon}{2}\right)\mathbb{E}_0\left[\sum_{t=1}^{T}\sum_{h=1}^{m}\mathbf{1}_{\{(v_0^h,v_1^h)\in Z_t\}}\right]$$

$$\leq\frac{4\epsilon^2}{1-\epsilon^2}OT \tag{28}$$

where OT is the expected number of edges observed until epoch T.

Fourth Step: ϵ-tuning

Substituting (28) into (23) and setting $\epsilon \leq \frac{1}{2}$, we get

$$\sup_{h>0} R_h(T) \geq \epsilon T \left(1 - \frac{k}{m} - 4\epsilon\sqrt{\frac{O}{6m}T}\right). \tag{29}$$

Letting $\epsilon = \alpha\sqrt{\frac{m}{OT}}$ for some $\alpha > 0$ and $k' = \frac{m}{k}$ we obtain

$$\sup_{h>0} R_h(T) \geq \sqrt{\frac{mT}{O}} \left(\alpha - \frac{\alpha}{k'} - \frac{4\alpha^2}{\sqrt{6}}\right). \tag{30}$$

For $\alpha = \left(1 - \frac{1}{k'}\right)\frac{\sqrt{6}}{8}$, we have

$$\begin{aligned}\sup_{h>0} R_h(T) &\geq \left[\frac{\sqrt{6}}{8}\left(1 - \frac{1}{k'}\right)^2 - \frac{4}{\sqrt{6}}\frac{6}{64}\left(1 - \frac{1}{k'}\right)^2\right]\sqrt{\frac{mT}{O}} \\ &= \frac{\sqrt{6}}{16}\left(1 - \frac{1}{k'}\right)^2\sqrt{\frac{mT}{O}}.\end{aligned}$$

Hence, by the definition of $R_h(T)$ in (20), we have

$$\mathbb{E}_h[R^{(1,1)}(T)] - cOT \geq \frac{\sqrt{6}}{16}\left(1 - \frac{1}{k'}\right)^2\sqrt{\frac{mT}{O}} \tag{31}$$

for at least one problem instance.

This shows that for at least one problem instance h

$$\begin{aligned}&\mathbb{E}_h[R^{(1,1)}(T)] = R_h(T) + cOT \\ &\geq \frac{\sqrt{6}}{16}k_0\sqrt{\frac{mT}{O}} + cOT \\ &\geq \max\left\{1.88 \times \left(\frac{\sqrt{6}}{16}\right)^{2/3} k_0^{2/3}c^{1/3}m^{1/3}T^{2/3}, \frac{\sqrt{6}}{16}k_0\sqrt{T}\right\}\end{aligned} \tag{32}$$

where $k_0 = (1 - \frac{1}{k'})^2$, by setting $O = (\frac{\sqrt{6}}{32})^{2/3}k_0^{2/3}c^{-2/3}$ $m^{1/3}T^{-1/3}$ in (32) and using the fact that O is upper bounded by m.

REFERENCES

[1] A. O. Saritac, A. Karakurt, and C. Tekin, "Online contextual influence maximization in social networks," in *Proc. 54th Annu. Allerton Conf. Commun., Control, Comput.*, 2016, pp. 1204–1211.

[2] X. Cao, Y. Chen, C. Jiang, and K. J. R. Liu, "Evolutionary information diffusion over heterogeneous social networks," *IEEE Trans. Signal Inf. Process. Over Netw.*, vol. 2, no. 4, pp. 595–610, Dec. 2016.

[3] X. Cao, Y. Chen, and K. J. R. Liu, "Understanding popularity dynamics: Decision-making with long-term incentives," *IEEE Trans. Signal Inf. Process. Over Netw.*, vol. 3, no. 1, pp. 91–103, Mar. 2017.

[4] S. Chen, J. Fan, G. Li, J. Feng, K.-l. Tan, and J. Tang, "Online topic-aware influence maximization," in *Proc. Very Large Data Base Endow.*, 2015, vol. 8, pp. 666–677.

[5] N. Barbieri, F. Bonchi, and G. Manco, "Topic-aware social influence propagation models," in *Proc. IEEE 12th Int. Conf. Data Mining*, 2012, pp. 81–90.

[6] Ç. Aslay, N. Barbieri, F. Bonchi, and R. Baeza-Yates, "Online topic-aware influence maximization queries," in *Proc. 17th Int. Conf. Extending Database Technol.*, 2014, pp. 295–306.

[7] W. Chen, T. Lin, and C. Yang, "Real-time topic-aware influence maximization using preprocessing," *Comput. Soc. Netw.*, vol. 3, no. 1, pp. 1–19, 2016.

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

[9] Y. Tang, X. Xiao, and Y. Shi, "Influence maximization: Near-optimal time complexity meets practical efficiency," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2014, pp. 75–86.

[10] K. Jung, W. Heo, and W. Chen, "IRIE: Scalable and robust influence maximization in social networks," in *Proc. IEEE 12th Int. Conf. Data Mining*, 2012, pp. 918–923.

[11] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, "Cost-effective outbreak detection in networks," in *Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2007, pp. 420–429.

[12] A. Goyal, W. Lu, and L. V. Lakshmanan, "CELF++: Optimizing the greedy algorithm for influence maximization in social networks," in *Proc. 20th Int. Conf. Companion World Wide Web*, 2011, pp. 47–48.

[13] Y. Tang, Y. Shi, and X. Xiao, "Influence maximization in near-linear time: A martingale approach," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2015, pp. 1539–1554.

[14] W. Chen, Y. Wang, and S. Yang, "Efficient influence maximization in social networks," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2009, pp. 199–208.

[15] Z. Wen, B. Kveton, M. Valko, and S. Vaswani, "Online influence maximization under independent cascade model with semi-bandit feedback," in *Proc. Advances Neural Inf. Process. Syst.*, 2017, pp. 3026–3036.

[16] S. Vaswani and L. V. S. Lakshmanan, "Influence maximization with bandits," 2015, arXiv:1503.00024.

[17] W. Chen, Y. Wang, Y. Yuan, and Q. Wang, "Combinatorial multi-armed bandit and its extension to probabilistically triggered arms," *J. Mach. Learn. Res.*, vol. 17, no. 50, pp. 1–33, 2016.

[18] T. Lin, J. Li, and W. Chen, "Stochastic online greedy learning with semi-bandit feedbacks," in *Proc. Advances Neural Inf. Process. Syst.*, 2015, pp. 352–360.

[19] W. Chen, C. Wang, and Y. Wang, "Scalable influence maximization for prevalent viral marketing in large-scale social networks," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2010, pp. 1029–1038.

[20] J. Kim, S. K. Kim, and H. Yu, "Scalable and parallelizable processing of influence maximization for large-scale social networks," in *Proc. 29th Int. Conf. Data Eng.*, 2013, pp. 266–277.

[21] C. Borgs, M. Brautbar, J. Chayes, and B. Lucier, "Maximizing social influence in nearly optimal time," in *Proc. 25th Annu. ACM-SIAM Symp. Discrete Algorithms*, 2014, pp. 946–957.

[22] Q. Wang and W. Chen, "Improving regret bounds for combinatorial semi-bandits with probabilistically triggered arms and its applications," in *Proc. Advances Neural Inf. Process. Syst.*, 2017, pp. 1161–1171.

[23] Y. Bao, X. Wang, Z. Wang, C. Wu, and F. C. Lau, "Online influence maximization in non-stationary social networks," in *Proc. IEEE/ACM 24th Int. Symp. Quality Service*, 2016, pp. 1–6.

[24] S. Lei, S. Maniu, L. Mo, R. Cheng, and P. Senellart, "Online influence maximization," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2015, pp. 645–654.

[25] L. Li, W. Chu, J. Langford, and R. E. Schapire, "A contextual-bandit approach to personalized news article recommendation," in *Proc. 19th Int. Conf. World Wide Web*, 2010, pp. 661–670.

[26] L. Qin, S. Chen, and X. Zhu, "Contextual combinatorial bandit and its application on diversified online recommendation," in *Proc. SIAM Int. Conf. Data Mining*, 2014, pp. 461–469.

[27] S. Li, B. Wang, S. Zhang, and W. Chen, "Contextual combinatorial cascading bandits," in *Proc. 33rd Int. Conf. Mach. Learn.*, 2016, pp. 1245–1253.

[28] Y. Seldin, P. Bartlett, K. Crammer, and Y. Abbasi-Yadkori, "Prediction with limited advice and multiarmed bandits with paid observations," in *Proc. 31st Int. Conf. Mach. Learn.*, 2014, vol. 32, pp. 280–287.

[29] N. Cesa-Bianchi, Y. Freund, D. Haussler, D. P. Helmbold, R. E. Schapire, and M. K. Warmuth, "How to use expert advice," *J. ACM*, vol. 44, no. 3, pp. 427–485, 1997.

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

[31] N. Cesa-Bianchi and G. Lugosi, *Prediction, Learning, and Games*. New York, NY, USA: Cambridge Univ. Press, 2006.

[32] J.-Y. Audibert and S. Bubeck, "Regret bounds and minimax policies under partial monitoring," *J. Mach. Learn. Res.*, vol. 11, pp. 2785–2836, Oct. 2010.

[33] W. Chen, W. Hu, F. Li, J. Li, Y. Liu, and P. Lu, "Combinatorial multi-armed bandit with general reward functions," in *Proc. Advances Neural Inf. Process. Syst.*, 2016, pp. 1651–1659.

[34] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications*, vol. 8. New York, NY, USA: Cambridge Univ. Press, 1994.

Anıl Ömer Sarıtaç received the B.Sc. degree in industrial engineering from Bilkent University, Ankara, Turkey, in 2015, where he is currently working toward the Master's degree with the Industrial Engineering Department. His research interests include multi-armed bandit problems and applied machine learning and their business applications.

Altuğ Karakurt received the B.Sc. degree in electrical and electronics engineering from Bilkent University, Ankara, Turkey, in 2016, and the M.S. degree in electrical and computer engineering from The Ohio State University, Columbus, OH, USA, in 2018. His research interests include information theory, multi-armed bandit problems, communication networks and applied machine learning. He received the Best Paper Award from WiOpt 2018.

Cem Tekin (M'13) received the B.Sc. degree in electrical and electronics engineering from the Middle East Technical University, Ankara, Turkey, in 2008, and the M.S.E. degree in electrical engineering: systems, the M.S. degree in mathematics, and the Ph.D. degree in electrical engineering: systems from the University of Michigan, Ann Arbor, MI, USA, in 2010, 2011, and 2013, respectively. He is an Assistant Professor with the Electrical and Electronics Engineering Department, Bilkent University, Ankara. From February 2013 to January 2015, he was a Postdoctoral Scholar with the University of California, Los Angeles, CA, USA. His research interests include reinforcement learning, multi-armed bandit problems, data mining, multi-agent systems, and smart healthcare. He received the University of Michigan Electrical Engineering Departmental Fellowship in 2008, and the Fred W. Ellersick award for the best paper in MILCOM 2009.