Learning Peer Influence Probabilities with Linear Contextual Bandits

Ahmed Sayeed Faruk
afaruk2@uic.edu
University of Illinois Chicago
Chicago, USA

Mohammad Shahverdikondori
mohammad.shahverdikondori@epfl.ch
EPFL
Switzerland

Elena Zheleva
ezheleva@uic.edu
University of Illinois Chicago
Chicago, USA

Abstract

In networked environments, users frequently share recommendations about content, products, services, and courses of action with others. The extent to which such recommendations are successful and adopted is highly contextual, dependent on the characteristics of the sender, recipient, their relationship, the recommended item, and the medium, which makes peer influence probabilities highly heterogeneous. Accurate estimation of these probabilities is key to understanding information diffusion processes and to improving the effectiveness of viral marketing strategies. However, learning these probabilities from data is challenging; static data may capture correlations between peer recommendations and peer actions but fails to reveal influence relationships. Online learning algorithms can learn these probabilities from interventions but either waste resources by learning from random exploration or optimize for rewards, thus favoring exploration of the space with higher influence probabilities. In this work, we study learning peer influence probabilities under a contextual linear bandit framework. We show that a fundamental trade-off can arise between regret minimization and estimation error, characterize all achievable rate pairs, and propose an uncertainty-guided exploration algorithm that, by tuning a parameter, attains any pair within this trade-off. Our experiments on semi-synthetic network datasets show the advantages of our method over static methods and contextual bandits that ignore this trade-off.

1 Introduction

Influence occurs when the action of one user affects the actions of other users in a social network, spreading information, behaviors, and attitudes. Influence is an important driver of user behavior in many digital platforms, including social media, e-commerce, and content platforms [19, 25, 45]. Influence probability, the extent to which one user can influence another, varies with users' network positions, connections, and individual traits. For example, in a social network, sharing a recommendation can result in adoption by some friends and in lack of interest by others. Predicting and leveraging these heterogeneous influence probabilities is very important for understanding information diffusion processes and for improving the effectiveness of viral marketing strategies.

Estimating influence probabilities from correlations between user actions is challenging because correlations can be attributed to other potential sources, such as homophily, confounding factors, and mere coincidence [3, 36]. Homophily refers to the tendency of similar individuals to form connections and exhibit similar behaviors [32]. Confounding factors, on the other hand, can simultaneously affect the formation of friendships and subsequent actions [31, 36]. Peer influence probabilities are higher among strong ties than weak ties, as individuals with stronger connections are more likely to influence each other [13]. Tie strengths have been estimated from correlations [16, 27, 34] but without disentangling their source. Nevertheless, many current influence-learning methods [9, 16, 27, 34, 38, 47, 49] rely on correlational or observational data, which are inherently non-causal.

Contextual multi-armed bandits (CMABs) [10] and online learning [35] algorithms have the potential to learn influence probabilities from interventions rather than pure correlations. However, these algorithms optimize for different objectives. CMABs offer a principled framework for sequential, context-aware interventions but they aim to minimize cumulative regret, which may limit exploration to the regions with high influence probabilities, this resulting in high influence probability estimation error. Online learning methods can optimize for estimation error by exploring at random, but this approach is resource-inefficient and lacks regret guarantees. A few works have considered multi-armed bandits in networked settings where actions can propagate through peer influence [2, 12, 20, 22, 23, 39, 42–44, 48]. Some of these works leverage heterogeneous influence probabilities, assuming that they are known, for tasks such as predicting recommendations [12] and influence maximization [20, 43]. Others learn the influence probabilities in the context of influence maximization, thus focusing on learning and leverage high influence probabilities [39, 42]. More recently, research on MABs with interference has studied the setting when treatments are simultaneously assigned to all nodes and rewards depend on neighbors' actions [2, 22, 23, 44, 48]. However, none of these studies address the task of learning peer influence probabilities as an independent problem.

In this paper, we study the problem of learning peer influence probabilities from k interventions per round in a network, where interventions may be selected either from the whole network or from the neighbors of specific nodes. Here, an intervention refers to the system showing the action of one user (e.g., social media post or product recommendation) to other user(s) to see whether the recipient(s) would take a subsequent action on it (e.g., share the post or buy the product). Alternatively, the intervention can be the user sharing their action with peer(s) suggested by the system. In both cases the system chooses the peers. Note that in general, such interventions can have privacy concerns if the original action was private. In this work, we focus on actions that are visible to the peer (e.g., the post/recommendation are public) and use features that do not have privacy concerns (e.g., from publicly available information). We cast this problem as a linear contextual bandit task and prove the existence of a fundamental trade-off between cumulative regret and influence probability estimation error. We characterize all achievable rate pairs for this trade-off and show that no algorithm can achieve the optimal rates for both objectives simultaneously.

Building on this theory, we propose **Influence Contextual Bandit** ($InfluenceCB$), a framework that alternates between uncertainty-guided exploration and reward-oriented exploitation. $InfluenceCB$ uses an objective-driven uncertainty threshold to decide when to explore uncertain edges versus exploit high-probability edges and employs a parameter β to navigate the trade-off frontier. Our framework can incorporate various features, including node, edge, and network attributes, into the learning process.

Key idea and highlights. To summarize, this paper makes the following contributions:

- We formulate the problem of learning heterogeneous peer influence probabilities in networks as a bi-objective contextual multi-armed bandit, aiming to simultaneously minimize cumulative regret and estimation error.
- We theoretically establish a fundamental trade-off between regret minimization and estimation error and characterize the achievable rate pairs.
- We propose $InfluenceCB$, a CMAB framework that incorporates uncertainty-guided exploration to accelerate influence probability learning while minimizing resource waste.
- We provide theoretical guarantees for linear settings and show that $InfluenceCB$ can achieve any point on the trade-off curve by tuning a single parameter.
- We evaluate $InfluenceCB$ on semi-synthetic network datasets and show its effectiveness over methods based on observational data or conventional contextual bandits.

2 Related Work

Link weight prediction focuses on estimating the numerical value (e.g., strength, capacity) associated with an edge in a network [14]. Previous studies have explored influence probability estimation either offline from historical cascades using propagation models or likelihood-based inference [9, 16, 27, 34], or have focused on structural patterns [47] and influence tracking or maximization without learning edge-level probabilities [38, 49]. In contrast, our work learns edge-level influence probabilities online through interventions, explicitly modeling the exploration–exploitation trade-off.

Learning influence probabilities with bandits has been studied for problems such as probabilistic maximum coverage and social influence maximization in viral marketing [39, 42]. However, they are interested in learning and leveraging high influence probabilities, rather than the full range of probabilities. A related recent direction is multi-armed bandits with network interference [2, 22, 23, 44, 48], where treatments are simultaneously assigned across all nodes and rewards depend on neighbors' treatments. The trade-off between regret and estimation error has also been investigated in this context of node interventions [48]. In contrast, our framework learns peer influence probabilities by selecting a small subset of edges per round and observing samples of their influence.

While extensive research exists on top-k recommendation problems (e.g., [21, 37, 46]), including with CMABs [4], limited attention has been given to identifying top-k edges in social networks. A recommendation model for socialized e-commerce focuses on recommending top-k relevant products to a user for sharing with all neighbors [15] but no work focuses on recommending top-k neighbors for sharing a fixed product.

3 Problem Description

We represent the data as a bidirected attributed network $G = (V, E)$, where $V = \{v_1, v_2, \ldots, v_n\}$ is the set of n nodes, and $E = \{e_{ij} \mid 1 \leq i, j \leq n\}$ is the set of edges, with e_{ij} denoting an edge between nodes v_i and v_j. The neighborhood of a node v_i is defined as $\mathcal{N}_i = \{v_j \in V \mid e_{ij} \in E\}$, and each $v_j \in \mathcal{N}_i$ is a neighbor of v_i. The attribute vector $e_{ij}.X$ is referred to as the context vector for edge $e_{ij} \in E$. It includes attributes of nodes v_i and v_j and other relevant information, such as the recommended item attributes and network structural features (e.g., number of common neighbors). Let $v_i.y \in \{0, 1\}$ denote the activation status of node v_i. Specifically, $v_i.y = 1$ indicates that v_i is activated, while $v_i.y = 0$ indicates no activation.

In our setup, a node v_i takes an action (e.g., creates a post or buys a product) and shares their action with a neighbor based on a peer recommendation from the system. The system then observes whether the neighbor gets activated by taking a subsequent action (e.g., shares the post or uses referral link to buy the product), reflecting peer influence. Each edge $e_{ij} \in E$ is associated with a peer influence probability $e_{ij}.p \in [0, 1]$, which represents the probability of activating node v_j (recipient node) due to peer influence from active node v_i (source node). The peer influence probability can be asymmetric ($e_{ij}.p \neq e_{ji}.p$) and heterogeneous across edges ($e_{ij}.p \neq e_{il}.p$, $e_{il}.p \neq e_{jl}.p$). In our problem setup, these probabilities are unknown.

We consider a stochastic M-armed contextual linear bandit setting over T rounds. The set of arms is denoted by $\mathcal{A} = \{X_1, X_2, \ldots, X_M\} \subseteq \mathbb{R}^d$, where each X_m corresponds to the context vector for an edge $e_{i,j}$ and M corresponds to the number of edges. At each round $t \in \{1, \ldots, T\}$, the agent observes a pool of available actions $\mathcal{A}_t \subseteq \mathcal{A}$ and selects k actions $\{X_{t,1}, \ldots, X_{t,k}\}$ where $k \leq |A_t|$ and $X_{t,i}$ denotes the i-th action chosen at round t. The action pool $\mathcal{A}_t$ may include edges from one node's neighborhood or from multiple nodes' neighborhoods, and its size may vary over different rounds. After receiving the binary rewards $\{r_{t,1}, \ldots, r_{t,k}\}$, the agent updates its selection policy.

Each reward corresponds to a successful activation event: a reward of 1 indicates that an inactive node v_j was activated due to influence from an active node v_i across edge $e_{i,j}$, and 0 otherwise. We assume a linear reward model with an unknown parameter $\theta^* \in \mathbb{R}^d$, such that

$$\mathbb{E}[r_{t,m}] = X_{t,m}^\top \theta^*.$$

The estimated peer influence probability thus corresponds to the probability of observing reward 1. The set of top-k edges in round t is defined as

$$\mathcal{E}_t^* \leftarrow \underset{X_t \in \mathcal{A}_t}{\arg\operatorname{top}_k} \; X_t^\top \theta^*.$$

Hence, any problem instance is characterized by $(\mathcal{A}, \theta^*)$. We further assume bounded norms for actions and parameters: there exist constants $c, c' > 0$ such that $\|X\|_2 \leq c$ for all $X \in \mathcal{A}$ and $\|\theta^*\|_2 \leq c'$. We also assume that the set of actions spans $\mathbb{R}^d$. These are standard assumptions in the linear bandit literature [28]. The set of all problem instances satisfying these conditions with d-dimensional actions and binary rewards is denoted by $\mathcal{E}$. We can now define our two objectives of interest.

Regret. The expected cumulative regret of a policy π interacting with an environment characterized by $(\mathcal{A}, \theta^*)$ over T rounds is

defined as

$$\text{Regret}_T(\pi, \mathcal{A}, \theta^*) \;=\; \sum_{t=1}^{T} \left(\sum_{X \in \mathcal{E}_t^*} X^\top \theta^* \;-\; \sum_{i=1}^{k} X_{t,i}^\top \theta^* \right),$$

where $\mathcal{E}_t^*$ denotes the set of top-k optimal actions at round t. The goal in standard contextual bandits is to minimize $\text{Regret}_T(\pi, \mathcal{A}, \theta^*)$. Since θ^* is unknown during the learning process, different bandit algorithms employ various strategies to minimize regret by selecting high-reward actions $X_{t,i}$. However, focusing solely on regret minimization does not necessarily lead to an accurate estimation of peer influence probabilities. We therefore introduce a second objective that explicitly measures estimation accuracy.

RMSE. The expected root mean squared error (RMSE) of the estimated influence probabilities over all edges serves as our second objective. For a policy π, its value after T rounds is defined as

$$\text{RMSE}_T(\pi, \mathcal{A}, \theta^*) \;=\; \mathbb{E}\left[\sqrt{\frac{1}{|\mathcal{A}|} \sum_{X \in \mathcal{A}} \left(X^\top (\hat{\theta}_T - \theta^*) \right)^2} \right],$$

where $\hat{\theta}_T$ denotes the parameter estimate obtained after T rounds.

The two objectives—minimizing cumulative regret and minimizing estimation error—are inherently in conflict. Minimizing regret requires focusing on high-reward edges, biasing exploration toward frequently activated neighbors and leaving large regions of the influence-probability space underexplored. Conversely, reducing estimation error necessitates exploring uncertain or low-reward edges, which inevitably increases regret. As a result, no single policy can simultaneously minimize both objectives to their global optima.

Motivated by this intuition, we measure the performance of any sampling policy π by a pair $(\text{Regret}_T(\pi), \text{RMSE}_T(\pi))$, where each term denotes the worst-case value of the corresponding objective:

$$\text{Regret}_T(\pi) = \max_{(\mathcal{A}, \theta^*) \in \mathcal{E}} \text{Regret}_T(\pi, \mathcal{A}, \theta^*),$$
$$\text{RMSE}_T(\pi) = \max_{(\mathcal{A}, \theta^*) \in \mathcal{E}} \text{RMSE}_T(\pi, \mathcal{A}, \theta^*).$$

To compare policies, we characterize optimality in terms of *Pareto domination*. A policy π^* is Pareto-optimal if there exists no other policy π such that

$$\text{Regret}_T(\pi) \le \text{Regret}_T(\pi^*) \quad \text{and} \quad \text{RMSE}_T(\pi) \le \text{RMSE}_T(\pi^*),$$

with at least one inequality strict. The set of all Pareto-optimal policies defines the *Pareto frontier*, representing the fundamental boundary of achievable trade-offs: any improvement in one objective beyond this frontier necessarily worsens the other. Our goal is therefore not to find a single globally optimal policy but to design bandit algorithms that enable controlled movement along the Pareto frontier, allowing practitioners to select policies aligned with desired trade-offs that are application-specific.

Problem 1. ***Learning Peer Influence Probabilities under Bi-Objective Trade-off*** *Given an attributed network $G = (V, E)$, the goal is to learn peer influence probabilities $e_{ij}.p$ over T rounds by selecting top-k edges in each round to minimize both regret and estimation error.*

These two objectives – cumulative regret Regret_T and estimation error RMSE_T – are inherently conflicting and cannot be simultaneously minimized. The solution space is therefore characterized by a Pareto frontier: $\min_\pi F(\pi) = \left[\text{Regret}_T(\pi),\ \text{RMSE}_T(\pi) \right]$, where each Pareto-optimal policy represents a trade-off where improving one objective inevitably worsens the other. The problem thus requires designing algorithms that enable controlled movement along this frontier, allowing selection of policies that best align with desired trade-offs that are application-specific.

4 Estimation-Regret Trade-Off

In this section, we present theoretical results demonstrating the existence of a fundamental trade-off, in the worst case, between two objectives: minimizing cumulative regret and minimizing the RMSE of estimating the influence probabilities. We begin by establishing the achievable rates for each objective individually, thereby illustrating the inherent difficulty of both tasks. We then prove that there exist instances in which a trade-off arises between these two objectives, showing that no algorithm can achieve the optimal rate for both simultaneously. Finally, we propose an algorithm that, by tuning a parameter, can interpolate between these objectives and attain any achievable pair of rates implied by the lower bound.

Note that in the described problem setting, it is not possible to generally establish a worst-case bound on the objectives, particularly for the RMSE. The main difficulty arises because, at each round t, nature selects the set of available actions $\mathcal{A}_t$ (e.g., which specific users have taken an action to be shared with their friends), and the algorithm must then choose k of its members to play. Since $\mathcal{A}_t$ is determined by nature and may vary across rounds, a uniform worst-case bound on the RMSE cannot be guaranteed.

For example, suppose the actions can be partitioned into two sets S_1 and S_2 such that all vectors $X_m \in S_1$ are nearly collinear (i.e., point in almost the same direction), while the vectors in S_2 lie far from that direction. If nature selects $\mathcal{A}_t \subseteq S_1$ for most rounds, the estimation of the parameter vector θ^* will be poor in directions corresponding to S_2. Consequently, the estimation errors $|X_m^\top \hat{\theta}_T - X_m^\top \theta^*|$ will remain large for each $X_m \in S_2$, resulting in a high RMSE.

Motivated by this limitation, we establish our theoretical results in a slightly modified setting where nature cannot restrict the action set across rounds, and the agent has access to a fixed set of actions throughout. Our theoretical bounds and guarantees are derived for this fixed-action setting, while our experiments are conducted under the more general setting introduced earlier. Formally, for the theoretical analysis, we assume that the set of available actions at each round is constant, i.e., $\mathcal{A}_t = \mathcal{A}$ for all t. We now proceed to provide bounds for each objective under this setting to highlight their inherent difficulty.

For the regret minimization objective, the agent may choose any set of k actions from $\mathcal{A}$ and then observe binary rewards indicating whether activation is successful for each of the selected arms. This setup corresponds to the well-studied *combinatorial linear bandit* problem with semi-bandit feedback [5, 8, 11, 41]. For this problem, the CombLinUCB algorithm introduced in [41] achieves the following worst-case regret upper bound over T rounds: $\tilde{O}\left(kd\sqrt{T}\right)$ where $\tilde{O}$ hides constant and logarithmic factors. Regarding tightness, a

lower bound of $O\left(\sqrt{kdT}\right)$ was established in [5]. Although these bounds do not match, they are close and can capture the inherent difficulty of the regret minimization problem.

For the objective of minimizing RMSE, we propose an algorithm and prove a worst-case upper bound on its RMSE over T rounds. At each round t, the agent selects the edges for activation, together with their feature vectors $\{X_{t,1}, X_{t,2}, \ldots, X_{t,k}\}$. We define

$$V_t = \lambda \mathbb{I} + \sum_{s=1}^{t} \sum_{r=1}^{k} X_{s,r} X_{s,r}^\top, \tag{1}$$

where $\lambda > 0$ is a regularization constant. It is well known in the linear bandit literature that, for each $X_m \in \mathcal{A}$, the quantity $\sqrt{X_m^\top V_t^{-1} X_m}$ would characterize the estimation uncertainty of the influence probability associated with feature vector X_i after t rounds. Motivated by this intuition, to minimize the maximum estimation uncertainty over all actions, the classical linear bandit problem (corresponding to the case $k = 1$) relies on the G-optimal design, which is obtained as the solution to the following optimization problem:

$$\min_{\mathbf{w} \in \Delta^{M-1}} \max_{X_i \in \mathcal{A}} X_i^\top \left(\sum_{j=1}^{M} w_j X_j X_j^\top \right)^{-1} X_i, \tag{2}$$

where Δ^{M-1} denotes the $(M-1)$-dimensional simplex. In this setting, given the optimal solution $\mathbf{w}^*$, the algorithm that plays each arm X_i for Tw_i^* rounds minimizes the maximum estimation error of mean rewards.

Our setting can be viewed as a linear bandit problem with kT total plays. The key difference is that at each round, the agent must choose k distinct arms, meaning no arm can be played more than T times out of the total kT selections. Consequently, if we were to adopt the G-optimal design, some coordinates of the optimal design vector $\mathbf{w}^*$ might exceed $\frac{1}{k}$, which is infeasible in our setting. To address this, we consider a constrained design that minimizes the maximum uncertainty while ensuring that no arm is selected more than T times. Formally, we define the optimization problem

$$f^*(\mathcal{A}, k) = \min_{\mathbf{w} \in \Delta_{1/k}^{M-1}} \max_{X_i \in \mathcal{A}} X_i^\top \left(\sum_{j=1}^{M} w_j X_j X_j^\top \right)^{-1} X_i, \tag{3}$$

where $\Delta_{1/k}^{M-1} = \left\{ \mathbf{w} \in \Delta^{M-1} \mid \forall i : w_i \leq \frac{1}{k} \right\}$ is the subset of the simplex in which all entries are bounded above by $\frac{1}{k}$.

Lemma 4.1 (RMSE Upper Bound). *For any instance $(\mathcal{A}, \theta^*) \in \mathcal{E}$, consider the algorithm that allocates its kT selections proportionally to $\mathbf{w}_k^*$, the optimizer of* (3)*. Then the final (expected) RMSE satisfies*

$$\mathrm{RMSE}_T(\pi, \mathcal{A}, \theta^*) \in \tilde{O}\left(\sqrt{\frac{f^*(\mathcal{A}, k)}{kT}} \right),$$

where $\tilde{O}(\cdot)$ hides logarithmic factors.

All the theoretical proofs are provided in Appendix 8. In the special case where the optimal solution of the unconstrained problem (2) already lies in $\Delta_{1/k}^{M-1}$ and $\mathrm{span}(X_1, \ldots, X_M) = \mathbb{R}^d$, we have $f^*(\mathcal{A}, k) = d$; otherwise, $f^*(\mathcal{A}, k)$ is larger.

The bounds above imply that the optimal rates in T for cumulative regret and RMSE are $T^{\frac{1}{2}}$ and $T^{-\frac{1}{2}}$, respectively. We now show that there exist instances in which these two objectives conflict and no algorithm can achieve both optimal rates simultaneously.

Theorem 4.2 (RMSE–Regret Trade-Off). *For any $M > k$, there exists an instance $(\mathcal{A}, \theta^*)$ such that, for any policy π with*

$$\mathrm{Regret}_T(\pi, \mathcal{A}, \theta^*)) \in O\left(T^{2\beta}\right),$$

we have

$$\mathrm{RMSE}_T(\pi, \mathcal{A}, \theta^*)) \in \Omega\left(T^{-\beta}\right),$$

where $\beta \leq \frac{1}{2}$, and by the regret lower bound, $\frac{1}{4} \leq \beta$.

By Theorem 4.2, if an algorithm achieves the optimal rate for cumulative regret (i.e., $\beta = \frac{1}{4}$, giving $R_T = O(T^{1/2})$), then its RMSE rate must be non-optimal, namely $\Omega\left(T^{-\frac{1}{4}}\right)$. Conversely, if an algorithm achieves the optimal RMSE rate (i.e., $\beta = \frac{1}{2}$, giving $\mathrm{RMSE}_T = \Theta(T^{-1/2})$), then its cumulative regret is necessarily linear.

An important question is whether one can achieve any pair of rates satisfying the condition in Theorem 4.2. In the next section, we propose an algorithm that, based on the parameter β, attains any feasible pair of rates implied by the theorem.

5 Influence Contextual Bandit framework

We propose an **Influence C**ontextual **B**andit (*InfluenceCB*) framework that simultaneously learns peer influence probabilities and leverages them to recommend top-k edges for intervention. In each round, the framework chooses between uncertainty-guided exploration and reward-guided exploitation. The details of the two phases are presented next, with the pseudo-code provided in Algorithm 1. Then we present a theorem on the optimality of *InfluenceCB*.

5.1 Uncertainty-guided Exploration

Contextual bandits face the well-known cold-start problem: in early rounds, limited feedback makes reward estimation highly unreliable, particularly in the setting of influence probability learning. To mitigate this, *InfluenceCB* employs an uncertainty-guided exploration strategy. In each round, the framework computes the estimation uncertainty for all candidate edges from the action pool. It considers two parameters, β and C which determine different strategies for the tradeoff between regrets and RMSE. As discussed in the previous section, β reflects our preference for prioritizing each of the objectives; for example, $\beta = 1/4$ optimizes for regret, and $\beta = 1/2$, for RMSE. The second parameter C is learned and the learning procedure reflects whether we would like to achieve a tradeoff between regrets and RMSE or optimize fully for one of the objectives. If the maximum uncertainty across edges exceeds the threshold C/t^β in a particular round t, the algorithm selects the exploration phase in that round. This adaptive thresholding mechanism ensures that actions with high uncertainty are explored sufficiently often, progressively reducing their uncertainty in subsequent rounds. As a result, the uncertainties of all actions remain at most on the order of $O(T^{-\beta})$, which leads to the corresponding upper bound on the RMSE. In this phase, the top-k edges with the highest uncertainty scores are recommended for intervention. The rewards are observed for the selected edges, and the CMAB parameters are updated accordingly. By prioritizing uncertain edges, the

Algorithm 1 Influence Contextual Bandit Framework

Input: rounds T; CMAB hyperparameter α; interventions per round k; exploration parameter β; objective $O \in \{\text{Regret}, \text{RMSE}\}$, linear CMAB $f(\mathcal{F}_t)$.
Output: Set of k edges with features $\mathbf{X}_t = \{X_{t,1}, X_{t,2}, \ldots, X_{t,k}\}$ to intervene on at each round t.
for each round $t = 1, 2, \ldots, T$ **do**
  Agent receives the pool of available actions $\mathcal{A}_t \in \mathcal{A}$.
  for each $X \in \mathcal{A}_t$: $U_t(X) \leftarrow \sqrt{X^\top V_{t-1}^{-1} X}$, where V_{t-1} is defined in (1).
  Set $u_t \leftarrow \max_{X \in \mathcal{A}_t} U_t(X)$, $au_t \leftarrow mean_{X \in \mathcal{A}_t} U_t(X)$.
  $C_t \leftarrow UpdateC(O, au_t, \text{activation rate})$ ▷ *Algorithm 2*
  if $u_t > C_t / t^\beta$ **then** ▷ *Exploration (uncertainty-guided)*
    $\mathbf{X}_t \leftarrow \arg\underset{X \in \mathcal{A}_t}{top_k} U_t(X)$
  else ▷ *Exploitation (reward-guided)*
    $\mathbf{X}_t \leftarrow f(\mathcal{F}_{t-1})$
  end if
  Observe rewards $r_{t,1}, r_{t,2}, \ldots, r_{t,k}$, and update $\hat{\theta}_t$, V_t.
end for

framework increases edge diversity and accelerates the estimation of influence probabilities, albeit at the cost of potentially higher short-term regret.

5.2 Reward-guided Exploitation

When the maximum edge uncertainty falls below the threshold C/t^β in a particular round t, $InfluenceCB$ is more confident in its predictions and selects the exploitation phase in that round. Given a fixed β, increasing C biases the bandit towards exploitation, leading to lower cumulative regret $Regret_T$ but higher estimation error $RMSE_T$. Here, the goal is to only focus on minimizing the regret. This step is performed by incorporating a linear CMAB algorithm designed for regret minimization, which is the input to our algorithm, and our algorithm plays based on its policy. We denote the sampling policy of the CMAB algorithm at round t by $f(\mathcal{F}_{t-1})$, where $\mathcal{F}_{t-1}$ is the history of interaction until round $t-1$. The rewards from the corresponding edge actions are observed, and the CMAB parameters are updated to refine subsequent reward estimates. By exploiting the learned knowledge, the framework gradually improves the quality of interventions while reducing regret over rounds.

5.3 Illustration of the $InfluenceCB$ Algorithm

Algorithm 1 shows the pseudo-code for $InfluenceCB$ which we explain next. The $InfluenceCB$ algorithm operates over discrete rounds $t = 1, 2, \ldots, T$. In each round t, the following steps are executed:

(1) **Action Pool Observation.** The agent receives the set of candidate edge contexts $\mathcal{A}_t$ from the network, representing the pool of available actions for round t *[Line 4]*.
(2) **Uncertainty Estimation.** For each edge e_{ij} with feature vector $X \in \mathcal{A}_t$, the algorithm computes the associated uncertainty $U_t(X) = \sqrt{X^\top V_{t-1}^{-1} X}$ *[Line 5]*.
(3) **Maximum Uncertainty Computation.** The maximum uncertainty $u_t = \max_{X \in \mathcal{A}_t} U_t(X)$ is computed to quantify the exploration potential of the current action pool *[Line 6]*.
(4) **Threshold Computation.** A dynamic threshold C_t is computed according to the optimization objective O *[Lines 7]*:
   - If O = RMSE, C_t is derived from uncertainty statistics to encourage exploration (Algorithm 2 in Appendix).
   - If O = Regret, C_t is computed from activation statistics to bias exploitation (Algorithm 2 in Appendix).

   C_t adapts over time based on recent batch-level statistics. In the uncertainty-based version, it decreases when current uncertainty exceeds its historical mean to encourage exploration and increases when it falls below the historical mean to favor exploitation. In the activation-based version, it increases when current activation rates drop below their historical mean, steering the model toward exploitation.
(5) **Exploration–Exploitation Decision.** If $u_t > C_t / t^\beta$, the algorithm enters the *exploration phase*, selecting the top-k edges with the highest uncertainty scores. Otherwise, it enters the *exploitation phase*, where the algorithm plays by the policy of the input linear CMAB algorithm *[Lines 8–12]*. For the CMAB algorithm, we can incorporate CombLinUCB, CombLinTS [41], or any other algorithm with similar regret bounds.
(6) **Reward Observation and Parameter Update.** After selecting the intervention set X_t, the algorithm observes the rewards associated with the corresponding edges and model parameters are updated with the new feedback *[Lines 13]*.

The computational complexity of $InfluenceCB$ is governed primarily by the size of $\mathcal{A}_t$. Its adaptive exploration–exploitation mechanism, modulated by β and C_t, enables continuous navigation along the Pareto frontier, balancing regret minimization against estimation accuracy.

The next theorem shows that, by choosing β, the algorithm attains all rates on the Pareto frontier (in T) implied by the lower bound in Theorem 4.2. The statement below assumes constant action sets and thresholds ($\forall t : \mathcal{A}_t = \mathcal{A}$ and $C_t = C$); in Section 6, we show that the algorithm performs well even when $\mathcal{A}_t$ varies over time and we propose adaptive procedures to set the C_t's.

Theorem 5.1 (Achievability of the Pareto-Frontier). *For any $\beta \in \left[\frac{1}{4}, \frac{1}{2}\right]$, Algorithm 1 initialized with parameter β on any instance $(\mathcal{A}, \theta^*)$ with $\forall t : \mathcal{A}_t = \mathcal{A}, C_t = C$, where C is a positive constant, satisfies*

$$\text{Regret}_T\left(\textit{InfluenceCB}, \mathcal{A}, \theta^*\right) \in \tilde{O}\left(T^{2\beta}\right),$$
$$\text{RMSE}_T\left(\textit{InfluenceCB}, \mathcal{A}, \theta^*\right) \in \tilde{O}\left(T^{-\beta}\right),$$

where $\tilde{O}(\cdot)$ hides logarithmic factors.

Proof Sketch. Here, we provide a high-level overview of the proof with full proof in the Appendix). We first show that the uncertainty of any arm $X_m \in \mathcal{A}$ decreases at least on the order of $\frac{1}{\sqrt{N_t(X_m)}}$, where $N_t(X_m)$ denotes the number of times arm X_m has been selected up to round t. Next, we show that, under the exploration condition of our algorithm, the total number of exploration

rounds is bounded by $O(T^{2\beta})$, which, combined with the regret bound of the regret minimization subroutine, implies the regret upper bound. Finally, we prove that the proposed exploration mechanism guarantees that the uncertainty of each arm remains at most on the order of $O(T^{-\beta})$, yielding an estimation accuracy of the same order and establishing the RMSE upper bound.

6 Experiments

We evaluate the performance of *InfluenceCB* on semi-synthetic network datasets against state-of-the-art contextual bandit algorithms, demonstrating its advantages and revealing the trade-off between minimizing regret and minimizing estimation error.

6.1 Data representation

We generate ground truth peer influence probabilities for two real-world attributed network datasets. BlogCatalog[1] is a network of social interactions among bloggers listed on the BlogCatalog website. It consists of 5,196 nodes, 343,486 edges, 8,189 attributes, and 6 labels. Labels correspond to topic categories inferred from blogger metadata. Flickr[1] is a benchmark social network dataset containing 7,575 nodes, 479,476 edges, 12,047 attributes, and 9 labels. Nodes represent users, edges denote following relationships, and labels indicate user group interests.

6.1.1 Construction of edge feature representations. The feature representation of an edge $e_{ij} \in E$ is constructed by concatenating the feature vectors of its endpoints v_i and v_j, capturing both node attributes and network information. To represent the attributes of each node in the edge, we apply truncated SVD [17] and project its attributes into a 64-dimensional space. To capture structural properties, we generate 64-dimensional node embeddings using a two-layer GraphSAGE model with mean aggregation [18]. Finally, we concatenate the 64-dimensional attribute representation with the 64-dimensional node embedding to form a unified 128-dimensional feature vector for each node, yielding a 256-dimensional edge feature vector.

6.1.2 Generation of synthetic ground truth peer influence probabilities (linear). The peer influence probability associated with edge e_{ij} is synthetically generated as a linear combination of the features:

$$e_{ij}.p = \sigma\left(\mathbf{w}^\top e_{ij}.X\right),$$

where $\mathbf{w} \in \mathbb{R}^{256}$ is a weight vector with entries sampled uniformly at random from $[-1, 1]$, and $\sigma(\cdot)$ is a min-max scaling function ensuring $e_{ij}.p \in (0, 1)$.

6.2 Evaluation metrics

We evaluate the performance of the *InfluenceCB* using the following metrics:

6.2.1 $Regret_T$. To evaluate regret, we compute the expected activations of the k recipient nodes influenced by their corresponding source nodes across the k $\{e_{ij}\}$, selected by *InfluenceCB* in round t. These activations are estimated through simulations using the ground-truth peer influence probabilities. We then compare this value against the expected activations obtained from the top-k edges $\{e^*_{ij}\}$, representing the maximum achievable expected activations under the same simulation setup in round t. The cumulative regret over T rounds, $Regret_T$, is then computed following the formulation defined in Section 3.

6.2.2 $RMSE_T$. To evaluate the accuracy of the learned peer influence probabilities, we compute the root mean squared error in round T, $RMSE_T$, over a randomly sampled subset of edges $\overline{E} \subset E$ from the network. The estimation error is calculated as: $RMSE_T = \sqrt{\frac{\sum_{e_{ij} \in \overline{E}} (\hat{e}_{ij}.p - e_{ij}.p)^2}{|\overline{E}|}}$, where $\hat{e}_{ij}.p = min(1, max(0, X^\top \theta^*))$ is the estimated peer influence probability in round T and , computed following the formulation defined in Section 3.

6.3 Main Algorithms and Baselines

6.3.1 ***Static Models.*** We compare our proposed *InfluenceCB* framework against several representative static baselines that do not learn peer influence probabilities over rounds but use the attributes from 6.1.1.

Random. Each edge in the network is pre-assigned a random score sampled uniformly from $[0, 1]$. In every round, this baseline selects k edges with the highest random scores from the candidate pool.

Similarity. Following [24], this method selects edges based on the similarity of their endpoint features. In each round, it selects the k edges with the smallest Euclidean distance between the feature vectors of the two endpoint nodes.

Link Prediction (Ridge). This baseline adopts a supervised learning approach to predict edge weights, following the learning based link prediction framework in [26]. Existing edges E are labeled as 1, while an equal number of randomly sampled non-existing edges are labeled as 0. Each edge (existing or non-existing) is represented by the concatenation of the 256-dimensional edge feature vector from 6.1.1. A ridge regression model is trained on this labeled data to estimate link scores for all edges $e_{ij} \in E$. In each round, the k edges with the highest predicted scores are selected.

Link Prediction (GraphSAGE). This method leverages node embeddings generated by the GraphSAGE algorithm [18]. Similar to the Ridge baseline, the concatenated embeddings of node pairs are passed through a neural network-based scoring function to produce link scores. The top-k edges ranked by these scores are selected in each round. We follow a hyperparameter setting [30] for this baseline.

Link Prediction (NCNC). This baseline employs the Neural Common Neighbor (NCNC) model [40], a state-of-the-art link prediction approach to generate the link score of all edges in the network, where each node is represented by the 128-dimensional feature vectors from 6.1.1. In each round, candidate edges are ranked by their link scores, and the top-k edges are selected. We follow the authors' default configuration [40] for hyperparameter setting.

6.3.2 ***Online Learning Models*** . We employ an online learning baseline in which model parameters are updated incrementally using Stochastic Gradient Descent (SGD) [7]. Each of the k selected edges triggers a gradient update, allowing the model to adapt continuously without retraining from scratch. Specifically, we adopt `SGDClassifier` with the `partial_fit` routine [33], where each

[1] All datasets available at https://renchi.ac.cn/datasets/

of the k selected edges is labeled as positive if the recipient node becomes activated and negative otherwise. In each round, edges in the pool are ranked by their SGD-estimated probability of being positive, and the k edges are selected. The model parameters are then updated online by treating edges with activated and inactive recipient nodes as positive and negative examples, respectively. We consider two variants of this baseline:

$SGD_{Exploit}$. A purely exploitative variant where, in each round, the top-k edges with the highest estimated probabilities are selected.

$SGD_{Explore}$. A purely exploratory variant where, in each round, k edges are selected uniformly at random from the candidate pool, regardless of the estimated probabilities.

6.3.3 ***Bandit Models.*** We also evaluate two contextual multi-armed bandit (CMAB) algorithms. Unlike static or purely online learning approaches, these models aim to minimize cumulative regret by adaptively selecting the top-k edges in each round.

LinUCB. This baseline assumes that the expected reward (i.e., the influence probability) is a linear function of the edge feature vector [29]. This makes it well aligned with our experimental setting, as influence probabilities are generated as linear combinations of edge features. We set $\alpha = 2.0$ for *LinUCB*.

EENet. This baseline does not rely on the linearity assumption [6]. Instead, it learns low-dimensional representations to improve reward estimation and accelerate convergence compared to standard *LinUCB*, making it suitable for capturing more complex feature–influence relationships. We follow the authors' default configuration [6] and perform a grid search over learning rates for each dataset, reporting the best-performing results.

6.3.4 ***Bandit Models for Influence (*** $InfluenceCB$***).*** Our proposed framework, $InfluenceCB$, leverages CombLinUCB [41] as the CMAB algorithm to learn edge-level peer influence probabilities. It dynamically learns C_t over rounds to optimize a specific objective O as $InfluenceCB(\beta, O)$. When C_t is fixed across rounds, we denote the variant as $InfluenceCB(\beta, C)$.

6.4 Experimental setup

In each round, for each action, a reward is generated based on the true peer influence probability (unknown to the algorithms). To compute $RMSE_T$, we reserve a fixed subset of $|\overline{E}| = 500$ edges, removed from the network in advance to ensure an unbiased evaluation. For all online and bandit algorithms, we conduct experiments under two edge selection settings in each round: (i) selecting k edges from a pool of 500 edges randomly sampled from the network (*network-based*), and (ii) selecting k edges from the pool of a randomly selected node's neighbors (*neighbor-based*) in each round. Due to space constraints, we only include the results for the network-based setting, noting notable differences between them when needed. In the Appendix, we include the Pareto frontier trade-off for both selection settings.

For $InfluenceCB$, we consider four experiments. In the first experiment, we evaluate $InfluenceCB(\beta, C)$ under varying $\beta \in \{0.0, 0.25, 0.30, 0.35, 0.40, 0.45, 0.50, 0.75, 1.0\}$ and fixed $C \in \{1, 3, 5, 7, 9\}$ to show the $Regret_T$ and $RMSE_T$ tradeoff. We also run an experiment where the ground truth influence probabilities are generated from a non-linear model, in order to check the robustness of our method when the function is misspecified. The third experiment evaluates $InfluenceCB(\beta, O)$ with dynamically optimized C_t for objectives, $O \in \{Regret, RMSE\}$ using $\beta \in \{0.25, 0.50\}$ and $k = 5$. We compare our variants against baseline methods. In the fourth experiment, we compare $InfluenceCB(\beta, O)$ variants with $O \in \{Regret, RMSE\}$ and $\beta \in \{0.25, 0.50\}$ across $k \in \{5, 10, 15, 20, 25\}$ for both settings. All experiments are repeated 10 times, and we report the average results across these repetitions.

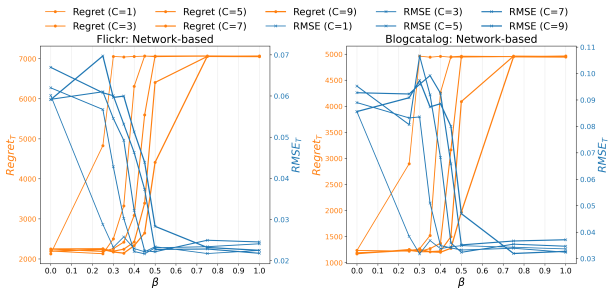

Figure 1: Trade-off between cumulative regret ($Regret_T$) and estimation error ($RMSE_T$) in $InfluenceCB(\beta, C)$ across linear network-based settings on the semi-synthetic datasets. The results illustrate how increasing C shifts the trade-off: larger C consistently reduces $Regret_T$ but leads to higher $RMSE_T$, with the trade-off depending on the choice of β.

6.5 Experimental results

6.5.1 Effect of β and C on $Regret_T$ and $RMSE_T$ in $InfluenceCB(\beta, C)$ under linear setting. Figure 1 shows the trade-off in our network-based experiments: when $\beta = 0.25$ (favoring optimal regret), increasing C from 1 to 9 reduces $Regret_T$ (as expected) by 53.7% and 57.5% on the Flickr and BlogCatalog datasets, respectively, while increasing $RMSE_T$ by 142.1% and 136.7%. Similarly, for $\beta = 0.50$ (favoring optimal RMSE), the same increase in C results in a 37.6% and 60.2% reduction in $Regret_T$ on Flickr and BlogCatalog, respectively, accompanied by a 21.2% and 34.7% increase in $RMSE_T$. A similar pattern is observed in the neighbor-based experiments.

Furthermore, the choice of C strongly influences the trade-off between regret minimization and estimation accuracy. When $\beta = 0.25$, $C = 3$ yields the lowest $Regret_T$, while $C = 1$ achieves the lowest $RMSE_T$ across both datasets. For $\beta = 0.50$, $C = 9$ leads to lowest $Regret_T$, whereas $C = 5$ leads to the lowest $RMSE_T$. Overall, $C = 3$ provides a good balance between $Regret_T$ and $RMSE_T$ for $\beta = 0.25$, whereas $C = 7$ achieves a good trade-off for $\beta = 0.50$. A similar pattern is observed in the neighbor-based experiments.

Overall, these results demonstrate that given β, careful tuning of C is crucial for achieving a good $Regret_T$–$RMSE_T$ tradeoff.

6.5.2 Effect of β and C on $Regret_T$ and $RMSE_T$ in $InfluenceCB(\beta, C)$ under non-linear setting. When the underlying function is misspecified, the results exhibit a similar overall pattern to those in Section 6.5.1: $Regret_T$ behaves as expected for the most part but $RMSE_T$ has a more idiosyncratic behavior. The full experimental setup and additional results are provided in Appendix 8.4, along with Figure 4.

6.5.3 Comparing $InfluenceCB(\beta, O)$ to the baselines. Table 1 summarizes the performance of $InfluenceCB(\beta, O)$ variants compared

to the *LinUCB* baseline. All static methods have higher $RMSE_T$ and $Regret_T$ than all four variants of $InfluenceCB$. In Figure 3 in the Appendix, we provide the detailed Pareto frontier visualization for our method vs. online and bandit baselines.

In network-based setting, on Flickr, $InfluenceCB(0.25, Regret)$ achieves 4.5% lower $Regret_T$ than *LinUCB* with 4.5% increase in $RMSE_T$, whereas $InfluenceCB(0.25, RMSE)$ reduces $RMSE_T$ by 31.1% at the cost of a 24.2% increase in $Regret_T$. Increasing β further biases the model towards exploration: $InfluenceCB(0.50, Regret)$ and $InfluenceCB(0.50, RMSE)$ reduce $RMSE_T$ by 60.3% and 60.4%, respectively, but incur 98.5% and 180.1% higher $Regret_T$. A similar trade-off pattern is observed on BlogCatalog. The $InfluenceCB(0.25, Regret)$ variant outperforms *LinUCB* in $Regret_T$ by 1.3% with a slight (3.5%) $RMSE_T$ increase, while $InfluenceCB(0.25, RMSE)$ achieves a 25.4% reduction in $RMSE_T$ with a 14.8% $Regret_T$ increase. High-β configurations further improve estimation accuracy, with $InfluenceCB(0.50, Regret)$ and $InfluenceCB(0.50, RMSE)$ lowering $RMSE_T$ by 49.3% and 63.1% at the cost of 81.2% and 226.9% higher $Regret_T$, respectively. These results demonstrate that $InfluenceCB$ expands the Pareto frontier beyond *LinUCB*, providing tunable trade-offs between $Regret_T$ and $RMSE_T$ depending on the application's priorities. A similar pattern is observed in the neighbor-based experiments.

6.5.4 Effect of k on the Trade-off Between $Regret_T$ and $RMSE_T$ in $InfluenceCB(\beta, O)$. Figure 2 shows how the number of interventions k affects the trade-off between $Regret_T$ and $RMSE_T$ for $InfluenceCB(\beta, O)$ relative to *LinUCB* in network based setting. Across both Flickr and BlogCatalog, increasing k amplifies the trade-off, with estimation error reductions becoming more pronounced but accompanied by larger regret increases as the k grows. For example, $InfluenceCB(0.25, Regret)$ consistently achieves comparable or lower $Regret_T$ than *LinUCB*, with up to 4.54% decrease on Flickr and 8.43% on BlogCatalog, and this advantage remains stable as k increases. However, the corresponding $RMSE_T$ increases slightly (about 3–4%). In contrast, $InfluenceCB(0.25, RMSE)$ exhibits a stronger dependence on k: $RMSE_T$ reduction grows from around 30% at $k = 5$ to more than 45% at $k = 25$, but this improvement is accompanied by a $Regret_T$ increase from roughly 14% to 46%. The increase in k is even more evident at $\beta = 0.50$. $InfluenceCB(0.50, Regret)$ reduces $RMSE_T$ by 33%–60% as k grows, but $Regret_T$ rises sharply from 81% to over 116%. Similarly, $InfluenceCB(0.50, RMSE)$ achieves up to 60% reduction in $RMSE_T$, but $Regret_T$ increases dramatically, from about 180% to over 330% as k increases from 5 to 25. This indicates that a larger k intensifies exploration-driven effects.

Overall, these results reveal that the influence of k on performance is tightly coupled with β. $\beta = 0.25$ settings scale more gracefully with k, maintaining near-optimal regret while offering moderate estimation improvements, whereas $\beta = 0.50$ configurations leverage larger k to substantially reduce estimation error but at steep regret costs. This highlights $InfluenceCB$'s adaptability to varying k, enabling practitioners to balance exploration and exploitation according to deployment needs.

Notably, the trends in the neighbor-based setting follow the same qualitative pattern but with more volatile changes in both $Regret_T$ and $RMSE_T$, indicating that the sensitivity to k is stronger.

Table 1: $Regret_T$ and $RMSE_T$ achieved by $InfluenceCB(\beta, O)$ on the Flickr and BlogCatalog datasets with network-based edge selection.

Method	Flickr		BlogCatalog	
	$Regret_T \downarrow$	$RMSE_T \downarrow$	$Regret_T \downarrow$	$RMSE_T \downarrow$
Static Models				
Random	5412	0.3296	5426	0.3018
Similarity	5504	0.3439	5585	0.4757
Link Prediction (Ridge)	5083	0.1066	6447	0.1810
Link Prediction (GraphSAGE)	4772	0.4325	5620	0.2430
Link Prediction (NCNC)	4579	0.2325	4920	0.2130
Bandit Models (CMAB)				
LinUCB	2245	0.0599	1144	0.0959
EENet	2146	0.0991	1718	0.1517
Online Learning				
$SGD_{Explore}$	5426	0.0665	5428	0.0894
$SGD_{Exploit}$	920	0.0968	1052	0.1781
Ours: InfluenceCB				
$InfluenceCB(0.25, Regret)$	**2143**	**0.0626**	**1129**	**0.0993**
$InfluenceCB(0.25, RMSE)$	**2788**	**0.0413**	**1313**	**0.0715**
$InfluenceCB(0.50, Regret)$	**4456**	**0.0238**	**2074**	**0.0486**
$InfluenceCB(0.50, RMSE)$	**6288**	**0.0237**	**3742**	**0.0354**

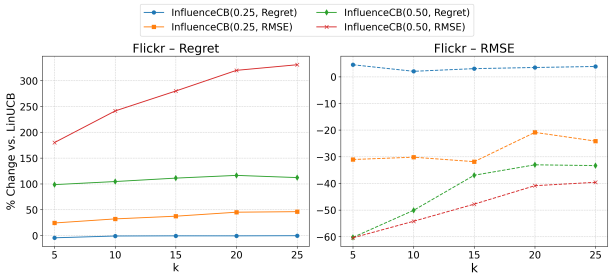

Figure 2: Tradeoff between $Regret_T$ (solid) and $RMSE_T$ (dotted) for $InfluenceCB$ variants compared to *LinUCB* for varying k.

7 Conclusion and Future Work

This work introduced a novel formulation of learning heterogeneous peer influence probabilities in networked environments under a contextual linear bandit framework. We proved the existence of a fundamental trade-off between cumulative regret and estimation error, characterized the achievable rate pairs, and showed that no algorithm can achieve the optimal rates for both simultaneously. Building on this theoretical insight, we proposed $InfluenceCB$ that leverages uncertainty-guided exploration to navigate the Pareto frontier and enables explicit control over the exploration–exploitation balance through a parameter. Our empirical evaluation on semi-synthetic network datasets corroborates the theoretical results, demonstrating that $InfluenceCB$ can achieve good trade-off between estimation accuracy and regret, unlike static, online learning, and contextual bandit baselines. Future work will focus on extending this framework beyond the linear setting to capture non-linear influence dynamics and adapting the approach to dynamic or partially observed networks.

References

[1] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. 2011. Improved algorithms for linear stochastic bandits. Advances in neural information processing systems 24 (2011).

[2] Abhineet Agarwal, Anish Agarwal, Lorenzo Masoero, and Justin Whitehouse. 2024. Mutli-Armed Bandits with Network Interference. Advances in Neural Information Processing Systems 37 (2024), 36414–36437.

[3] Aris Anagnostopoulos, Ravi Kumar, and Mohammad Mahdian. 2008. Influence and correlation in social networks. In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. 7–15.

[4] Nicolás Aramayo, Mario Schiappacasse, and Marcel Goic. 2023. A multiarmed bandit approach for house ads recommendations. Marketing Science 42, 2 (2023), 271–292.

[5] Jean-Yves Audibert, Sébastien Bubeck, and Gábor Lugosi. 2014. Regret in online combinatorial optimization. Mathematics of Operations Research 39, 1 (2014), 31–45.

[6] Yikun Ban, Yuchen Yan, Arindam Banerjee, and Jingrui He. 2022. EE-Net: Exploitation-Exploration Neural Networks in Contextual Bandits. In International Conference on Learning Representations.

[7] Léon Bottou. 2010. Large-scale machine learning with stochastic gradient descent. In Proceedings of COMPSTAT'2010: 19th International Conference on Computational StatisticsParis France, August 22-27, 2010 Keynote, Invited and Contributed Papers. Springer, 177–186.

[8] Nicolo Cesa-Bianchi and Gábor Lugosi. 2012. Combinatorial bandits. J. Comput. System Sci. 78, 5 (2012), 1404–1422.

[9] Pritish Chakraborty, Sayan Ranu, Krishna Sri Ipsit Mantri, and Abir De. 2023. Learning and maximizing influence in social networks under capacity constraints. In Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining. 733–741.

[10] Wei Chen, Yajun Wang, and Yang Yuan. 2013. Combinatorial multi-armed bandit: General framework and applications. In International conference on machine learning. PMLR, 151–159.

[11] Richard Combes, Mohammad Sadegh Talebi Mazraeh Shahi, Alexandre Proutiere, et al. 2015. Combinatorial bandits revisited. Advances in neural information processing systems 28 (2015).

[12] Ahmed Sayeed Faruk and Elena Zheleva. 2025. Leveraging Heterogeneous Spillover in Maximizing Contextual Bandit Rewards. In Proceedings of the ACM on Web Conference 2025 (Sydney NSW, Australia) (WWW '25). Association for Computing Machinery, New York, NY, USA, 3049–3060. doi:10.1145/3696410.3714706

[13] Jonathan Frenzen and Kent Nakamoto. 1993. Structure, cooperation, and the flow of market information. Journal of consumer research 20, 3 (1993), 360–375.

[14] Chenbo Fu, Minghao Zhao, Lu Fan, Xinyi Chen, Jinyin Chen, Zhefu Wu, Yongxiang Xia, and Qi Xuan. 2018. Link weight prediction using supervised learning methods and its application to yelp layered network. IEEE Transactions on Knowledge and Data Engineering 30, 8 (2018), 1507–1518.

[15] Chen Gao, Chao Huang, Donghan Yu, Haohao Fu, Tzh-Heng Lin, Depeng Jin, and Yong Li. 2022. Item recommendation for word-of-mouth scenario in social E-commerce. IEEE Transactions on Knowledge and Data Engineering 34, 6 (2022), 2798–2809.

[16] Amit Goyal, Francesco Bonchi, and Laks VS Lakshmanan. 2010. Learning influence probabilities in social networks. In Proceedings of the third ACM international conference on Web search and data mining. 241–250.

[17] Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. 2009. Finding structure with randomness: Stochastic algorithms for constructing approximate matrix decompositions. (2009).

[18] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. Advances in neural information processing systems 30 (2017).

[19] Linda D Hollebeek, Viktorija Kulikovskaja, Marco Hubert, and Klaus G Grunert. 2023. Exploring a customer engagement spillover effect on social media: the moderating role of customer conscientiousness. Internet Research 33, 4 (2023), 1573–1596.

[20] Alexandra Iacob, Bogdan Cautis, and Silviu Maniu. 2022. Contextual Bandits for Advertising Campaigns: A Diffusion-Model Independent Approach. In Proceedings of the 2022 SIAM International Conference on Data Mining (SDM). SIAM, 513–521.

[21] Mohsen Jamali and Martin Ester. 2009. Using a trust network to improve top-n recommendation. In Proceedings of the third ACM conference on Recommender systems. 181–188.

[22] Fateme Jamshidi, Mohammad Shahverdikondori, and Negar Kiyavash. 2025. Graph-dependent regret bounds in multi-armed bandits with interference. arXiv preprint arXiv:2503.07555 (2025).

[23] Su Jia, Peter Frazier, and Nathan Kallus. 2024. Multi-armed bandits with interference. arXiv preprint arXiv:2402.01845 (2024).

[24] Maosheng Jiang, Yonxiang Chen, and Ling Chen. 2015. Link prediction in networks with nodes attributes by similarity propagation. arXiv preprint arXiv:1502.04380 (2015).

[25] Lini Kuang, Ni Huang, Yili Hong, and Zhijun Yan. 2019. Spillover effects of financial incentives on non-incentivized user engagement: Evidence from an online knowledge exchange platform. Journal of Management Information Systems 36, 1 (2019), 289–320.

[26] Ajay Kumar, Shashank Sheshar Singh, Kuldeep Singh, and Bhaskar Biswas. 2020. Link prediction techniques, applications, and performance: A survey. Physica A: Statistical Mechanics and its Applications 553 (2020), 124289.

[27] Konstantin Kutzkov, Albert Bifet, Francesco Bonchi, and Aristides Gionis. 2013. Strip: stream learning of influence probabilities. In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. 275–283.

[28] Tor Lattimore and Csaba Szepesvári. 2020. Bandit algorithms. Cambridge University Press.

[29] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In Proceedings of the 19th international conference on World wide web. 661–670.

[30] Haohui Lu and Shahadat Uddin. 2024. A parameterised model for link prediction using node centrality and similarity measure based on graph embedding. Neurocomputing 593 (2024), 127820.

[31] Cameron Marlow, Mor Naaman, Danah Boyd, and Marc Davis. 2006. HT06, tagging paper, taxonomy, Flickr, academic article, to read. In Proceedings of the seventeenth conference on Hypertext and hypermedia. 31–40.

[32] Mark EJ Newman. 2003. Mixing patterns in networks. Physical review E 67, 2 (2003), 026126.

[33] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. the Journal of machine Learning research 12 (2011), 2825–2830.

[34] Kazumi Saito, Ryohei Nakano, and Masahiro Kimura. 2008. Prediction of information diffusion probabilities for independent cascade model. In International conference on knowledge-based and intelligent information and engineering systems. Springer, 67–75.

[35] Shai Shalev-Shwartz et al. 2012. Online learning and online convex optimization. Foundations and Trends® in Machine Learning 4, 2 (2012), 107–194.

[36] Cosma Rohilla Shalizi and Andrew C Thomas. 2011. Homophily and contagion are generically confounded in observational social network studies. Sociological methods & research 40, 2 (2011), 211–239.

[37] Dongjin Song, David A Meyer, and Dacheng Tao. 2015. Top-k link recommendation in social networks. In 2015 IEEE International Conference on Data Mining. IEEE, 389–398.

[38] Karthik Subbian, Charu C Aggarwal, and Jaideep Srivastava. 2016. Querying and tracking influencers in social streams. In Proceedings of the ninth ACM international conference on Web search and data mining. 493–502.

[39] Sharan Vaswani, Branislav Kveton, Zheng Wen, Mohammad Ghavamzadeh, Laks VS Lakshmanan, and Mark Schmidt. 2017. Model-independent online learning for influence maximization. In International Conference on Machine Learning. PMLR, 3530–3539.

[40] Xiyuan Wang, Haotong Yang, and Muhan Zhang. 2024. Neural Common Neighbor with Completion for Link Prediction. In The Twelfth International Conference on Learning Representations.

[41] Zheng Wen, Branislav Kveton, and Azin Ashkan. 2015. Efficient learning in large-scale combinatorial semi-bandits. In International Conference on Machine Learning. PMLR, 1113–1122.

[42] Zheng Wen, Branislav Kveton, Michal Valko, and Sharan Vaswani. 2017. Online influence maximization under independent cascade model with semi-bandit feedback. Advances in neural information processing systems 30 (2017).

[43] Bryan Wilder, Nicole Immorlica, Eric Rice, and Milind Tambe. 2018. Maximizing influence in an unknown social network. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32.

[44] Yang Xu, Wenbin Lu, and Rui Song. 2024. Linear contextual bandits with interference. arXiv preprint arXiv:2409.15682 (2024).

[45] Yukuan Xu, Juan Luis Nicolau, and Peng Luo. 2022. Travelers' reactions toward recommendations from neighboring rooms: Spillover effect on room bookings. Tourism Management 88 (2022), 104427.

[46] Xiwang Yang, Harald Steck, Yang Guo, and Yong Liu. 2012. On top-k recommendation using social networks. In Proceedings of the sixth ACM conference on Recommender systems. 67–74.

[47] Jing Zhang, Jie Tang, Yuanyi Zhong, Yuchen Mo, Juanzi Li, Guojie Song, Wendy Hall, and Jimeng Sun. 2017. Structinf: Mining structural influence from social streams. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 31.

[48] Zhiheng Zhang and Zichen Wang. 2024. Online experimental design with estimation-regret trade-off under network interference. arXiv preprint arXiv:2412.03727 (2024).

[49] Zizhu Zhang, Weiliang Zhao, Jian Yang, Cecile Paris, and Surya Nepal. 2019. Learning influence probabilities and modelling influence diffusion in twitter. In Companion Proceedings of The 2019 World Wide Web Conference. 1087–1094.

8 Omitted Proofs

In this section, we present the omitted proofs of the theoretical results.

8.1 Proof of Lemma 4.1

Here, we present the proof of Lemma 4.1.

Proof. We first introduce two notations:

$$\forall \mathbf{w} \in \Delta_{1/k}^{M-1}: \quad M(\mathbf{w}) \;=\; \sum_{j=1}^{M} w_j X_j X_j^\top,$$

$$\forall \mathbf{u} \in \mathbb{R}^d,\ V \in \mathbb{R}^{d\times d}: \quad \|\mathbf{u}\|_V = \sqrt{\mathbf{u}^\top V \mathbf{u}}.$$

If the kT total selections are allocated proportionally to $\mathbf{w}_k^*$, then the design matrix satisfies

$$V_T \;=\; \lambda \mathbb{I} + \sum_{t=1}^{T}\sum_{r=1}^{k} X_{t,r} X_{t,r}^\top \;=\; \lambda \mathbb{I} + kT\, M(\mathbf{w}_k^*).$$

Since $A \succeq B \Rightarrow A^{-1} \preceq B^{-1}$, for any $X_i \in \mathcal{A}$,

$$X_i^\top V_T^{-1} X_i = X_i^\top \big(\lambda \mathbb{I} + kT\, M(\mathbf{w}_k^*)\big)^{-1} X_i$$
$$\le \frac{1}{kT}\, X_i^\top M(\mathbf{w}_k^*)^{-1} X_i \;\le\; \frac{f^*(\mathcal{A}, k)}{kT}.$$

Let $\hat{\theta}_T$ be the ridge estimator with parameter $\lambda > 0$. Since rewards are binary, they are $\frac{1}{2}$-sub-Gaussian. From the self-normalized bound in [1], for any $\delta \in (0,1)$, with probability at least $1-\delta$,

$$\|\hat{\theta}_T - \theta^*\|_{V_T} \;\le\; \beta_T(\delta),$$

where

$$\beta_T(\delta) \;=\; \tfrac{1}{2}\sqrt{d \log\!\Big(\tfrac{1+kTL^2/\lambda}{d}\Big)} \;+\; \sqrt{\lambda}\,\|\theta^*\|_2, \tag{4}$$

and $L = \max_i \|X_i\|_2$.

To move from parameter error to prediction error for X_i, we apply Cauchy–Schwarz in the V_T-inner product:

$$|X_i^\top(\hat{\theta}_T - \theta^*)| = \big|\langle V_T^{-1/2} X_i,\ V_T^{1/2}(\hat{\theta}_T - \theta^*)\rangle\big| \;\le\; \|X_i\|_{V_T^{-1}}\, \|\hat{\theta}_T - \theta^*\|_{V_T}.$$

Thus, with probability at least $1-\delta$,

$$|X_i^\top(\hat{\theta}_T - \theta^*)| \;\le\; \beta_T(\delta)\sqrt{X_i^\top V_T^{-1} X_i} \;\le\; \beta_T(\delta)\sqrt{\tfrac{f^*(\mathcal{A},k)}{kT}}.$$

By definition,

$$\mathrm{RMSE}_T = \mathbb{E}\left[\sqrt{\frac{1}{M}\sum_{i=1}^{M}\big(X_i^\top(\hat{\theta}_T - \theta^*)\big)^2}\right].$$

The uniform bound above implies

$$\mathrm{RMSE}_T \;\le\; \beta_T(\delta)\sqrt{\tfrac{f^*(\mathcal{A},k)}{kT}}.$$

Finally, choosing $\delta = \frac{1}{kT}$ and taking expectations yields

$$\mathbb{E}[\mathrm{RMSE}_T] \;\le\; \tilde{O}\left(\sqrt{\frac{f^*(\mathcal{A},k)}{kT}}\right),$$

where $\tilde{O}(\cdot)$ hides logarithmic factors in d, kT, and confidence parameters. This completes the proof. □

8.2 Proof of Theorem 4.2

Proof. We begin by presenting a known lower bound in the literature on the expected estimation error for Bernoulli random variables.

Lemma 8.1 (Expected estimation error for the Bernoulli mean)**.** *Let $X_1, \ldots, X_t$ be i.i.d. samples from a Bernoulli distribution with success probability $p \in [\frac{1}{4}, \frac{3}{4}]$, and let $\hat{p}$ be any estimator of p. Then there exists a universal constant $C > 0$ such that, for all $t \ge 1$ and all $p \in [\frac{1}{4}, \frac{3}{4}]$,*

$$\mathbb{E}_p\big[\,|\hat{p} - p|\,\big] \;\ge\; C\sqrt{\frac{p(1-p)}{t}}.$$

We construct an instance $(\mathcal{A}_0, \theta_0^*)$ as follows. Let the action set be

$$\mathcal{A}_0 = \{X_1, \ldots, X_M\}, \quad X_m = \begin{cases} (1, 0, x_{m,3}, \ldots, x_{m,d}), & 1 \le m \le k, \\ (0, 1, x_{m,3}, \ldots, x_{m,d}), & k < m \le M, \end{cases}$$

and set the true parameter vector to

$$\theta_0^* = \Big(\tfrac{3}{4},\ \tfrac{1}{4},\ 0, \ldots, 0\Big).$$

The expected reward of arm m is $X_m^\top \theta_0^*$, so the first k arms are high-reward arms with mean $3/4$, and the remaining $M-k$ arms are low-reward arms with mean $1/4$. The expected reward gap is thus $1/2$.

Fix any policy π that selects k arms per round for T rounds, for a total of kT plays. Let $\alpha \in [0,1]$ denote the expected fraction of plays allocated to high-reward arms, so that a fraction $1-\alpha$ corresponds to plays of low-reward arms. Define $\beta = \frac{\ln(kT(1-\alpha))}{2\ln T}$, equivalently $kT(1-\alpha) = T^{2\beta}$. The total expected regret is then

$$\mathrm{Regret}_T(\pi, \mathcal{A}_0, \theta_0^*) = \frac{1}{2} \times \big(\#\text{ of plays on low-reward arms}\big)$$
$$= \frac{1}{2}(kT)(1-\alpha) \;\in\; O(T^{2\beta}).$$

Let N_{low} denote the (random) number of plays of low-reward arms, so that $\mathbb{E}[N_{\mathrm{low}}] = kT(1-\alpha)$. Each time a low-reward arm is played, the observed reward follows Bernoulli$(1/4)$ and provides information only about the second coordinate $\theta_{0,2}^* = \frac{1}{4}$. Conditioned on the (possibly adaptive) history of actions, these N_{low} samples are i.i.d. Bernoulli$(1/4)$.

By the definition of RMSE,

$$\mathrm{RMSE}_T(\pi, \mathcal{A}_0, \theta_0^*) = \mathbb{E}\left[\sqrt{\frac{1}{|\mathcal{A}|}\sum_{X\in\mathcal{A}}\big(X^\top(\hat{\theta}_T - \theta_0^*)\big)^2}\right]$$
$$\ge \mathbb{E}\left[\sqrt{\frac{1}{|\mathcal{A}|}\sum_{X\ \text{low-reward}}\big(X^\top(\hat{\theta}_T - \theta_0^*)\big)^2}\right]$$
$$\ge \sqrt{\frac{M-k}{M}}\;\mathbb{E}\Big[\big|\hat{\theta}_{T,2} - \tfrac{1}{4}\big|\Big],$$

where $\hat{\theta}_{T,2}$ denotes the estimate of the second coordinate of θ_0^*, and the last inequality follows from the Cauchy–Schwarz inequality.

Applying Lemma 8.1 with $p = \frac{1}{4}$ yields

$$\mathbb{E}\Big[\,\big|\hat{\theta}_{T,2} - \tfrac{1}{4}\big| \;\Big|\; N_{\mathrm{low}} = n\Big] \;\ge\; C'\,\frac{1}{\sqrt{n}}, \quad \text{for all } n \ge 1,$$

for some universal constant $C' > 0$. Taking expectations and applying Jensen's inequality to the convex function $n \mapsto 1/\sqrt{n}$, we obtain

$$\mathbb{E}\Big[|\hat{\theta}_{T,2} - \tfrac{1}{4}|\Big] \;\geq\; \frac{C'}{\sqrt{\mathbb{E}[N_{\text{low}}]}} \;=\; \frac{C'}{\sqrt{kT(1-\alpha)}}.$$

Combining these inequalities gives

$$\text{RMSE}_T(\pi, \mathcal{A}_0, \theta_0^*) \;\geq\; C'\sqrt{\frac{M-k}{MkT(1-\alpha)}} \;\in\; \Omega(T^{-\beta}),$$

where constants depending on k are absorbed into the asymptotic notation. This completes the proof. □

8.3 Proof of Theorem 5.1

Proof. Let the set of available actions be fixed as $\mathcal{A}$ for all rounds, and let $\forall t : C_t = C$. For each round t and arm $X \in \mathcal{A}$, denote by $N_t(X)$ the number of times Algorithm 1 has selected arm X up to round t. For any $\mathbf{u} \in \mathbb{R}^d$ and $V \in \mathbb{R}^{d\times d}$, define

$$\|\mathbf{u}\|_V = \sqrt{\mathbf{u}^\top V \mathbf{u}}.$$

The uncertainty of arm X at round t is given by $U_t(X) = \|X\|_{V_{t-1}^{-1}}$. Let $m_t(X)$ denote the number of times $s \leq t$ such that $u_s > C/s^\beta$ and $\arg\max_{X'\in\mathcal{A}} \|X'\|_{V_{s-1}^{-1}} = X$.

We first show that for all t and $X \in \mathcal{A}$,

$$\|X\|_{V_t^{-1}} \leq \frac{1}{\sqrt{N_t(X)}}. \tag{5}$$

Recall

$$\begin{aligned} V_t \;&=\; \lambda\mathbb{I} \;+\; \sum_{s=1}^{t}\sum_{r=1}^{k} X_{s,r}X_{s,r}^\top \\ &=\; \lambda\mathbb{I} \;+\; \sum_{X'\in\mathcal{A}} N_t(X')\,X'X'^\top \;\succeq\; N_t(X)\,XX^\top. \end{aligned}$$

For any positive semidefinite matrix A and any vector u, the Cauchy–Schwarz inequality in the A-inner product yields

$$u^\top A^{-1} u \;\leq\; \frac{(u^\top u)^2}{u^\top A\, u}.$$

Apply this with $A = V_t$ and $u = X$:

$$\begin{aligned} X^\top V_t^{-1} X \;&\leq\; \frac{\|X\|_2^4}{X^\top V_t X} \;\leq\; \frac{\|X\|_2^4}{X^\top\big(N_t(X)\,XX^\top\big)X} \\ &=\; \frac{\|X\|_2^4}{N_t(X)\,(X^\top X)^2} \;=\; \frac{1}{N_t(X)}. \end{aligned}$$

Equivalently,

$$\|X\|_{V_t^{-1}} \;=\; \sqrt{X^\top V_t^{-1} X} \;\leq\; \frac{1}{\sqrt{N_t(X)}}.$$

Then, we prove the following upper bound on $m_t(X)$:

$$m_t(X) \leq \frac{2t^{2\beta}}{C^2}. \tag{6}$$

Let $s_1 < s_2 < \cdots < s_{m_t(X)}$ be the rounds up to t at which X is (i) the maximizer of uncertainty and (ii) has uncertainty above the threshold, i.e.,

$$\arg\max_{X'\in\mathcal{A}} \|X'\|_{V_{s_j-1}^{-1}} = X, \qquad \|X\|_{V_{s_j-1}^{-1}} > \frac{C}{s_j^\beta}.$$

By the algorithm's exploration rule, X is selected at each such round, hence the play count increases strictly:

$$N_{s_j}(X) \;\geq\; N_{s_j-1}(X) + 1.$$

Inductively, $N_{s_j-1}(X) \geq j-1$ for all $j \geq 1$. Therefore, by Step 1,

$$\|X\|_{V_{s_j-1}^{-1}} \;\leq\; \frac{1}{\sqrt{N_{s_j-1}(X)}} \;\leq\; \frac{1}{\sqrt{j-1}}.$$

Combining with the threshold condition gives, for every $j \geq 2$,

$$\frac{1}{\sqrt{j-1}} \;>\; \frac{C}{s_j^\beta} \quad\Longrightarrow\quad j-1 \;<\; \frac{s_j^{2\beta}}{C^2} \;\leq\; \frac{t^{2\beta}}{C^2}.$$

Thus $j \leq 1 + t^{2\beta}/C^2$ for all such indices. Including the base case $j = 1$ (which trivially satisfies the same bound), we obtain

$$m_t(X) \;\leq\; \frac{t^{2\beta}}{C^2} \;+\; 1 \;\leq\; \frac{2\,t^{2\beta}}{C^2} \qquad \text{for all sufficiently large } t.$$

Summing over $X \in \mathcal{A}$ then gives

$$\sum_{X\in\mathcal{A}} m_T(X) \;\leq\; \frac{2M}{C^2}\,T^{2\beta},$$

which shows the total number of rounds that our algorithm performs the exploration. On all of the other rounds, we are playing based on an optimal regret minimization subroutine, like the CombLinUCB algorithm[41], where we can use their regret upper bound to bound our regret as

$$Regret_T\big(\text{InfluenceCB}, \mathcal{A}, \theta^*\big) \tag{7}$$

$$\leq \tilde{O}\big(ckd\sqrt{T}\big) + \frac{N}{C^2}T^{2\beta} \in \tilde{O}\left(T^{2\beta}\right), \tag{8}$$

where we use $\beta \geq \frac{1}{4}$.

We now prove that

$$\forall X \in \mathcal{A} : \quad \|X\|_{V_T^{-1}} \leq \frac{2C}{T^\beta}. \tag{9}$$

Since $V_t \succeq V_{t-1}$, the uncertainty $\|X\|_{V_t^{-1}}$ is monotonically decreasing in t. If $\|X\|_{V_T^{-1}} > \frac{2C}{T^\beta}$, then for all $t < T$,

$$\|X\|_{V_t^{-1}} > \frac{2C}{T^\beta} > \frac{C}{(T/2)^\beta},$$

implying that for all $t \in [T/2, T]$, we have $u_t > C/t^\beta$ and the algorithm explores. However, this would require

$$\frac{T}{2} \leq \frac{2M}{C^2}T^{2\beta},$$

which fails for large T for $\beta < \frac{1}{2}$, and for $\beta = \frac{1}{2}$, the constant C can be chosen to be large enough to fail this inequality. Thus, for sufficiently large T, inequality (9) must hold. For small T, the constant in the final bound can be adjusted accordingly.

From the self-normalized bound of [1], with probability at least $1-\delta$, we have for all $X \in \mathcal{A}$:

$$|X^\top(\hat{\theta}_T - \theta^*)| \;\leq\; \beta_T(\delta)\sqrt{X^\top V_T^{-1} X},$$

where $\beta_T(\delta)$ is defined in (4) and is logarithmic in the problem parameters. By setting $\delta = \frac{1}{kT}$ and taking the expectation, and combining this with (9) implies

$$|X^\top(\hat{\theta}_T - \theta^*)| \;\leq\; \tilde{O}\Big(T^{-\beta}\Big),$$

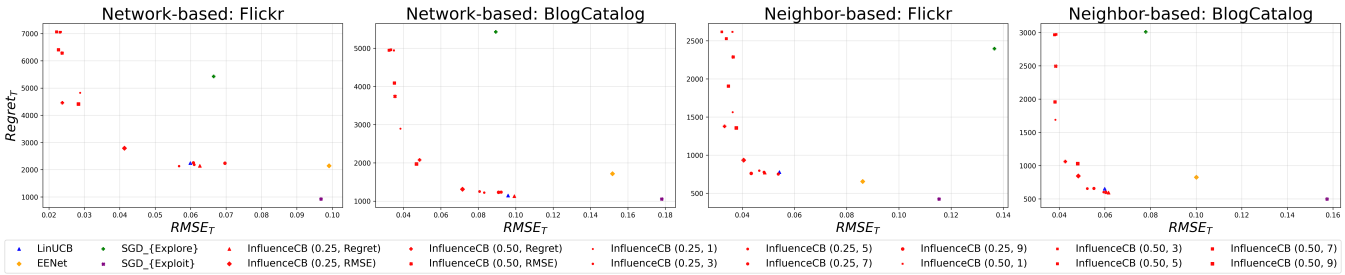

Figure 3: Pareto frontiers of cumulative regret $Regret_T$ versus estimation error $RMSE_T$ for InfluenceCB(β, O) compared to baseline bandit and online models on semi-synthetic datasets. InfluenceCB expands the Pareto frontier and enables tunable trade-offs, with a notably smoother frontier on BlogCatalog. Varying β and C shifts the curve between low-regret and low-error regimes.

Algorithm 2 UpdateC: Adaptive Computation of C_t

Input: objective $O \in \{\text{RMSE}, \text{Regret}\}$, activation rate, average uncertainty, $C_{max}, C_{min}, \gamma, warmup, \epsilon$
$m_t \leftarrow$ activation rate if $O = Regret$ else average uncertainty
Append m_t to history buffer $hist$
$baseline \leftarrow mean(hist)$ if $|hist| > warmup$ else m_t
$z \leftarrow (baseline - m_t)/(std(hist) + \epsilon)$
if $O =$ Regret **then** $z \leftarrow \max(0, z)$
end if
$C_{new} \leftarrow C_{min} + (C_{max} - C_{min})/(1 + e^{-\gamma z})$
if $O =$ Regret **then**
    $C_t \leftarrow \max(C_{prev}, C_{new})$, $C_{prev} \leftarrow C_t$
else
    $C_t \leftarrow C_{new}$
end if
return C_t

and hence for any instance $(\mathcal{A}, \theta^*)$,

$$RMSE_T\left(\text{InfluenceCB}, \mathcal{A}, \theta^*\right) = \mathbb{E}\left[\sqrt{\frac{1}{M}\sum_{i=1}^{M}\left(X_i^\top(\hat{\theta}_T - \theta^*)\right)^2}\right] \in \tilde{O}\left(T^{-\beta}\right).$$

This completes the proof. □

8.4 Experiments with Non-linear Ground-truth Influence Model.

To capture non-linear interactions among edge features, we synthetically generate peer influence probabilities using a feed-forward neural network. Given the feature vector $e_{ij}.X \in \mathbb{R}^{256}$ for edge e_{ij}, the probability $e_{ij}.p$ is computed as

$$h_1 = \tanh\left(W_1 e_{ij}.X + \mathbf{b}_1\right), \quad h_2 = \tanh(W_2 h1 + \mathbf{b}_1),$$
$$z = W_3 h2 + b_3, \quad e_{ij}.p = \sigma\left(\frac{z}{\tau}\right),$$

where $W_1 \in \mathbb{R}^{128\times256}$, $W_2 \in \mathbb{R}^{64\times128}$, and $W_3 \in \mathbb{R}^{1\times64}$ are weight matrices with entries drawn uniformly at random from $[-1, 1]$, $\mathbf{b}_1 \in \mathbb{R}^{128}$, $\mathbf{b}_2 \in \mathbb{R}^{64}$ and $b_2 \in \mathbb{R}$ are bias terms, $\tanh(\cdot)$ denotes the hyperbolic tangent activation, and $\sigma(\cdot)$ is the sigmoid function ensuring $e_{ij}.p \in (0, 1)$. The scalar $\tau = 4.0$ is a temperature parameter that controls the sharpness of the probability distribution.

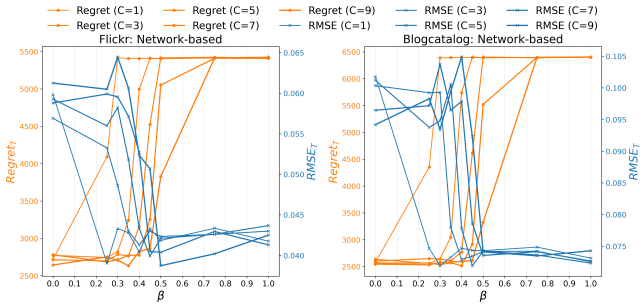

Figure 4: Trade-off between cumulative regret ($Regret_T$) and estimation error ($RMSE_T$) in $InfluenceCB(\beta, C)$ under a non-linear network-based setting. Peer influence probabilities are generated via a feed-forward neural network to simulate model misspecification. Results on Flickr and BlogCatalog show that while $Regret_T$ trends remain consistent with the linear case, $RMSE_T$ exhibits more irregular patterns as β and C vary.

This experiment follows the same setup as the second experiment but employs the synthetically generated peer influence probabilities described above to investigate how the generation model misspecification impacts the trade-off between cumulative regret $Regret_T$ and estimation error $RMSE_T$. The observed trade-off are presented in Figure 4. A similar pattern is observed in the neighbor-based experiments.