Artificial Intelligence 354 (2026) 104500

Contents lists available at ScienceDirect

Artificial Intelligence

journal homepage: www.elsevier.com/locate/artint

Exploring multi-layered networks through random walks: bridging offline optimization and online learning

Xiangxiang Dai [a], Xutong Liu [b,*], Jinhang Zuo [c], Xiaowei Chen [d], Wei Chen [e], John C S Lui [a]

[a] Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong SAR, China
[b] Department of Electrical and Computer Engineering, University of Washington, Tacoma, USA
[c] Department of Computer Science, City University of Hong Kong, Hong Kong SAR, China
[d] Bytedance, USA
[e] Microsoft Research Asia, China

ARTICLE INFO

Keywords:
Network exploration
Random walk
Offline optimization
Online learning
Combinatorial multi-armed bandit

ABSTRACT

The multi-layered network exploration problem MuLaNE is a significant challenge derived from various practical applications. In MuLaNE, there are multiple layers of the network, where each node is assigned an importance weight, and each layer is explored through random walks. The objective is to allocate a total random walk budget B across the network layers to maximize the cumulative weights of the unique nodes visited. We systematically approach this problem by addressing both offline optimization and online learning scenarios. For the offline optimization case, where the network structure and node weights are known, we propose constant-ratio approximation algorithms based on greedy strategies for overlapping networks, and exact optimal solutions using greedy or dynamic programming for non-overlapping networks. In the online learning scenario, where neither the network structure nor the node weights are initially known, we adapt the combinatorial multi-armed bandit framework. We develop algorithms with two distinct confidence radius designs to simultaneously learn random walk parameters and node weights, while optimizing budget allocation across multiple rounds, achieving logarithmic regret bounds. Finally, experimental results on real-world social and computer networks validate the practical applicability of MuLaNE and our theoretical findings.

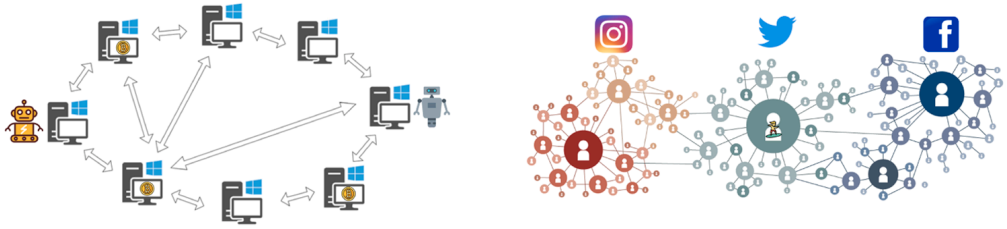
1. Introduction

Network exploration is a fundamental process for discovering information and resources located at nodes within a network, where random walks are frequently employed as an effective method [1–3]. In this paper, we investigate the problem of multi-layered network exploration via random walks, a model applicable to various real-world scenarios such as resource discovery in peer-to-peer (P2P) networks and web navigation in online social networks (OSNs), as depicted in Fig. 1.

In P2P networks, as shown in Fig. 1a, a user often needs to find valuable resources (like files or data) that are scattered across many different peers, which may belong to several distinct networks. A common and effective decentralized search strategy is to launch multiple parallel random walkers [1,4], where each walker is a query that propagates from peer to peer. Critically, these

* Corresponding author.
E-mail addresses: xxdai23@cse.cuhk.edu.hk (X. Dai), xutongl@uw.edu (X. Liu), jinhang.zuo@cityu.edu.hk (J. Zuo), weic@microsoft.com (W. Chen).

https://doi.org/10.1016/j.artint.2026.104500
Received 28 November 2024; Received in revised form 11 September 2025; Accepted 12 February 2026
Available online 26 February 2026



(a) Resource Searching in P2P Networks. (b) Web Surfing in OSNs.

Fig. 1. Applications on network exploration via random walks.

search queries don't run indefinitely; they operate under strict constraints, such as a maximum hop count or a time-to-live limit. Given that resources vary in importance and are distributed unevenly, the central challenge becomes one of resource allocation: *how should a user with a limited total search capacity strategically divide this effort among different walkers in various networks to maximize the total value of the unique, important resources they discover?*

A similar challenge emerges in the automated exploration of online social networks (OSNs), where autonomous explorers navigate social connections to uncover information [5,6]. Unlike tasks with predefined targets, these explorers lack a feasible list of nodes due to OSNs' vast scale and dynamism; their core objective is discovering unknown content. Instead, they traverse platforms like Instagram, Twitter, and Facebook through real-time programmatic navigation, guided by emerging connections (Fig. 1b). This path-dependent process–starting at a seed node, following edges (e.g., likes or connections) to profiles, and continuing onward–naturally models as a random walk. Yet, it is limited by a finite computational budget, stemming from practical constraints like API rate limits and per-step processing costs. This poses a central optimization question: *How can an automated system optimally allocate its budget across platforms to maximize high-value information discovery?*

We formalize the above application scenarios as the **Mu**lti-**La**yered **N**etwork **E**xploration (`MuLaNE`) problem. In `MuLaNE`, the overall network to be explored (e.g., a combination of different OSNs) is modeled as a multi-layered network $\mathcal{G}$ with m layers $L_1, \ldots, L_m$. Each layer L_i (e.g., an individual OSN) is a weighted directed graph $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)$, where $\mathcal{V}_i$ is the set of nodes (e.g., user pages with importance weights σ_u), and $\mathcal{E}_i$ is the set of directed edges with associated edge weight function w_i. Each layer is explored by an explorer (or random walker) W_i, which starts according to a given distribution $\boldsymbol{\alpha}_i$ and follows the outgoing edges with probabilities proportional to their edge weights. Each step consumes one unit of budget[1] Given a total budget B and per-layer limits c_i, the goal is to determine the optimal budget allocation $\boldsymbol{k} = (k_1, \ldots, k_m)$ within the budget constraints to maximize the expected total weight of *unique* visited nodes.

In practical applications, the network structure $\mathcal{G}$, the node weights $\boldsymbol{\sigma}$, and the starting distributions $\boldsymbol{\alpha}_i$ may not be known in advance. Therefore, we consider both the *offline optimization* scenario, where $\mathcal{G}$, $\boldsymbol{\sigma}$, and $\boldsymbol{\alpha}_i$ are known, and the *online learning* scenario, where these elements are unknown. Furthermore, different layers may have overlapping vertices (e.g., the same user's Homepage may appear in multiple OSNs), and the starting distributions $\boldsymbol{\alpha}_i$ may or may not correspond to stationary distributions. This paper provides a systematic study of all these combinations of cases:

- For the offline optimization process, we first address the general case with overlapping layers and propose constant approximation algorithms. These are based on the lattice submodularity property for non-stationary starting distributions $\boldsymbol{\alpha}_i$, and the diminishing returns (DR) submodularity property for stationary $\boldsymbol{\alpha}_i$. For the special case of non-overlapping layers, we design a dynamic programming algorithm that computes the exact optimal solution for `MuLaNE`.
- In the online learning setting, exploration is conducted over multiple rounds, each constrained by the same budget B and individual layer budgets c_i. After each round, the reward is the total weight of the unique nodes visited, and the trajectory of each explorer, along with the importance weights of the visited nodes, is observed. This feedback informs future exploration rounds by providing valuable insights into the network's structure.

We adapt the combinatorial multi-armed bandit (CMAB) framework [7] to minimize regret, defined as the difference between the cumulative reward achieved by "*the exact or approximate offline algorithm*" and that achieved by "*the online learning algorithm*" over T rounds. For the offline optimization setting, the core challenge is to solve a complex combinatorial optimization problem for budget allocation, given full knowledge of the network. The problem is NP-hard in general, as the reward function may not possess desirable properties. To address this, we leverage the property of lattice submodularity to design **constant-ratio approximation algorithms** for the general case, and we provide **the exact optimal solutions** via greedy or dynamic programming methods for important special cases. For the online learning setting, the network structure and node weights are initially unknown, and the challenge lies in sequentially allocating the budget over multiple rounds. This requires balancing the exploration of unknown parameters with the exploitation of current knowledge, with the goal of minimizing cumulative regret against the offline solution. To tackle this, we developing **novel online algorithms** that learn random walk parameters, i.e., visiting probabilities, and node weights simultaneously to achieve **logarithmic regret bounds**.

[1] The model can be extended to allow multiple steps per budget unit.

Contributions. In summary, our contributions are as follows: (a) We model the multi-layered network exploration via the random walks problem (MuLaNE) as an abstraction for various real-world applications. (b) We conduct a systematic study of the MuLaNE problem, designing specialized algorithms and providing rigorous theoretical analysis for both offline and online settings, considering overlapping and non-overlapping multi-layered networks. (c) We perform empirical validation on real-world social and computer network datasets, demonstrating the practical applicability of MuLaNE and the effectiveness of our proposed algorithms.

Paper Organization. Section 3 presents the formal problem setting of MuLaNE. In Section 4, we introduce the equivalent bipartite coverage model to derive the explicit form of the reward function. Section 5 discusses offline algorithms for four distinct scenarios, providing provable approximation guarantees and running-time analyses. In Section 6, we present different online learning algorithms along with regret analysis for both overlapping and non-overlapping cases. Empirical results are provided in Section 7, followed by related work in Section 2. Finally, Section 8 concludes the paper.

2. Related work

Network exploration via random walks has been studied in various application contexts such as centrality measuring [2], large-scale network sampling [8], and influence maximization [3]. Multiple random walks has also been used, e.g. in Lv et al. [1] for query resolution in peer-to-peer networks. However, none of these studies addresses the budgeted MuLaNE problem. MuLaNE is also related to the influence maximization (IM) problem [9,10] and can be viewed as a special variant of IM. Different from the standard Independent Cascade (IC) model [11] in IM, which randomly broadcasts to all its neighbors, the propagation process of MuLaNE is a random walk that selects one neighbor. Moreover, each unit of budget in MuLaNE is used to propagate one random-walk step, not to select one seed node as in IC.

While the MuLaNE problem formulation is novel, its offline version can be framed as an instance of a budget allocation problem. Our work, therefore, builds upon the theoretical tools developed for this class of problems. The offline budget allocation problem has been studied by Alon et al. [12], Soma et al. [13], the latter of which propose the lattice submodularity and a constant approximation algorithm based on this property. Our offline overlapping MuLaNE setting is based on a similar approach, but we provide a better approximation ratio compared to the original analysis in Alon et al. [12]. Moreover, we further show that the stationary setting enjoys DR-submodularity, leading to an efficient algorithm with a better approximation ratio.

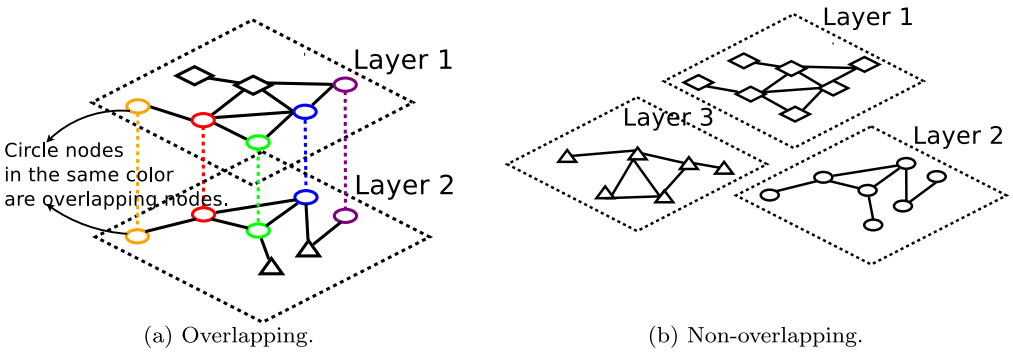
The multi-armed bandit (MAB) problem is first studied by Robbins [14] and then extended by many studies (cf. [15]). Our online setting fits into the general Combinatorial MAB (CMAB) framework of Chen et al. [7], Gai et al. [16]. CUCB is proposed as a general algorithm for CMAB [7,17]. Our study includes several adaptations, such as handling unknown node weights, defining intermediate random variables as base arms, and so on. The community exploration problem, as addressed by Chen et al. [18], represents a special case of MuLaNE that is restricted to non-overlapping complete graphs. Consequently, their approach is inherently different from and simpler than the generalized methods we propose. While Chen et al. [7], Wang and Chen [17], Liu et al. [19] can also tackle the MuLaNE problem, their algorithms' regret upper bound scales linearly with the number of explorers m. Our explorer-relaxed online algorithm improves upon this result, achieving a tighter bound by a factor of $\tilde{O}\left(\frac{m}{\log m}\right)$.

3. Preliminaries

Basic Notations. In this paper, we use $\mathbb{R}$ and $\mathbb{Z}$ to denote the sets of real numbers and integers, respectively, with subscripts ≥ 0 and > 0 representing their non-negative and positive subsets. We consider a multi-layered network $\mathcal{G}(\mathcal{V}, \mathcal{E}, \boldsymbol{\sigma})$, consisting of m layers $L_1, \ldots, L_m$ and N unique nodes in the set $\mathcal{V}$, with fixed node weights $\boldsymbol{\sigma} \in [0,1]^{\mathcal{V}}$. Let $[m]$ denote the set $1, 2, \ldots, m$. Each layer L_i, for $i \in [m]$, is a subgraph of $\mathcal{G}$ and is represented as a weighted directed graph $\mathcal{G}_i(\mathcal{V}_i, \mathcal{E}_i; w_i)$, where $\mathcal{V}_i \subseteq \mathcal{V}$ is the set of vertices in layer L_i, $\mathcal{E}_i \subseteq \mathcal{V}_i \times \mathcal{V}_i$ is the set of edges, and $w_i : \mathcal{E}_i \to \mathbb{R}_{>0}$ is the edge weight function for the edges in $\mathcal{G}_i$. Two layers L_i and L_j are considered *non-overlapping* if $\mathcal{V}_i \cap \mathcal{V}_j = \emptyset$. The network $\mathcal{G}$ is called a *non-overlapping* multi-layered network if all layers are non-overlapping; otherwise, it is an *overlapping* multi-layered network. Fig. 2 illustrates both types of networks. Let $n_i = |\mathcal{V}_i|$ denote the size of layer L_i, and the adjacency matrix of $\mathcal{G}_i$ is defined as $\boldsymbol{A_i} \in \mathbb{R}_{\geq 0}^{n_i \times n_i}$, where $\boldsymbol{A_i}[u, v] = w_i((u, v))$ if $(u, v) \in \mathcal{E}_i$ and 0 otherwise. Based on the definition of $\boldsymbol{A_i}$, we also define the transition probability matrix $\boldsymbol{P}_i \in \mathbb{R}_{\geq 0}^{n_i \times n_i}$ as $\boldsymbol{P}_i[u, v] := \boldsymbol{A_i}[u, v] / (\sum_{w:(u,w)\in\mathcal{E}_i} \boldsymbol{A_i}[u, w])$.

Exploration Rule. Each layer L_i is associated with an explorer W_i, a budget $k_i \in \mathbb{Z}_{\geq 0}$ and an initial starting distribution $\boldsymbol{\alpha}_i = (\alpha_{i,u})_{u\in\mathcal{V}_i} \in \mathbb{R}_{\geq 0}^{n_i}$ with $\sum_{u\in\mathcal{V}_i} \alpha_{i,u} = 1$. The exploration process of W_i is identical to applying *weighted random walks* within layer L_i. Specifically, the explorer W_i starts the exploration from a random node $v_1 \in \mathcal{V}_i$ with probability α_{i,v_1}; it consumes one unit of budget and continues the exploration by walking from v_1 to one of its out-neighbors, say v_2, with the transition probability matrix $\boldsymbol{P}_i[v_1, v_2]$. Consider visiting the initial node v_1 as the first step, the process is repeated until the explorer W_i walks k_i steps and visits k_i nodes (with possibly duplicated nodes). Note that our results can be easily generalized by allowing W_i to take $\lambda_i \in \mathbb{Z}_{>0}$ steps per unit of budget. It can be readily verified that this modification does not affect the algorithm or its analysis. For simplicity, we set $\lambda_i = 1$.

Reward Function. We define the exploration trajectory for explorer W_i after exploring k_i nodes as $\Phi(i, k_i) := (X_{i,1}, \ldots, X_{i,k_i})$, where $X_{i,j} \in \mathcal{V}_i$ denotes the node visited at j-th step, and $\Pr(X_{i,1}{=}u) = \alpha_{i,u}, u \in \mathcal{V}_i$. Note that $\alpha_{i,u}$ denotes the probability of node u being the initial starting node for W_i. The reward for $\Phi(i, k_i)$ is defined as the total weights of *unique* nodes visited by W_i, i.e., $\sum_{v\in\bigcup_{j=1}^{k_i}\{X_{i,j}\}} \sigma_v$. Considering trajectories of all W_i's, the total reward is the total weights of *unique* nodes visited by all random walkers, i.e., $\sum_{v\in\bigcup_{i=1}^{m}\bigcup_{j=1}^{k_i}\{X_{i,j}\}} \sigma_v$. For notational simplicity, let $\mathcal{G} := (\mathcal{G}_1, \ldots, \mathcal{G}_m)$, $\boldsymbol{\alpha} := (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_m)$ and $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_{|\mathcal{V}|})$ be the parameters of a



(a) Overlapping. (b) Non-overlapping.

Fig. 2. Two types of multi-layered networks.

problem instance and $\boldsymbol{k}:=(k_1,\dots,k_m)$ be the allocation vector. For overlapping multi-layered network $\mathcal{G}$, the total expected reward is

$$r_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(\boldsymbol{k}):=\mathbb{E}_{\Phi(1,k_1),\dots,\Phi(m,k_m)}\left[\sum_{v\in\bigcup_{i=1}^{m}\bigcup_{j=1}^{k_i}\{X_{i,j}\}}\sigma_v\right], \tag{1}$$

where its explicit formula will be discussed later in Section 4. For non-overlapping $\mathcal{G}$, there are no overlapping nodes among different layers, i.e., a node can only be visited by the random explorer from the same layer. Hence, the reward function can be simplified and written as the summation over separated layers,

$$r_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(\boldsymbol{k}) := \sum_{i=1}^{m}\mathbb{E}_{\Phi(i,k_i)}\left[\sum_{v\in\bigcup_{j=1}^{k_i}\{X_{i,j}\}}\sigma_v\right]. \tag{2}$$

Problem Formulation. We are interested in the *budget allocation problem*: how to distribute a total budget $B\in\mathbb{Z}_{\geq 0}$ among m explorers to maximize the total weight of unique nodes visited. Additionally, each explorer $i\in[m]$ has an individual budget constraint c_i, meaning the allocated budget k_i must satisfy $k_i\leq c_i$. We assume $c_i\leq B$ and $\sum_{i=1}^{m}c_i\geq B$, as the problem becomes trivial otherwise. Specifically, if $\sum_{i=1}^{m}c_i<B$, the optimal solution is simply to allocate $k_i=c_i$ for all explorer i, which satisfies the total budget constraint.

Definition 1. Given graph structures $\mathcal{G}:=(\mathcal{G}_1,\dots,\mathcal{G}_m)$, total budget B, starting distributions $\boldsymbol{\alpha}:=(\boldsymbol{\alpha}_1,\dots,\boldsymbol{\alpha}_m)$, and budget constraints $\boldsymbol{c}:=(c_1,\dots,c_m)$, the **Mu**lti-**La**yered **N**etwork **E**xploration problem, denoted as `MuLaNE`, is formulated as the following optimization problem,

$$\max_{\boldsymbol{k}}\ r_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(\boldsymbol{k})\ s.t.\ \boldsymbol{k}\in\mathbb{Z}_{\geq 0}^{m}\leq\boldsymbol{c},\ \sum_{i=1}^{m}k_i\leq B, \tag{3}$$

where $r_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(\boldsymbol{k})$ is given by Eq. (1) or Eq. (2).

We consider the following two settings for the `MuLaNE` problem:

`Offline Setting`. For the offline setting, all the parameters of the problem instance $(\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma},\boldsymbol{c},B)$ are given in advance. The objective is to determine the optimal allocation of the budget $\boldsymbol{k}^*$ by solving the optimization problem defined in Eq. (3).

`Online Setting`. For the online setting, we consider T-round explorations. Prior to exploration, only an upper bound on the total number of nodes in $\mathcal{G}$ and the number of layers m are known[2] The network structure $\mathcal{G}$, starting distributions $\boldsymbol{\alpha}$, and node weights $\boldsymbol{\sigma}$ are initially unknown. In round $t\in[T]$, we choose the budget allocation $\boldsymbol{k}_t:=(k_{t,1},\dots,k_{t,m})$ based solely on observations from previous rounds, where $k_{t,i}\in\mathbb{Z}_{\geq 0}$ is the budget allocated to the i-th layer and $\sum_{i=1}^{m}k_{t,i}\leq B,\boldsymbol{k}_t\leq\boldsymbol{c}$. By choosing *action* $\boldsymbol{k}_t$, each explorer W_i performs $k_{t,i}$ exploration steps on layer L_i, generating an exploration trajectory $\Phi(i,k_{t,i})=(X_{i,1},\dots,X_{i,k_{t,i}})$. After the action, the exploration trajectories $\Phi(i,k_{t,i})$ and the fixed importance weights σ_u for all nodes $u\in\Phi(i,k_{t,i})$ are revealed as *feedback* (Random node weights with unknown mean vector $\boldsymbol{\sigma}$ is considered in Appendices F and G). Feedback results are used to learn parameters related to the graph structures $\mathcal{G}$, the starting distribution $\boldsymbol{\alpha}$, and the node weights $\boldsymbol{\sigma}$. This information guides the selection of better budget allocations in subsequent rounds. In round t, the reward obtained is the total weights of unique nodes visited by all random explorers. Our objective is to design an efficient online learning algorithm, denoted as A, to guide the budget allocation process and maximize the cumulative reward over T rounds.

A key challenge for the online algorithm is managing the exploration-exploitation tradeoff. The performance of an online learning algorithm is commonly measured by cumulative regret. Formally, the T-round (ξ,β)-approximation regret of algorithm A is defined as:

$$Reg_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(T)=\xi\beta T\cdot r_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(\boldsymbol{k}^*)-\mathbb{E}\left[\sum_{t=1}^{T}r_{\mathcal{G},\boldsymbol{\alpha},\sigma}(\boldsymbol{k}_t^A)\right], \tag{4}$$

where $r_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(\boldsymbol{k}^*)$ is the reward value for the optimal budget allocation $\boldsymbol{k}^*$, $\boldsymbol{k}_t^A$ is budget allocation selected by the learning algorithm A in round t, and the expectation is taken over the randomness of both the algorithm and the exploration trajectories across all T rounds. The constants (ξ,β) represent the approximation guarantee of an offline oracle, as explained below. Similar to other online learning

[2] More precisely we only need to know the number of independent explorers. If multiple explorers operate on the same layer, this is equivalent to treating them as separate layers with identical graph structures.

frameworks [7,17,20], we assume that the online learning algorithm has access to an offline (ξ, β)-approximation oracle, which for problem instance $(\mathcal{G}, \boldsymbol{\alpha}, \boldsymbol{\sigma}, \boldsymbol{c}, B)$ outputs an action $\boldsymbol{k}$ such that $\Pr\left(r_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(\boldsymbol{k}) \geq \xi \cdot r_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(\boldsymbol{k}^*)\right) \geq \beta$. Importantly, in our actual online learning algorithms, we do not try to recover the graph structure $\mathcal{G}$ and the starting distribution $\boldsymbol{\alpha}$ but take certain intermediate parameters as inputs.

Submodularity and DR-Submodularity Over Integer Lattices. To address the `MuLaNE` problem, we exploit the submodular and diminishing returns (DR)-submodular properties of the reward function. Let $\mathbb{R}$ and $\mathbb{Z}_{\geq 0}$ denote the set of reals and non-negative integers. For any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{Z}^m_{\geq 0}$, we denote $\boldsymbol{x} \vee \boldsymbol{y}, \boldsymbol{x} \wedge \boldsymbol{y} \in \mathbb{Z}^m_{\geq 0}$ as the coordinate-wise maximum and minimum of these two vectors, i.e., $(\boldsymbol{x} \vee \boldsymbol{y})_i = \max\{x_i, y_i\}, (\boldsymbol{x} \wedge \boldsymbol{y})_i = \min\{x_i, y_i\}$. We define a function $f : \mathbb{Z}^m_{\geq 0} \to \mathbb{R}$ over the integer lattice $\mathbb{Z}^m_{\geq 0}$ as a *submodular* function if the following inequality holds for any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{Z}^m_{\geq 0}$:

$$f(\boldsymbol{x} \vee \boldsymbol{y}) + f(\boldsymbol{x} \wedge \boldsymbol{y}) \leq f(\boldsymbol{x}) + f(\boldsymbol{y}). \tag{5}$$

Let $\text{supp}^+(\boldsymbol{x} - \boldsymbol{y})$ denote the set $I = \{i \in [m] : x_i > y_i\}$, $\chi_i = (0, \ldots, 1, \ldots, 0)$ be the one-hot vector whose i-th element is 1 and 0 otherwise, and $\boldsymbol{x} \geq \boldsymbol{y}$ means $x_i \geq y_i$ for all $i \in [m]$. We define a function $f : \mathbb{Z}^m_{\geq 0} \to \mathbb{R}$ as a *DR-submodular* function if the following inequality holds for any $\boldsymbol{x} \leq \boldsymbol{y}$ and $i \in [m]$,

$$f(\boldsymbol{y} + \chi_i) - f(\boldsymbol{y}) \leq f(\boldsymbol{x} + \chi_i) - f(\boldsymbol{x}). \tag{6}$$

We say a function f is *monotone* if for any $\boldsymbol{x} \leq \boldsymbol{y}$, $f(\boldsymbol{x}) \leq f(\boldsymbol{y})$. Without loss of generality, we assume monotone functions are normalized, i.e., $f(\boldsymbol{0}) = 0$. For a function $f : \{0,1\}^m \to \mathbb{R}$ defined on a set, submodularity and DR-submodularity are equivalent. On general integer lattices, however, the distinction is critical. A DR-submodular function $f : \mathbb{Z}^m_{\geq 0} \to \mathbb{R}$ is always a submodular function, but the converse is not necessarily true. Specifically, DR-submodularity implies that the marginal gain from a single-unit increase in any coordinate is non-increasing. The more general lattice submodularity does not require this strict, per-unit diminishing return property.

4. Equivalent bipartite coverage model

In this section, we derive the formulation of the reward function, a crucial step for both offline optimization and online learning. We first introduce an equivalent bipartite coverage model to formalize the problem, then use this framework to express the reward function mathematically, and finally analyze the properties of its core component: the node visiting probability.

4.1. Formulation of reward function

To derive the explicit formulation of the reward function $r_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(\boldsymbol{k})$ in Eq. (1), we construct an undirected bipartite coverage graph $\mathcal{B}(\mathcal{W}, \mathcal{V}, \mathcal{E}')$, where $\mathcal{W} = \{W_1, \ldots, W_m\}$ denotes m random explorers, $\mathcal{V} = \bigcup_{i\in[m]} \mathcal{V}_i$ denotes all possible distinct nodes to be explored in $\mathcal{G}$, and the edge set $\mathcal{E}'=\{(W_i, u) | u \in \mathcal{V}_i, i \in [m]\}$ indicating whether node u could be visited by W_i. For each edge $(W_i, u) \in \mathcal{E}'$, we associate it with $c_i + 1$ visiting probabilities denoted as $P_{i,u}(k_i)$ for $k_i \in \{0\} \cup [c_i]$. $P_{i,u}(k_i)$ represents the probability that the node u is visited by the random walker W_i given the budget k_i, i.e., $P_{i,u}(k_i) = \Pr(u \in \Phi(i, k_i))$. We will provide the analytical formula and detailed properties of these visiting probabilities later on. Since W_i can never visit the node outside the i-th layer (i.e., $u \notin \mathcal{G}_i$), we set $P_{i,u}(k_i) = 0$ if $(W_i, u) \notin \mathcal{E}'$. Then, given budget allocation $\boldsymbol{k}$, the probability of a node u visited by at least one random explorer is $\Pr(u, \boldsymbol{k})=1 - \prod_{i\in[m]}(1 - P_{i,u}(k_i))$ because each W_i has the independent probability $P_{i,u}(k_i)$ to visit u. By summing over all possible nodes, the reward function is

$$r_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(\boldsymbol{k}) = \sum_{u\in\mathcal{V}} \sigma_u \left(1 - \prod_{i\in[m]} \left(1 - P_{i,u}(k_i)\right)\right) \tag{7}$$

According to Eq. (2), for a non-overlapping multi-layered network, we can rewrite the reward function as:

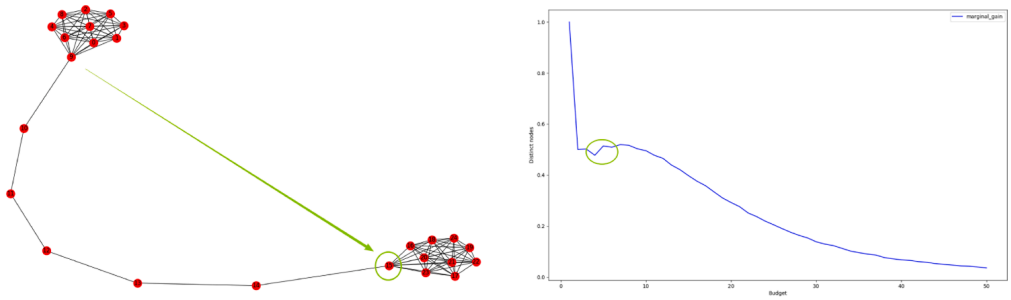
$$r_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(\boldsymbol{k}) = \sum_{i\in[m]} \sum_{u\in\mathcal{V}_i} \sigma_u P_{i,u}(k_i). \tag{8}$$

Remark. The bipartite coverage model abstracts the integration of graph structure and random-walk exploration into $P_{i,u}(k_i)$, thereby determining the reward function. Its generality allows seamless extensions, e.g., adding a new random walker to a layer corresponds to inserting a node in $\mathcal{W}$ and associated edges in $\mathcal{E}'$, without altering the fundamental properties of the reward function or the applicability of our optimization and learning techniques.

4.2. Properties of the visiting probability

The quantities $P_{i,u}(k_i)$'s in Eqs. (7) and (8) are crucial for both our offline and online algorithms, and thus we provide their analytical formulas and properties here. To analyze the property of the reward function, we apply the absorbing Markov Chain technique to calculate $P_{i,u}(k_i)$. For $u \notin \mathcal{G}_i$, $P_{i,u}(k_i) = 0$; For $u \in \mathcal{G}_i$, we create an absorbing Markov Chain $\boldsymbol{P_i(u)} \in \mathbb{R}^{n_i \times n_i}$ by setting the target node u as the absorbing node. We derive the corresponding transition matrix $\boldsymbol{P_i(u)}$ as $\boldsymbol{P_i(u)}[v, \cdot] = \chi_u^\top$ if $v = u$ and $\boldsymbol{P_i(u)}[v, \cdot] = \boldsymbol{P_i}[v, \cdot]$ otherwise, where $\boldsymbol{P_i}[v, \cdot]$ denotes the row vector corresponding to the node v of $\boldsymbol{P_i}$, and $\chi_u = (0, \ldots, 0, 1, 0, \ldots, 0)^\top$ denotes the one-hot vector with 1 at the u-th entry and 0 elsewhere.[3] Intuitively, $\boldsymbol{P_i(u)}$ corresponds to the transition probability matrix of $\mathcal{G}_i$ after removing all out-edges of u and adding a self-loop to itself in the original graph $\mathcal{G}_i$. The random walker W_i will be trapped in u if

[3] When related to matrix operations, we treat all vectors as column vectors by default.



(a) Barbell graph. (b) Marginal gain.

Fig. 3. An example showing $g_{i,u}(k_i)$ is not monotone.

it ever visits u. We observe that $P_{i,u}(k_i)$ equals to the probability the random walker stays in the absorbing node u at step $k_i \in \mathbb{Z}_{>0}$ (trivially, $P_{i,u}(0) = 0$), if it starts from the initial distribution $\boldsymbol{\alpha}_i$, i.e.,

$$P_{i,u}(k_i) = \boldsymbol{\alpha}_i^\top \boldsymbol{P}_i(\boldsymbol{u})^{k_i-1} \boldsymbol{\chi}_u. \tag{9}$$

To see this, the probability that node u has not been visited before (included) step k equals to $\boldsymbol{\alpha}\boldsymbol{P}(u)^{k-1}(\mathbf{1} - \boldsymbol{\chi}_u)$, where $\mathbf{1} \in \mathbb{R}^{1\times n}$ is the all 1 vector. From this, we have $1 - \Pr(u, k) = 1 - \boldsymbol{\alpha}\boldsymbol{P}(u)^{k-1}\boldsymbol{\chi}_u$. Then define the marginal gain of $P_{i,u}(\cdot)$ at step $k_i \geq 1$ as,

$$g_{i,u}(k_i) = P_{i,u}(k_i) - P_{i,u}(k_i - 1). \tag{10}$$

The physical meaning of $g_{i,u}(k_i)$ is the probability that node u is visited exactly at the k_i-th step and not visited before k_i. Now, we can show $g_{i,u}(k_i)$ is non-negative,

Lemma 1. *$g_{i,u}(k_i) \geq 0$ for any $i \in [m]$, $u \in \mathcal{V}_i$, $k_i \in \mathbb{Z}_{>0}$.*

This means $P_{i,u}(k_i)$ is non-decreasing with respect to step k_i. In other words, the more budgets we allocate to W_i, the higher probability W_i can visit u in k_i steps.

Starting from Arbitrary Distributions: Although $P_{i,u}(k_i)$ is monotone, i.e., non-decreasing, w.r.t k_i, $g_{i,u}(k_i)$ may not be monotonic non-increasing under arbitrary staring distributions. Fig. 3 illustrates this using a classic barbell graph, shown in Fig. 3a, with two densely connected clusters joined by a single path. A random walk starting at one end is likely confined to the first cluster initially, yielding negligible marginal gain for reaching a target node in the distant cluster. However, once the walk crosses the connecting path–a low-probability event requiring sufficient budgets–it gains access to the entire second cluster, causing a sharp spike in marginal gain. This non-monotonic behavior is depicted in Fig. 3b, which shows that $g_{i,u}(k_i)$ is not always non-increasing. Intuitively, there may exist one critical step k_i such that $P_{i,u}(k_i)$ suddenly increases by a large value (see Section B.1). This means that $P_{i,u}(k_i)$ lacks the diminishing return (or "discretely concave") property, which many problems rely on to provide good optimization [21,22]. In other words, we are dealing with a more challenging non-concave optimization problem for the discrete budget allocation.

Starting from Stationary Distributions: If our walkers start with the stationary distribution, the following property holds: $g_{i,u}(k_i)$ is non-increasing w.r.t k_i.

Lemma 2. *$g_{i,u}(k_i + 1) - g_{i,u}(k_i) \leq 0$ for any $i \in [m]$, $u \in \mathcal{V}_i$, $k_i \in \mathbb{Z}_{>0}$, if $\boldsymbol{\alpha}_i = \boldsymbol{\pi}_i$, where $\boldsymbol{\pi}_i^\top \boldsymbol{P}_i = \boldsymbol{\pi}_i^\top$.*

More interestingly, the stationary distribution $\boldsymbol{\pi}_i$ is the only starting distribution for $\boldsymbol{P}_i$ such that for any $u \in \mathcal{V}_i, k_i \in \mathbb{Z}_{>0}$, $g_{i,u}(k_i + 1) - g_{i,u}(k_i) \leq 0$ when $\boldsymbol{P}_i$ is ergodic (the random walker can visit any node in finite steps starting from any node) and there are no self loops (the random walker will not stay in the same node in any two consecutive steps) in $\mathcal{G}_i$. This means that the diminishing return property is missing for most starting distributions. The proofs are included in the Appendix B.

5. Offline optimization for `MuLaNE`

This section develops our optimization algorithms for the offline `MuLaNE` problem, which serve as the foundation and performance benchmark for the online learning setting. Our approach is to systematically analyze different scenarios, starting with the most general case and then moving to more specialized settings where stronger theoretical guarantees can be achieved. First, we consider the general case, where layers are overlapping and starting distributions are arbitrary. Next, we consider the starting distribution as the stationary distribution, and give solutions with better solution quality and time complexity for the approximation algorithms. Then we analyze special cases where layers are non-overlapping and give exact optimal solutions for arbitrary distributions. Table 1 summarizes the offline models and algorithmic results.

5.1. Offline algorithm for overlapping mulane

Starting from Arbitrary Distributions. Based on the equivalent bipartite coverage model, one can observe that our problem formulation is a generalization of the Probabilistic Maximum Coverage (PMC) [7,23,24] problem, which is NP-hard and has a $(1 - 1/e)$

Table 1
Summary of the offline models and algorithms.

Overlapping?	Starting distribution	Algorithm	Apprx ratio	Time complexity
✓	*Arbitrary*	Budget Effective Greedy	$(1-e^{-\eta})$ [a]	$O(B\|\|\boldsymbol{c}\|\|_\infty mn_{\max} + \|\|\boldsymbol{c}\|\|_\infty mn_{\max}^3)$
✓	*Stationary*	Myopic Greedy	$(1-1/e)$	$O(Bmn_{\max} + \|\|\boldsymbol{c}\|\|_\infty mn_{\max}^3)$
×	*Arbitrary*	Dynamic Programming	1	$O(B\|\|\boldsymbol{c}\|\|_\infty m + \|\|\boldsymbol{c}\|\|_\infty mn_{\max}^3)$
×	*Stationary*	Myopic Greedy	1	$O(B\log m + \|\|\boldsymbol{c}\|\|_\infty mn_{\max}^3)$

[a] $1-e^{-\eta} \approx 0.357$, where η is the solution of $e^\eta = 2-\eta$.

Algorithm 1 Budget effective greedy (BEG) algorithm for the overlapping MuLaNE.

Input: Network $\mathcal{G}$, starting distributions $\boldsymbol{\alpha}$, node weights $\boldsymbol{\sigma}$, budget B, constraints $\boldsymbol{c}$.
Output: Budget allocation $\boldsymbol{k}$.
1: Compute visiting probabilities $(P_{i,u}(b))_{i\in[m],u\in\mathcal{V},b\in[c_i]}$ according to Eq. (9).
2: $\boldsymbol{k} \leftarrow \text{BEG}((P_{i,u}(b))_{i\in[m],u\in\mathcal{V},b\in[c_i]}, \boldsymbol{\sigma}, B, \boldsymbol{c})$.
3: **Procedure** BEG($(P_{i,u}(b))_{i\in[m],u\in\mathcal{V},b\in[c_i]}$, $\boldsymbol{\sigma}$, B, $\boldsymbol{c}$)
4: Let $\boldsymbol{k} := (k_1,\dots,k_m) \leftarrow \boldsymbol{0}$, $K \leftarrow B$.
5: Let $\mathcal{Q} \leftarrow \{(i,b_i) \mid i\in[m], 1\le b_i \le c_i\}$.
6: **while** $K>0$ and $\mathcal{Q}\neq\emptyset$ **do**
7: $(i^*,b^*) \leftarrow arg\,max_{(i,b)\in\mathcal{Q}}\delta(i,b,\boldsymbol{k})/b$. {Eq. (11)}
8: $k_{i^*} \leftarrow k_{i^*}+b^*$, $K \leftarrow K-b^*$.
9: Modify all pairs $(i,b)\in\mathcal{Q}$ to $(i,b-b^*)$.
10: Remove all paris $(i,b)\in\mathcal{Q}$ such that $b\le 0$.
11: **end while**
12: **for** $i\in[m]$ **do**
13: **if** $r_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(c_i\chi_i) > r_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(\boldsymbol{k})$, **then** $\boldsymbol{k} \leftarrow c_i\chi_i$.
14: **end for**
15: **return** $\boldsymbol{k} := (k_1,\dots,k_m)$.
16: **end Procedure**

approximation based on submodular set function maximization. However, our problem is more general in that we want to select multi-sets from $\mathcal{W}$ with budget constraints, and the reward function in general does not have the DR-submodular property for an arbitrary starting distribution. Nevertheless, we have the following lemma to solve our problem.

Lemma 3. *For any network $\mathcal{G}$, distribution $\boldsymbol{\alpha}$ and weights $\boldsymbol{\sigma}$, $r_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(\cdot) : \mathbb{Z}_{\ge 0}^m \to \mathbb{R}$ is monotone and submodular.*

Leveraging on the monotone submodular property, we design a Budget Effective Greedy algorithm (Algorithm 1). The core of Algorithm 1 is the procedure of BEG. Let $\delta(i,b,\boldsymbol{k})$ be the *per-unit marginal gain* $(r_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(\boldsymbol{k}+b\chi_i) - r_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(\boldsymbol{k}))/b$ for allocating b more budgets to layer i, which equals to

$$\sum_{u\in\mathcal{V}} \sigma_u \prod_{j\neq i}(1-P_{j,u}(k_j))(P_{i,u}(k_i+b)-P_{i,u}(k_i))/b. \tag{11}$$

The BEG procedure consists of two parts and maintains a queue $\mathcal{Q}$, where any pair $(i,b)\in\mathcal{Q}$ represents a tentative plan of allocating b more budgets to layer i. The first part is built around the while loop (lines 6–10), where each iteration greedily selects the (i,b) pair in $\mathcal{Q}$ such that the *per-unit marginal gain* $\delta(i,b,\boldsymbol{k})$ is maximized. If b is within the remaining budget, we allocate b to layer i; otherwise, we proceed to the next iteration. The second part is a for loop (lines 12–13), where in the i-th round we attempt to allocate all c_i budgets to layer i and replace the current best budget allocation if we have a larger reward.

Theorem 1. *Algorithm 1 obtains a $(1-e^{-\eta}) \approx 0.357$-approximate solution, where η is the solution of equation $e^\eta = 2-\eta$, to the overlapping MuLaNE problem with an arbitrary starting distribution.*

Proof. See Section C.1 for details. □

Line 1 uses $O(m\|\boldsymbol{c}\|_\infty n_{\max}^3)$ time to pre-calculate visiting probabilities based on Eq. (9), where $n_{\max} = \max_i |\mathcal{V}_i|$. In the procedure of BEG, the while loop contains B iterations, the size of queue $\mathcal{Q}$ is $O(m\|\boldsymbol{c}\|_\infty)$, and line 7 uses $O(n_{\max})$ to calculate $\delta(i,b,\boldsymbol{k})$ by proper pre-computation and update, thus the time complexity of Algorithm 1 is $O(B\|\boldsymbol{c}\|_\infty mn_{\max} + m\|\boldsymbol{c}\|_\infty n_{\max}^3)$.

Remark 1. The idea of combining the greedy algorithm with enumerating solutions on one layer is also adopted in previous works [12, 25], but they only give a $\frac{1}{2}(1-e^{-1}) \approx 0.316$-approximation analysis. In this paper, we provide a novel analysis with a better $(1-e^{-\eta}) \approx 0.357$-approximation, .

Remark 2. Another algorithm proposed by Alon et al. [12], based on partial enumeration techniques (i.e., BEGE), achieves a $(1-1/e)$ approximation ratio. This ratio is the best possible in polynomial time unless P = NP. However, the algorithm incurs prohibitively high computational cost, with a time complexity of $O(B^4m^4\|\boldsymbol{c}\|_\infty n_{\max} + m\|\boldsymbol{c}\|_\infty n_{\max}^3)$. Similarly, the algorithm in Soma et al. [13] also

Algorithm 2 Myopic greedy (MG) algorithm for the non-overlapping MuLaNE.

1: Same input, output and lines 1–2 as in Algorithm 1, except replacing BEG with MG.
2: **Procedure** MG($(P_{i,u}(b))_{i\in[m],u\in\mathcal{V},b\in[c_i]}$, $\boldsymbol{\sigma}$, B, $\boldsymbol{c}$)
3: Let $\boldsymbol{k} := (k_1, \dots, k_m) \leftarrow \mathbf{0}$, $K \leftarrow B$.
4: **while** $K > 0$ **do**
5: $i^* \leftarrow arg\ max_{i\in[m],k_i+1\le c_i} \delta(i, 1, \boldsymbol{k})$. {Eq. (11)}
6: $k_{i^*} \leftarrow k_{i^*} + 1$, $K \leftarrow K - 1$.
7: **end while**
8: **return** $\boldsymbol{k} = (k_1, \dots, k_m)$.
9: **end Procedure**

achieves a $(1 - 1/e)$ approximation ratio but with a time complexity of $O(mB^5 n_{\max}^4)$. We also provide the algorithm with a $(1 - 1/e)$-approximate solution for the overlapping MuLaNE problem with arbitrary starting distributions, as detailed in the analysis in D.

Starting from Stationary Distributions. We also consider the special case where each random explorer W_i starts from the stationary distribution $\boldsymbol{\pi}_i$ with $\boldsymbol{\pi}_i^\top \boldsymbol{P} = \boldsymbol{\pi}_i^\top$. If a Markov chain $\boldsymbol{P}$ admits a stationary distribution $\boldsymbol{\pi}$, the random walk will gradually approach $\boldsymbol{\pi}$ from arbitrary starting distributions. Furthermore, in the random walk literature, the stationary distribution often gives nice properties for the theoretical analysis. In this case, we have the following stronger DR-submodularity.

Lemma 4. *For any network $\mathcal{G}$, stationary distributions $\boldsymbol{\pi}$ and node weights $\boldsymbol{\sigma}$, function $r_{\mathcal{G},\boldsymbol{\pi},\boldsymbol{\sigma}}(\cdot) : \mathbb{Z}_{\ge 0}^m \to \mathbb{R}$ is monotone and DR-submodular, i.e., $r_{\mathcal{G}}(\boldsymbol{\pi}, \boldsymbol{y} + \boldsymbol{\chi}_j) - r_{\mathcal{G}}(\boldsymbol{\pi}, \boldsymbol{y}) \le r_{\mathcal{G}}(\boldsymbol{\pi}, \boldsymbol{x} + \boldsymbol{\chi}_j) - r_{\mathcal{G}}(\boldsymbol{\pi}, \boldsymbol{x})$, and $r_{\mathcal{G}}(\boldsymbol{\pi}, \boldsymbol{x}) \le r_{\mathcal{G}}(\boldsymbol{\pi}, \boldsymbol{y})$ for any $\boldsymbol{x} \le \boldsymbol{y}$, $j \in [m]$.*

Since the reward function is DR-submodular, the procedure of BEG can be replaced by the simple MG procedure in Algorithm 2 with a better approximation ratio. The time complexity is also improved to $O(Bmn_{\max} + m\|\boldsymbol{c}\|_\infty n_{\max}^3)$. As shown in Algorithm 2, first, line 5 always select the pair (i^*, b^*) such that $b^* = 1$ because $r_{\mathcal{G}}(\boldsymbol{\alpha}, \boldsymbol{k} + b\boldsymbol{\chi}_i) - r_{\mathcal{G}}(\boldsymbol{\alpha}, \boldsymbol{k} + (b-1)\boldsymbol{\chi}_i) \le r_{\mathcal{G}}(\boldsymbol{\alpha}, \boldsymbol{k} + \boldsymbol{\chi}_i) - r_{\mathcal{G}}(\boldsymbol{\alpha}, \boldsymbol{k})$ for arbitrary $\boldsymbol{k}$, i and b. Thus, Algorithm 1 can always extend the current budget allocation $\boldsymbol{k}$ by 1 and never enters line 10. Therefore, line 10 and lines 12–13 of Algorithm 1 can be safely removed in Algorithm 2. Following a similar argument to the proof of Theorem 1, we can show that the following theorem holds.

Theorem 2. *Algorithm 2 obtains a $(1 - 1/e)$-approximate solution to the overlapping MulaNE with the stationary starting distributions.*

Proof. See Section C.2 for details. □

5.2. Offline algorithm for non-overlapping MuLaNE

For the non-overlapping MuLaNE setting, we are able to obtain the exact optimal solution rather than merely an approximate one. Specifically, under the stationary starting distribution, a slight modification of the greedy algorithm (Algorithm 2) yields the exact optimum; for arbitrary starting distributions, we further design a dynamic programming algorithm that computes the exact optimal solution (see E).

6. Online learning for MuLaNE

This section addresses the online overlapping and non-overlapping MuLaNE problem, where the structure of the network $\mathcal{G}$, the distribution $\boldsymbol{\alpha}$, and node weights $\boldsymbol{\sigma}$ *are not known a priori*. Instead, the only information revealed to the decision maker includes total budget B, the number of layers m, the number of the target nodes $|\mathcal{V}|$ (or an upper bound of it), and the budget constraint $\boldsymbol{c}$. For the unknown network structure, our solution is to estimate certain intermediate parameters, instead of estimating the graph structure (namely the transition probability matrix $\boldsymbol{P}_i$) and the starting distribution $\boldsymbol{\alpha}$. While one could try to estimate each transition probability matrix $\boldsymbol{P}_i$ directly, this method presents two major issues. First, it is difficult to analyze how the estimated $\hat{\boldsymbol{P}}_i$ affects the performance of online algorithms, since the reward function given by Eq. (7) is highly non-linear. Second, this method requires extensive matrix calculations in each round, which is inefficient for large networks. Subsequently, our solution bypasses the transition matrices $\boldsymbol{P}_i$ and instead directly estimate the visiting probabilities $P_{i,u}(b) \in [0, 1]$. For unknown node weight σ_u, we maintain an optimistic weight before we first visit u. With a little abuse of the notation, we use $\boldsymbol{\mu}$ to denote the unknown visiting probability parameters and $r_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\boldsymbol{k})$ to denote the reward $r_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(\boldsymbol{k})$.

Specifically, we maintain a set of base arms $\mathcal{A} = \{(i, u, b) | i \in [m], u \in \mathcal{V}, b \in [c_i]\}$, where the total number $|\mathcal{A}| = \sum_{i\in[m]} c_i|\mathcal{V}|$. For each base arm $(i, u, b) \in \mathcal{A}$, we denote $\mu_{i,u,b}$ as the true value of each base arm, i.e., $\mu_{i,u,b} = P_{i,u}(b)$. For the unknown fixed node weights, we maintain the optimistic weight $\bar{\sigma}_u = 1$ if $u \in \mathcal{V}$ has not been visited. After u is first visited and its true value σ_u is revealed, we replace $\bar{\sigma}_u$ with σ_u. Beyond the fixed-node-weight scenario, we also consider the more complex case of random node weights, which will be detailed in subsequent sections.

To provide a high-level overview of our online learning framework, we present the complete workflow in Fig. 4. The process is an iterative feedback loop designed to operate on an initially unknown network structure. In each round, the agent begins with its

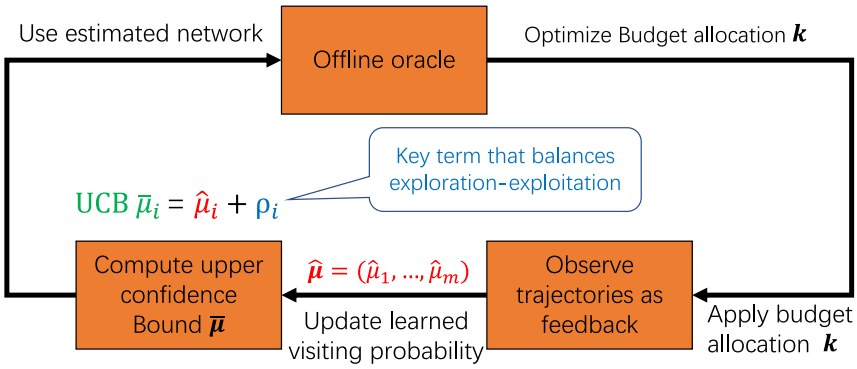


Fig. 4. Complete iterative process of MuLaNE.

current estimates of the network parameters (e.g., the visiting probabilities). It first calls an offline oracle to determine an optimized budget allocation $\boldsymbol{k}$, based on these estimates. The agent then executes the exploration by performing random walks according to this budget and observes the resulting trajectories as feedback. This new information is used to update and refine the learned parameters for the next round. This cycle of budget optimization, exploration, observation, and model refinement continues for T rounds, with the overarching goal of maximizing the total cumulative reward by progressively learning a more accurate representation of the network's dynamics.

6.1. *Online algorithm for overlapping* `MuLaNE`

We present our algorithm in Algorithm 3, which is an adaptation of the CUCB algorithm for the general Combinatorial Multi-arm Bandit (CMAB) framework [7,26] to our setting. Notice that the set of base arms $\mathcal{A}$ is defined using $\mathcal{V}$, the set of node IDs, which is not known before the learning process begins. This is handled by operating on a known size of the node set $|\mathcal{V}|$. At the start, the algorithm creates $|\mathcal{V}|$ placeholders for the node IDs. As new, previously unseen nodes are visited during exploration, they are dynamically assigned to these placeholders. Therefore, the use of $\mathcal{V}$ in the definition of $\mathcal{A}$ is for notational convenience and refers to this dynamic placeholder set.

In Algorithm 3, we maintain an unbiased estimation of the visiting probability $P_{i,u}(b)$, denoted as $\hat{\mu}_{i,u,b}$. Let $T_{i,u,b}$ record the number of times arm (i,u,b) is played so far and $\bar{\sigma}_v$ denotes the optimistic importance weight. In each round $t \geq 1$, we compute the radius $\rho_{i,u,b}$ in line 5, which controls the level of exploration. The confidence radius is larger when the arm (i,u,b) is not explored often (i.e. $T_{i,u,b}$ is small), and thus motivates more exploration. In line 6, we compute the upper confidence bound (UCB) value $\bar{\mu}_{i,u,b}$. Due to the randomness of the exploration process, the upper confidence bound (UCB) value $\tilde{\mu}_{i,u,b}$ could be decreasing w.r.t b, but our offline oracle BEG can only accept non-decreasing UCB values (otherwise the $1-e^{-\eta}$ approximation is not guaranteed). Therefore, in line 8, we increase the UCB value $\tilde{\mu}_{i,u,b}$ and set it to be $\max_{j\in[b]} \tilde{\mu}_{i,u,j}$. This is the adaption of the CUCB algorithm to fix our case, and thus we name our algorithm as CUCB-MAX. Consider two base arms $a_1 = (i,u,b_1)$ and $a_2 = (i,u,b_2)$, $b_1 < b_2$, arm a_2 has both larger true value and larger confidence radius (T_{i,u,b_2} is smaller due to line 13), so it is safe to increase a_2's UCB value if it is smaller than a_1's. After we apply the $1-e^{-\eta}$ approximate solution $\boldsymbol{k}$ given by the oracle (i.e., Algorithm 1), we get m trajectories as the feedback. In line 11, we can update the unknown weights of visited nodes. In line 13, for base arms (i,u,b) with $b \leq k_i$, we update corresponding statistics by the Bernoulli random variable $Y_{i,u,b} \in \{0,1\}$ indicating whether node u is visited by W_i in first b steps.

Regret Bound Analysis. To state the regret bound, we first give some definitions followed by our main result. We define the reward gap $\Delta_{\boldsymbol{k}}$ as $\max(0, \xi r_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\boldsymbol{k}^*) - r_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\boldsymbol{k}))$ for all feasible action $\boldsymbol{k}$ satisfying $\sum_{i=1}^m k_i = B$, $0 \leq k_i \leq c_i$, where $\boldsymbol{k}^*$ is the optimal solution for parameters $\boldsymbol{\mu}$, $\boldsymbol{\sigma}$ and $\xi = 1-e^{-\eta}$ is the approximation ratio of the $(\xi,1)$-approximate offline oracle. For each base arm (i,u,b), we define $\Delta^{i,u,b}_{\min} = \min_{\Delta_{\boldsymbol{k}}>0, k_i=b} \Delta_{\boldsymbol{k}}$ and $\Delta^{i,u,b}_{\max} = \max_{\Delta_{\boldsymbol{k}}>0, k_i=b} \Delta_{\boldsymbol{k}}$. As a convention, if there is no action $\boldsymbol{k}$ with $k_i = b$ such that $\Delta_{\boldsymbol{k}}>0$, we define $\Delta^{i,u,b}_{\min} = \infty$ and $\Delta^{i,u,b}_{\max} = 0$. Let $\Delta_{\min} = \min_{(i,u,b)\in\mathcal{A}} \Delta^{i,u,b}_{\min}$ and $\Delta_{\max} = \max_{(i,u,b)\in\mathcal{A}} \Delta^{i,u,b}_{\max}$.

Theorem 3. *Algorithm 3 has the following distribution-dependent* $(1-e^{-\eta},1)$ *approximation regret,* $O\left(\sum_{(i,u,b)\in\mathcal{A}} \frac{m|\mathcal{V}|\log T}{\Delta^{i,u,b}_{\min}}\right)$.

Proof. See Section F.2 for details. □

Remark 1. Looking at the above distribution-dependent bound, we have the $O(\log T)$ approximation regret, which is asymptotically tight. Coefficient $m|\mathcal{V}|$ in the leading term corresponds to the number of edges in the complete bipartite coverage graph. Notice that we cannot use the true edge number $\sum_{i\in[m]} |\mathcal{V}_i|$, because the learning algorithm does not know which nodes are contained in each layer, and has to explore all visiting possibilities given by the default complete bipartite graph. For the non-overlapping case, we will remove the coefficient $m|\mathcal{V}|$ to achieve a tighter regret bound.

Remark 2. The $(1-e^{-\eta},1)$-approximate regret arises from the offline oracle BEG used in line 9. This can be improved to $(1-1/e,1)$ regret by employing BEGE, or even to the exact regret if the oracle computes the exact optimal budget allocation. The usage of BEG is a trade-off we make between computational efficiency and learning efficiency, and empirically, it performs well as we shall see in Section 7.

Algorithm 3 CUCB-MAX algorithm for the MuLaNE.

Input: Budget B, number of layers m, number of nodes $|\mathcal{V}|$, constraints $\boldsymbol{c}$, offline oracle BEG.
1: For each arm $(i,u,b) \in \mathcal{A}$, $T_{i,u,b} \leftarrow 0$, $\hat{\mu}_{i,u,b} \leftarrow 0$.
2: For each node $v \in \mathcal{V}$, $\bar{\sigma}_v \leftarrow 1$.
3: **for** $t = 1, 2, 3, \ldots, T$ **do**
4: **for** $(i,u,b) \in \mathcal{A}$ **do**
5: $\rho_{i,u,b} \leftarrow \sqrt{3 \ln t / (2T_{i,u,b})}$.
6: $\tilde{\mu}_{i,u,b} \leftarrow \min\{\hat{\mu}_{i,u,b} + \rho_{i,u,b}, 1\}$.
7: **end for**
8: For $(i,u,b) \in \mathcal{A}$, $\bar{\mu}_{i,u,b} \leftarrow \max_{j \in [b]} \tilde{\mu}_{i,u,j}$.
9: $\boldsymbol{k} \leftarrow \mathrm{BEG}((\bar{\mu}_{i,u,b})_{(i,u,b)\in\mathcal{A}}, (\bar{\sigma}_v)_{v\in\mathcal{V}}, \boldsymbol{B}, \boldsymbol{c})$.
10: Apply budget allocation $\boldsymbol{k}$, which gives trajectories $\boldsymbol{X} := (X_{i,1}, \ldots, X_{i,k_i})_{i\in[m]}$ as feedbacks.
11: For any visited node $v \in \bigcup_{i\in[m]} \{X_{i,1}, \ldots, X_{i,k_i}\}$, receive its node weight σ_v and set $\bar{\sigma}_v \leftarrow \sigma_v$.
12: For any $(i,u,b) \in \tau := \{(i,u,b) \in \mathcal{A} \mid b \le k_i\}$,
$Y_{i,u,b} \leftarrow 1$ if $u \in \{X_{i,1}, \ldots, X_{i,b}\}$ and 0 otherwise.
13: For $(i,u,b) \in \tau$, update $T_{i,u,b}$ and $\hat{\mu}_{i,u,b}$:
$T_{i,u,b} \leftarrow T_{i,u,b} + 1$, $\hat{\mu}_{i,u,b} \leftarrow \hat{\mu}_{i,u,b} + (Y_{i,u,b} - \hat{\mu}_{i,u,b})/T_{i,u,b}$.
14: **end for**

Remark 3. The full proof of the above theorem is included in the F, where we rely on the following properties of $r_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\boldsymbol{k})$ to bound the regret. We further consider random node weights with unknown mean vector $\boldsymbol{\sigma}$ (Please refer to F.3).

Property 1. *(Monotonicity). The reward $r_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\boldsymbol{k})$ is monotonically increasing, i.e., for any budget allocation $\boldsymbol{k}$, any two vectors $\boldsymbol{\mu}=(\mu_{i,u,b})_{(i,u,b)\in\mathcal{A}}$, $\boldsymbol{\mu'}=(\mu'_{i,u,b})_{(i,u,b)\in\mathcal{A}}$ and any node weights $\boldsymbol{\sigma}$, $\boldsymbol{\sigma}'$, we have $r_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\boldsymbol{k}) \le r_{\boldsymbol{\mu'},\boldsymbol{\sigma'}}(\boldsymbol{k})$, if $\mu_{i,u,b} \le \mu'_{i,u,b}$, $\sigma_v \le \sigma'_v$, $\forall (i,u,b) \in \mathcal{A}, v \in \mathcal{V}$.*

Property 2. *(1-Norm Bounded Smoothness). The reward function $r_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\boldsymbol{k})$ satisfies the 1-norm bounded smoothness condition, i.e., for any budget allocation $\boldsymbol{k}$, any two vectors $\boldsymbol{\mu}=(\mu_{i,u,b})_{(i,u,b)\in\mathcal{A}}$, $\boldsymbol{\mu'}=(\mu'_{i,u,b})_{(i,u,b)\in\mathcal{A}}$ and any node weights $\boldsymbol{\sigma}$, $\boldsymbol{\sigma}'$, we have $|r_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\boldsymbol{k}) - r_{\boldsymbol{\mu'},\boldsymbol{\sigma'}}(\boldsymbol{k})| \sum_{i\in[m],u\in\mathcal{V},b=k_i} (\sigma_u |\mu_{i,u,b} - \mu'_{i,u,b}| + |\sigma_u - \sigma'_u| \mu'_{i,u,b})$.*

Analysis Technique. We emphasize that our algorithm and analysis differ from the original CUCB algorithm [7] as follows. First, we have the additional regret caused by the over-estimated weights for unvisited nodes, i.e., $|\sigma_u - \sigma'_u| \mu'_{i,u,b}$ term in Property 2. We carefully bound this term on the basis of the observation that $\mu'_{i,u,b}$ is small and rapidly decreases before u is first visited. Next, we have to take the maximum (line 8) to guarantee that the UCB value is monotone w.r.t b since our BEG oracle can only output the $(1 - e^{-\eta}, 1)$ approximation with monotone input. Due to the above operation, the value of the (i,u,b)' UCB depends on the feedback from all arms (i,u,j) for $j \le b$ (set τ in line 12). So we should update all these arms (line 13) to guarantee that the estimates for all these arms are accurate enough. Finally, directly following the standard CMAB result would have a greater regret upper bound, because arms in τ are defined as triggered arms, but only arms in $\tau' = \{(i,u,b) \in \mathcal{A} | k_i = b\}$ affect the rewards. So we conceptually view arms in τ' as triggered arms and use a tighter 1-Norm Bounded Smoothness condition as given above to derive a tighter regret bound. This improves the coefficient of the leading $\ln T$ term in the distribution-dependent regret bound by a factor of $|\tau|/|\tau'| = O(B/m)$, and the $1/\Delta^{i,u,b}_{\min}$ term is smaller since the original definition would have $\Delta^{i,u,b}_{\min} = \min_{\Delta_{\boldsymbol{k}}>0, b\le k_i} \Delta_{\boldsymbol{k}}$.

6.2. Explorer-relaxed online algorithm for overlapping MuLaNE

As shown in Algorithm 4, we now introduce an explorer-relaxed online algorithm called ECUCB-MAX. By "explorer-relaxed", we mean an algorithm whose regret scales sub-linearly in the number of explorers m, rather than linearly as in previous work, making it substantially more scalable for large network layers. The overall structure of Algorithm 4 is similar to Algorithm 3. However, in addition to maintaining the empirical unbiased estimate $\hat{\mu}_{i,u,b}$ for the visiting probability $P_{i,u}(b)$, Algorithm 4 also tracks the empirical variance estimate $\hat{V}_{i,u,b}$ for each tuple (i,u,b). In contrast to Algorithm 3, which constructs the confidence interval $\rho_{i,u,b} = \sqrt{\frac{3\log t}{2T_{i,u,b}}}$ using the Chernoff-Hoeffding inequality, ECUCB-MAX employs the empirical variance $\hat{V}_{i,u,b}$ to form a more refined confidence interval based on the Bernstein inequality:

$$\rho_{i,u,b} = \sqrt{\frac{6\hat{V}_{i,u,b}\log t}{T_{i,u,b}}} + \frac{9\log t}{T_{i,u,b}}. \tag{12}$$

For the first term in Eq. (12), $\hat{V}_{i,u,b}$ is approximately equal to the true variance $V_{i,u,b}$. With $V_{i,u,b} \le (1-\mu_{i,u,b})\mu_{i,u,b}$[4], the estimation of $\mu_{i,u,b}$ becomes more precise when $\mu_{i,u,b}$ approaches 0 or 1, effectively mitigating the influence of $|\mathcal{V}|$, which will be discussed in detail

[4] For a bounded random variable $X \in [0,1]$ with mean μ, variance $V = \mathbb{E}[X^2] - \mathbb{E}[X]^2 \le \mathbb{E}[X] - (\mathbb{E}[X])^2 \le (1-\mu)\mu$, with equality for a Bernoulli random variable.

Algorithm 4 ECUCB-MAX algorithm for the MuLaNE.

Input: Budget B, number of layers m, number of nodes $|\mathcal{V}|$, constraints $\boldsymbol{c}$, offline oracle BEG.

For each arm $(i,u,b) \in \mathcal{A}$, $T_{i,u,b} \leftarrow 0$, $\hat{\mu}_{i,u,b} \leftarrow 0$.
For each node $v \in \mathcal{V}$, $\bar{\sigma}_v \leftarrow 1$.
for $t = 1, 2, 3, \ldots, T$ **do**
  for $(i,u,b) \in \mathcal{A}$ **do**
    $\rho_{i,u,b} \leftarrow \sqrt{\frac{6\hat{V}_{i,u,b}\log t}{T_{i,u,b}}} + \frac{9\log t}{T_{i,u,b}}$.
    $\tilde{\mu}_{i,u,b} \leftarrow \min\{\hat{\mu}_{i,u,b} + \rho_{i,u,b}, 1\}$.
  end for
  For $(i,u,b) \in \mathcal{A}$, $\bar{\mu}_{i,u,b} \leftarrow \max_{j\in[b]} \tilde{\mu}_{i,u,j}$.
  $\boldsymbol{k} \leftarrow \mathrm{BEG}((\bar{\mu}_{i,u,b})_{(i,u,b)\in\mathcal{A}}, (\bar{\sigma}_v)_{v\in\mathcal{V}}, B, \boldsymbol{c})$.
  Apply budget allocation $\boldsymbol{k}$, which gives trajectories $\boldsymbol{X} := (X_{i,1}, \ldots, X_{i,k_i})_{i\in[m]}$ as feedbacks.
  For any visited node $v \in \bigcup_{i\in[m]}\{X_{i,1}, \ldots, X_{i,k_i}\}$, receive its node weight σ_v and set $\bar{\sigma}_v \leftarrow \sigma_v$.
  For any $(i,u,b) \in \tau := \{(i,u,b) \in \mathcal{A} \mid b \le k_i\}$,
  $Y_{i,u,b} \leftarrow 1$ if $u \in \{X_{i,1}, \ldots, X_{i,b}\}$ and 0 otherwise.
  For every $(i,u,b) \in \tau$, update $T_{i,u,b}$, $\hat{V}_{i,u,b}$ and $\hat{\mu}_{i,u,b}$: $T_{i,u,b} \leftarrow T_{i,u,b} + 1$, $\hat{\mu}_{i,u,b} \leftarrow \hat{\mu}_{i,u,b} + (X_{i,u,b} - \hat{\mu}_{i,u,b})/T_{i,u,b}$, $\hat{V}_{i,u,b} \leftarrow \frac{T_{i,u,b}-1}{T_{i,u,b}}\left(\hat{V}_{i,u,b} + \frac{1}{T_{i,u,b}}\left(\hat{\mu}_{i,u,b} - X_{i,u,b}\right)^2\right)$.
end for

later. The second term of Eq. (12) compensates for using the empirical variance $\hat{V}_{i,u,b}$ instead of the unknown true variance $V_{i,u,b}$, which will be shown in G.2.1.

Regret Bound Analysis. Following the definition in Section 6.1, we have the following theorem.

Theorem 4. *Algorithm 4 has the following distribution-dependent* $(1-e^{-\eta}, 1)$ *approximation regret,* $O\Big(\sum_{(i,u,b)\in\mathcal{A}}\big(\frac{|\mathcal{V}|\log(m|\mathcal{V}|)\log T}{\Delta^{i,u,b}_{\min}} + \log\left(\frac{m|\mathcal{V}|}{\Delta^{i,u,b}_{\min}}\right)\log T\big)\Big)$.

Proof. See Section G.2 for details. □

Remark 1. Taking into account the upper bound of regret above, the dependence on the number of explorers m is $\tilde{O}(\log m)$. In comparison to Theorems 3 and 4 provides a strictly better regret bound, with an improvement factor of $O\Big(\frac{m}{\log m|\mathcal{V}|}\Big)$.

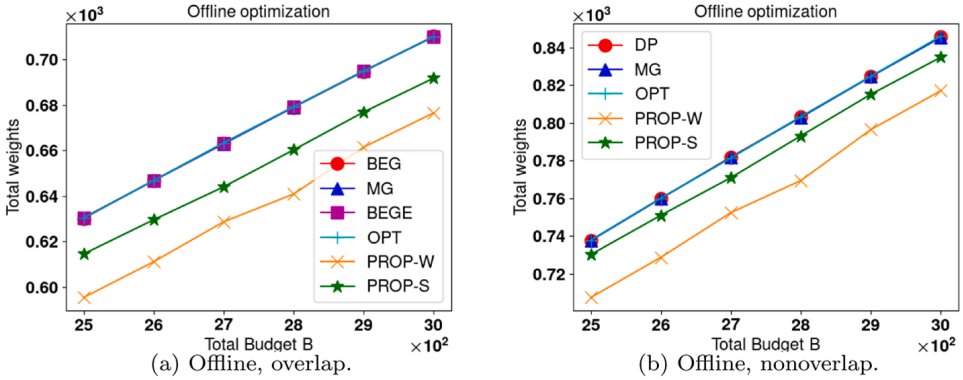
Remark 2. In comparison to Wang and Chen [17], our results match theirs up to a factor of $O(\log m|\mathcal{V}|)$. Regarding the lower bound, based on the findings of Merlis and Mannor [27], our regret bound remains tight within the same factor. Compared to Liu et al. [28], we have removed a factor $O(\log m|\mathcal{V}|)$ from the second main term in the regret upper bound by effectively reducing the influence of observation randomness.

We then introduce a new property discovered for the MuLaNE problem based on Algorithm 4.

Property 3. *(Precision-Balanced Bounded Smoothness). The reward function* $r_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\boldsymbol{k})$ *satisfies the precision-balanced bounded smoothness condition, i.e., for any budget allocation* $\boldsymbol{k}$*, any two vectors* $\boldsymbol{\mu}=(\mu_{i,u,b})_{(i,u,b)\in\mathcal{A}}$, $\boldsymbol{\mu'} = \boldsymbol{\mu} + \boldsymbol{\zeta}_\mu + \boldsymbol{\eta}_\mu = (\mu'_{i,u,b})_{(i,u,b)\in\mathcal{A}}$ *and any node weights* $\boldsymbol{\sigma}$, $\boldsymbol{\sigma'} = \boldsymbol{\sigma} + \boldsymbol{\zeta}_\sigma + \boldsymbol{\eta}_\sigma$, *where* $\boldsymbol{\zeta}, \boldsymbol{\eta} \in [-1,1]^m$ *represent the parameter change, we have the inequality* $|r_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\boldsymbol{k}) - r_{\boldsymbol{\mu'},\boldsymbol{\sigma'}}(\boldsymbol{k})| \le \sqrt{1.25|\mathcal{V}|}\sqrt{\sum_{u\in\mathcal{V}}\big(\sum_{i\in[m],b=k_i}\big(\frac{\zeta^2_{\mu,i,u}}{(1-\mu_{i,u,b})\mu_{i,u,b}}\big) + \frac{\zeta^2_{\sigma,u}}{(1-\sigma_u)\sigma_u}\big)} + \sum_{u\in\mathcal{V}}\big(\sum_{i\in[m],b=k_i}\left|\eta_{\mu,i,u,b}\right| + \left|\eta_{\sigma,u}\right|\big)$.

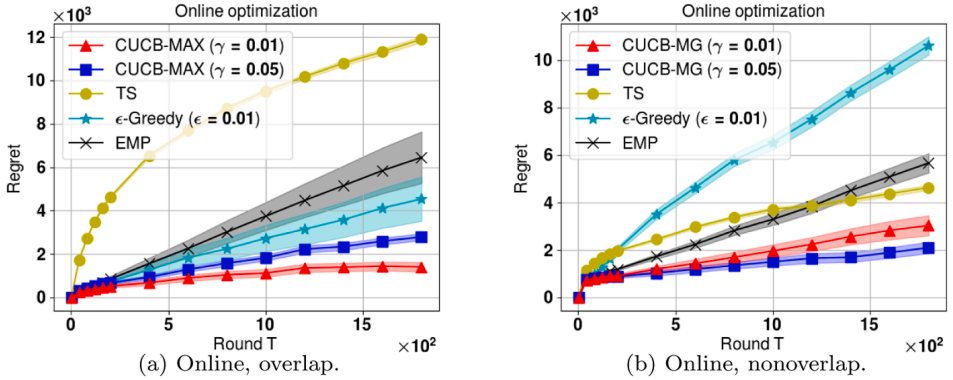
Remark 3. Although the impact of $|\mathcal{V}|$ can increase significantly with abrupt reward changes when $\mu_{i,u,b}$ approaches 0 or 1, introducing the term $1/(1-\mu_{i,u,b})\mu_{i,u,b}$ amplifies the square root term in these cases, thereby mitigating the negative effect. Moreover, as $\mu_{i,u,b}$ nears 0 or 1, the variance $V_{i,u,b}$ diminishes, allowing for "a more precise estimation" of $\mu_{i,u,b}$. This high estimation precision leads to a smaller estimation gap, introducing a variance-related term that offsets the challenging $(1-\mu_{i,u,b})\mu_{i,u,b}$ term in the denominator. This reveals an inherent balance: regions where the reward function is less smooth are precisely where its underlying parameters can be estimated with higher precision. Since this property captures the *balance between estimation precision and the smoothness bound*, we refer to it as "Precision-Balanced Bounded Smoothness".

Analysis Technique. In addition to the analysis technique in Section 6.1, the proof for Theorem 4 is more challenging than that for Theorem 3. Drawing inspiration from Liu et al. [28], our approach filters the total regret through several events and bounds the regret for each filtered event separately. As shown in G.2, the event contributing most to the regret is $E_t = \{\Delta_{\boldsymbol{k}_t} \le e_t(\boldsymbol{k}_t)\}$, where the error term $e_t(\boldsymbol{k}_t) = O(|\mathcal{V}|\sqrt{\sum_{u\in\mathcal{V}}\sum_{i\in[m],b=k_i}\left(\frac{\log t}{T_{i,u,b}}\right)} + \sum_{u\in\mathcal{V}}\sum_{i\in[m],b=k_i}\left(\frac{\log t}{T_{i,u,b}}\right))$. To bound the leading regret filtered by E_t, we use the reverse amortization trick [17,29], and adaptively allocate each arm's regret contribution, carefully selecting thresholds for the error term $e_t(\boldsymbol{k}_t)$ specific to the MuLaNE problem rather than relying on those in Wang and Chen [17], Liu et al. [28]. Additionally, our analysis holds for both the primary model of "fixed but unknown" node weights and the more general stochastic model where weights are "random variables" with an unknown mean (as detailed in G). In both settings, we avoid the triggering probability group technique from Wang and Chen [17], Liu et al. [28] and instead minimize observation uncertainty.



(a) Offline, overlap. (b) Offline, nonoverlap.

Fig. 5. Total weights of unique nodes visited for offline algorithms.



(a) Online, overlap. (b) Online, nonoverlap.

Fig. 6. Cumulative regret for online algorithms when $B = 3000$.

Table 2
Statistics for FF-TW-YT network.

Layer	FriendFeed	Twitter	YouTube
# of vertices	5540	5702	663
# of edges	31,921	42,327	614

6.3. Online algorithm for non-overlapping case

For the non-overlapping case, we set *layer-wise marginal gains* as our base arms. Concretely, we maintain a set of base arms $\mathcal{A} = \{(i,b)|i \in [m], b \in [c_i]\}$. For each base arm $(i,b) \in \mathcal{A}$, let $\mu_{i,b} = \sum_{u \in \mathcal{V}} \sigma_u(P_{i,u}(b) - P_{i,u}(b-1))$ be the true marginal gain of assigning budget b in layer i. We apply the standard CUCB algorithm to this setting and call the resulting algorithm CUCB-MG (Algorithm 8 in H). Note that in the non-overlapping setting, we can solve the offline problem exactly so that we can use the exact offline oracle to solve the online problem and achieve an exact regret bound. This is the major advantage over the overlapping setting where we can only achieve an approximate bound. Define $\Delta_{\boldsymbol{k}} = r_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\boldsymbol{k}^*) - r_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\boldsymbol{k})$ for all feasible action $\boldsymbol{k}$, and $\Delta^{i,b}_{\min} = \min_{\Delta_{\boldsymbol{k}}>0, k_i \geq b} \Delta_{\boldsymbol{k}}$, CUCB-MG has a $O(\sum_{(i,b)\in\mathcal{A}} 48B \ln T / \Delta^{i,b}_{\min})$ regret upper bound.

7. Experiments

Experimental Settings. We firstly conduct experiments on a real-world multi-layered network FF-TW-YT dataset, which contains $m = 3$ layers representing users' social connections in FriendFeed (FF), Twitter (TW) and YouTube (YT) [30]. This multi-layered network is built upon the FF network, and the TW and YT layers are generated by exploiting the feature of FriendFeed as a social media aggregator. In total, FF-TW-YT has $6,407$ distinct vertices representing users and $74,836$ directed edges representing connections ("who follows whom") among users. The statistics for each layer is summarized in Table 2. We transform the FF-TW-YT network $\mathcal{G}(\mathcal{V},\mathcal{E})$ into a symmetric directed network (by adding a new edge (v,u) if $(u,v) \in \mathcal{E}$ but $(v,u) \notin \mathcal{E}$) because a user can be visited via her followers or followees. Each edge weight is set to be 1 and the node weights are set to be $\sigma_u \in \{0, 0.5, 1\}$ uniformly at random. Each random walker always starts from the smallest node-id in each layer and we set constraints c_i equal to the total budget B. Note that in order to test the non-overlapping case, we use the same network but relabel node-ids so that they do not overlap between different layers.

Table 3
Running time (seconds) for offline algorithms with different budgets B.

(a) Offline, overlapping case.			
Algorithm	B = 2.6k	B = 2.8k	B = 3.0k
BEG	0.274	0.316	0.363
BEGE	34.37	45.51	59.20
OPT	91.16	98.42	105.63
(b) Offline, non-overlapping case.			
Algorithm	B = 2.6k	B = 2.8k	B = 3.0k
DP	0.038	0.044	0.050
MG	0.008	0.009	0.010
OPT	70.10	75.78	81.11

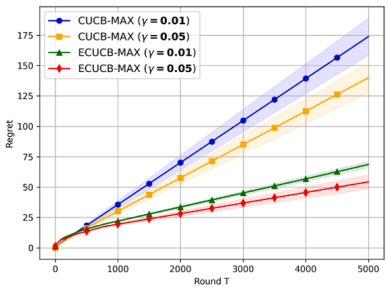


Fig. 7. Comparison of online algorithms with same offline algorithm in the complex overlapping case.

Similarly, we utilize the RocketFuel dataset [31], which represents a real-world computer network. Our analysis focuses on a randomly selected network from this dataset, comprising 300 nodes and 549 edges. We calculate and normalize the centrality of the interconnectivity for each node to represent the edge weight, a metric that indicates the functional importance of the node within the network [32]. A node's capacity is positively correlated with its functional importance; the greater the capacity, the higher the likelihood that the node remains operational. Identifying such nodes is crucial for a random walker aiming to maximize the discovery of important nodes, with m also set to 3. To handle the randomness, we repeat $10,000$ times and present the averaged total weights of the unique nodes visited for offline optimization. We calculate the cumulative regret by comparing with the *optimal* solution, which is stronger than comparing with the $(1 - e^{-\eta}, 1)$-approximate solution as defined in Eq. (4).

Benckmarks. For the offline setting, we present the results for Algorithm 1 (denoted as BEG), Algorithm 2 (denoted as MG), BEG with partial enumeration (denoted as BEGE) and the dynamic programming algorithm in the supplementary material (denoted as DP). We provide two baselines PROP-S and PROP-W, which allocate the budget proportional to the layer size and proportional to the total weights if we allocate $B/3$ budgets to that layer, respectively. The optimal solution (denoted as OPT) is also provided by enumerating all possible budget allocations. For online settings, we consider CUCB-MAX (Algorithm 3), ECUCB-MAX (Algorithm 4) for the overlapping case, and CUCB-MG for the non-overlapping case (Algorithm 8). We shrink the confidence interval by γ, i.e., $\rho_{i,u,b} \leftarrow \gamma \rho_{i,u,b}$, to speed up the learning, though our theoretical regret bound requires $\gamma = 1$. For baselines, we consider the EMP algorithm which always allocates according to the empirical mean, and the ϵ-Greedy algorithm which allocates budgets according to empirical mean with probability $1 - \epsilon$ and allocates all B budgets to the i-th layer with probability ϵ/m. We also compare with the Thompson sampling (TS) method [29], which uses Beta distribution Beta(α, β) (where $\alpha = \beta = 1$ initially) as prior distribution for each base arm.

Experimental Results. We first present the results of the social network dataset for the `MuLaNE` problem under the offline setting in Fig. 5. For the offline overlapping case, both BEG and MG outperform two baselines PROP-W and PROP-S in receiving total weights. Although not guaranteed by the theory, BEG is empirically close to BEGE and the optimal solution (OPT). For computational efficiency, in Table 3a, BEG is at least two orders of magnitude (e.g., 163 times when B = 3.0k) faster than BEGE and OPT. Combining the result that the reward of BEG is empirically close to BEGE and the optimal solution, this shows that BEG is empirically better than BEGE and OPT. For the offline non-overlapping case, the results presented in Table 3b are similar; however, a key distinction is that we provide a theoretical guarantee for the optimality of DP.

For the online setting, as shown in Fig. 6, all of our proposed algorithms consistently outperform the baseline methods. For the online overlapping case, the curve of CUCB-MAX grows sub-linearly w.r.t round T under the social network dataset, because our offline oracle achieves near-optimal solutions. Moreover, the cumulative regret decreases and consistently outperforms baseline algorithms when $\gamma = 0.1, 0.05$. The resulting sub-linear cumulative regret curve demonstrates empirically that the CUCB-MAX algorithm is effectively learning to make near-optimal budget allocation decisions while optimizing the objective. Moreover, as shown in Fig. 7 on the computer network dataset with the offline algorithm, ECUCB-MAX exhibits lower regret compared to CUCB-MAX with the same

number of random walkers, and shows less fluctuation due to the consideration of variance. In the non-overlapping case, the CUCB-MG algorithm outperforms both baseline approaches. Additional experimental results, including variations in budgets, stationary starting distributions, and the computational efficiency of different online learning algorithms, can be found in I.

8. Conclusion

This paper explores the multi-layered networks through random walks (MuLaNE), where the objective is to maximize the total weight of distinct nodes visited across multiple layers. We systematically study the MuLaNE problem in both offline optimization and online learning settings, considering both overlapping and non-overlapping networks. For the offline setting, we propose four algorithms based on the network's structure (overlapping or non-overlapping) and the nature of the starting distributions (arbitrary or stationary). Each algorithm is accompanied by provable guarantees on approximation factors and running time. In the online setting, where the network structure and node weights are unknown, we develop two distinct online learning algorithms for the more complex overlapping MuLaNE, while also addressing the non-overlapping case. All of these algorithms achieve a logarithmic order regret bound on the total rounds. Finally, we validate the empirical performance of our algorithms through experiments on real-world social and computer network datasets.

Several promising directions remain for future work. One interesting extension would be to allow joint optimization of the starting distributions and budget allocations. Additionally, exploring adaptive versions of MuLaNE, where feedback from earlier exploration informs future strategies, presents a compelling avenue for further research.

CRediT authorship contribution statement

Xiangxiang Dai: Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Conceptualization; **Xutong Liu:** Writing – review & editing, Writing – original draft, Validation, Formal analysis, Conceptualization; **Jinhang Zuo:** Investigation, Formal analysis, Conceptualization; **Xiaowei Chen:** Methodology, Conceptualization; **Wei Chen:** Supervision, Methodology, Formal analysis; **John C S Lui:** Writing – review & editing, Supervision, Funding acquisition.

Data availability

This paper primarily makes theoretical contributions. The validation dataset used are open-source datasets, which have been specifically referenced in the paper.

Declaration of competing interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The work of Xiangxiang Dai was supported in part by the National Natural Science Foundation of China (625B2163). The work of John C.S. Lui was supported in part by the RGC GRF-14202923.

Appendix A. Appendix overview

The appendix is organized as follows. We provide proofs and examples for properties of the visiting probability in B. Proofs of offline optimization for overlapping MuLaNE are provided in C. We provide the detailed budget-effective greedy algorithm with partial enumeration (BEGE) and its analysis in D. Proofs of offline optimization for non-overlapping MuLaNE are provided in E. We state the detailed analysis of online learning and explorer-relaxed online learning for overlapping MuLaNE in F and G, respectively. We state the analysis of online learning for non-overlapping MuLaNE in H. Supplemental experiments are provided in I.

Appendix B. Proofs and examples for properties of visiting probability

B.1. Starting from arbitrary distributions

Proof. For analysis, we use the following equivalent formulation for $k_i \geq 2$, (trivially $P_{i,u}(0) = 0$ and $P_{i,u}(1) = \alpha_{i,u}$),

$$P_{i,u}(k_i) = ((\boldsymbol{\alpha_i})^{\top}_{-u}(\boldsymbol{\Gamma_i})_{uk_i} + \alpha_{i,u}), \tag{B.1}$$

where $(\boldsymbol{\Gamma_i})_{uk_i} = (\boldsymbol{I} + \boldsymbol{T_i(u)} + \ldots + \boldsymbol{T_i(u)}^{k_i-2})(\boldsymbol{p_i})_u$, $(\boldsymbol{p_i})_u = (\boldsymbol{P_i}[\cdot, u])_{-u}$, $\boldsymbol{T_i(u)} = (\boldsymbol{P_i})_{-u,-u}$, (we use Eq. (2.67) in Kijima [33] to derive $\boldsymbol{\Gamma}_{uk_i}$). Note that $\boldsymbol{p}_{-u} \in \mathbb{R}^{n-1}$ is the vector obtained by deleting the u-th element from $\boldsymbol{p} \in \mathbb{R}^n$, and $\boldsymbol{P}_{-u,-v} \in \mathbb{R}^{(n-1)\times(n-1)}$ is the matrix obtained by deleting the u-th row and the v-th column from $\boldsymbol{P} \in \mathbb{R}^{n\times n}$.

We further derive the marginal gain of $P_{i,u}(\cdot)$ at step $k_i \geq 2$ (trivially $g_{i,u}(1) = \alpha_{i,u}$) as,

$$g_{i,u}(k_i) = (\boldsymbol{\alpha_i})_{-u}^{\top} \boldsymbol{T_i(u)}^{k_i-2} (\boldsymbol{p_i})_u. \tag{B.2}$$

Now, we can show $g_{i,u}(k_i)$ is non-negative because any element of $(\boldsymbol{\alpha_i})_{-u}^{\top}$, $\boldsymbol{T_i(u)}$ and $(\boldsymbol{p_i})_u$ are non-negative, which means $P_{i,u}(k_i)$ is non-decreasing with respect to step k_i. □

An example showing $g_{i,u}(k_i)$ is not monotone: Consider a path P_3 with three nodes as the i-th layer $\mathcal{G}_i(\mathcal{V}_i, \mathcal{E}_i)$, where $\mathcal{V}_i = \{u, v, w\}$ and $\mathcal{E}_i = \{(u, v), (v, u), (v, w), (w, v)\}$. If the W_i always starts from the left-most node u and chooses the right-most node w as our target node. Then, $g_{i,w}(1) = g_{i,w}(2) = 0$ but $g_{i,w}(3) > 0$ since at least three steps are needed to visit the node w, which shows that $g_{i,w}(k_i)$ is not always non-increasing.

B.2. Starting from the stationary distribution

Proof. Consider any layer L_i with transition probability matrix $\boldsymbol{P_i}$, if we start from the stationary distribution $\boldsymbol{\pi_i}$, the probability that node $u \in \mathcal{V}_i$ is ever visited in the first k_i steps is $P_{i,u}(k_i) = (\boldsymbol{\pi_i})_{-u}^{\top} (\boldsymbol{\Gamma_i})_{uk_i} + \pi_{i,u}$, where $(\boldsymbol{\Gamma_i})_{uk_i} = (\boldsymbol{I} + \boldsymbol{T_i(u)} + \ldots + \boldsymbol{T_i(u)}^{k-2})(\boldsymbol{p_i})_u$, $\boldsymbol{T_i(u)} = (\boldsymbol{P_i})_{-u,-u}$, $(\boldsymbol{p_i})_u = (\boldsymbol{P_i}[\cdot, u])_{-u}$. Then the marginal gain for node u is $g_{i,u}(k_i) = P_{i,u}(k_i) - P_{i,u}(k_i - 1) = (\boldsymbol{\pi_i})_{-u}^{\top} \boldsymbol{T_i(u)}^{k_i-2} (\boldsymbol{p_i})_u$ for $k_i \geq 2$ and $g_{i,u}(k_i) = \pi_{i,u}$ when $k_i = 1$.

Define the margin of the marginal gain as $\Delta_{i,u}(k_i) = g_{i,u}(k_i + 1) - g_{i,u}(k_i)$. When $k = 1$, $\Delta(u, 1) = g_{i,u}(2) - g_{i,u}(1) = (\boldsymbol{\pi_i})_{-u}^{\top} \cdot (\boldsymbol{p_i})_u - \pi_{i,u} = \pi_{i,u} - \pi_{i,u}\boldsymbol{P}[u, u] - \pi_{i,u} = -\pi_{i,u}\boldsymbol{P}[u, u] \leq 0$. When $k_i \geq 2$, $\Delta_{i,u}(k_i) = g_{i,u}(k_i + 1) - g_{i,u}(k_i) = (\boldsymbol{\pi_i})_{-u}(\boldsymbol{T_i(u)}^{k_i-1} - \boldsymbol{T_i(u)}^{k_i-2})(\boldsymbol{p_i})_u$. Because $\boldsymbol{\pi_i}^{\top} \boldsymbol{P_i} = \boldsymbol{\pi_i}^{\top}$, we have $(\boldsymbol{\pi_i})_{-u}^{\top} - (\boldsymbol{\pi_i})_{-u}^{\top} \boldsymbol{P_i}(u) = \pi_{i,u}(\boldsymbol{q_i})_u^{\top}$, where $(\boldsymbol{q_i})_u^{\top} = (\boldsymbol{P_i}[u, \cdot])_{-u}$. Thus, $\Delta_{i,u}(k_i) = -(\pi_{i,u})(\boldsymbol{q_i})_u^{\top} \boldsymbol{T_i(u)}^{k_i-2} (\boldsymbol{p_i})_u \leq 0$ because any element in $\boldsymbol{\pi_i}$, $(\boldsymbol{q_i})_u^{\top}$, $(\boldsymbol{p_i})_u$ and $\boldsymbol{P_i(u)}$ is non-negative. □

More interestingly, the stationary distribution $\boldsymbol{\pi_i}$ is the *only* starting distribution for $\boldsymbol{P_i}$ such that any $u \in \mathcal{V}_i, k_i \in \mathbb{Z}_{>0}$, $g_{i,u}(k_i + 1) - g_{i,u}(k_i) \leq 0$ when $\boldsymbol{P_i}$ is ergodic and there are no self loops in $\mathcal{G}_i$.

Proof. With a little abuse of the notation, we use $\boldsymbol{P}$ to denote the transition probability matrix $\boldsymbol{P_i} \in \mathbb{R}^{n \times n}$ and let $\boldsymbol{P}_{i,j}$ be the element in the i-th row and the j-th column. Since $\boldsymbol{P}$ is ergodic, according to Theorem 54 in Serfozo [34], there exists a unique and positive stationary distribution $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_n)$, i.e., $\boldsymbol{\pi P} = \boldsymbol{\pi}$ and $\boldsymbol{\pi} > \boldsymbol{0}$. Any starting distribution $\boldsymbol{\alpha}$ can be represented by $\boldsymbol{\pi} + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_n)$ is a perturbation vector, and $-\pi_j \leq \epsilon_j \leq 1 - \pi_j, j \in [n]$. We have the following equation for the margin of marginal gains for node u in the first two steps,

$$\begin{aligned}
\Delta_u = g_{i,u}(1) - g_{i,u}(2) &= (\pi_u + \epsilon_u) - \sum_{j \neq u} (\pi_j + \epsilon_j) \boldsymbol{P}_{j,u} \\
&= \pi_u \boldsymbol{P}_{uu} + \epsilon_u - \sum_{j \neq u} \epsilon_j \boldsymbol{P}_{j,u}.
\end{aligned}$$

Since $\mathcal{G}$ has no self loops, i.e., $\boldsymbol{P}_{u,u} = 0$, we have $\Delta_u = \epsilon_u - \sum_{j \neq u} \epsilon_i \boldsymbol{P}_{j,u}$, and we can verify that $\sum_{u \in [n]} \Delta_u = \sum_{u \in [n]} (1 - \sum_{j \neq u} \boldsymbol{P}_{u,j}) \epsilon_u = 0$. Hence, we have to guarantee $\Delta_u = 0$ for all u, otherwise there will exist a node u such that $\Delta_u < 0$. To ensure $\Delta_u = 0$, we need to ensure $\boldsymbol{\epsilon P} = \boldsymbol{\epsilon}$ by rephrasing the equations $\Delta_u = 0$ for all u. Again, since there exists a unique and positive stationary distribution for $\boldsymbol{P}$ and $\sum_{u \in [n]} \epsilon_u = 0$, we can derive $\boldsymbol{\epsilon} = \beta \boldsymbol{\pi}$, where β has to be 0. Therefore, combined with Lemma 2, the stationary distribution is the only starting distribution such that for any $\boldsymbol{P_i}$, any $u \in \mathcal{V}_i, k_i \in \mathbb{Z}_{>0}$, $g_{i,u}(k_i + 1) - g_{i,u}(k_i) \leq 0$. □

Appendix C. Proofs of Offline Optimization for Overlapping `MuLaNE`

C.1. Starting from the arbitrary distribution

Proof. By definition, we need to show $r_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(\boldsymbol{x} \wedge \boldsymbol{y}) + r_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(\boldsymbol{x} \vee \boldsymbol{y}) \leq r_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(\boldsymbol{x}) + r_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(\boldsymbol{y})$ for any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{Z}_{\geq 0}^m$, and $r_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(\boldsymbol{x}) \leq r_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(\boldsymbol{y})$ if $\boldsymbol{x} \leq \boldsymbol{y}$.

(**Monotonicity.**) By Eq. (7), $r_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(\boldsymbol{k}) = \sum_{v \in \mathcal{V}} \sigma_v (1 - \prod_{i \in [m]} (1 - P_{i,v}(k_i)))$. Since $P_{j,v}(x_j) \leq P_{j,v}(x_j + 1)$, for any $j \in [m], v \in \mathcal{V}$, $x_j \in \mathbb{Z}_{\geq 0}$, we have

$$\begin{aligned}
&r_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(\boldsymbol{x} + \chi_j) - r_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(\boldsymbol{x}) \\
&= \sum_{v \in \mathcal{V}} \sigma_v \Big(\big(\prod_{i \neq j} \big(1 - P_{i,v}(x_i)\big) \big) \big(P_{j,v}(x_j + 1) - P_{j,v}(k_j) \big) \Big) \\
&\geq 0,
\end{aligned}$$

for any $\boldsymbol{x} \in \mathbb{Z}_{\geq 0}^m, j \in [m]$. Then we can use the above inequality repeatedly to show that $r_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(\boldsymbol{x}) \leq r_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(\boldsymbol{y})$ when $\boldsymbol{x} \leq \boldsymbol{y}$.

(**Submodularity.**) For submodular property, it suffices to prove that $(1 - \prod_{i \in [m]} (1 - P_{i,v}(k_i)))$ is submodular for any $v \in \mathcal{V}$, because a positive weighted sum of submodular function is still submodular. We will rely on the following lemma to prove the submodularity of $r_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(\boldsymbol{k})$.

Lemma 5. *Function $f : \mathbb{Z}^m_{\geq 0} \to \mathbb{R}$ is submodular if and only if*

$$f(\boldsymbol{x}+\chi_i)-f(\boldsymbol{x}) \geq f(\boldsymbol{x}+\chi_j+\chi_i)-f(\boldsymbol{x}+\chi_j), \quad \text{for any } \boldsymbol{x} \in \mathbb{Z}^m_{\geq 0} \text{ and } i \neq j. \tag{C.1}$$

Proof.

(If part.) We first prove, if Eq. (C.1) holds, the following inequality holds,

$$f(\boldsymbol{x}+\chi_i)-f(\boldsymbol{x}) \geq f(\boldsymbol{y}+\chi_i)-f(\boldsymbol{y}) \tag{C.2}$$

for any $\boldsymbol{x} \leq \boldsymbol{y}$ and $i \in [m]$ such that $x_i = y_i$.

Let $I_0 = \{i \in [m] : x_i = y_i\}, I_1 = \{i \in [m] : x_i < y_i\}$. For any $\boldsymbol{x} \leq \boldsymbol{y}$, we denote the elements in I_1 by $i_1, \ldots, i_s$ and write $\boldsymbol{y} = \boldsymbol{x} + \sum_{j=1}^{s} \alpha_j \chi_{i_j}$, where $\alpha_j = y_j - x_j$. For any $i \in I_0$, we have

$$\begin{aligned}
&f(\boldsymbol{x}+\chi_i)-f(\boldsymbol{x})\\
&\geq f(\boldsymbol{x}+\chi_{i_1}+\chi_i)-f(\boldsymbol{x}+\chi_{i_1})\\
&\geq f(\boldsymbol{x}+2\chi_{i_1}+\chi_i)-f(\boldsymbol{x}+2\chi_{i_1})\\
&\geq \cdots\\
&\geq f(\boldsymbol{x}+\alpha_{i_1}\chi_{i_1}+\chi_i)-f(\boldsymbol{x}+\alpha_{i_1}\chi_{i_1})\\
&\geq f(\boldsymbol{x}+\alpha_{i_1}\chi_{i_1}+\chi_{i_2}+\chi_i)-f(\boldsymbol{x}+\alpha_{i_1}\chi_{i_1}+\chi_{i_2})\\
&\geq f(\boldsymbol{x}+\alpha_{i_1}\chi_{i_1}+\alpha_{i_2}\chi_{i_2}+\chi_i)-f(\boldsymbol{x}+\alpha_{i_1}\chi_{i_1}+\alpha_{i_2}\chi_{i_2})\\
&\geq \cdots\\
&\geq f(\boldsymbol{x}+\sum_{j=1}^{s}\alpha_j\chi_{i_j}+\chi_i)-f(\boldsymbol{x}+\sum_{j=1}^{s}\alpha_j\chi_{i_j})\\
&= f(\boldsymbol{y}+\chi_i)-f(\boldsymbol{y}).
\end{aligned} \tag{C.3}$$

Then for any $i \in I_0$ and $a \in \mathbb{Z}_{\geq 0}$, we have

$$\begin{aligned}
&f(\boldsymbol{x}+a\chi_i)-f(\boldsymbol{x})\\
&=\sum_{j=1}^{a}\left(f(\boldsymbol{x}+j\chi_i)-f(\boldsymbol{x}+(j-1)\chi_i)\right)\\
&\geq \sum_{j=1}^{a}\left(f(\boldsymbol{y}+j\chi_i)-f(\boldsymbol{y}+(j-1)\chi_i)\right)\\
&\geq f(\boldsymbol{y}+a\chi_i)-f(\boldsymbol{y}).
\end{aligned} \tag{C.4}$$

The first inequality holds because of Eq. (C.3), the fact $\boldsymbol{x}+(j-1)\chi_i \leq \boldsymbol{y}+(j-1)\chi_i$ and $x_i + j - 1 = y_i + j - 1$.

Then for any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{Z}^m_{\geq 0}$, let $I_2 = \{i \in [m] : x_i > y_i\} = \{i_1, \ldots, i_s\}$. We have

$$\begin{aligned}
&f(\boldsymbol{x})-f(\boldsymbol{x}\wedge\boldsymbol{y})\\
&=\sum_{l=1}^{s}\left(f\left(\boldsymbol{x}\wedge\boldsymbol{y}+\sum_{j=1}^{l}(x_{i_j}-y_{i_j})\chi_{i_j}\right)\right.\\
&\left.-f\left(\boldsymbol{x}\wedge\boldsymbol{y}+\sum_{j=1}^{l-1}(x_{i_j}-y_{i_j})\chi_{i_j}\right)\right)\\
&\geq\sum_{l=1}^{s}\left(f\left(\boldsymbol{y}+\sum_{j=1}^{l}(x_{i_j}-y_{i_j})\chi_{i_j}\right)\right.\\
&\left.-f\left(\boldsymbol{y}+\sum_{j=1}^{l-1}(x_{i_j}-y_{i_j})\chi_{i_j}\right)\right)\\
&=f(\boldsymbol{x}\vee\boldsymbol{y})-f(\boldsymbol{y}).
\end{aligned}$$

The inequality is derived from Eq. (C.4) because $\boldsymbol{y}+\sum_{j=1}^{l-1}(x_{i_j}-y_{i_j})\chi_{i_j} \geq \boldsymbol{x}\wedge\boldsymbol{y}+\sum_{j=1}^{l-1}(x_{i_j}-y_{i_j})\chi_{i_j}$ for any $0 \leq l \leq s$ and $(\boldsymbol{x}\wedge\boldsymbol{y})_{i_l} = \boldsymbol{y}_{il}$, which concludes the if part.

(Only if part.) Assume f is submodular, let $\boldsymbol{a} = \boldsymbol{x}+\chi_i, (\subsetneq)\boldsymbol{x}+\chi_j, i \neq j$, we have $\boldsymbol{a} \vee (\subsetneq)\boldsymbol{x}+\chi_i+\chi_j$, $\boldsymbol{a} \wedge (\subsetneq)\boldsymbol{x}$. $f(\boldsymbol{x}+\chi_i)-f(\boldsymbol{x}) = f(\boldsymbol{a})-f(\boldsymbol{a}\wedge(\subsetneq) \geq f(\boldsymbol{a}\vee(\subsetneq)-f((\subsetneq) = f(\boldsymbol{x}+\chi_j+\chi_i)-f(\boldsymbol{x}+\chi_j)$. □

Then, by Lemma 5 and the explicit formula of the reward function given by Eq. (7), we can prove $g(\boldsymbol{x}, v)-g(\boldsymbol{x}+\chi_l, v) \geq g(\boldsymbol{x}+\chi_j, v)-g(\boldsymbol{x}+\chi_l+\chi_j, v)$ for any $\boldsymbol{x} \in \mathbb{Z}^m_{\geq 0}$, $v \in \mathcal{V}$ and $l \neq j \in [m]$, where $g(\boldsymbol{x}, v) = \prod_{i\in[m]}(1-P_{i,v}(x_i))$. This holds due to the

fact that the left hand side equals to $\left(\prod_{i\in[m]\setminus\{j,l\}}(1-P_{i,v}(x_i))\right)(1-P_{j,v}(k_j))(P_{l,v}(k_l+1)-P_{l,v}(k_l))$ and the right hand side equals to $\left(\prod_{i\in[m]\setminus\{j,l\}}(1-P_{i,v}(x_i))\right)(1-P_{j,v}(k_j+1))(P_{l,v}(k_l+1)-P_{l,v}(k_l))$, and the left hand side is larger or equal to the right hand side because $(1-P_{j,v}(k_j))\geq(1-P_{j,v}(k_j+1))$. By summation over all nodes $v\in\mathcal{V}$ with node weights $\sigma_v\in[0,1]$, we can prove the reward function is submodular. □

Proof. For theoretical analysis, we first give a modified version of Algorithm 1, which both provide the same solution $\boldsymbol{k}$ given the same problem instance $(\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma},B,\boldsymbol{c})$. To see this fact, the modified Algorithm 1 considers invalid tentative allocations (i^*,b^*) (adding it will exceed the total budget constraint B) and remove them if $b^*>K$, while Algorithm 1 only considers valid allocations by directly removing invalid allocations in advance in line 10 of Algorithm 1.

With little abuse of the notation, we use $r(\boldsymbol{k})$ to represent the reward $r_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(\boldsymbol{k})$ of a given problem instance. Let $\boldsymbol{k}^*\in\mathbb{Z}^m_{\geq 0}$ denote the optimal budget allocation and $\boldsymbol{k}^j\in\mathbb{Z}^m_{\geq 0}$ denote the budget allocation before entering the j-th iteration of the while loop in the modified Algorithm 1. After entering the j-th iteration, the algorithm tries to extend the current budget allocation $\boldsymbol{k}^j$ by choosing the pair (i^*,b^*), which we denote as (i^j,b^j). Let s be the first iteration we can not extend the current solution, i.e., $\boldsymbol{k}^s=\boldsymbol{k}^{s+1}$ and $\boldsymbol{k}^j<\boldsymbol{k}^{j+1}$ for $j=1,\dots,s-1$. If we can always extend the current solution, we set s to be $B+1$. For analysis, we temporarily add (i^s,b^s) (which is removed later on in the s-th iteration) to form a "virtual" budget allocation $\boldsymbol{k}^{s+1}=\boldsymbol{k}^s+b^s\boldsymbol{\chi}_{i^s}$. Let $\boldsymbol{k}_g\in\mathbb{Z}^m_{\geq 0}$ denote the solution returned by the modified Algorithm 1.

We first introduce lemmas describing two important properties given by the submodularity over the integer lattice. □

Lemma 6. *Soma* et al. *[13]. Let $f:\mathbb{Z}^m_{\geq 0}\to\mathbb{R}$ be a submodular function. For any $\boldsymbol{x},\boldsymbol{y}\in\mathbb{Z}^m_{\geq 0}$, we have,*

$$\begin{aligned} &f(\boldsymbol{x}\vee\boldsymbol{y})\\ &\leq f(\boldsymbol{x})+\sum_{i\in supp^+(\boldsymbol{y}-\boldsymbol{x})}\left(f\left(\boldsymbol{x}+(y_i-x_i)\boldsymbol{\chi}_i\right)-f(\boldsymbol{x})\right). \end{aligned} \tag{C.5}$$

Lemma 7. *Soma* et al. *[13]. Let $f:\mathbb{Z}^m_{\geq 0}\to\mathbb{R}$ be a monotone submodular function. For any $\boldsymbol{x},\boldsymbol{y}\in\mathbb{Z}^m_{\geq 0}$ with $\boldsymbol{x}\leq\boldsymbol{y}$ and $i\in[m]$ we have,*

$$f(\boldsymbol{x}\vee k\boldsymbol{\chi}_i)-f(\boldsymbol{x})\geq f(\boldsymbol{y}\vee k\boldsymbol{\chi}_i)-f(\boldsymbol{y}). \tag{C.6}$$

Then, we have the following lemma.

Lemma 8. *For $j=1,\dots,s$,*

$$r(\boldsymbol{k}^{j+1})\geq(1-\frac{b^j}{B})r(\boldsymbol{k}^j)+\frac{b^j}{B}r(\boldsymbol{k}^*). \tag{C.7}$$

Proof. This is because

$$\begin{aligned} &r(\boldsymbol{k}^*)\\ &\leq r(\boldsymbol{k}^*\vee\boldsymbol{k}^j)\\ &\leq r(\boldsymbol{k}^j)+\sum_{i\in\text{supp}^+(\boldsymbol{k}^*-\boldsymbol{k}^j)}\left(r(\boldsymbol{k}^j+(k_i^*-k_i^j)\boldsymbol{\chi}_i)-r(\boldsymbol{k}^j)\right)\\ &=r(\boldsymbol{k}^j)+\sum_{i\in\text{supp}^+(\boldsymbol{k}^*-\boldsymbol{k}^j)}\left(r(\boldsymbol{k}^j+\alpha_i\boldsymbol{\chi}_i)-r(\boldsymbol{k}^j)\right)\quad(\text{Let }\alpha_i=k_i^*-k_i^j)\\ &\leq r(\boldsymbol{k}^j)+\sum_{i\in\text{supp}^+(\boldsymbol{k}^*-\boldsymbol{k}^j)}\left(\alpha_i\frac{r(\boldsymbol{k}^{j+1})-r(\boldsymbol{k}^j)}{b^j}\right)\\ &\leq r(\boldsymbol{k}^j)+B\left(\frac{r(\boldsymbol{k}^{j+1})-r(\boldsymbol{k}^j)}{b^j}\right). \end{aligned} \tag{C.8}$$

The second inequality comes from Lemma 6, the third inequality holds because of the greedy procedure, and the last inequality holds because $\sum_{i\in\text{supp}^+(\boldsymbol{k}^*-\boldsymbol{k}^j)}\alpha_i\leq B$. By rearranging terms, Eq. (C.7) holds. □

Next, We can prove the following lemma.

Lemma 9. *For $l=1,\dots,s$,*

$$\begin{aligned} r(\boldsymbol{k}^{l+1})\geq\ & r(\boldsymbol{k}^1)\prod_{j=1}^{l}\left(1-\frac{b^j}{B}\right)\\ &+r(\boldsymbol{k}^*)\left(1-\prod_{j=1}^{l}\left(1-\frac{b^j}{B}\right)\right). \end{aligned} \tag{C.9}$$

Proof. We can prove this lemma by induction on l. When $l=1$, the lemma holds due to Lemma 8. Assume that the lemma holds for $l-1$, we have the following inequality holds,

$$r(\boldsymbol{k}^{l+1})$$

$$
\begin{aligned}
&\geq (1-\frac{b^l}{B})r(\boldsymbol{k}^l)+\frac{b^l}{B}r(\boldsymbol{k}^*)\\
&\geq (1-\frac{b^l}{B})r(\boldsymbol{k}^1)\prod_{j=1}^{l-1}\left(1-\frac{b^j}{B}\right)\\
&+(1-\frac{b^l}{B})r(\boldsymbol{k}^*)\left(1-\prod_{j=1}^{l-1}\left(1-\frac{b^j}{B}\right)\right)+\frac{b^l}{B}r(\boldsymbol{k}^*)\\
&=r(\boldsymbol{k}^1)\prod_{j=1}^{l}\left(1-\frac{b^j}{B}\right)+r(\boldsymbol{k}^*)\left(1-\prod_{j=1}^{l}\left(1-\frac{b^j}{B}\right)\right),
\end{aligned}
\tag{C.10}
$$

where the first inequality is due to Lemma 8 by setting $j=l$, and the second inequality is by the assumption for $l-1$. By induction, Lemma 9 holds.

We then consider the following cases.

Case 1. Suppose the total budget used for $\boldsymbol{k}^s$ is larger or equal to ηB, i.e., $\sum_{j=1}^{s-1} b^j \geq \eta B$, where $\eta \in [0,1]$.

We have the following inequality.

$$
r(\boldsymbol{k}^s) \geq (1-e^{-\eta})r(\boldsymbol{k}^*). \tag{C.11}
$$

This is due to Lemma 9 by setting $l=s-1$, combined with the fact that $\boldsymbol{k}^1=\boldsymbol{0}$ and $\prod_{j=1}^{s-1}\left(1-\frac{b^j}{B}\right)\leq e^{-\eta}$. The later fact holds because,

$$
\begin{aligned}
\log\left(\prod_{j=1}^{s-1}\left(1-\frac{b^j}{B}\right)\right)&=(s-1)\sum_{j=1}^{s-1}\frac{1}{s-1}\log\left(1-\frac{b^j}{B}\right)\\
&\leq (s-1)\log\left(1-\frac{1}{s-1}\sum_{j=1}^{s-1}\frac{b^j}{B}\right)\\
&\leq (s-1)\log(1-\frac{\eta}{s-1}),
\end{aligned}
$$

where the first inequality holds because of the Jensen's Inequality [35] and the second inequality holds because $\sum_{j=1}^{s-1}\frac{b^j}{B}\geq\eta$. Then we can easily check $\prod_{j=1}^{s-1}\left(1-\frac{b^j}{B}\right)\leq(1-\frac{\eta}{s-1})^{s-1}\leq e^{-\eta}$.

Case 2. Suppose the total budget $\sum_{j=1}^{s-1} b^j \leq \eta B$. Then, we have $b^s > (1-\eta)B$. We can prove the following inequality holds,

$$
r(\boldsymbol{k}_g)\geq(1-\frac{1}{2-\eta})r(\boldsymbol{k}^*). \tag{C.12}
$$

This is due to Eq. (C.8), we have

$$
\begin{aligned}
r(\boldsymbol{k}^*)&\leq r(\boldsymbol{k}^s)+B\left(\frac{r(\boldsymbol{k}^{s+1})-r(\boldsymbol{k}^s)}{b^s}\right)\\
&\leq r(\boldsymbol{k}^s)+\left(\frac{r(\boldsymbol{k}^{s+1})-r(\boldsymbol{k}^s)}{1-\eta}\right)\\
&\leq (1+\frac{1}{1-\eta})r(\boldsymbol{k}_g),
\end{aligned}
$$

where the second inequality is due to $b^s>(1-\eta)B$ and the last equality is due to the fact $r(\boldsymbol{k}^{s+1})-r(\boldsymbol{k}^s)\leq r(\boldsymbol{k}_g)$, $r(\boldsymbol{k}^s)\leq r(\boldsymbol{k}_g)$. To see the above fact, we assume without loss of generality the pair (i^s,b^s) improves $\boldsymbol{k}^s$ towards the optimal budget allocation, i.e., $k^{s+1}_{i_s}=b^s+k^s_{i^s}\leq k^*_{i^s}$. Otherwise, if $b^s+k^s_{i^s}>k^*_{i^s}$, we can safely delete (i^s,b^s) in the queue Q and does not affect the greedy solution, the optimal solution and the analysis. Let $r_{\max}=\max_{i\in[m]} r(c_i\chi_i)$, we have $r(\boldsymbol{k}^{s+1})-r(\boldsymbol{k}^s)\leq r(\boldsymbol{k}^s\vee k^*_{i^s}\chi_{i^s})-r(\boldsymbol{k}^s)\leq r(k^*_{i^s}\chi_{i^s})-r(\boldsymbol{0})\leq r_{\max}\leq r(\boldsymbol{k}_g)$. Also, we can obtain that $r(\boldsymbol{k}^s)\leq r(\boldsymbol{k}_g)$ since $\boldsymbol{k}^s\leq\boldsymbol{k}_g$. By rearranging the terms, Eq. (C.12) holds.

Combining Eqs. (C.11) and (C.12), we have

$$
\begin{aligned}
r(\boldsymbol{k}_g)&\geq \min_{\eta\in[0,1]}\max\,(1-e^{-\eta},1-\frac{1}{2-\eta})r(\boldsymbol{k}^*)\\
&\geq (1-e^{-\eta})r(\boldsymbol{k}^*),
\end{aligned}
$$

where η is the solution for equation $e^{\eta}=2-\eta$. □

C.2. Starting from the stationary distribution

Proof. By definition, we need to show $r_{\mathcal{G},\boldsymbol{\pi},\sigma}(\boldsymbol{y}+\chi_j)-r_{\mathcal{G},\boldsymbol{\pi},\sigma}(\boldsymbol{y})\leq r_{\mathcal{G},\boldsymbol{\pi},\sigma}(\boldsymbol{x}+\chi_j)-r_{\mathcal{G},\boldsymbol{\pi},\sigma}(\boldsymbol{x})$, and $r_{\mathcal{G},\boldsymbol{\pi},\sigma}(\boldsymbol{x})\leq r_{\mathcal{G},\boldsymbol{\pi},\sigma}(\boldsymbol{y})$ for any $\boldsymbol{x}\leq\boldsymbol{y}$, $j\in[m]$.

We first introduce the following lemma to help us to prove the DR-submodularity. □

Lemma 10. *Function f is DR-submodular if and only if*

$$f(\boldsymbol{x}+\chi_i)-f(\boldsymbol{x}) \geq f(\boldsymbol{x}+\chi_j+\chi_i)-f(\boldsymbol{x}+\chi_j), \quad \text{for any } \boldsymbol{x} \in \mathbb{Z}^m_{\geq 0}. \tag{C.13}$$

(If part.) We can easily check this direction holds by using the similar argument for Eq. (C.3), where the only difference is we consider $i \in [m]$ instead of $i \in I_0$.

(Only if part.) We can set $\boldsymbol{y} := \boldsymbol{x}' + \chi_j$, $\boldsymbol{x} := \boldsymbol{x}' + \chi_i$, for any $i, j \in [m], \boldsymbol{x}' \in \mathbb{Z}^m_{\geq 0}$, and use Eq. (C.2) to show the only if part holds.

Since $r_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(\boldsymbol{k})$ is submodular for arbitrary starting distributions $\boldsymbol{\alpha}$, Eq. (C.1) holds. Then consider any layer $j \in [m]$ and budget allocation $\boldsymbol{x} \in \mathbb{Z}^m_{\geq 0}$, it is sufficient to show $r_{\mathcal{G},\boldsymbol{\pi},\boldsymbol{\sigma}}(\boldsymbol{x}+2\chi_j) - r_{\mathcal{G},\boldsymbol{\pi},\boldsymbol{\sigma}}(\boldsymbol{x}+\chi_j) \leq r_{\mathcal{G},\boldsymbol{\pi},\boldsymbol{\sigma}}(\boldsymbol{x}+\chi_j) - r_{\mathcal{G},\boldsymbol{\pi},\boldsymbol{\sigma}}(\boldsymbol{x})$. Since the left hand side minus the right hand side equals to $\sum_{v\in\mathcal{V}} \sigma_v[\left(\prod_{i\in[m],i\neq j}\left(1-P_{i,v}(x_i)\right)\right)\cdot(P_{j,v}(x_j)-2P_{j,u}(x_j+1)+P_{j,v}(x_j+2))]$, we only need to show $\sum_{v\in\mathcal{V}} \sigma_v[\left(\prod_{i\in[m],i\neq j}\left(1-P_{i,v}(x_i)\right)\right)\cdot(g_{j,v}(x_j+2)-g_{j,v}(x_j+1))] \leq 0$. The above inequality holds because of Lemma 2.

Proof. We can observe that the modified Algorithm 1 always select the pair (i^*, b^*) with $b^* = 1$ because $r_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(\boldsymbol{k}+b\chi_i) - r_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(\boldsymbol{k}+(b-1)\chi_i) \leq r_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(\boldsymbol{k}+\chi_i) - r_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(\boldsymbol{k})$ for arbitrary $\boldsymbol{k}$, i and b. Thus, we have $s = B+1$ and by the similar argument for Eq. (C.11), we have $r(\boldsymbol{k}_g) = r(\boldsymbol{k}^{B+1}) \geq (1-1/e)r(\boldsymbol{k}^*)$, which completes the proof. □

C.3. Efficiently evaluating the reward function

One key issue to derive the budget allocation is to efficiently evaluate the reward function $r_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(\boldsymbol{k})$ and its marginal gains $\delta(i,b,\boldsymbol{k})$. Since we need to repetitively use the visiting probabilities $P_{i,u}(k_i)$, we pre-calculate $P_{i,u}(k_i)$ in $O(m\|\boldsymbol{c}\|_\infty n^3_{\max})$ time in our algorithms based on Eq. (9), where $n_{\max} = \max_i |\mathcal{V}_i|$. For the overlapping case, given the current budget allocation $\boldsymbol{k}$, we maintain a value $p_u = \prod_{i\in[m]}\left(1-P_{i,u}(k_i)\right)$ for each node $u \in \mathcal{V}$. The marginal gain $\delta(i,b,\boldsymbol{k}) = \sum_{u\in\mathcal{V}} \sigma_u\left[p_u\left(1-\frac{1-P_{i,u}(k_i+b)}{1-P_{i,u}(k_i)}\right)\right]/b$ can be evaluated in $O(n_{\max})$ time. Then, we update all $p_u = p_u\left(\frac{1-P_{i,u}(k_i+b)}{1-P_{i,u}(k_i)}\right)$ in $O(n_{\max})$ after we allocate b more budgets to layer i. Therefore, we can use $O(n_{\max})$ in total to evaluate $\delta(i,b,\boldsymbol{k})$. In practice, lazy evaluation [36] and parallel computing can be used to further accelerate our algorithm.

Appendix D. Budget effective greedy algorithm with partial enumeration and its analysis

D.1. Algorithm

The algorithm is shown in Algorithm 5.

Algorithm 5 Budget effective greedy algorithm with partial enumeration (BEGE).

Input: Graph $\mathcal{G}$, starting distributions $\boldsymbol{\alpha}$, node weights $\boldsymbol{\sigma}$, budget B, constraints $\boldsymbol{c}$.
Output: Budget allocation $\boldsymbol{k}$.

Compute visiting probabilities $(P_{i,u}(b))_{i\in[m],u\in\mathcal{V},b\in[c_i]}$ according to Eq. (7).
$\boldsymbol{k} \leftarrow$ BEGE($(P_{i,u}(b))_{i\in[m],u\in\mathcal{V},b\in[c_i]}$, $\boldsymbol{\sigma}$, B, $\boldsymbol{c}$).

Procedure BEGE($(P_{i,u}(b))_{i\in[m],u\in\mathcal{V},b\in[c_i]}$, $\boldsymbol{\sigma}$, B, $\boldsymbol{c}$)
  Let $\boldsymbol{k}_{\max} \leftarrow \boldsymbol{0}$.
  $\mathcal{S} \leftarrow \{\boldsymbol{k} = (k_1,\dots,k_m) | 0 \leq k_i \leq c_i, \sum_{i\in[m]} k_i \leq B, \sum_{i\in[m]} \mathbb{I}\{k_i > 0\} \leq 3\}$. {$\mathcal{S}$ contains all partial solutions which allocate partial budgets to at most three layers}
  for $\boldsymbol{k} \in \mathcal{S}$ **do**
    $K \leftarrow B - \sum_{i\in[m]} k_i$.
    Let $\mathcal{Q} \leftarrow \{(i,b_i) \mid i \in [m], 1 \leq b_i \leq c_i - k_i\}$.
    while $K > 0$ and $\mathcal{Q} \neq \emptyset$ **do**
      $(i^*, b^*) \leftarrow arg\,max_{(i,b)\in\mathcal{Q}} \delta(i,b,\boldsymbol{k})/b$. {Eq. (11)}
      if $b^* \leq K$ **then**
        $k_{i^*} \leftarrow k_{i^*} + b^*$, $K \leftarrow K - b^*$.
        Modify all pairs $(i^*, b) \in \mathcal{Q}$ to $(i^*, b-b^*)$.
        Remove all pairs $(i^*, b) \in \mathcal{Q}$ such that $b \leq 0$.
      else
        Remove (i^*, b^*) from $\mathcal{Q}$.
      end if
    end while
    if $r_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(\boldsymbol{k}) > r_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(\boldsymbol{k}_{\max})$, **then** $\boldsymbol{k}_{\max} \leftarrow \boldsymbol{k}$.
  end forreturn $\boldsymbol{k}_{\max} := (k_1,\dots,k_m)$.
end Procedure

D.2. Analysis

Theorem 5. *The Algorithm 5 obtains a $(1-1/e)$-approximate solution to the overlapping MuLaNE with arbitrary starting distributions.*

Proof. Suppose that we start from a partial solution $\boldsymbol{k}^1 \in \mathbb{Z}^m_{\geq 0} \neq \mathbf{0}$. Let us first reorder the optimal solution $\boldsymbol{k}^*$ according to a non-increasing marginal gain ordering. Namely, the marginal gain of the pair (s_1^*, b_1^*) with respect to the empty pair is the highest among all other pairs, the marginal gain of the pair (s_2^*, b_2^*) with respect to the solution consisting of pair (s_1^*, b_1^*) is the highest among all remaining pairs, and so on. To be concrete, $\boldsymbol{k}^* = \{(s_1^*, b_1^*), \ldots, (s_m^*, b_m^*)\}$, where $s_i^* \in [m] \setminus \{s_1^*, \ldots, s_{i-1}^*\}$ is selected to maximize the following equation:

$$\left(r\left(\sum_{j=1}^{i-1} b_j^* \chi_{s_j^*} + b^*{}_i \chi_{s_i^*} \right) - r\left(\sum_{j=1}^{i-1} b_j^* \chi_{s_j^*} \right) \right). \tag{D.1}$$

Then, we try to bound the "virtual" marginal gain $\Delta^s = r(\boldsymbol{k}^{s+1}) - r(\boldsymbol{k}^s)$, and recall s is the first iteration we can not extend the current solution. Without loss of generality, we assume the virtual pair (i^s, b^s) improves $\boldsymbol{k}^s$ towards the optimal budget allocation, i.e., $b^s + k_{i^s}^s \leq k_{i^s}^*$. Otherwise, if $b^s + k_{i^s}^s > k_{i^s}^*$, we can safely delete (i^s, b^s) in the queue $\mathcal{Q}$ and does not affect the greedy solution, the optimal solution and the analysis.

If we start from the initial solution $\boldsymbol{k}^1$ such that $\boldsymbol{k}^1$ matches (s_1^*, b_1^*), $(s_2^*, b_2^*), (s_3^*, b_3^*)$, i.e., $\boldsymbol{k}^1{}_{s_i^*} = b_i^*$ for $i = 1, 2, 3$, we have $\Delta^s = r(\boldsymbol{k}^{s+1}) - r(\boldsymbol{k}^s) \leq r(\boldsymbol{k}^s \vee k_{i^s}^* \chi_{i^s}) - r(\boldsymbol{k}^s) \leq r(k_{i^s}^* \chi_{i^s}) - r(\mathbf{0}) \leq r(b_1^* \chi_{s_1^*})$. Moreover, $\Delta^s = r(\boldsymbol{k}^{s+1}) - r(\boldsymbol{k}^s) \leq r(\boldsymbol{k}^s \vee k_{i^s}^* \chi_{i^s}) - r(\boldsymbol{k}^s) = r(\boldsymbol{k}^s \vee b_1^* \chi_{s_1^*} \vee k_{i^s}^* \chi_{i^s}) - r(\boldsymbol{k}^s \vee b_1^* \chi_{s_1^*}) \leq r(b_1^* \chi_{s_1^*} \vee k_{i^s}^* \chi_{i^s}) - r(b_1^* \chi_{s_1^*}) \leq r(b_1^* \chi_{s_1^*} + b_2^* \chi_{s_2^*}) - r(b_1^* \chi_{s_1^*})$. Similarly, we have $\Delta^s \leq r(b_1^* \chi_{s_1^*} + b_2^* \chi_{s_2^*} + b_3^* \chi_{s_3^*}) - r(b_1^* \chi_{s_1^*} + b_2^* \chi_{s_2^*})$. By adding above inequalities, we have $\Delta^s \leq r(\boldsymbol{k}^1)/3$.

Now, we can use Lemma 9 by setting $l = s$ and the fact $\sum_{j=1}^s b^j \geq B$ to show $r(\boldsymbol{k}^{s+1}) \geq r(\boldsymbol{k}^1) + (1-1/e)(r(\boldsymbol{k}^*) - r(\boldsymbol{k}^1))$. Combining $\Delta^s \leq r(\boldsymbol{k}^1)/3$, we have $r(\boldsymbol{k}^s) \geq (1-1/3)r(\boldsymbol{k}^1) + (1-1/e)(r(\boldsymbol{k}^*) - r(\boldsymbol{k}^1)) \geq (1-1/e)r(\boldsymbol{k}^*)$, which completes the proof. □

The time complexity of Algorithm 5 is $O(B^4 m^4 \|\boldsymbol{c}\|_\infty n_{\max} + \|\boldsymbol{c}\|_\infty m n_{\max}^3)$. This is because the number of all partial solutions is $|S| = O(B^3 m^3)$, and for any partial enumeration $\boldsymbol{k} \in S$, the time complexity is the same order as the Algorithm 1, i.e., $O(B\|\boldsymbol{c}\|_\infty m n_{\max})$, where $O(n_{\max})$ is the time to evaluate $\delta(i, b, \boldsymbol{k})$ as discussed before.

Appendix E. Algorithms and analysis of offline optimization for non-overlapping MuLaNE

E.1. Starting from the stationary distribution

Algorithm 6 Myopic greedy algorithm for the non-overlapping MuLaNE.

Input: Network $\mathcal{G}$, starting distributions $\boldsymbol{\alpha}$, budget B, constraints $\boldsymbol{c}$.
Output: Budget allocation $\boldsymbol{k}$.
1: Compute visiting probabilities $(P_{i,u}(b))_{i\in[m],u\in\mathcal{V},b\in[c_i]}$ according to Eq. (8).
2: Compute layer-level marginal gain $g_i(b)_{i\in[m],b\in[c_i]}$ according to Eq. (E.2).
3: $\boldsymbol{k} \leftarrow \mathrm{MG}((g_i(b))_{i\in[m],b\in[c_i]}, B, \boldsymbol{c})$
4: **Procedure** $\mathrm{MG}((g_i(b))_{i\in[m],b\in[c_i]}, B, \boldsymbol{c})$
5: Let $\boldsymbol{k} := \{k_1, \ldots, k_m\} \leftarrow \mathbf{0}$, $K \leftarrow B$.
6: **while** $K > 0$ **do**
7: $i^* \leftarrow arg\,max_{i\in[m],k_i+1\leq c_i} g_i(k_i+1)$. {$O(\log m)$ using the priority queue}
8: $k_{i^*} \leftarrow k_{i^*} + 1$, $K \leftarrow K - 1$.
9: **end whilereturn** $\boldsymbol{k} = (k_1, \ldots, k_m)$.
10: **end Procedure**

According to Eqs. (8) and (10), the reward function can be rewritten as

$$r_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(\boldsymbol{k}) = \sum_{i\in[m]} \sum_{b\in[k_i]} g_i(b), \tag{E.1}$$

where $g_i(b)$ represents the *layer-level marginal gain* when we allocate one more budget (from $b-1$ to b) to layer i, which is given by

$$g_i(b) = \sum_{v\in\mathcal{V}_i} \sigma_v (P_{i,v}(b) - P_{i,v}(b-1)). \tag{E.2}$$

From the definition of the reward function, we have two observations. First, budgets allocated to a layer will not affect the reward of other layers because layers are non-overlapping. Second, the layer-level marginal gain $g_i(b)$ for the i-th layer is non-increasing with respect to budget b. Based on these two observation, we propose the myopic greedy algorithm (Algorithm 6), which is a slight modification of Algorithm 2. Algorithm 6 takes the multi-layered network $\mathcal{G} = (\mathcal{G}_1, \ldots, \mathcal{G}_m)$, starting distributions $(\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_m)$, node weights $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_{|\mathcal{V}|})$ and total budget B as inputs. It consists of B rounds to compute the optimal budget allocation $\boldsymbol{k}^*$. In each round, line 7 in Algorithm 6 selects the layer i^* with the largest layer-level marginal gain and allocate one budget to layer i^* until total B budgets are used up.

Algorithm 7 Dynamic programming (DP) algorithm for the non-overlapping MuLaNE.

Input: Network $\mathcal{G}$, starting distributions $\boldsymbol{\alpha}$, node weights $\boldsymbol{\sigma}$, budget B, constraints $\boldsymbol{c}$.
Output: Budget allocation $\boldsymbol{k}$.
1: Compute visiting probabilities $(P_{i,u}(b))$ according to Eq. (8).
2: Compute layer-level marginal gain $(g_i(b))$ according to Eq. (E.2).
3: $\boldsymbol{k} \leftarrow \mathrm{DP}((g_i(b))_{i\in[m],b\in[c_i]}, B, \boldsymbol{c})$
4: **Procedure** $\mathrm{DP}(g_i(b))_{i\in[m],b\in c_i}, B, \boldsymbol{c}$
5: **for** $i \in \{0\} \cup [m], b \in \{0\} \cup [B]$ **do**
6: Set $A[i,b] \leftarrow \mathbf{0}$, $V[i,b] \leftarrow 0$.
7: **end for**
8: **for** $i \leftarrow 0$ **to** $m-1$ **do**
9: **for** $b \in [B]$ **do**
10: $j^* \leftarrow arg\,max_{j\in\{0\}\cup[c_{i+1}]}\{V[i,b-j] + r_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(j\boldsymbol{\chi_{i+1}})\}$.
11: $A[i+1,b] \leftarrow A[i,b-j^*] + j^*\boldsymbol{\chi_{i+1}}$.
12: $V[i+1,b] \leftarrow V[i,b-j^*] + r_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(j^*\boldsymbol{\chi_{i+1}})$.
13: **end for**
14: **end for**
15: **return** $A[m,B]$.
16: **end Procedure**

Theorem 6. *The Algorithm 6 obtains the optimal budget allocation to the non-overlapping MuLaNE with the stationary starting distributions.*

Proof. Define a two dimensional array $\boldsymbol{M} \in \mathbb{R}^{c_1+c_2+\ldots+c_m}$, where (i,j)-th entry is the j-th step layer-level marginal gain for layer i, i.e., $\boldsymbol{M}[i,j] = \sum_{v\in\mathcal{V}_i} g_{i,v}(j)$, for $i \in [m], j \in [c_i]$. Given the budget allocation $\boldsymbol{k} = \{k_1, \ldots, k_m\}$, the expected reward $r_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(\boldsymbol{k})$ can be written as the sum of elements in $\boldsymbol{M}$, i.e., $r_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(\boldsymbol{k}) = \sum_{i=1}^{m}\sum_{j=1}^{k_i} \boldsymbol{M}[i,j]$. Because $g_{i,v}(k_i)$ is non-increasing with respect to k_i, the element in each row or the layer-level marginal gain is non-increasing. Hence, the greedy method at step s choose the s-th largest element in $\boldsymbol{M}$. At step $s = B$, the greedy policy selects all B largest elements and the corresponding budget allocation maximizes the expected reward $r_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(\boldsymbol{k})$. □

Algorithm 6 uses $O(\|\boldsymbol{c}\|_\infty mn_{\max}^3)$ to compute visiting probabilities $P_{i,u}(k_i)$ and $O(\|\boldsymbol{c}\|_\infty mn_{\max})$ to compute layer-wise marginal gains $g_i(b)$. Then $\delta(i,1,\boldsymbol{k})$ can be evaluated in $O(\log m)$ using the priority queue, which is repeated for B iterations. Therefore, the time complexity of Algorithm 6 is $O(B\log m + \|\boldsymbol{c}\|_\infty mn_{\max}^3)$.

E.2. Starting from arbitrary distributions

When random explorers start from any arbitrary distribution, Algorithm 6 can not obtain the optimal solution. This is because the layer-level marginal gain $g_i(b)$ is *not* non-increasing with respect to b. So we have to adopt a more general technique, dynamic programming (DP), to solve the budget allocation problem. The key idea is to keep a DP budget allocation table $A \in \mathbb{R}^{(m+1)\times(B+1)}$, where the (i,b)-th entry $A[i,b]$ saves the optimal budget allocation by allocating b budgets to the first i layers. Another DP table $V \in R^{(m+1)\times(B+1)}$ saves the value of the optimal reward, where the (i,b)-th entry $V[i,b]$ corresponds to the optimal reward when setting $A[i,b]$ as the budget allocation. In the i'-th outer loop and b'-th inner loop in Algorithm 7, we have already obtained the optimal budget allocation for $A[i,b]$ with $i = 0,\ldots,i'$, $b \in [B]$, which help us to find the optimal amount of budget j^* to $(i'+1)$-th layer (in line 10), and thus we can obtain the $A[i'+1,b']$ and $V[i'+1,b']$ accordingly.

Theorem 7. *Algorithm 7 obtains the optimal budget allocation to the non-overlapping MuLaNE with an arbitrary starting distribution.*

Algorithm 7 uses $O(\|\boldsymbol{c}\|_\infty mn_{\max}^3)$ to compute visiting probabilities $P_{i,u}(k_i)$ and $O(\|\boldsymbol{c}\|_\infty mn_{\max})$ to compute layer-wise marginal gains $g_i(b)$. In the DP procedure, we can also pre-calculate all rewards $r_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(j\boldsymbol{\chi_{i+1}})$ in $O(\|\boldsymbol{c}\|_\infty m)$. Then DP procedure takes $O(\|\boldsymbol{c}\|_\infty)$ to find j^*, which is repeated for Bm iterations. Therefore, the Algorithm 7 has the time complexity of $O(B\|\boldsymbol{c}\|_\infty m + \|\boldsymbol{c}\|_\infty mn_{\max}^3)$.

Appendix F. Analysis of online learning for overlapping MuLaNE

F.1. Proof of 1-Norm bounded smoothness

Proof. According to Eq. (7), and since $\mu_{i,u,b} \le \mu'_{i,u,b}$, for $(i,u,b) \in \mathcal{A}$, we have $(1 - \prod_{i\in[m]}(1-\mu_{i,u,k_i})) \le (1 - \prod_{i\in[m]}(1-\mu'_{i,u,k_i}))$. Therefore for any $\sigma_u \le \sigma'_u$, we have $\sigma_u(1 - \prod_{i\in[m]}(1-\mu_{i,u,k_i})) \le \sigma'_u(1 - \prod_{i\in[m]}(1-\mu'_{i,u,k_i}))$

By summarizing both sides with $u \in \mathcal{V}$, we have $r_{\boldsymbol{k}}(\boldsymbol{\mu},\boldsymbol{\sigma}) \le r_{\boldsymbol{k}}(\boldsymbol{\mu'},\boldsymbol{\sigma'})$. □

Proof. The left-hand side:

$$\left| r_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\boldsymbol{k}) - r_{\boldsymbol{\mu'},\boldsymbol{\sigma'}}(\boldsymbol{k}) \right|$$

$$
\begin{aligned}
&= \Big| \sum_{u\in\mathcal{V}} [\sigma_u\Big(1-\prod_{i\in[m]}\Big(1-\mu_{i,u,k_i}\Big)\Big)\\
&- \sigma'_u\Big(1-\prod_{i\in[m]}\Big(1-\mu'_{i,u,k_i}\Big)\Big)]\Big|\\
&\le \sum_{u\in\mathcal{V}} \Big|\sigma_u(1-\prod_{i\in[m]} q_{i,u}) - \sigma'_u(1-\prod_{i\in[m]} q'_{i,u})\Big|\\
&= \sum_{u\in\mathcal{V}} |\sigma_u(1-\prod_{i\in[m]} q_{i,u}) - \sigma_u(1-\prod_{i\in[m]} q'_{i,u})\\
&+ \sigma_u(1-\prod_{i\in[m]} q'_{i,u}) - \sigma'_u(1-\prod_{i\in[m]} q'_{i,u})|\\
&\le \sum_{u\in\mathcal{V}} \sigma_u|\prod_{i\in[m]} q_{i,u} - \prod_{i\in[m]} q'_{i,u}|\\
&+ \sum_{u\in\mathcal{V}} |\sigma_u-\sigma'_u|(1-\prod_{i\in[m]} q'_{i,u}),
\end{aligned}
$$

where $q_{i,u} = (1-\mu_{i,u,k_i}), q'_{i,u} = (1-\mu'_{i,u,k_i})$.

For the first term, it can be derived that,

$$
\begin{aligned}
&\sigma_u\Big|\prod_{i\in[m]} q_{i,u} - \prod_{i\in[m]} q'_{i,u}\Big|\\
&= \sigma_u\Big|\sum_{i=1}^{m}\Big(\Big(\prod_{j=1}^{i-1} q'_{j,u}\Big)(q_{i,u}-q'_{i,u})\Big(\prod_{j=i+1}^{m} q_{j,u}\Big)\Big)\Big|\\
&\le \sigma_u \sum_{i=1}^{m}\Big|\Big(\Big(\prod_{j=1}^{i-1} q'_{j,u}\Big)(q_{i,u}-q'_{i,u})\Big(\prod_{j=i+1}^{m} q_{j,u}\Big)\Big)\Big|\\
&\le \sigma_u \sum_{i=1}^{m}\Big|q_{i,u}-q'_{i,u}\Big|\\
&= \sigma_u \sum_{i\in[m]}\Big|\mu_{i,u,k_i}-\mu'_{i,u,k_i}\Big|,
\end{aligned}
$$

where last inequality holds because $q_{j,u}, q'_{j,u} \le 1$ and $\sigma_u \in [0,1]$, for any $j \in [m], u \in \mathcal{V}$.

For the second term, we have

$$
\begin{aligned}
&|\sigma_u-\sigma'_u|(1-\prod_{i\in[m]} q'_{i,u})\\
&= |\sigma_u-\sigma'_u|(1-\prod_{i\in[m]}(1-\mu'_{i,u,k_i}))\\
&\le |\sigma_u-\sigma'_u| \sum_{i\in[m]} \mu'_{i,u,k_i},
\end{aligned}
$$

where the inequality holds because of the Weierstrass product inequality. Plugging the above two terms into the previous inequality, 1-Norm Bounded Smoothness is satisfied. □

F.2. Regret proof analysis

In this section, we give the regret analysis for overlapping `MuLaNE`.

Let $\mathcal{A}$ denote the set containing all base arms, i.e., $\mathcal{A} = \{(i,u,b)|i \in [m], u \in \mathcal{V}, b \in [c_i]\}$. We add subscript t to denote the value of a variable at the end of round $t \in [T]$, e.g. $\bar{\sigma}_t, \hat{\mu}_{i,u,b,t}$, where T is the total number of rounds. For example, $T_{i,u,b,t}$ denotes the total number of times that arm (i,u,b) is played at the end of round t. Let us first introduce a definition of an unlikely event, where $\hat{\mu}_{i,u,b,t-1}$ is not accurate as expected.

Definition 2. We say that the sampling is nice at the beginning of round t, if for each arm $(i,u,b) \in \mathcal{A}$, $|\hat{\mu}_{i,u,b,t-1} - \mu_{i,u,b}| < \rho_{i,u,b,t}$, where $\rho_{i,u,b,t} = \sqrt{\frac{3\ln t}{2T_{i,u,b,t-1}}}$. Let $\mathcal{N}_t$ be such event.

Lemma 11. *For each round $t \ge 1$, $Pr\{\neg N_t\} \le 2|\mathcal{A}|t^{-2}$*

Proof. For each round $t \ge 1$, we have

$$Pr\{\neg\mathcal{N}_t\}$$

$$
\begin{aligned}
&= Pr\{\exists (i,u,b) \in \mathcal{A}, |\hat{\mu}_{i,u,b,t-1} - \mu_{i,u,b}| \geq \sqrt{\frac{3\ln t}{2T_{i,u,b,t-1}}}\} \\
&\leq \sum_{(i,u,b)\in\mathcal{A}} Pr\{|\hat{\mu}_{i,u,b,t-1} - \mu_{i,u,b}| \geq \sqrt{\frac{3\ln t}{2T_{i,u,b,t-1}}}\} \\
&= \sum_{(i,u,b)\in\mathcal{A}} \sum_{k=1}^{t-1} Pr\{T_{i,u,b,t-1} = k \\
&\wedge |\hat{\mu}_{i,u,b,t-1} - \mu_{i,u,b}| \geq \sqrt{\frac{3\ln t}{2T_{i,u,b,t-1}}}\} \\
&\leq \sum_{(i,u,b)\in\mathcal{A}} \sum_{k=1}^{(t-1)} \frac{2}{t^3} \leq 2|\mathcal{A}|t^{-2}.
\end{aligned}
$$

Given $T_{i,u,b,t-1} = k$, $\hat{\mu}_{i,u,b,t-1}$ is the average of k i.i.d. random variables $Y_{i,u,b}^{[1]}, \ldots, Y_{i,u,b}^{[k]}$, where $Y_{i,u,b}^{[j]}$ is the Bernoulli random variable (computed in Algorithm 3 line 12) when the arm (i,u,b) is played for the j-th time during the execution. That is, $\hat{\mu}_{i,u,b,t-1} = \sum_{j=1}^{k} Y_{i,u,b}^{[j]}/k$. Note that the independence of $Y_{i,u,b}^{[1]}, \ldots, Y_{i,u,b}^{[k]}$ comes from the fact we select a new starting position following the starting distribution $\boldsymbol{\alpha}_i$ at the beginning of each round t. And the last inequality uses the Hoeffding Inequaility [37]. □

Then we can use monotonicity and 1-norm bounded smoothness properties (Properties 1 and 2) to bound the reward gap $\boldsymbol{\Delta}_{\boldsymbol{k}_t} = \xi r_{\boldsymbol{\mu},\sigma}(\boldsymbol{k}^*) - r_{\boldsymbol{\mu},\sigma}(\boldsymbol{k}_t)$ between the optimal action $\boldsymbol{k}^*$ and the action $\boldsymbol{k}_t = (k_{t,1}, \ldots, k_{t,m})$ selected by our algorithm A. To achieve this, we introduce a positive real number $M_{i,u,b}$ for each arm (i,u,b) and define $M_{\boldsymbol{k}_t} = \max_{(i,u,b)\in S_t} M_{i,u,b}$, where $S_t = \{(i,u,b) \in \mathcal{A} | b = k_{t,i}\}$. Define $|\mathcal{E}'| = |S_t| = \sum_{i\in[m]} |\mathcal{V}| = m|\mathcal{V}|$, and let

$$
\kappa_T(M,s) = \begin{cases} 2, & \text{if } s = 0, \\ 3\sqrt{\frac{6\ln T}{s}}, & \text{if } 1 \leq s \leq \ell_T(M), \\ 0, & \text{if } s \geq \ell_T(M) + 1, \end{cases}
$$

where

$$
\ell_T(M) = \lfloor \frac{54|\mathcal{E}'|^2 \ln T}{M^2} \rfloor.
$$

Lemma 12. *If $\{\Delta_{\boldsymbol{k}_t} \geq M_{\boldsymbol{k}_t}\}$ and $\mathcal{N}_t$ holds, we have*

$$
\Delta_{\boldsymbol{k}_t} \leq \sum_{(i,u,b)\in S_t} \kappa_T(M_{i,u,b}, T_{i,u,b,t-1}). \tag{F.1}
$$

Proof. First, we can observe, for $(i,u,b) \in \mathcal{A}$, $\mu_{i,u,b}$ is upper bounded by $\bar{\mu}_{i,u,b,t}$, i.e., $\bar{\mu}_{i,u,b,t} \geq \mu_{i,u,b}$, when $\mathcal{N}_t$ holds. This is due to

$$
\begin{aligned}
\bar{\mu}_{i,u,b,t} &= \max_{j\in[b]} \tilde{\mu}_{i,u,j,t} \\
&\geq \tilde{\mu}_{i,u,b,t} \\
&= \min\{\hat{\mu}_{i,u,b,t-1} + \rho_{i,u,b,t}, 1\} \\
&\geq \mu_{i,u,b},
\end{aligned}
$$

where the last inequality comes from the fact that $|\hat{\mu}_{i,u,b,t-1} - \mu_{i,u,b}| < \rho_{i,u,b,t}$ when $\mathcal{N}_t$ holds.

Then, notice that the right hand of Eq. (F.1) is non-negative, it is trivially satisfied when $\Delta_{\boldsymbol{k}_t} = 0$. We only need to consider $\Delta_{\boldsymbol{k}_t} > 0$. By $\mathcal{N}_t$ and Property 1, We have the following inequalities,

$$
r_{\bar{\boldsymbol{\mu}}_t,\bar{\sigma}}(\boldsymbol{k}_t) \geq \xi r_{\bar{\boldsymbol{\mu}}_t,\bar{\sigma}}(\boldsymbol{k}^*) \geq \xi r_{\boldsymbol{\mu},\sigma}(\boldsymbol{k}^*) = \Delta_{\boldsymbol{k}_t} + r_{\boldsymbol{\mu},\sigma}(\boldsymbol{k}_t).
$$

The first inequality is due to the *oracle* outputs the (ξ,β)-approximate solution given parameters $\bar{\boldsymbol{\mu}}_t$, where $\xi = 1 - e^{-\eta}, \beta = 1$ by definition. The second inequality is due to Condition 1 (monotonicity).

Then, by Condition 2, we have

$$
\begin{aligned}
\Delta_{\boldsymbol{k}_t} &\leq r_{\bar{\boldsymbol{\mu}}_t,\bar{\sigma}}(\boldsymbol{k}_t) - r_{\boldsymbol{\mu},\sigma}(\boldsymbol{k}_t) \\
&\leq \sum_{(i,u,b)\in S_t} (\sigma_u |\bar{\mu}_{i,u,b,t} - \mu_{i,u,b}| + |\bar{\sigma}_u - \sigma_u| \bar{\mu}_{i,u,b,t}) \\
&\leq \sum_{(i,u,b)\in S_t} (|\bar{\mu}_{i,u,b,t} - \mu_{i,u,b}| + \rho_{i,u,b,t}),
\end{aligned}
$$

where the last inequality is due to the following observation: $\bar{\sigma}_u \neq \sigma_u$ if and only if u has not been visited so far, so $\hat{\mu}_{i,u,b,t-1} = 0$ for any $i \in [m], b \in [B]$ and $\bar{\mu}_{i,u,b,t} = \max_{j\in[b]}\{\hat{\mu}_{i,u,j,t-1} + \rho_{i,u,j,t}\} = \rho_{i,u,b,t}$.

Next, we can bound $\Delta_{\boldsymbol{k}_t}$ by bounding $\bar{\mu}_{i,u,b,t} - \mu_{i,u,b} + \rho_{i,u,b,t}$ by applying a transformation. As $\{\Delta_{\boldsymbol{k}_t} \geq M_{\boldsymbol{k}_t}\}$ holds by assumption, so $\sum_{(i,u,b)\in S_t} |\bar{\mu}_{i,u,b,t} - \mu_{i,u,b}| + \rho_{i,u,b,t} \geq \Delta_{\boldsymbol{k}_t} \geq M_{\boldsymbol{k}_t}$. We have

$$\begin{aligned}
\Delta_{\boldsymbol{k}_t} &\leq \sum_{(i,u,b)\in S_t} (|\bar{\mu}_{i,u,b,t} - \mu_{i,u,b}| + \rho_{i,u,b,t}) \\
&\leq -M_{\boldsymbol{k}_t} + 2\sum_{(i,u,b)\in S_t} |\bar{\mu}_{i,u,b,t} - \mu_{i,u,b}| + \rho_{i,u,b,t} \\
&= 2\sum_{(i,u,b)\in S_t} (|\bar{\mu}_{i,u,b,t} - \mu_{i,u,b}| + \rho_{i,u,b,t} - \frac{M_{\boldsymbol{k}_t}}{2|\mathcal{E}'|}) \\
&\leq 2\sum_{(i,u,b)\in S_t} (|\bar{\mu}_{i,u,b,t} - \mu_{i,u,b}| + \rho_{i,u,b,t} - \frac{M_{i,u,b}}{2|\mathcal{E}'|}).
\end{aligned} \tag{F.2}$$

By $\mathcal{N}_t$, for all $(i,u,b) \in \mathcal{A}$, we have:

$$\begin{aligned}
&|\bar{\mu}_{i,u,b,t} - \mu_{i,u,b}| + \rho_{i,u,b,t} - \frac{M_{i,u,b}}{2|\mathcal{E}'|} \\
&= \tilde{\mu}_{i,u,l,t} - \mu_{i,u,b} + \rho_{i,u,b,t} - \frac{M_{i,u,b}}{2|\mathcal{E}'|} \quad (l = \arg\max_{j\in[b]} \tilde{\mu}_{i,u,j,t}) \\
&\leq \tilde{\mu}_{i,u,l,t} - \mu_{i,u,l} + \rho_{i,u,b,t} - \frac{M_{i,u,b}}{2|\mathcal{E}'|} \\
&\leq \min\{2\rho_{i,u,l,t} + \rho_{i,u,b,t}, 1\} - \frac{M_{i,u,b}}{2|\mathcal{E}'|} \\
&\leq \min\{3\rho_{i,u,b,t}, 1\} - \frac{M_{i,u,b}}{2|\mathcal{E}'|} \\
&\leq \min\left\{3\sqrt{\frac{3\ln T}{2T_{i,u,b,t-1}}}, 1\right\} - \frac{M_{i,u,b}}{2|\mathcal{E}'|}.
\end{aligned}$$

where the first inequality is due to $\mu_{i,u,j} \leq \mu_{i,u,b}$ for $j \leq b$, the second inequality is due to the event $\mathcal{N}_t$ and the third inequality holds because $T_{i,u,j,t-1} \geq T_{i,u,b,t-1}$ for $j \leq b$.

Now consider the following cases:

Case 1. If $T_{i,u,b,t-1} \leq \ell_T(M_{i,u,b})$, we have $\bar{\mu}_{i,u,b,t} - \mu_{i,u,b} + \rho_{i,u,b,t} - \frac{M_{i,u,b}}{2|\mathcal{E}'|} \leq \min\left\{3\sqrt{\frac{3\ln T}{2T_{i,u,b,t-1}}}, 1\right\} \leq \frac{1}{2}\kappa_T(M_{i,u,b}, T_{i,u,b,t-1})$.

Case 2. If $T_{i,b,t-1} \geq \ell_T(M_{i,u,b,t}) + 1$, then $3\sqrt{\frac{3\ln T}{2T_{i,u,b,t-1}}} \leq \frac{M_{i,u,b}}{2|\mathcal{E}'|}$, so $\bar{\mu}_{i,u,b,t} - \mu_{i,u,b} + \rho_{i,u,b,t} - \frac{M_{i,u,b}}{2|\mathcal{E}'|} \leq 0 = \frac{1}{2}\kappa_T(M_{i,u,b}, T_{i,u,b,t-1})$.

Combine these two cases and Eq. (F.2), we have the following inequality, which concludes the lemma.

$$\Delta_{\boldsymbol{k}_t} \leq \sum_{(i,u,b)\in S_t} \kappa_T(M_{i,u,b}, T_{i,u,b,t-1}).$$

□

Proof. By above lemmas, we have

$$\begin{aligned}
&\sum_{t=1}^{T} \mathbb{I}\Big(\{\Delta_{\boldsymbol{k}_t} \geq M_{\boldsymbol{k}_t}\} \wedge \mathcal{N}_t\Big)\Delta_{\boldsymbol{k}_t} \\
&\leq \sum_{t=1}^{T}\sum_{(i,u,b)\in S_t} \kappa_T(M_{i,u,b}, T_{i,u,b,t-1}) \\
&= \sum_{(i,u,b)\in\mathcal{A}}\sum_{t'\in\{t|t\in[T],\boldsymbol{k}_{t,i}=b\}} \kappa_T(M_{i,u,b}, T_{i,u,b,t'-1}) \\
&\leq \sum_{(i,u,b)\in\mathcal{A}}\sum_{s=0}^{T_{i,u,b,T}} \kappa_T(M_{i,u,b}, s) \\
&\leq 2|\mathcal{A}| + \sum_{(i,u,b)\in\mathcal{A}}\sum_{s=1}^{\ell_T(M_{i,u,b})} 3\sqrt{6\frac{\ln T}{s}} \\
&\leq 2|\mathcal{A}| + \sum_{(i,u,b)\in\mathcal{A}}\int_{s=0}^{\ell_T(M_{i,u,b})} 3\sqrt{6\frac{\ln T}{s}}\,ds \\
&\leq 2|\mathcal{A}| + \sum_{(i,u,b)\in\mathcal{A}} 6\sqrt{6\ln T\,\ell_T(M_{i,u,b})}
\end{aligned}$$

$$\leq 2|\mathcal{A}| + \sum_{(i,u,b)\in\mathcal{A}} \frac{108|\mathcal{E}'|\ln T}{M_{i,u,b}}.$$

For distribution dependent bound, let us set $M_{i,u,b} = \Delta^{i,u,b}_{\min}$ and thus the event $\{\Delta_{\boldsymbol{k}_t} \geq M_{\boldsymbol{k}_t}\}$ always holds, i.e., $\mathbb{I}\{\Delta_{\boldsymbol{k}_t} < M_{\boldsymbol{k}_t}\} = 0$. We have

$$\sum_{t=1}^{T} \mathbb{I}\Big(\mathcal{N}_t \wedge \{\Delta_{\boldsymbol{k}_t} \geq M_{\boldsymbol{k}_t}\}\Big)\Delta_{\boldsymbol{k}_t}$$
$$\leq 2|\mathcal{A}| + \sum_{(i,u,b)\in\mathcal{A}} \frac{108|\mathcal{E}'|\ln T}{\Delta^{i,u,b}_{\min}}.$$

By Lemma 11, $Pr\{\neg N_t\} \leq 2|\mathcal{A}|t^{-2}$, then

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{I}(\neg\mathcal{N}_t)\Delta_{\boldsymbol{k}_t}\right] \leq \sum_{t=1}^{T} 2|\mathcal{A}|t^{-2}\Delta_{\max} \leq \frac{\pi^2}{3}|\mathcal{A}|\Delta_{\max}.$$

By our choice of $M_{i,u,b}$,

$$\sum_{t=1}^{T} \mathbb{I}\Big(\Delta_{\boldsymbol{k}_t} < M_{\boldsymbol{k}_t}\Big)\Delta_{\boldsymbol{k}_t} = 0.$$

By the definition of the regret,

$$\begin{aligned}
Reg_{\boldsymbol{\mu}}(T) &= \mathbb{E}\left[\sum_{t=1}^{T} \Delta_{\boldsymbol{k}_t}\right] \\
&\leq \mathbb{E}\Bigg[\sum_{t=1}^{T}\Bigg(\mathbb{I}(\neg\mathcal{N}_t) + \mathbb{I}\Big(\Delta_{\boldsymbol{k}_t} < M_{\boldsymbol{k}_t}\Big) \\
&\quad + \mathbb{I}\Big(\mathcal{N}_t \wedge \{\Delta_{\boldsymbol{k}_t} \geq M_{\boldsymbol{k}_t}\}\Big)\Bigg)\Delta_{\boldsymbol{k}_t}\Bigg] \\
&\leq \sum_{(i,u,b)\in\mathcal{A}} \frac{108|\mathcal{E}'|\ln T}{\Delta^{i,u,b}_{min}} + 2|\mathcal{A}| + \frac{\pi^2}{3}|\mathcal{A}|\Delta_{max} \\
&= \sum_{(i,u,b)\in\mathcal{A}} \frac{108m|\mathcal{V}|\ln T}{\Delta^{i,u,b}_{min}} + 2|\mathcal{A}| + \frac{\pi^2}{3}|\mathcal{A}|\Delta_{max}.
\end{aligned}$$

For distribution independent bound, we take $M_{i,u,b} = M = \sqrt{108|\mathcal{E}'||\mathcal{A}|\ln T/T}$, then we have $\sum_{t=1}^{T}\mathbb{I}\{\Delta_{\boldsymbol{k}_t} \leq M_{\boldsymbol{k}_t}\} \leq TM$. Then

$$\begin{aligned}
&Reg_{\boldsymbol{\mu}}(T) \\
&\leq \sum_{(i,u,b)\in\mathcal{A}} \frac{108|\mathcal{E}'|\ln T}{M_{i,u,b}} + 2|\mathcal{A}| + \frac{\pi^2}{3}|\mathcal{A}|\Delta_{max} + TM \\
&\leq \frac{108|\mathcal{E}'||\mathcal{A}|\ln T}{M} + 2|\mathcal{A}| + \frac{\pi^2}{3}|\mathcal{A}|\Delta_{max} + TM \\
&= 2\sqrt{108|\mathcal{E}'||\mathcal{A}|T\ln T} + \frac{\pi^2}{3}|\mathcal{A}|\Delta_{max} + 2|\mathcal{A}| \\
&\leq 21\sqrt{m|\mathcal{V}||\mathcal{A}|T\ln T} + \frac{\pi^2}{3}|\mathcal{A}|\Delta_{max} + 2|\mathcal{A}|. \qquad \square
\end{aligned}$$

F.3. Extension for random node weights

In this section, we extend the deterministic node weight σ_u for node $u \in \mathcal{V}$ to a random weight. Specifically, we denote $\tilde{\sigma}_u \in [0,1]$ as the random weight when we first visit u (after the first visit we will not get any reward so on so forth), and denote $\sigma_u := \mathbb{E}[\tilde{\sigma}_u]$.

For the offline setting, the reward function in Eq. (7) remains the same since $\tilde{\sigma}_u$ is independent of all other random variables. For the online setting, since the reward function remains unchanged, Properties 1 and 2 still hold. However, we need to change the way we maintain the optimistic node weight $\bar{\sigma}_u$. Compared with Algorithm 3, the major difference is that $\bar{\sigma}_u$ now represent the upper confidence bound (UCB) value of σ_u.

Regret analysis. Let $\mathcal{A}$ denote the set containing all base arms for visiting probabilities, i.e., $\mathcal{A} = \{(i,u,b)|i \in [m], u \in \mathcal{V}, b \in [c_i]\}$. With a bit of abuse of notation, we also use $\mathcal{V}$ to denote the set of base arms corresponding to the node weights. Therefore, $\mathcal{A} \cup \mathcal{V}$ is the set of all base arms, which is different from the Section F.2 that only has $\mathcal{A}$ as base arms and does not have base arms for random weights. Following the notations of Wang and Chen [17], we define $X_a \in [0,1]$ be the random outcome of base arm $a \in \mathcal{A} \cup \mathcal{V}$ and

define D as the unknown joint distribution of random outcomes $\boldsymbol{X} = (X_a)_{a\in\mathcal{A}\cup\mathcal{V}}$ with unknown mean $\mu_a = \mathbb{E}_{\boldsymbol{X}\sim D}[X_a]$. Let $\boldsymbol{k}$ be any feasible budget allocation, we denote $p_a^{D,\boldsymbol{k}}$ the probability that budget allocation $\boldsymbol{k}$ triggers arm a (i.e. outcome X_a is observed) when the unknown environment distribution is D. Specifically, $p_a^{D,\boldsymbol{k}} = 1$ for $a \in \{(i,u,b)|i\in[m], u\in\mathcal{V}, b = k_i\}$, $p_a^{D,\boldsymbol{k}} = 1 - \prod_{i\in[m]}(1-\mu_{i,a,k_i})$ for $a\in\mathcal{V}$ and $p_a^{D,\boldsymbol{k}} = 0$ for the rest of base arms. Therefore, we can rewrite the inequality in Property 2 as

$$\begin{aligned}
&|r_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\boldsymbol{k}) - r_{\boldsymbol{\mu}',\boldsymbol{\sigma}'}(\boldsymbol{k})| \\
&\le \sum_{i\in[m],u\in\mathcal{V},b=k_i} (\sigma'_u|\mu_{i,u,b} - \mu'_{i,u,b}|) \\
&+ \sum_{u\in\mathcal{V}} |\sigma_u - \sigma'_u| \left(1 - \prod_{i\in[m]}(1-\mu_{i,u,b})\right) \\
&\le \sum_{a\in\mathcal{A}} p_a^{D,\boldsymbol{k}} |\mu_a - \mu'_a|,
\end{aligned} \tag{F.3}$$

where we abuse the notation to denote μ_a as σ_a and μ'_a as σ'_a if $a\in\mathcal{A}_2$ in the last inequality. We can observe that the above rewritten Property 2 can be viewed as a special case of the 1-norm TPM bounded smoothness condition in Wang and Chen [17].

Now we can apply the similar proof in Section B.3 in Wang and Chen [17], together with how we deal with the $|\bar{\mu}_{i,u,b,t} - \mu_{i,u,b}|$ in Appendix F.2 to get a distribution-dependent regret bound.

Let $K = m|\mathcal{V}| + |\mathcal{V}|$ be the maximum number of triggered arms, $|\mathcal{A}\cup\mathcal{V}| = Bm|\mathcal{V}| + |\mathcal{V}|$ be the number of base arms and $\lceil x\rceil_0 = \max\{\lceil x\rceil, 0\}$. For any feasible budget allocation $\boldsymbol{k}$ and any base arm $a\in\mathcal{A}\cup\mathcal{V}$, the gap $\Delta_{\boldsymbol{k}} = \max\{0, \xi r_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\boldsymbol{k}^*) - r_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\boldsymbol{k})\}$ and we define $\Delta^a_{\min} = \inf_{\boldsymbol{k}: p_a^{D,\boldsymbol{k}}>0, \Delta_{\boldsymbol{k}}>0} \Delta_{\boldsymbol{k}}$, $\Delta^a_{\max} = \sup_{\boldsymbol{k}: p_a^{D,\boldsymbol{k}}>0, \Delta_{\boldsymbol{k}}>0} \Delta_{\boldsymbol{k}}$. As convention, if there is no budget allocation $\boldsymbol{k}$ such that $p_a^{D,\boldsymbol{k}} > 0$ and $\Delta_{\boldsymbol{k}} > 0$, we define $\Delta^a_{\min} = +\infty$, $\Delta^a_{\max} = 0$. We define $\min_{a\in\mathcal{A}\cup\mathcal{V}} \Delta^a_{\min}$ and $\max_{a\in\mathcal{A}\cup\mathcal{V}} \Delta^a_{\max}$.

Theorem 8. *Algorithm 3 with random node weights has the following distribution-dependent* $(1-e^{-\eta}, 1)$ *approximation regret,*

$$\begin{aligned}
Reg_{\boldsymbol{\mu},\boldsymbol{\sigma}}(T) &\le \sum_{a\in\mathcal{A}\cup\mathcal{V}} \frac{576K\ln T}{\Delta^a_{\min}} \\
&+ 4|\mathcal{A}\cup\mathcal{V}| + \sum_{a\in\mathcal{A}\cup\mathcal{V}} \left(\lceil \log_2 \frac{2K}{\Delta^a_{\min}}\rceil_0 + 2\right)\frac{\pi^2}{6}\Delta_{\max},
\end{aligned}$$

We can also have a distribution-independent bound,

$$\begin{aligned}
Reg_{\boldsymbol{\mu},\boldsymbol{\sigma}}(T) &\le 12\sqrt{|\mathcal{A}\cup\mathcal{V}|KT\ln T} \\
&+ 2|\mathcal{A}\cup\mathcal{V}| + \left(\lceil \log_2 \frac{T}{18\ln T}\rceil_0 + 2\right)\frac{\pi^2}{6}\Delta_{\max},
\end{aligned}$$

Remark. Compared with the regret bound in Theorem 3 in the main text, we can see the base arms now become $\mathcal{A}\bigcup\mathcal{V}$ instead of $\mathcal{A}$, which is worse than the regret bound of the fixed unknown weight setting. This is a trade-off since we need more estimation to handle the uncertain random node weight, which incurs additional regrets. It also shows our non-trivial analysis of the fixed unknown weights carefully merges the error term into base arms $\mathcal{A}$ and incurs only a constant factor of extra regrets.

Appendix G. Analysis of explorer-relaxed online learning for overlapping MuLaNE

G.1. Proof of precision-balanced bounded smoothness

To prove this property while conserving space, we directly address a more general case, the MuLaNE problem with random node weights, rather than just fixed node weights.

Similar to Section F.3, let $\mathcal{A}$ denote the set containing all base arms for visiting probabilities, i.e., $\mathcal{A} = \{(i,u,b)|i\in[m], u\in\mathcal{V}, b\in[c_i]\}$. With a bit of abuse of notation, we also use $\mathcal{V}$ to denote the set of base arms corresponding to the node weights. Therefore, $\mathcal{A}\cup\mathcal{V}$ is the set of all base arms. Let $\boldsymbol{k}$ be any feasible budget allocation. We denote $p_a^{D,\boldsymbol{k}}$ the probability that action $\boldsymbol{k}$ triggers arm a (i.e. outcome X_a is observed) when the unknown environment distribution is D. Specifically, $p_a^{D,\boldsymbol{k}} = 1$ for $a\in\{(i,u,b)|i\in[m], u\in\mathcal{V}, b = k_i\}$, $p_a^{D,\boldsymbol{k}} = 1 - \prod_{i\in[m]}(1-\mu_{i,a,k_i})$ for $a\in\mathcal{V}$ and $p_a^{D,\boldsymbol{k}} = 0$ for the rest of base arms.

Property 4. *(General Precision-Balanced Bounded Smoothness). The reward function* $r_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\boldsymbol{k})$ *satisfies the following general precision-balanced bounded smoothness condition, i.e., for any budget allocation* $\boldsymbol{k}$*, any two vectors* $\boldsymbol{\mu}=(\mu_{i,u,b})_{(i,u,b)\in\mathcal{A}}$, $\boldsymbol{\mu}' = \boldsymbol{\mu} + \boldsymbol{\zeta}_\mu + \boldsymbol{\eta}_\mu = (\mu'_{i,u,b})_{(i,u,b)\in\mathcal{A}}$ *and any node weights* $\boldsymbol{\sigma}$, $\boldsymbol{\sigma}' = \boldsymbol{\sigma} + \boldsymbol{\zeta}_\sigma + \boldsymbol{\eta}_\sigma$, *where* $\boldsymbol{\zeta}, \boldsymbol{\eta} \in [-1,1]^m$ *represent the parameter change, we have* $|r_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\boldsymbol{k}) - r_{\boldsymbol{\mu}',\boldsymbol{\sigma}'}(\boldsymbol{k})| \le \sum_{u\in\mathcal{V}} \left(\sum_{i\in[m],b=k_i} \left|\eta_{\mu,i,u,b}\right| + |\eta_{\sigma,u}| p_u^{D,\boldsymbol{k}}\right) + \sqrt{1.25|\mathcal{V}|}\sqrt{\sum_{u\in\mathcal{V}} \left(\sum_{i\in[m],b=k_i} \left(\frac{\zeta^2_{\mu,i,u,b}}{(1-\mu_{i,u,b})\mu_{i,u,b}}\right) + \frac{\zeta^2_{\sigma,u}(p_u^{D,\boldsymbol{k}})^2}{(1-\sigma_u)\sigma_u}\right)}$.

Unlike Condition 3 in Liu et al. [28] (Directional TPVM Bounded Smoothness), Property 4 represents the undirectional version (i.e., allowing $\boldsymbol{\zeta}$ and $\boldsymbol{\eta}$ to be negative). We now provide the proof for Property 4. For fixed node weights, the same procedure applies by setting the triggering probability to 1 for the effective base arm.

Proof. Denote $\bar{p}_u^{D,\boldsymbol{k}} = 1 - \prod_{i\in[m]}(1-\mu'_{i,u,k_i})$ for $u\in\mathcal{V}$. To better-fit layout constraints and present the following complex proof clearly, we split the formula into two parts.

In the first part, we preliminary decompose and simplify the utility function difference:

$$\begin{aligned}
&\left|r_{\mu,\sigma}(\boldsymbol{k}) - r_{\mu',\sigma'}(\boldsymbol{k})\right| \\
=&\left|\sum_{u\in\mathcal{V}}\sigma'_u\left(\prod_{i\in[m]}(1-\mu_{i,u,k_i}) - \prod_{i\in[m]}(1-\mu'_{i,u,k_i})\right) + \sum_{u\in\mathcal{V}}(\sigma'_u-\sigma_u)p_u^{D,\boldsymbol{k}}\right| && \text{(G.1)}\\
\leq&\left|\sum_{u\in\mathcal{V}}\sum_{i\in[m]}(\mu'_{i,u,k_i}-\mu_{i,u,k_i})(\prod_{\ell=1}^{i-1}(1-\mu_{\ell,u,k_\ell})\prod_{\ell=i+1}^{m}(1-\mu_{\ell,u,k_\ell}))\right| + \left|\sum_{u\in\mathcal{V}}(\sigma'_u-\sigma_u)p_u^{D,\boldsymbol{k}}\right| \\
\leq&\sum_{u\in\mathcal{V}}\sum_{i\in[m],b=k_i}(\left|\zeta_{\mu,i,u,b}\right| + \left|\eta_{\mu,i,u,b}\right|)\prod_{\ell=1}^{i-1}(1-\mu_{\ell,u,k_\ell})\sigma'_u + \sum_{u\in\mathcal{V}}(|\zeta_{\sigma,u}| + |\eta_{\sigma,u}|)p_u^{D,\boldsymbol{k}}, && \text{(G.2)}
\end{aligned}$$

where Eq. (G.1) is by telescoping the difference of $\prod_{i\in[m]}(1-\mu_{i,u,k_i}) - \prod_{i\in[m]}(1-\mu'_{i,u,k_i}))$, and Eq. (G.2) is due to $\mu'_{i,u,k_i}\in[0,1]$ for any $i\in[m], u\in\mathcal{V}$.

Then, we apply some common inequalities and deductions to bound the difference:

$$\begin{aligned}
&\sum_{u\in\mathcal{V}}\sum_{i\in[m],b=k_i}(\left|\zeta_{\mu,i,u,b}\right| + \left|\eta_{\mu,i,u,b}\right|)(\prod_{\ell=1}^{i-1}(1-\mu_{\ell,u,k_\ell})\sigma'_u + \sum_{u\in\mathcal{V}}(|\zeta_{\sigma,u}| + |\eta_{\sigma,u}|)p_u^{D,\boldsymbol{k}} \\
\leq&\sum_{u\in\mathcal{V}}\sum_{i\in[m],b=k_i}(\frac{\left|\zeta_{\mu,i,u,b}\right|}{\sqrt{(1-\mu_{i,u,b})\mu_{i,u,b}}})\sqrt{(\prod_{\ell=1}^{i-1}(1-\mu_{\ell,u,k_\ell})\mu_{i,u,b}\sigma'_u} \\
&+\sum_{u\in\mathcal{V}}\left(\frac{|\zeta_{\sigma,u}|p_u^{D,\boldsymbol{k}}}{\sqrt{(1-\sigma_u)\sigma_u}}\right)\sqrt{(1-\sigma_u)\sigma_u} + \left(\sum_{u\in\mathcal{V}}\sum_{i\in[m],b=k_i}\left|\eta_{\mu,i,u,b}\right| + \sum_{u\in\mathcal{V}}|\eta_{\sigma,u}|p_u^{D,\boldsymbol{k}}\right) && \text{(G.3)}\\
\leq&\sqrt{\sum_{u\in\mathcal{V}}\sum_{i\in[m],b=k_i}(\frac{\zeta^2_{\mu,i,u,b}}{(1-\mu_{i,u,b})\mu_{i,u,b}}) + \sum_{u\in\mathcal{V}}\frac{\zeta^2_{\sigma,u}(p_u^{D,\boldsymbol{k}})^2}{(1-\sigma_u)\sigma_u}} \\
&\cdot\sqrt{\sum_{u\in\mathcal{V}}\sum_{i\in[m],b=k_i}\prod_{\ell=1}^{i-1}(1-\mu_{\ell,u,k_\ell})\mu_{i,u,b}\sigma'^2_u + \sum_{u\in\mathcal{V}}(1-\sigma_u)\sigma_u} \\
&+\left(\sum_{u\in\mathcal{V}}\sum_{i\in[m],b=k_i}\left|\eta_{\mu,i,u,b}\right| + \sum_{u\in\mathcal{V}}|\eta_{\sigma,u}|p_u^{D,\boldsymbol{k}}\right). && \text{(G.4)}
\end{aligned}$$

In Eq. (G.3), we multiply the first term by $\sqrt{\mu_{i,u,b}}$ (where $b=k_i$) and then divide it by $\sqrt{\mu_{i,u,b}(1-\mu_{i,u,b})}$. Similarly, we multiply the second term by $\sqrt{\sigma_u}$ and divide it by $\sqrt{\sigma_u(1-\sigma_u)}$. In Eq. (G.4), we apply the Cauchy-Schwarz inequality simultaneously to both terms.

Then, with $\mu_{i,u,b}\in[0,1],\forall i,u$ and $\sigma_u,\sigma'_u\in[0,1],\forall u$, we have

$$\begin{aligned}
&Eq.(G.4) && \text{(G.5)}\\
&\leq\sqrt{\sum_{u\in\mathcal{V}}\sum_{i\in[m],b=k_i}(\frac{\zeta^2_{\mu,i,u,b}}{(1-\mu_{i,u,b})\mu_{i,u,b}}) + \sum_{u\in\mathcal{V}}\frac{\zeta^2_{\sigma,u}(p_u^{D,\boldsymbol{k}})^2}{(1-\sigma_u)\sigma_u}}\sqrt{\sum_{u\in\mathcal{V}}\sigma'^2_u + (1-\sigma_u)\sigma_u} + \\
&\left(\sum_{u\in\mathcal{V}}\sum_{i\in[m],b=k_i}\left|\eta_{\mu,i,u,b}\right| + \sum_{u\in\mathcal{V}}|\eta_{\sigma,u}|p_u^{D,\boldsymbol{k}}\right) && \text{(G.6)}\\
&\leq\sqrt{\sum_{u\in\mathcal{V}}\sum_{i\in[m],b=k_i}(\frac{\zeta^2_{\mu,i,u,b}}{(1-\mu_{i,u,b})\mu_{i,u,b}}) + \sum_{u\in\mathcal{V}}\frac{\zeta^2_{\sigma,u}(p_u^{D,\boldsymbol{k}})^2}{(1-\sigma_u)\sigma_u}}\cdot\sqrt{1.25|\mathcal{V}|} \\
&+\left(\sum_{u\in\mathcal{V}}\sum_{i\in[m],b=k_i}\left|\eta_{\mu,i,u,b}\right| + \sum_{u\in\mathcal{V}}|\eta_{\sigma,u}|p_u^{D,\boldsymbol{k}}\right). && \text{(G.7)}
\end{aligned}$$

where Eq. (G.6) is due to the support of $\mu_{i,u,b}$ are bounded with $[0,1]$ for any $i\in[m], u\in\mathcal{V}$ and Eq. (G.7) is due to the support of σ_u,σ'_u are bounded with $[0,1]$ for any $u\in\mathcal{V}$. □

G.2. Regret proof analysis

In this section, we provide detailed proofs for Theorem 4 under the more general case, i.e., random node weight, where we present an alternative, more precise proof for the main regret term compared to Liu et al. [28].

Let $\mathcal{A}$ denote the set that contains all base arms for visiting probabilities, i.e., $\mathcal{A} = \{(i,u,b)|i \in [m] \cup \mathcal{V}, u \in \mathcal{V}, b \in [c_i]\}$. With some abuse of notation, we also use $\mathcal{V}$ to denote the set of base arms corresponding to the node weights. Therefore, $\mathcal{A} \cup \mathcal{V}$ is the set of all base arms. Following the notations of Wang and Chen [17], we define $X_a \in [0,1]$ be the random outcome of base arm $a \in \mathcal{A} \cup \mathcal{V}$ and define D as the unknown joint distribution of random outcomes $\boldsymbol{X} = (X_a)_{a \in \mathcal{A} \cup \mathcal{V}}$ with unknown mean $\mu_a = \mathbb{E}_{\boldsymbol{X} \sim D}[X_a]$. Let $\boldsymbol{k}$ be any feasible budget allocation, we denote $p_a^{D,\boldsymbol{k}}$ the probability that budget allocation $\boldsymbol{k}$ triggers arm a (i.e. outcome X_a is observed) when the unknown environment distribution is D. Specifically, $p_a^{D,\boldsymbol{k}} = 1$ for $a \in \{(i,u,b)|i \in [m], u \in \mathcal{V}, b = k_i\}$, $p_a^{D,\boldsymbol{k}} = 1 - \prod_{i \in [m]}(1 - \mu_{i,a,k_i})$ for $a \in \mathcal{V}$ and $p_a^{D,\boldsymbol{k}} = 0$ for the rest of base arms. Let $|\mathcal{A} \cup \mathcal{V}| = Bm|\mathcal{V}| + |\mathcal{V}|$ be the number of base arms and $\lceil x \rceil_0 = \max\{\lceil x \rceil, 0\}$. For any feasible budget allocation $\boldsymbol{k}$ and any base arm $a \in \mathcal{A} \cup \mathcal{V}$, the gap $\Delta_{\boldsymbol{k}} = \max\{0, \xi r_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\boldsymbol{k}^*) - r_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\boldsymbol{k})\}$ and we define $\Delta_{\min}^a = \inf_{\boldsymbol{k}: p_a^{D,\boldsymbol{k}} > 0, \Delta_{\boldsymbol{k}} > 0} \Delta_{\boldsymbol{k}}$, $\Delta_{\max}^a = \sup_{\boldsymbol{k}: p_a^{D,\boldsymbol{k}} > 0, \Delta_{\boldsymbol{k}} > 0} \Delta_{\boldsymbol{k}}$. By convention, if there is no action $\boldsymbol{k}$ such that $p_a^{D,\boldsymbol{k}} > 0$ and $\Delta_{\boldsymbol{k}} > 0$, we define $\Delta_{\min}^a = +\infty$, $\Delta_{\max}^a = 0$. We define $\Delta^{\min} = \min_{a \in \mathcal{A} \cup \mathcal{V}} \Delta_{\min}^a$ and $\Delta^{\max} = \max_{a \in \mathcal{A} \cup \mathcal{V}} \Delta_{\max}^a$. Denote $S = \{(i,u,b) \in \mathcal{A} \mid b = k_i\}$, and let $\tilde{S}$ be the set of base arms that can be triggered by S, i.e., $\{a \in \mathcal{A} \cup \mathcal{V} \mid p_a^{D,\boldsymbol{k}} > 0\}$ under random node weights. The triggered arm set τ is extended to $\{(i,u,b) \in \mathcal{A} | k_i = b\} \cup \mathcal{V}$ here.

For fixed node weights, this can be considered a special case where $p_a^{D,\boldsymbol{k}} = 1$ if $a \in \tilde{S}$, and $p_a^{D,\boldsymbol{k}} = 0$ otherwise. Similarly to Section F.3, the set of base arms simplifies to $\mathcal{A} = \{(i,u,b) \mid i \in [m], u \in \mathcal{V}, b \in [c_i]\}$ under the fixed node weight setting. Consequently, due to the complexity of the proof for Theorem 4, which requires extensive detail, we will instead prove the more general and complex scenario with random node weights.

G.2.1. *Useful concentration bounds, definitions and inequalities*

We use the following tail bound for the construction of the confidence radius and our analysis.

Lemma 13 (Empirical Bernstein Inequality [38]). *Let $(X_a)_{a \in \mathcal{A} \cup \mathcal{V}}$ be n i.i.d random variables with bounded support $[0,1]$ and mean $\mathbb{E}[X_a] = \mu$. Let $\hat{X}_n \triangleq \frac{1}{n}\sum_{a \in \mathcal{A} \cup \mathcal{V}} X_a$ and $\hat{V}_n \triangleq \frac{1}{n}\sum_{i \in [n]}(X_a - \hat{X}_n)^2$ be the empirical mean and empirical variance of $(X_a)_{a \in \mathcal{A} \cup \mathcal{V}}$. Then for any $n \in \mathbb{N}$ and $y > 0$, it holds that*

$$\Pr\left[|\hat{X}_n - \mu| \geq \sqrt{\frac{2\hat{V}_n y}{n}} + \frac{3y}{n}\right] \leq 3e^{-y} \tag{G.8}$$

We use the following Bernstein Inequality to bound the difference between the empirical variance and the true variance.

Lemma 14 (Bernstein Inequality [37]). *Let $(X_a)_{a \in \mathcal{A} \cup \mathcal{V}}$ be n independent random variables in $[0,1]$ with mean $\mathbb{E}[X_a] = \mu$ and variance $Var[X_a] \triangleq \mathbb{E}[X_i^2] - (\mathbb{E}[X_a])^2 = V$. Then with probability $1 - \delta$:*

$$\frac{1}{n}\sum_{a \in \mathcal{A} \cup \mathcal{V}} X_a \leq \mu + \frac{2\log 1/\delta}{3n} + \sqrt{\frac{2V \log 1/\delta}{n}}. \tag{G.9}$$

Following a similar approach to Wang and Chen [17], we define event-filtered regret to aid our analysis. However, since our proof does not require the definitions related to the triggering group analysis technique, we simplify the mathematical definitions accordingly.

Definition 3 (Event-Filtered Regret). For any series of events $(\mathcal{E}_t)_{t \geq [T]}$ indexed by round number t, we define the $Reg_{\xi,\boldsymbol{\mu},\boldsymbol{\sigma}}(T, (\mathcal{E}_t)_{t \geq [T]})$ as the regret filtered by events $(\mathcal{E}_t)_{t \geq [T]}$, or the regret is only counted in t if $\mathcal{E}$ happens in t. Formally,

$$Reg_{\xi,\boldsymbol{\mu},\boldsymbol{\sigma}}(T, (\mathcal{E}_t)_{t \geq [T]}) = \mathbb{E}\left[\sum_{t \in [T]} \mathbb{I}(\mathcal{E}_t)(\xi r_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\boldsymbol{k}^*) - r_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\boldsymbol{k}_t)\right]. \tag{G.10}$$

We will omit ξ, T and rewrite $Reg_{\xi,\boldsymbol{\mu},\boldsymbol{\sigma}}(T, (\mathcal{E}_t)_{t \geq [T]})$ as $Reg_{\boldsymbol{\mu},\boldsymbol{\sigma}}(T, \mathcal{E}_t)$ when contexts are clear for simplicity.

Definition 4. We say that the sampling is nice at the beginning of round t if: (1) for every base arm $a \in \mathcal{A} \cup \mathcal{V}$, $|\hat{\mu}_{a,t-1} - \mu_a| \leq \rho_{t,a}$, where $\rho_{t,a} = \sqrt{\frac{6\hat{V}_{a,t-1}\log t}{T_{a,t-1}}} + \frac{9\log t}{T_{a,t-1}}$; (2) for every base arm $a \in \mathcal{A} \cup \mathcal{V}$, $\hat{V}_{a,t-1} \leq 2\mu_a(1-\mu_a) + \frac{3.5\log t}{T_{a,t-1}}$. We denote such event as $\mathcal{N}_t^s$.

The following lemma bounds the probability that $\mathcal{N}_t^s$ does not happen.

Lemma 15. *For each round t, $\Pr[\neg\mathcal{N}_t^s] \leq 4|\mathcal{A} \cup \mathcal{V}|t^{-2}$.*

Proof. Let $\mathcal{N}_t^{s,1}, \mathcal{N}_t^{s,2}$ be the event (1) and event (2), where $\mathcal{N}_t^s = \mathcal{N}_t^{s,1} \bigcap \mathcal{N}_t^{s,2}$. We first bound the probability that $\mathcal{N}_t^{s,1}$ does not happen, we have

$$\Pr[\neg\mathcal{N}_t^{s,1}] = \Pr\left[\exists a \in \mathcal{A} \cup \mathcal{V} \text{ s.t. } |\hat{\mu}_{a,t-1} - \mu_a| > \sqrt{\frac{6\hat{V}_{a,t-1}\log t}{T_{a,t-1}}} + \frac{9\log t}{T_{a,t-1}}\right] \tag{G.11}$$

$$\leq \sum_{a \in \mathcal{A} \cup \mathcal{V}} \sum_{\tau \in [t]} \Pr\left[|\hat{\mu}_{a,t-1} - \mu_a| > \sqrt{\frac{6\hat{V}_{a,t-1}\log t}{\tau}} + \frac{9\log t}{\tau}, T_{a,t-1} = \tau\right] \tag{G.12}$$

$$\leq 3|\mathcal{A} \cup \mathcal{V}|t^{-2}, \tag{G.13}$$

where Eq. (G.12) is due to the union bound over i, τ, Eq. (G.13) is due to Lemma 13 by setting $y = 3\log t$ and when $T_{a,t-1} = \tau$, $\hat{\mu}_{a,t-1}$ and $\hat{V}_{a,t-1}$ are the empirical mean and empirical variance of τ i.i.d random variables with mean μ_a.

We then bound the probability that the second event $\mathcal{N}_t^{s,2}$ does not happen using the similar proof of [39, Eq. (7)]. Fix $T_{a,t-1} = \tau$ and consider $(Y_a^1, \dots, Y_a^\tau)$, where $Y_a^k = (X_a^k - \mu_a)^2 \in [0,1]$ and X_a^k is the random outcome of the k-th i.i.d trial. Since X_a^k are independent across k, Y_a^k are independent across k as well. In this case, one can verify that $\hat{V}_{a,t-1} = \frac{1}{\tau}\sum_{k=1}^{\tau}(X_a^k - \mu_a)^2 - (\frac{1}{\tau}\sum_{k=1}^{\tau} X_a^k - \mu_a)^2 \le \frac{1}{\tau}\sum_{k=1}^{\tau}(X_a^k - \mu_a)^2 = \frac{1}{\tau}\sum_{k=1}^{\tau}\sigma_i^k$; $\mathbb{E}[Y_a^k] = \mathbb{E}[(X_a^k)^2] - \mu_a^2 \le \mathbb{E}[X_a^k]\cdot 1 - \mu_a^2 = (1-\mu_a)\mu_a$; and $\mathrm{Var}[Y_a] = \mathbb{E}[(Y_a^k)^2] - (\mathbb{E}[Y_a^k])^2 \le \mathbb{E}[(Y_a^k)^2] \le \mathbb{E}[Y_a^k] \le (1-\mu_a)\mu_a$. By Lemma 14 over τ i.i.d random variable $(Y_a^k)_{k\in\tau}$, it holds with probability at least $1 - t^{-3}$ that

$$\frac{1}{\tau}\sum_{k=1}^{\tau} Y_a^k \le \mathbb{E}[Y_a^k] + \frac{2\log t}{\tau} + \sqrt{\frac{6\mathrm{Var}[Y_a^k]\log t}{\tau}} \tag{G.14}$$

This implies

$$\hat{V}_{a,t-1} \le \frac{1}{\tau}\sum_{k=1}^{\tau} Y_a^k \le \mathbb{E}[Y_a^k] + \frac{2\log t}{\tau} + \sqrt{\frac{6\mathrm{Var}[Y_a^k]\log t}{\tau}} \tag{G.15}$$

$$\le \mu_a(1-\mu_a) + \frac{2\log t}{\tau} + \sqrt{\frac{6(1-\mu_a)\mu_a\log t}{\tau}} \tag{G.16}$$

$$\le \mu_a(1-\mu_a) + \frac{2\log t}{\tau} + \mu_a(1-\mu_a) + \frac{3\log t}{2\tau} \tag{G.17}$$

$$= 2\mu_a(1-\mu_a) + \frac{3.5\log t}{\tau} \tag{G.18}$$

where Eq. (G.16) is using $2xy \le x^2 + y^2$ and $x = \sqrt{2\mu_a(1-\mu_a)}, y = \sqrt{\frac{3\log t}{n}}$.

Now by applying union bound over $a \in \mathcal{A}\cup\mathcal{V}$ and $\tau \in [t]$, we have $\Pr[\neg\mathcal{N}_t^{s,2}] \le |\mathcal{A}\cup\mathcal{V}|t^{-2}$. Lastly, applying union bound over $\mathcal{N}_t^{s,1}$ and $\mathcal{N}_t^{s,2}$, we have $\Pr[\neg\mathcal{N}_t^{s}] \le 4|\mathcal{A}\cup\mathcal{V}|t^{-2}$. □

After setting up all above definitions, we can prove Lemma 16 about the confidence radius, which appears in the main content.

Lemma 16. *Fix every base arm $(i,u,b) \in \mathcal{A}\cup\mathcal{V}$ and every time t, with probability at least $1 - 4|\mathcal{A}\cup\mathcal{V}|t^{-3}$, it holds that*

$$\mu_{i,u,b} \le \min\{\hat{\mu}_{i,u,b,t-1} + \rho_{i,u,b,t}, 1\} =: \tilde{u}_{i,u,b,t} \le \min\{\mu_{i,u,b} + 2\rho_{i,u,b,t}, 1\}$$
$$\le \min\left\{\mu_{i,u,b} + 4\sqrt{3}\sqrt{\frac{\mu_{i,u,b}(1-\mu_{i,u,b})\log t}{T_{i,u,b,t-1}}} + \frac{28\log t}{T_{i,u,b,t-1}}, 1\right\}. \tag{G.19}$$

Proof. Recall that $\min\{\hat{\mu}_{i,u,b,t-1} + \rho_{i,u,b,t}, 1\} = \min\{\hat{\mu}_{i,u,b,t-1} + \sqrt{\frac{6\hat{V}_{i,u,b,t-1}\log t}{T_{i,u,b,t-1}}} + \frac{9\log t}{T_{i,u,b,t-1}}, 1\}$. Under event $N_t^{s,1}$, we have $|\mu_{i,u,b} - \hat{\mu}_{i,u,b,t}| \le \rho_{i,u,b,t}$ by Lemma 15, hence the first and second inequality in Lemma 16 holds.

For the last inequality, under event $N_t^{s,2}$, it holds that

$$\mu_{i,u,b} + 2\rho_{i,u,b,t} = \mu_{i,u,b} + 2\left(\sqrt{\frac{6\hat{V}_{i,u,b,t-1}\log t}{T_{i,u,b,t-1}}} + \frac{9\log t}{T_{i,u,b,t-1}}\right) \tag{G.20}$$

$$\le \mu_{i,u,b} + 2\left(\sqrt{\frac{6\cdot(2\mu_{i,u,b}(1-\mu_{i,u,b}) + \frac{3.5\log t}{T_{i,u,b,t-1}})\log t}{T_{i,u,b,t-1}}} + \frac{9\log t}{T_{i,u,b,t-1}}\right) \tag{G.21}$$

$$\le \mu_{i,u,b} + 4\sqrt{3}\sqrt{\frac{\mu_{i,u,b}(1-\mu_{i,u,b})\log t}{T_{i,u,b,t-1}}} + 2\sqrt{21}\frac{\log t}{T_{i,u,b,t-1}} + \frac{18\log t}{T_{i,u,b,t-1}} \tag{G.22}$$

$$\le \mu_{i,u,b,t-1} + 4\sqrt{3}\sqrt{\frac{\mu_{i,u,b}(1-\mu_{i,u,b})\log t}{T_{i,u,b,t-1}}} + \frac{28\log t}{T_{i,u,b,t-1}}, \tag{G.23}$$

where Eq. (G.22) uses $\sqrt{x+y} \le \sqrt{x} + \sqrt{y}$.

Since $\mathcal{N}_t^s = \mathcal{N}_t^{s,1} \bigcap \mathcal{N}_t^{s,2}$ and by Lemma 15, Eq. (G.19) holds with probability at least $1 - 4|\mathcal{A}\cup\mathcal{V}|t^{-2}$. □

G.2.2. Decompose the total regret to event-filtered regrets

We firstly decompose the regret $Reg_{\boldsymbol{\mu},\boldsymbol{\sigma}}(T,\{\}) = Reg_{\boldsymbol{\mu},\boldsymbol{\sigma}}(T,\mathcal{N}_t^s,\mathcal{N}_t^o) + Reg_{\boldsymbol{\mu},\boldsymbol{\sigma}}(T,\neg(\mathcal{N}_t^s\bigcap\mathcal{N}_t^o)) \le Reg_{\boldsymbol{\mu},\boldsymbol{\sigma}}(T,\mathcal{N}_t^s,\mathcal{N}_t^o) + Reg_{\boldsymbol{\mu},\boldsymbol{\sigma}}(T,\neg\mathcal{N}_t^s) + Reg_{\boldsymbol{\mu},\boldsymbol{\sigma}}(T,\neg\mathcal{N}_t^o)$, where $\mathcal{N}_t^s$ is defined in Lemma 4, $\mathcal{N}_t^o$ denotes the event where oracle successfully outputs an ξ-approximate solution (with probability at least β). We have the following lemma to do the decomposition.

Lemma 17 (Leading Regret Term). *Define the error term*

$$e_t(\boldsymbol{k}_t) = 4\sqrt{3.75|\mathcal{V}|}\sqrt{\sum_{a\in\tilde{S}_t}(\frac{\log t}{T_{a,t-1}}\wedge\frac{1}{28})(p_a^{D,\boldsymbol{k}_t})^2} + 28\sum_{a\in\tilde{S}_t}(\frac{\log t}{T_{a,t-1}}\wedge\frac{1}{28})(p_a^{D,S_t}) \tag{G.24}$$

and event $E_t = \mathbb{I}\{\Delta_{\boldsymbol{k}_t} \le e_t(\boldsymbol{k}_t)\}$. *The regret of Algorithm 4, when used with* $(1-e^{-\eta},1)$ *approximation oracle is bounded by*

$$Reg_{\boldsymbol{\mu},\boldsymbol{\sigma}}(T) \le Reg_{\boldsymbol{\mu},\boldsymbol{\sigma}}(T,E_t) + \frac{2\pi^2}{3}|\mathcal{A}\cup\mathcal{V}|\Delta_{\max}. \tag{G.25}$$

Proof. Recall that $\boldsymbol{k}_t = (k_{t,1},\dots,k_{t,m})$, $S_t = \{(i,u,b)\in\mathcal{A} \mid b = k_{t,i}\}$, and $\tilde{S}_t$ represents the base arms that can be triggered by S_t under the random node weight setting. Under event $\mathcal{N}_t^s, \mathcal{N}_t^o$, by Lemma 16, it is easily to check that

$$\begin{aligned}
\min\{\mu_{a,t-1}+\rho_{a,t},1\} &\le \bar{\mu}_{a,t-1}\\
&\le \min\{\mu_{a,t-1}+4\sqrt{3}\sqrt{\frac{\mu_a(1-\mu_a)\log t}{T_{a,t-1}}}+\frac{28\log t}{T_{a,t-1}},1\}\\
&\le \mu_{a,t-1}+4\sqrt{3}\sqrt{\mu_a(1-\mu_a)(\frac{\log t}{T_{a,t-1}}\wedge\frac{1}{28})}\\
&\quad+28(\frac{\log t}{T_{a,t-1}}\wedge\frac{1}{28})
\end{aligned} \tag{G.26}$$

Therefore, it holds that

$$\xi r_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\boldsymbol{k}^*) \le \xi r_{\bar{\boldsymbol{\mu}}_t,\bar{\boldsymbol{\sigma}}}(\boldsymbol{k}^*) \le r_{\bar{\boldsymbol{\mu}}_t,\bar{\boldsymbol{\sigma}}}(\boldsymbol{k}_t) \tag{G.27}$$

$$\begin{aligned}
&\le r_{\boldsymbol{\mu}_t,\boldsymbol{\sigma}}(\boldsymbol{k}_t) + 4\sqrt{3.75|\mathcal{V}|}\sqrt{\sum_{a\in\tilde{S}_t}(\frac{\log t}{T_{a,t-1}}\wedge\frac{1}{28})(p_a^{D,\boldsymbol{k}_t})^2}\\
&\quad+28\sum_{a\in\tilde{S}_t}(\frac{\log t}{T_{a,t-1}}\wedge\frac{1}{28})p_a^{D,\boldsymbol{k}_t},
\end{aligned} \tag{G.28}$$

where the first inequality in Eq. (G.27) is due to monotonicity (Property 1), and second inequality in Eq. (G.27) is due to event $\mathcal{N}_t^o$, Eq. (G.28) is because of Eq. (G.26) and Property 4 by plugging in $\zeta_a = 4\sqrt{3}\sqrt{\mu_a(1-\mu_a)(\frac{\log t}{T_{a,t-1}}\wedge\frac{1}{28})}$ and $\eta_a = 28(\frac{\log t}{T_{a,t-1}}\wedge\frac{1}{28})$.

So $Reg_{\boldsymbol{\mu},\boldsymbol{\sigma}}(T,\mathcal{N}_t^s,\mathcal{N}_t^o) \le Reg_{\boldsymbol{\mu},\boldsymbol{\sigma}}(T,E_t)$. Now for $Reg_{\boldsymbol{\mu},\boldsymbol{\sigma}}(T,\neg\mathcal{N}_t^s)$, by Lemma 15 it holds that

$$Reg_{\boldsymbol{\mu},\boldsymbol{\sigma}}(T,\neg\mathcal{N}_t^s) \le \sum_{t=1}^{T}\Pr[\neg\mathcal{N}_t^s] \le \sum_{t=1}^{T}4|\mathcal{A}\cup\mathcal{V}|t^{-2} \le \frac{2\pi^2}{3}|\mathcal{A}\cup\mathcal{V}|\Delta_{\max}. \tag{G.29}$$

Similarly by definition, it holds that

$$Reg_{\boldsymbol{\mu},\boldsymbol{\sigma}}(T,\neg\mathcal{N}_t^o) \le (1-\beta)T\Delta_{\max}. \tag{G.30}$$

Therefore $Reg_{\boldsymbol{\mu},\boldsymbol{\sigma}}(T,\{\}) \le Reg_{\boldsymbol{\mu},\boldsymbol{\sigma}}(T,E_t) + \frac{2\pi^2}{3}|\mathcal{A}\cup\mathcal{V}|\Delta_{\max} + (1-\beta)T\Delta_{\max}$. And we have $Reg_{\boldsymbol{\mu},\boldsymbol{\sigma}}(T) = Reg_{\boldsymbol{\mu},\boldsymbol{\sigma}}(T,\{\}) - (1-\beta)T\Delta_{\max} \le Reg_{\boldsymbol{\mu},\boldsymbol{\sigma}}(T,E_t) + \frac{2\pi^2}{3}\Delta_{\max}$, which concludes Lemma 17. □

Recall event $E_t = \{\Delta_{\boldsymbol{k}_t} \le e_t(\boldsymbol{k}_t)\}$, where $e_t(\boldsymbol{k}_t) = 28\sum_{a\in\tilde{S}_t}(\frac{\log t}{T_{a,t-1}}\wedge\frac{1}{28})(p_a^{D,\boldsymbol{k}_t}) + 4\sqrt{3.75|\mathcal{V}|}\sqrt{\sum_{a\in\tilde{S}_t}(\frac{\log t}{T_{a,t-1}}\wedge\frac{1}{28})(p_a^{D,\boldsymbol{k}_t})^2}$. We will further decompose the event-filtered regret $Reg_{\boldsymbol{\mu},\boldsymbol{\sigma}}(T,E_t)$ into two event-filtered regret $Reg_{\boldsymbol{\mu},\boldsymbol{\sigma}}(T,E_{t,1})$ and $Reg_{\boldsymbol{\mu},\boldsymbol{\sigma}}(T,E_{t,2})$,

$$Reg_{\boldsymbol{\mu},\boldsymbol{\sigma}}(T,E_t) \le Reg_{\boldsymbol{\mu},\boldsymbol{\sigma}}(T,E_{t,1}) + Reg_{\boldsymbol{\mu},\boldsymbol{\sigma}}(T,E_{t,2}), \tag{G.31}$$

where event $E_{t,1} = \{\Delta_{\boldsymbol{k}_t} \le 2e_{t,1}(\boldsymbol{k}_t)\}$, event $E_{t,2} = \{\Delta_{\boldsymbol{k}_t} \le 2e_{t,2}(\boldsymbol{k}_t)\}$, $e_{t,1}(\boldsymbol{k}_t) = 4\sqrt{3.75|\mathcal{V}|}\cdot\sqrt{\sum_{a\in\tilde{S}_t}(\frac{\log t}{T_{a,t-1}}\wedge\frac{1}{28})(p_a^{D,\boldsymbol{k}_t})^2}$, $e_{t,2}(\boldsymbol{k}_t) = 28\sum_{a\in\tilde{S}_t}(\frac{\log t}{T_{a,t-1}}\wedge\frac{1}{28})(p_a^{D,\boldsymbol{k}_t})$. The above inequality holds since the following facts: We can observe $e_{t,1}(\boldsymbol{k}_t) + e_{t,2}(\boldsymbol{k}_t) = e_t(\boldsymbol{k}_t)$. From E_t, we know either $E_{t,1}$ holds or $E_{t,2}$ holds. So E_t implies $1 \le \mathbb{I}\{E_{t,1}\} + \mathbb{I}\{E_{t,2}\}$, and thus $\Delta_{\boldsymbol{k}_t}\mathbb{I}\{E_t\} \le \Delta_{\boldsymbol{k}_t}\mathbb{I}\{E_{t,1}\} + \Delta_{\boldsymbol{k}_t}\mathbb{I}\{E_{t,2}\}$, which concludes $Reg_{\boldsymbol{\mu},\boldsymbol{\sigma}}(T,E_t) \le Reg_{\boldsymbol{\mu},\boldsymbol{\sigma}}(T,E_{t,1}) + Reg_{\boldsymbol{\mu},\boldsymbol{\sigma}}(T,E_{t,2})$. The next section will provide two proofs for $Reg_{\boldsymbol{\mu},\boldsymbol{\sigma}}(T,E_{t,1})$, and $Reg_{\boldsymbol{\mu},\boldsymbol{\sigma}}(T,E_{t,2})$, respectively.

G.3. Refined regret bound analysis

In this section, we are going to bound the $Reg_{\boldsymbol{\mu},\boldsymbol{\sigma}}(T,E_{t,1})$ and $Reg_{\boldsymbol{\mu},\boldsymbol{\sigma}}(T,E_{t,2})$ separately under the event $\mathcal{N}_t^t$. The approach leverages a refined reverse amortization technique from Wang and Chen [17], Liu et al. [28] to assign regret $\Delta_{\boldsymbol{k}_t}$ to each base arm based on carefully designed thresholds. Deriving the correct thresholds and regret allocation strategy is highly non-trivial, as it aims to minimize the factors m, $|\mathcal{V}|$, and T for the `MuLaNE` problem. In contrast to Liu et al. [28], we have eliminated a logarithmic factor related to the number of explorers from the second main term in their regret upper bound. Finally, we offer discussions on the distribution-independent regret bounds.

G.3.1. Bounding the regret on the first term

Let $c_1 = 4\sqrt{3}$, and $\tilde{\Delta}_{\boldsymbol{k}_t} = \Delta_{\boldsymbol{k}_t}/p_a^{D,\boldsymbol{k}_t}$. Given filtration $\mathcal{F}_{t-1}$ and under event $E_{t,1}$, we have

$$\Delta_{\boldsymbol{k}_t} \le \sum_{a\in\mathcal{A}\cup\mathcal{V}} \frac{5c_1^2|\mathcal{V}|(p_a^{D,\boldsymbol{k}_t})^2\frac{\log t}{T_{a,t-1}}}{\Delta_{\boldsymbol{k}_t}} \tag{G.32}$$

$$= -\Delta_{\boldsymbol{k}_t} + 2\sum_{a\in\mathcal{A}\cup\mathcal{V}} \frac{5c_1^2|\mathcal{V}|(p_a^{D,\boldsymbol{k}_t})^2\frac{\log t}{T_{a,t-1}}}{\Delta_{\boldsymbol{k}_t}} \tag{G.33}$$

$$= -\frac{\sum_{a\in\tilde{S}_t} p_a^{D,\boldsymbol{k}_t}\Delta_{\boldsymbol{k}_t}/p_a^{D,\boldsymbol{k}_t}}{m|\mathcal{V}|+|\mathcal{V}|} + 2\sum_{a\in\mathcal{A}\cup\mathcal{V}} \frac{5c_1^2|\mathcal{V}|(p_a^{D,\boldsymbol{k}_t})^2\frac{\log t}{T_{a,t-1}}}{\Delta_{\boldsymbol{k}_t}} \tag{G.34}$$

$$\le \sum_{a\in\mathcal{A}\cup\mathcal{V}} p_a^{D,\boldsymbol{k}_t}\left(\frac{10c_1^2|\mathcal{V}|\frac{\log t}{T_{a,t-1}}}{\Delta_{\boldsymbol{k}_t}/p_a^{D,\boldsymbol{k}_t}} - \frac{\Delta_{\boldsymbol{k}_t}/p_a^{D,\boldsymbol{k}_t}}{m|\mathcal{V}|+|\mathcal{V}|}\right) \tag{G.35}$$

$$= \sum_{a\in\mathcal{A}\cup\mathcal{V}} p_a^{D,\boldsymbol{k}_t}\left(\frac{10c_1^2|\mathcal{V}|\frac{\log t}{T_{a,t-1}}}{\tilde{\Delta}_{\boldsymbol{k}_t}} - \frac{\tilde{\Delta}_{\boldsymbol{k}_t}}{m|\mathcal{V}|+|\mathcal{V}|}\right). \tag{G.36}$$

$$\Delta_{\boldsymbol{k}_t} = \mathbb{E}_t[\Delta_{\boldsymbol{k}_t}] \le \mathbb{E}_t\left[\sum_{a\in\mathcal{A}\cup\mathcal{V}} p_a^{D,\boldsymbol{k}_t}\left(\frac{10c_1^2|\mathcal{V}|\frac{\log t}{T_{a,t-1}}}{\tilde{\Delta}_{\boldsymbol{k}_t}} - \frac{\tilde{\Delta}_{\boldsymbol{k}_t}}{m|\mathcal{V}|+|\mathcal{V}|}\right)\right] \tag{G.37}$$

$$= \mathbb{E}_t\left[\sum_{a\in\tau_t}\left(\frac{10c_1^2|\mathcal{V}|\frac{\log t}{T_{a,t-1}}}{\tilde{\Delta}_{\boldsymbol{k}_t}} - \frac{\tilde{\Delta}_{\boldsymbol{k}_t}}{m|\mathcal{V}|+|\mathcal{V}|}\right)\right] \tag{G.38}$$

$$\le \mathbb{E}_t\left[\sum_{a\in\tau_t}\kappa_{a,T}(T_{a,t-1})\right]. \tag{G.39}$$

Regarding the last inequality, we define a regret allocation function

$$\kappa_{a,T}(\ell) = \begin{cases} \frac{1.25c_1^2|\mathcal{V}|}{\tilde{\Delta}^a_{\min}}, & \text{if } \ell = 0, \\ 2\sqrt{\frac{5c_1^2|\mathcal{V}|\log T}{\ell}}, & \text{if } 1 \le \ell \le L_{a,T,1}, \\ \frac{10c_1^2|\mathcal{V}|\log T}{\tilde{\Delta}^a_{\min}}\frac{1}{\ell}, & \text{if } L_{a,T,1} < \ell \le L_{a,T,2}, \\ 0, & \text{if } \ell > L_{a,T,2}, \end{cases} \tag{G.40}$$

where $L_{a,T,1} = \frac{5c_1^2|\mathcal{V}|\log T}{(\tilde{\Delta}^a_{\min})^2}$, $L_{a,T,2} = \frac{10c_1^2|\mathcal{V}|(m|\mathcal{V}|+|\mathcal{V}|)\log T}{(\tilde{\Delta}^a_{\min})^2}$, and $\tilde{\Delta}^a_{\min} = \min_{\boldsymbol{k}:p_a^{D,\boldsymbol{k}}>0,\Delta_{\boldsymbol{k}}>0}\Delta_{\boldsymbol{k}}/p_a^{D,\boldsymbol{k}}$. Then, the reason why Eq. (G.39) holds is proved as follows.

Proof.

1. **When** $T_{a,t-1} > L_{a,T,2} = \frac{10c_1^2|\mathcal{V}|(m|\mathcal{V}|+|\mathcal{V}|)\log T}{(\tilde{\Delta}^a_{\min})^2}$,
   we have $(G.39, a) \le \frac{10c_1^2|\mathcal{V}|\log T}{T_{a,t-1}\cdot\tilde{\Delta}_{\boldsymbol{k}_t}} - \frac{\tilde{\Delta}_{\boldsymbol{k}_t}}{m|\mathcal{V}|+|\mathcal{V}|} < \frac{(\tilde{\Delta}^a_{\min})^2}{(m|\mathcal{V}|+|\mathcal{V}|)\tilde{\Delta}_{\boldsymbol{k}_t}} - \frac{\tilde{\Delta}_{\boldsymbol{k}_t}}{m|\mathcal{V}|+|\mathcal{V}|} \le 0 = \kappa_{a,T}(T_{a,t-1})$.
2. **When** $L_{a,T,1} < T_{a,t-1} \le L_{a,T,2}$,
   we have $(G.39, a) \le \frac{10c_1^2|\mathcal{V}|\log T}{T_{a,t-1}\cdot\tilde{\Delta}_{\boldsymbol{k}_t}} - \frac{\tilde{\Delta}_{\boldsymbol{k}_t}}{m|\mathcal{V}|+|\mathcal{V}|} < \frac{10c_1^2|\mathcal{V}|\log T}{T_{a,t-1}\cdot\tilde{\Delta}_{\boldsymbol{k}_t}} \le \frac{10c_1^2|\mathcal{V}|\log T}{T_{a,t-1}\cdot\tilde{\Delta}^a_{\min}} = \kappa_{a,T}(T_{a,t-1})$.
3. **When** $T_{a,t-1} \le L_{a,T,1}$,
   we further consider two different cases $T_{a,t-1} \le \frac{5c_1^2|\mathcal{V}|\log T}{(\tilde{\Delta}_{\boldsymbol{k}_t})^2}$ or $\frac{5c_1^2|\mathcal{V}|\log T}{(\tilde{\Delta}_{\boldsymbol{k}_t})^2} < T_{a,t-1} \le L_{a,T,1} = \frac{5c_1^2|\mathcal{V}|\log T}{(\tilde{\Delta}^a_{\min})^2}$.
   - For the former case, if there exists $a \in \tau_t$ so that $T_{a,t-1} \le \frac{5c_1^2|\mathcal{V}|\log T}{(\tilde{\Delta}_{\boldsymbol{k}_t})^2}$, then we know $\sum_{q\in\tilde{S}_t}\kappa_{q,T}(T_{t-1,q}) \ge \kappa_{a,T}(T_{a,t-1}) = 2\sqrt{\frac{5c_1^2|\mathcal{V}|\log T}{T_{a,t-1}}} \ge 2\sqrt{(\tilde{\Delta}_{\boldsymbol{k}_t})^2} \ge \Delta_{\boldsymbol{k}_t}$, which makes Eq. (G.39) valid no matter what.
   - For the latter case, when $\frac{5c_1^2|\mathcal{V}|\log T}{(\tilde{\Delta}_{\boldsymbol{k}_t})^2} < T_{a,t-1}$, we know that $(G.39, a) \le \frac{10c_1^2|\mathcal{V}|\log T}{\tilde{\Delta}_{\boldsymbol{k}_t}}\frac{1}{T_{a,t-1}} = 2\sqrt{\frac{5c_1^2|\mathcal{V}|\log T}{(\tilde{\Delta}_{\boldsymbol{k}_t})^2}\frac{1}{T_{a,t-1}}}\sqrt{\frac{5c_1^2|\mathcal{V}|\log T}{T_{a,t-1}}} \le 2\sqrt{\frac{\tilde{\Delta}_{\boldsymbol{k}_t}}{\tilde{\Delta}_{\boldsymbol{k}_t}}}\sqrt{\frac{5c_1^2|\mathcal{V}|\log T}{T_{a,t-1}}} = 2\sqrt{\frac{5c_1^2|\mathcal{V}|\log T}{T_{a,t-1}}} = \kappa_{a,T}(T_{a,t-1})$.

4. **When $\ell = 0$,**

We have $(G.39, a) \le \frac{10c_1^2|\mathcal{V}|}{\tilde{\Delta}_{\boldsymbol{k}_t}} \cdot \frac{1}{28} - \frac{\tilde{\Delta}_{\boldsymbol{k}_t}}{m|\mathcal{V}|+|\mathcal{V}|} \le \frac{1.25c_1^2|\mathcal{V}|}{\tilde{\Delta}_{\boldsymbol{k}_t}} \le \frac{1.25c_1^2|\mathcal{V}|}{\tilde{\Delta}^a_{\min}} = \kappa_{a,T}(T_{a,t-1})$.

Combining all the above cases, we have $\Delta_{\boldsymbol{k}_t} \le \mathbb{E}[\sum_{a\in\tau_t} \kappa_{a,T}(T_{a,t-1})]$. □

Based on Eq. (G.39), we now have the following inequalities on $Reg_{\boldsymbol{\mu},\boldsymbol{\sigma}}(T, E_{t,1})$:

$$Reg_{\boldsymbol{\mu},\boldsymbol{\sigma}}(T, E_{t,1}) = \mathbb{E}\left[\sum_{t=1}^{T} \Delta_{\boldsymbol{k}_t}\right] \tag{G.41}$$

$$\overset{(a)}{\le} \mathbb{E}\left[\sum_{t\in[T]} \mathbb{E}_t\left[\sum_{a\in\tau_t} \kappa_{a,T}(T_{a,t-1})\right]\right] \tag{G.42}$$

$$\overset{(b)}{=} \mathbb{E}\left[\sum_{t\in[T]}\sum_{a\in\tau_t} \kappa_{a,T}(T_{a,t-1})\right] \tag{G.43}$$

$$\overset{(c)}{=} \mathbb{E}\left[\sum_{a\in\mathcal{A}\cup\mathcal{V}}\sum_{s=0}^{T_{a,t-1}} \kappa_{a,T}(s)\right] \tag{G.44}$$

$$\le \sum_{a\in\mathcal{A}\cup\mathcal{V}} \frac{1.25c_1^2|\mathcal{V}|}{\kappa_{a,T}} + \sum_{a\in\mathcal{A}\cup\mathcal{V}}\sum_{s=1}^{L_{a,T,1}} 2\sqrt{\frac{5c_1^2|\mathcal{V}|\log T}{s}} + \sum_{a\in\mathcal{A}\cup\mathcal{V}}\sum_{s=L_{a,T,1}+1}^{L_{a,T,2}} \frac{10c_1^2|\mathcal{V}|\log T}{\kappa_{a,T}}\frac{1}{s} \tag{G.45}$$

$$\le \sum_{a\in\mathcal{A}\cup\mathcal{V}} \frac{1.25c_1^2|\mathcal{V}|}{\tilde{\Delta}^a_{\min}} + \sum_{a\in\mathcal{A}\cup\mathcal{V}}\int_{s=0}^{L_{a,T,1}} 2\sqrt{\frac{5c_1^2|\mathcal{V}|\log T}{s}}ds + \sum_{a\in\mathcal{A}\cup\mathcal{V}}\int_{s=L_{a,T,1}}^{L_{a,T,2}} \frac{10c_1^2|\mathcal{V}|\log T}{\tilde{\Delta}^a_{\min} s}ds \tag{G.46}$$

$$\le \sum_{a\in\mathcal{A}\cup\mathcal{V}} \frac{1.25c_1^2|\mathcal{V}|}{\tilde{\Delta}^a_{\min}} + \sum_{a\in\mathcal{A}\cup\mathcal{V}} \frac{10c_1^2|\mathcal{V}|\log T}{\tilde{\Delta}^a_{\min}}(1+\log(m|\mathcal{V}|+|\mathcal{V}|)) + \sum_{a\in\mathcal{A}\cup\mathcal{V}} \frac{20c_1^2|\mathcal{V}|\log T}{\tilde{\Delta}^a_{\min}}, \tag{G.47}$$

where (a) follows from Eq. (G.39), (b) follows from the tower rule, (c) follows from that $T_{a,t-1}$ is increased by 1 if and only if $a \in \tau_t$.

G.3.2. Bounding the regret on the second term

Let $c_2 = 28$. Given filtration $\mathcal{F}_{t-1}$ and event $E_{t,2}$, we have

$$\Delta_{\boldsymbol{k}_t} \overset{(a)}{\le} \sum_{a\in\bar{S}_t} 2c_2 p_a^{D,\boldsymbol{k}_t} \min\left\{1/28, \frac{\log T}{T_{a,t-1}}\right\}$$

$$\overset{(b)}{\le} -\Delta_{\boldsymbol{k}_t} + 2\sum_{a\in\bar{S}_t} 2c_2 p_a^{D,\boldsymbol{k}_t} \min\left\{1/28, \frac{\log T}{T_{a,t-1}}\right\}$$

$$= -\frac{\sum_{a\in\mathcal{A}\cup\mathcal{V}} p_a^{D,\boldsymbol{k}_t}\Delta_{\boldsymbol{k}_t}}{\sum_{a\in\mathcal{A}\cup\mathcal{V}} p_a^{D,\boldsymbol{k}_t}} + 2\sum_{i\in[m]} 2c_2 p_a^{D,\boldsymbol{k}_t} \min\left\{1/28, \frac{\log T}{T_{a,t-1}}\right\} \tag{G.48}$$

$$\overset{(c)}{\le} \sum_{a\in\mathcal{A}\cup\mathcal{V}} p_a^{D,\boldsymbol{k}_t}\left(-\frac{\Delta_{\boldsymbol{k}_t}}{m|\mathcal{V}|+|\mathcal{V}|} + 4c_2\min\left\{1/28, \frac{\log T}{T_{a,t-1}}\right\}\right), \tag{G.49}$$

where (a) follows from event $E_{t,2}$, (b) is by the reverse amortization trick [17] that multiplies two to both sides of (a) and rearranges the terms, and (c) follows from $p_a^{D,\boldsymbol{k}_t} \le 1$.

It follows that

$$\Delta_{\boldsymbol{k}_t} = \mathbb{E}_t[\Delta_{\boldsymbol{k}_t}] \overset{(a)}{\le} \mathbb{E}_t\left[\sum_{a\in\mathcal{A}\cup\mathcal{V}} p_a^{D,\boldsymbol{k}_t}\left(-\frac{\Delta_{\boldsymbol{k}_t}}{m|\mathcal{V}|+|\mathcal{V}|} + 4c_2\min\left\{1/28, \frac{\log T}{T_{a,t-1}}\right\}\right)\right]$$

$$\overset{(b)}{=} \mathbb{E}_t\left[\sum_{a\in\tau_t}\left(-\frac{\Delta_{\boldsymbol{k}_t}}{m|\mathcal{V}|+|\mathcal{V}|} + 4c_2\min\left\{1/28, \frac{\log T}{T_{a,t-1}}\right\}\right)\right]$$

$$\overset{(c)}{\leq} \mathbb{E}_t\left[\sum_{a\in\tau_t} \kappa_{a,T}(T_{a,t-1})\right], \tag{G.50}$$

Here, the regret allocation function is defined as

$$\kappa_{a,T}(\ell) = \begin{cases} \Delta^a_{\max}, & \text{if } 0 \leq \ell \leq L_{a,T,1} \\ \frac{4c_2 \log T}{\ell}, & \text{if } L_{a,1} < \ell \leq L_{a,2} \\ 0, & \text{if } \ell > L_{a,T,2}+1, \end{cases} \tag{G.51}$$

where $L_{a,T,1} = \frac{4c_2 \log T}{\Delta^a_{\max}}$, $L_{a,T,2} = \frac{4c_2(m|\mathcal{V}|+|\mathcal{V}|)\log T}{\Delta^a_{\min}}$. And (a) follows from Eq. (G.49), (b) from the triggering probability equivalence technique [40] that links the triggering probabilities with the random triggering event under expectation, and (c) is proven similarly to Section G.3.1 as follows.

Proof.

1. **When** $T_{a,t-1} > L_{a,T,2} = \frac{4c_2(m|\mathcal{V}|+|\mathcal{V}|)\log T}{\Delta^a_{\min}}$**,**
   we have $(G.50, i) \leq 4c_2 \frac{\log T}{T_{a,t-1}} - \frac{\Delta_{\boldsymbol{k}_t}}{m|\mathcal{V}|+|\mathcal{V}|} < \frac{\Delta^a_{\min}}{m|\mathcal{V}|+|\mathcal{V}|} - \frac{\Delta_{\boldsymbol{k}_t}}{m|\mathcal{V}|+|\mathcal{V}|} \leq 0 = \kappa_{a,T}(T_{a,t-1})$.
2. **When** $T_{a,t-1} \leq L_{a,T,2}$**,**
   we have $(G.50, i) \leq 4c_2 \frac{\log T}{T_{a,t-1}} - \frac{\Delta_{\boldsymbol{k}_t}}{m|\mathcal{V}|+|\mathcal{V}|} < \frac{4c_2 \log T}{T_{a,t-1}} = \kappa_{a,T}(T_{a,t-1})$.
3. **When** $T_{a,t-1} \leq L_{a,T,1}$**,**
   if there exists $a \in \tilde{S}_t$ so that $T_{a,t-1} \leq L_{a,T,1}$, then we know $\sum_{q\in\tilde{S}_t} \kappa_{q,T}(T_{t-1,q}) \geq \kappa_{a,T}(T_{a,t-1}) = \Delta^a_{\max} \geq \Delta_{\boldsymbol{k}_t}$, which makes Eq. (G.50) holds no matter what. This means we do not need to consider this case for good.
   Combining all above cases, we have $\Delta_{\boldsymbol{k}_t} \leq \mathbb{E}_t[\sum_{a\in\tau_t} \kappa_{a,T}(T_{a,t-1})]$. □

$$Reg_{\boldsymbol{\mu},\boldsymbol{\sigma}}(T, E_{t,2}) = \mathbb{E}\left[\sum_{t=1}^{T} \Delta_{\boldsymbol{k}_t}\right] \tag{G.52}$$

$$\overset{(a)}{\leq} \mathbb{E}\left[\sum_{t\in[T]} \mathbb{E}_t\left[\sum_{a\in\tau_t} \kappa_{a,T}(T_{a,t-1})\right]\right] \tag{G.53}$$

$$\overset{(b)}{=} \mathbb{E}\left[\sum_{t\in[T]}\sum_{a\in\tau_t} \kappa_{a,T}(T_{a,t-1})\right] \tag{G.54}$$

$$\overset{(c)}{=} \mathbb{E}\left[\sum_{a\in\mathcal{A}\cup\mathcal{V}}\sum_{s=0}^{T_{a,t-1}} \kappa_{a,T}(s)\right] \tag{G.55}$$

$$\leq |\mathcal{A}\cup\mathcal{V}|\Delta_{\max} + \sum_{a\in\mathcal{A}\cup\mathcal{V}}\sum_{\ell=1}^{L_{a,T,1}} \Delta^a_{\max} + \sum_{a\in\mathcal{A}\cup\mathcal{V}}\sum_{\ell=L_{a,T,1}+1}^{L_{a,T,2}} \frac{4c_2 \log T}{\ell} \tag{G.56}$$

$$\leq |\mathcal{A}\cup\mathcal{V}|\Delta_{\max} + \sum_{a\in\mathcal{A}\cup\mathcal{V}} 4c_2 \log T + \sum_{a\in\mathcal{A}\cup\mathcal{V}} 4c_2 \log\left(\frac{(m|\mathcal{V}|+|\mathcal{V}|)\Delta^a_{\max}}{\Delta^a_{\min}}\right)\log T \tag{G.57}$$

$$= |\mathcal{A}\cup\mathcal{V}|\Delta_{\max} + \sum_{a\in\mathcal{A}\cup\mathcal{V}} 4c_2\left(1 + \log\left(\frac{(m|\mathcal{V}|+|\mathcal{V}|)\Delta^a_{\max}}{\Delta^a_{\min}}\right)\right)\log T \tag{G.58}$$

$$\leq |\mathcal{A}\cup\mathcal{V}|\Delta_{\max} + \sum_{a\in\mathcal{A}\cup\mathcal{V}} 4c_2\left(1 + \log\left(\frac{(m|\mathcal{V}|+|\mathcal{V}|)\Delta^a_{\max}}{\Delta^a_{\min}}\right)\right)\log T, \tag{G.59}$$

where (a) follows from Eq. (G.50), (b) follows from the tower rule, (c) follows from that $T_{a,t-1}$ is increased by 1 if and only if $a \in \tau_t$.

In the case of fixed node weights, the problem degenerates into one where only edge weights need to be learned, as the node weights remain constant and do not require learning.

G.3.3. *Distribution-independent bound*

We fix a gap Δ, and consider two events for $\Delta_{\boldsymbol{k}_t}$: $\{\Delta_{\boldsymbol{k}_t} \leq \Delta\}$ and $\{\Delta_{\boldsymbol{k}_t} > \Delta\}$. In the former case, the regret is trivially bounded by $Reg(T, \{\Delta_{\boldsymbol{k}_t} \leq \Delta\}) \leq T\Delta$. For the latter case, under $\{\Delta_{\boldsymbol{k}_t} > \Delta\}$, we can easily replace all instances of $\Delta^a_{\min}$ with Δ and derive the bound $Reg(T, \{\Delta_{\boldsymbol{k}_t} > \Delta\}) \leq O\left(\frac{|\mathcal{A}\cup\mathcal{V}||\mathcal{V}|\log(m|\mathcal{V}|+|\mathcal{V}|)\log T}{\Delta} + |\mathcal{A}\cup\mathcal{V}|\log\left(\frac{m|\mathcal{V}|+|\mathcal{V}|}{\Delta}\right)\log T\right)$. Similar to Section F.3, we can also establish a

distribution-independent bound:

$$O\Big(\sqrt{|\mathcal{A}\cup\mathcal{V}||\mathcal{V}|T\log(m|\mathcal{V}|+|\mathcal{V}|)\log T}+|\mathcal{A}\cup\mathcal{V}|\log T\log((m|\mathcal{V}|+|\mathcal{V}|)T)\Big).$$

Appendix H. Online learning for non-overlapping MuLaNE

H.1. Online learning algorithm for non-overlapping MuLaNE

For non-overlapping MuLaNE, we estimate layer-level marginal gains as our parameters, rather than probabilities $P_{i,u}(b)$ and node weights $\boldsymbol{\sigma}$. By doing so, we can largely simply our analysis since we no longer need to handle the unknown weights and the extra constraint of $P_{i,u}(b)$ as in the overlapping setting. Concretely, we maintain a set of base arms $\mathcal{A}=\{(i,b)|i\in[m],b\in[c_i]\}$, and let $|\mathcal{A}|=\sum_{i\in[m]}c_i$ be the total number of these arms. For each base arm $(i,b)\in\mathcal{A}$., denote $\mu_{i,b}$ as the truth value of each base arm, i.e., $\mu_{i,b}=\sum_{u\in\mathcal{V}}\sigma_u(P_{i,u}(b)-P_{i,u}(b-1))$. Based on the definition of base arms, we can see that we do not need to explicitly estimate the node weights $\boldsymbol{\sigma}$ or probabilities $P_{i,u}(b)$, so we will use $r_{\boldsymbol{\mu}}(\boldsymbol{k})$ to denote the reward function $r_{\mathcal{G},\boldsymbol{\alpha},\boldsymbol{\sigma}}(\boldsymbol{k})$.

We present our algorithm in Algorithm 8, which has three differences compared with the Algorithm 3 for overlapping MuLaNE. First, the expected reward of action $\boldsymbol{k}$ is $r_{\boldsymbol{\mu}}(\boldsymbol{k})=\sum_{i=1}^{m}\sum_{b=1}^{k_i}\mu_{i,b}$, which is a linear reward function and satisfies two new Monotonicity and 1-Norm Bounded Smoothness conditions (see Condition 1 and 2). Second, we use the Dynamic Programming method (Algorithm 7) as our Oracle, which does not require the parameters $\mu_{i,b}$ to be increasing. Moreover, the offline oracle always outputs *optimal* solutions. Third, we use a new random variable $Y_{i,b}$, which indicates the gain of weights which depends on whether a new node is visited by random walker from the i-th layer exactly at the b-th step, i.e., $Y_{i,b}=\sum_{u\in\bigcup_{j=1}^{b}\{X_{i,j}\}}\sigma_u-\sum_{u\in\bigcup_{j=1}^{b-1}\{X_{i,j}\}}\sigma_u$ for base arms (i,b) such that $b\le k_i$.

Condition 1 (Monotonicity). *The reward $r_{\boldsymbol{\mu}}(\boldsymbol{k})$ is monotonically increasing, i.e., for any budget allocation $\boldsymbol{k}$ and any two vectors $\boldsymbol{\mu}=(\mu_{i,b})_{(i,b)\in\mathcal{A}}$, $\boldsymbol{\mu'}=(\mu'_{i,b})_{(i,b)\in\mathcal{A}}$, we have $r_{\boldsymbol{\mu}}(\boldsymbol{k})\le r_{\boldsymbol{\mu'}}(\boldsymbol{k})$, if $\mu_{i,b}\le\mu'_{i,b}$, for $(i,b)\in\mathcal{A}$.*

Condition 2 (1-Norm Bounded Smoothness). *The reward function $r_{\boldsymbol{\mu}}(\boldsymbol{k})$ satisfies the 1-norm bounded smoothness, i.e., for any budget allocation $\boldsymbol{k}$ and any two vectors $\boldsymbol{\mu}=(\mu_{i,b})_{(i,b)\in\mathcal{A}}$, $\boldsymbol{\mu'}=(\mu'_{i,b})_{(i,b)\in\mathcal{A}}$, we have $|r_{\boldsymbol{\mu}}(\boldsymbol{k})-r_{\boldsymbol{\mu'}}(\boldsymbol{k})|\le\sum_{b\le k_i}|\mu_{i,b}-\mu'_{i,b}|$.*

We define the reward gap $\Delta_{\boldsymbol{k}}=r_{\boldsymbol{\mu}}(\boldsymbol{k}^*)-r_{\boldsymbol{\mu}}(\boldsymbol{k})$ for all feasible action $\boldsymbol{k}$ satisfying $\sum_{i=1}^{m}k_i=B$, $0\le k_i\le c_i$. For each base arm (i,b), we define $\Delta_{\min}^{i,b}=\min_{\Delta_{\mathbf{k}}>0,k_i\ge b}\Delta_{\mathbf{k}}$ and $\Delta_{\max}^{i,b}=\max_{\Delta_{\mathbf{k}}>0,k_i\ge b}\Delta_{\mathbf{k}}$. As a convention, if there is no action with $k_i\ge b$ such that $\Delta_{\boldsymbol{k}}>0$, we define $\Delta_{\min}^{i,b}=\infty$ and $\Delta_{\max}^{i,b}=0$. Also, we define $\Delta_{\min}=\min_{(i,b)\in\mathcal{A}}\Delta_{\min}^{i,b}$ and $\Delta_{\max}=\max_{(i,b)\in\mathcal{A}}\Delta_{\max}^{i,b}$.

Theorem 9. *CUCB-MG has the following distribution-dependent regret bound,*

$$Reg_{\mu}(T)\le\sum_{(i,b)\in\mathcal{A}}\frac{48B\ln T}{\Delta_{\min}^{i,b}}+2|\mathcal{A}|+\frac{\pi^2}{3}|\mathcal{A}|\Delta_{\max}.$$

Algorithm 8 CUCB-MG: combinatorial upper confidence bound (CUCB) algorithm for non-overlapping MuLaNE, using Marginal Gains as the base arms.

Input: Budget B, number of layers m, offline oracle DP.
For each $(i,b)\in\mathcal{A}$, $T_{i,b}\leftarrow 0$, $\hat{\mu}_{i,b}\leftarrow 1$.
for $t=1,2,3,\dots,T$ **do**
  for $(i,b)\in\mathcal{A}$ **do**
    $\rho_{i,b}\leftarrow\sqrt{3\ln t/(2T_{i,b})}$.
    $\hat{\mu}_{i,b}=\min\{\hat{\mu}_{i,b}+\rho_{i,b},1\}$.
  end for
  $\boldsymbol{k}\leftarrow\mathrm{DP}((\bar{\mu}_{i,b})_{i,b},B,\boldsymbol{c})$
  Apply budget allocation $\boldsymbol{k}$, which gives trajectories $\boldsymbol{X}:=(X_{i,1},\dots,X_{i,k_i})_{i\in[m]}$ as feedbacks.
  For $(i,b)\in\tau:=\{(i,b)\in\mathcal{A}|\,b\le k_i\}$,
  $Y_{i,b}=\sum_{u\in\bigcup_{j=1}^{b}\{X_{i,j}\}}\sigma_u-\sum_{u\in\bigcup_{j=1}^{b-1}\{X_{i,j}\}}\sigma_u$.
  For $(i,b)\in\tau$, update $T_{i,b}$ and $\hat{\mu}_{i,b}$:
  $T_{i,b}=T_{i,b}+1$, $\hat{\mu}_{i,b}=\hat{\mu}_{i,b}+(Y_{i,b}-\hat{\mu}_{i,b})/T_{i,b}$.
end for

H.2. Regret analysis for non-overlapping MuLaNE

In this section, we give the regret analysis for CUCB-MG, which applies the standard CUCB algorithm [17] to this setting.

Let $\mathcal{A}$ denote the set containing all base arms, i.e., $\mathcal{A}=\{(i,b)|i\in[m],b\in[c_i]\}$. We add subscript t to denote the value of a variable at the end of t. For example, $T_{i,b,t}$ denotes the total times of arm $(i,b)\in\mathcal{A}$ is played at the end of round t. Let us first introduce a definition of an unlikely event that $\hat{\mu}_{i,b,t-1}$ is not accurate as expected.

Definition 5. We say that the sampling is nice at the beginning of round t, if for every arm $(i,b) \in \mathcal{A}$, $|\hat{\mu}_{i,b,t-1} - \mu_{i,b}| < \rho_{i,b,t}$, where $\rho_{i,b,t} = \sqrt{\frac{3\ln t}{2T_{i,b,t-1}}}$. Let $\mathcal{N}_t$ be such event.

Lemma 18. *For each round $t \geq 1$, $Pr\{\neg N_t\} \leq 2|\mathcal{A}|t^{-2}$*

Proof. For each round $t \geq 1$, we have

$$
\begin{aligned}
&Pr\{\neg\mathcal{N}_t\} \\
&= Pr\{\exists(i,b) \in \mathcal{A}, |\hat{\mu}_{i,b,t-1} - \mu_{i,b}| \geq \sqrt{\frac{3\ln t}{2T_{i,b,t-1}}}\} \\
&\leq \sum_{(i,b)\in\mathcal{A}} Pr\{|\hat{\mu}_{i,b,t-1} - \mu_{i,b}| \geq \sqrt{\frac{3\ln t}{2T_{i,b,t-1}}}\} \\
&= \sum_{(i,b)\in\mathcal{A}} \sum_{k=1}^{t-1} Pr\{T_{i,b,t-1} = k \\
&\wedge |\hat{\mu}_{i,b,t-1} - \mu_{i,b}| \geq \sqrt{\frac{3\ln t}{2T_{i,b,t-1}}}\} \\
&\leq \sum_{(i,b)\in\mathcal{A}} \sum_{k=1}^{(t-1)} \frac{2}{t^3} \leq 2|\mathcal{A}|t^{-2}.
\end{aligned}
$$

When $T_{i,b,t-1} = k$, $\hat{\mu}_{i,b,t-1}$ is the average of k i.i.d. random variables $Y_{i,b}^{[1]}, \ldots, Y_{i,b}^{[k]}$, where $Y_{i,b}^{[j]}$ is the Bernoulli random variable of arm (i,b) when it is played for the j-th time during the execution. That is, $\hat{\mu}_{i,b,t-1} = \sum_{j=1}^{k} Y_{i,b}^{[j]}/k$. Note that the independence of $Y_{i,b}^{[1]}, \ldots, Y_{i,b}^{[k]}$ comes from the fact we select a new starting position following the starting distribution $\boldsymbol{\alpha}_i$ at the beginning of each round t. The last inequality uses the Hoeffding Inequaility [37]. □

Then we can use monotonicity and 1-norm bounded smoothness propterties (Conditions 1 and 2)to bound the reward gap $\boldsymbol{\Delta}_{\boldsymbol{k}_t} = r_{\boldsymbol{\mu}}(\boldsymbol{k}^*) - r_{\boldsymbol{\mu}}(\boldsymbol{k}_t)$ between the optimal action $\boldsymbol{k}^*$ and the action $\boldsymbol{k}_t = (k_{t,1}, \ldots, k_{t,m})$ selected by our algorithm A. To achieve this, we introduce a positive real number $M_{i,b}$ for each arm (i,b) and define $M_{\boldsymbol{k}_t} = \max_{(i,b)\in S_t} M_{i,b}$, where $S_t = \{(i,b) \in \mathcal{A} | b = k_{t,i}\}$ and $|S_t| = B$. Define,

$$
\kappa_T(M,s) = \begin{cases} 2, & \text{if } s = 0, \\ 2\sqrt{\frac{6\ln T}{s}}, & \text{if } 1 \leq s \leq \ell_T(M), \\ 0, & \text{if } s \geq \ell_T(M) + 1, \end{cases}
$$

where

$$
\ell_T(M) = \lfloor \frac{24B^2 \ln T}{M^2} \rfloor.
$$

Lemma 19. *If $\{\Delta_{\boldsymbol{k}_t} \geq M_{\boldsymbol{k}_t}\}$ and $\mathcal{N}_t$ holds, we have*

$$
\Delta_{\boldsymbol{k}_t} \leq \sum_{(i,b)\in S_t} \kappa_T(M_{i,b}, T_{i,b,t-1}). \tag{H.1}
$$

Proof.

First, we can observe $\bar{\mu}_{i,b,t} \geq \mu_{i,b}$, for $(i,b) \in \mathcal{A}$, when $\mathcal{N}_t$ holds. This comes from the fact $\bar{\mu}_{i,b,t} = \min\{\hat{\mu}_{i,b,t-1} + \rho_{i,b,t}, 1\}$ and $|\hat{\mu}_{i,b,t-1} - \mu_{i,b}| < \rho_{i,b,t}$.

Notice that the right hand of Eq. (H.1) is non-negative, it is trivially satisfied when $\Delta_{\boldsymbol{k}_t} = 0$. We only need to consider $\Delta_{\boldsymbol{k}_t} > 0$. By $\mathcal{N}_t$ and Condition 1, We have the following inequalities,

$$
r_{\bar{\mu}_t}(\boldsymbol{k}_t) \geq r_{\bar{\mu}_t}(\boldsymbol{k}^*) \geq r_{\boldsymbol{\mu}}(\boldsymbol{k}^*) = \Delta_{\boldsymbol{k}_t} + r_{\boldsymbol{\mu}}(\boldsymbol{k}_t).
$$

The first inequality is due to the *oracle* outputs the optimal solution given parameters $\bar{\boldsymbol{\mu}}_t$,the second inequality is due to Condition 1 (monotonicity).

Then, by Condition 2, we have

$$
\Delta_{\boldsymbol{k}_t} \leq r_{\bar{\mu}_t}(\boldsymbol{k}_t) - r_{\boldsymbol{\mu}}(\boldsymbol{k}_t) \leq \sum_{(i,b)\in S_t} |\bar{\mu}_{i,b,t} - \mu_{i,b}|.
$$

Next, we can bound $\Delta_{\boldsymbol{k}_t}$ by bounding $\bar{\mu}_{i,b,t} - \mu_{i,b}$ by applying a transformation. As $\{\Delta_{\boldsymbol{k}_t} \geq M_{\boldsymbol{k}_t}\}$ holds by assumption, so $\sum_{(i,b)\in S_t} |\bar{\mu}_{i,b,t} - \mu_{i,b}| \geq \Delta_{\boldsymbol{k}_t} \geq M_{\boldsymbol{k}_t}$. We have

$$
\Delta_{\boldsymbol{k}_t} \leq \sum_{(i,b)\in S_t} |\bar{\mu}_{i,b,t} - \mu_{i,b}|
$$

$$\begin{aligned}
&\leq -M_{\boldsymbol{k}_t} + 2\sum_{(i,b)\in S_t} |\bar{\mu}_{i,b,t} - \mu_{i,b}| \\
&= 2\sum_{(i,b)\in S_t} (|\bar{\mu}_{i,b,t} - \mu_{i,b}| - \frac{M_{\boldsymbol{k}_t}}{2B}) \\
&\leq 2\sum_{(i,b)\in S_t} (|\bar{\mu}_{i,b,t} - \mu_{i,b}| - \frac{M_{i,b}}{2B}).
\end{aligned} \tag{H.2}$$

By $\mathcal{N}_t$, we have:

$$\begin{aligned}
|\bar{\mu}_{i,b,t} - \mu_{i,b}| - \frac{M_{i,b}}{2B} &\leq \min\{2\rho_{i,b,t}, 1\} - \frac{M_{i,b}}{2B} \\
&\leq \min\left\{2\sqrt{\frac{3\ln T}{2T_{i,b,t-1}}}, 1\right\} - \frac{M_{i,b}}{2B}.
\end{aligned}$$

Now consider the following cases:

Case 1. If $T_{i,b,t-1} \leq \ell_T(M_{i,b})$, we have $\bar{\mu}_{i,b,t} - \mu_{i,b} - \frac{M_{i,b}}{2B} \leq \min\left\{2\sqrt{\frac{3\ln T}{2T_{i,b,t-1}}}, 1\right\} \leq \frac{1}{2}\kappa_T(M_{i,b}, T_{i,b,t-1})$.

Case 2. If $T_{i,b,t-1} \geq \ell_T(M_{i,b,t}) + 1$, then $2\sqrt{\frac{3\ln T}{2T_{i,b,t-1}}} \leq \frac{M_{i,b}}{2B}$, so $\bar{\mu}_{i,b,t} - \mu_{i,b} - \frac{M_{i,b}}{2B} \leq 0 = \frac{1}{2}\kappa_T(M_{i,b}, T_{i,b,t-1})$.

Combining these two cases and Eq. (H.2), we have the following inequality, which concludes the lemma.

$$\Delta_{\boldsymbol{k}_t} \leq \sum_{(i,b)\in S_t} \kappa_T(M_{i,b}, T_{i,b,t-1}). \qquad \square$$

Proof. By above lemmas, we have

$$\begin{aligned}
&\sum_{t=1}^{T} \mathbb{I}\Big(\{\Delta_{\boldsymbol{k}_t} \geq M_{\boldsymbol{k}_t}\} \wedge \mathcal{N}_t\Big)\Delta_{\boldsymbol{k}_t} \\
&\leq \sum_{t=1}^{T} \sum_{(i,b):(i,b)\in S_t} \kappa_T(M_{i,b}, T_{i,b,t-1}) \\
&= \sum_{(i,b)\in\mathcal{A}} \sum_{t'\in\{t|t\in[T],k_{t,i}\geq b\}} \kappa_T(M_{i,b}, T_{i,b,t'-1}) \\
&\leq \sum_{(i,b)\in\mathcal{A}} \sum_{s=0}^{T_{i,b,T}} \kappa_T(M_{i,b}, s) \\
&\leq 2|\mathcal{A}| + \sum_{(i,b)\in\mathcal{A}} \sum_{s=1}^{\ell_T(M_{i,b})} 2\sqrt{6\frac{\ln T}{s}} \\
&\leq 2|\mathcal{A}| + \sum_{(i,b)\in\mathcal{A}} \int_{s=0}^{\ell_T(M_{i,b})} 2\sqrt{6\frac{\ln T}{s}}\,ds \\
&\leq 2|\mathcal{A}| + \sum_{(i,b)\in\mathcal{A}} 4\sqrt{6\ln T\ell_T(M_{i,b})} \\
&\leq 2|\mathcal{A}| + \sum_{(i,b)\in\mathcal{A}} \frac{48B\ln T}{M_{i,b}}.
\end{aligned}$$

Let us set $M_{i,b} = \Delta_{\min}^{i,b}$ and thus the event $\{\Delta_{\boldsymbol{k}_t} \geq M_{\boldsymbol{k}_t}\}$ always holds. We have

$$\begin{aligned}
&\sum_{t=1}^{T} \mathbb{I}\Big(\mathcal{N}_t \wedge \{\Delta_{\boldsymbol{k}_t} \geq M_{\boldsymbol{k}_t}\}\Big)\Delta_{\boldsymbol{k}_t} \\
&\leq 2|\mathcal{A}| + \sum_{(i,b)\in\mathcal{A}} \frac{48B\ln T}{\Delta_{\min}^{i,b}}.
\end{aligned}$$

By Lemma 18, $Pr\{\neg N_t\} \leq 2|\mathcal{A}|t^{-2}$, then we have

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{I}(\neg\mathcal{N}_t)\Delta_{\boldsymbol{k}_t}\right] \leq \sum_{t=1}^{T} 2|\mathcal{A}|t^{-2}\Delta_{\max} \leq \frac{\pi^2}{3}|\mathcal{A}|\Delta_{\max}.$$

By our choice of $M_{i,b}$,

$$\sum_{t=1}^{T} \mathbb{I}\Big(\Delta_{\boldsymbol{k}_t} < M_{\boldsymbol{k}_t}\Big)\Delta_{\boldsymbol{k}_t} = 0.$$

By the definition of the regret,

$$\begin{aligned} Reg_{\mu}(T) &= \mathbb{E}\left[\sum_{t=1}^{T} \Delta_{\boldsymbol{k}_t}\right] \\ &\le \mathbb{E}\Bigg[\sum_{t=1}^{T} \Big(\mathbb{I}(\neg\mathcal{N}_t) + \mathbb{I}(\Delta_{\boldsymbol{k}_t} < M_{\boldsymbol{k}_t}) \\ &+ \mathbb{I}(\mathcal{N}_t \wedge \{\Delta_{\boldsymbol{k}_t} \ge M_{\boldsymbol{k}_t}\})\Big)\Delta_{\boldsymbol{k}_t}\Bigg] \\ &\le \sum_{(i,b)\in\mathcal{A}} \frac{48B\ln T}{\Delta_{min}^{i,b}} + 2|\mathcal{A}| + \frac{\pi^2}{3}|\mathcal{A}|\Delta_{max}. \end{aligned}$$

For distribution independent bound, we take $M_{i,b} = M = \sqrt{48B|\mathcal{A}|\ln T/T}$, then we have $\sum_{t=1}^{T}\mathbb{I}\{\Delta_{\boldsymbol{k}_t} \le M_{\boldsymbol{k}_t}\} \le TM$. Then

$$\begin{aligned} Reg_{\mu}(T) &\le \sum_{(i,b)\in\mathcal{A}} \frac{48B\ln T}{M_{i,b}} + 2|\mathcal{A}| + \frac{\pi^2}{3}|\mathcal{A}|\Delta_{max} + TM \\ &\le \frac{48B|\mathcal{A}|\ln T}{M} + 2|\mathcal{A}| + \frac{\pi^2}{3}|\mathcal{A}|\Delta_{max} + TM \\ &= 2\sqrt{48B|\mathcal{A}|T\ln T} + \frac{\pi^2}{3}|\mathcal{A}|\Delta_{max} + 2|\mathcal{A}| \\ &\le 14\sqrt{B|\mathcal{A}|T\ln T} + \frac{\pi^2}{3}|\mathcal{A}|\Delta_{max} + 2|\mathcal{A}|. \qquad \square \end{aligned}$$

Appendix I. Supplemental experiments

I.1. Experimental results for stationary distributions

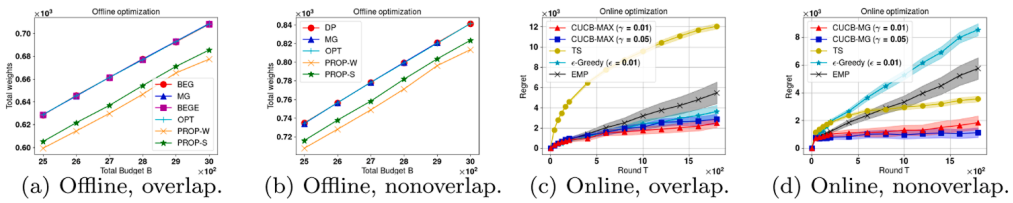
In this section, we show experimental results with explorers starting from stationary distributions in Fig. I.8. Specifically, each explorer W_i starts from the node u in layer L_i with probability proportional to the out-degree of u. For offline overlapping case, the total weights of unique nodes visited for BEG and MG are the same, and outperforms two baselines, which accords with our theoretical analysis. Similar to the Section 7, BEG and MG are empirically close to the optimal solution. For offline non-overlapping case, DP and MG give us the same *optimal* solution. For the online learning algorithms, note that we still use BEG rather than MG as the offline oracle, this is due to the UCB values do not have the diminishing return properties. The same applies for the non-overlapping case, where DP (instead of MG) is used as the offline oracle. The trend of the curves and the analysis are similar to that in the Section 7.

I.2. Online learning algorithms with different total budgets

As shown in Figs. I.9 and I.10, although there are some fluctuations, the trend of the curves and the analysis are almost the same as that when $B = 3000$, which shows our experimental results are consistent for different budgets.

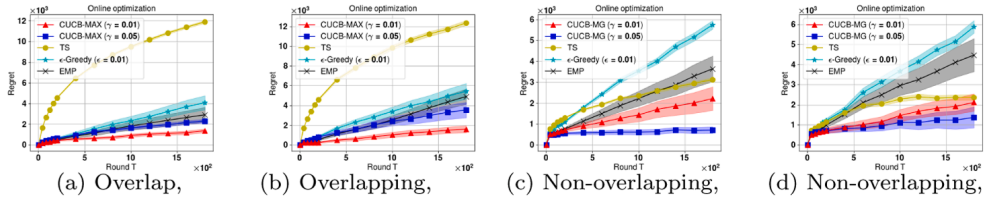
I.3. Computational efficiency of online learning algorithms

For the computational efficiency of online learning algorithms, since the running time for algorithms with the same oracle is similar, we first present the running time for CUCB-MAX (or CUCB-MG) with different oracles in Table I.4a under $B = 3.0$k. CUCB-MAX with BEG is 50 times faster than BEGE, which is consistent with our theoretical analysis. Table I.4b demonstrates the total running time of CUCB-MAX and ECUCB-MAX algorithms with the same BEG oracle on the RocketFuel dataset under $T = 10000$, where the running time of ECUCB-MAX is slightly longer than that of CUCB-MAX. This is expected because the variance-based adjustments in ECUCB-MAX introduce extra computational complexity, although the difference in runtime remains relatively minor.



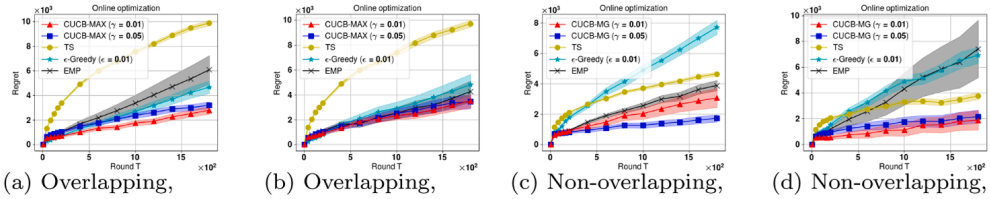
(a) Offline, overlap. (b) Offline, nonoverlap. (c) Online, overlap. (d) Online, nonoverlap.

Fig. I.8. Explorers start with stationary distributions. Above: total weights of unique nodes visited given by different offline algorithms. Below: regret for different online algorithms when $B = 3000$.



(a) Overlap, fix node. (b) Overlapping, stationary. (c) Non-overlapping, fix node. (d) Non-overlapping, stationary.

Fig. I.9. Regret for different online algorithms, when total budget $B = 2000$.



(a) Overlapping, fix node. (b) Overlapping, stationary. (c) Non-overlapping, fix node. (d) Non-overlapping, stationary.

Fig. I.10. Regret for different online algorithms, when total budget $B = 4000$.

Table I.4
Running time (seconds) for online algorithms on two different real-world datasets.

(a) With different offline oracles on the social network dataset.				
Scenario	BEG	BEGE	DP	OPT
Overlapping	1.22	60.03	NA	107.33
Non-overlapping	NA	NA	0.86	82.01
(b) With different parameters on the computer network dataset.				
Algorithm	$\gamma = 0.01$	$\gamma = 0.05$		
CUCB-MAX	97.6	96.9		
ECUCB-MAX	102.7	98.6		

I.4. Extended experimental settings and results

We extend the baseline experiments still on the FF-TW-YT dataset with its topology, by introducing more diverse weight assignments and randomized starting positions.

Settings. `Edge weights:` Instead of fixing all values to 1, we adopt two alternative schemes: (i) sample each edge weight independently from $\{0, 0.5, 1\}$ with equal probability, so that the network contains both weak, medium, and strong ties, reflecting heterogeneous interaction strengths among users; (ii) set the weight of edge $(u, v) \in \mathcal{E}$ to $\deg(u) + \deg(v)$, normalized by the maximum value in the network, where $\deg(\cdot)$ denotes the degree (number of incident edges) of a node. This scheme assigns larger weights to edges connecting highly connected users, modeling the intuition that interactions between central individuals are stronger or more influential. `Node weights:` Beyond the uniform random assignment $\{0, 0.5, 1\}$, we consider two alternatives: (i) *power-law weights*, where each node weight is drawn from a distribution $P(\sigma) \propto \sigma^{-\alpha}$ with $\alpha = 2.5$ and then linearly normalized to $[0, 1]$. This captures the skewness observed in real social systems, where a few nodes (e.g., influencers) are disproportionately important while the majority remain relatively unimportant; (ii) *centrality-correlated weights*, where $\sigma_u = \deg(u)/\max_{v \in \mathcal{V}} \deg(v)$, such that nodes with higher degree receive proportionally larger importance, mimicking the common assumption that well-connected users exert greater influence. For clarity, we denote the four settings as: (a) *Discrete edges & Power-law nodes*, (b) *Discrete edges & Degree-based nodes*, (c) *Degree-sum edges & Power-law nodes*, and (d) *Degree-sum edges & Degree-based nodes*. `Starting nodes:` Instead of always using the smallest node-id, the initial node in each trial is drawn uniformly at random from the vertex set of each layer. This avoids bias from a fixed entry point and better reflects realistic exploration where the walker may start from arbitrary users.

Baselines. We newly include CUCB [7] and BCUCB-T [17] as stronger baselines in our comparison. Notably, both [7,17] propose the same CUCB algorithm, with [17] providing improved regret bounds via new proof techniques. BCUCB-T [28] is designed for settings with probabilistically triggered arms and achieves batch-size independence.

Results. We summarize the outcomes in Fig. I.11, where each subfigure reports regret at $T = 1500$ for the four algorithms under one weight configuration (panels (a)–(d)). Overall, ECUCB-MAX consistently achieves the lowest regret, where the gains stem from its property of precision-balanced bounded smoothness and explorer-relaxed design. CUCB-MAX performs reliably as the second-best method, while CUCB and BCUCB-T incur higher regret under heterogeneous conditions. Both rely on properties such as monotonically increasing UCB estimates, which are not guaranteed to hold in the budget-constrained setting, leading to slower adaptation, especially

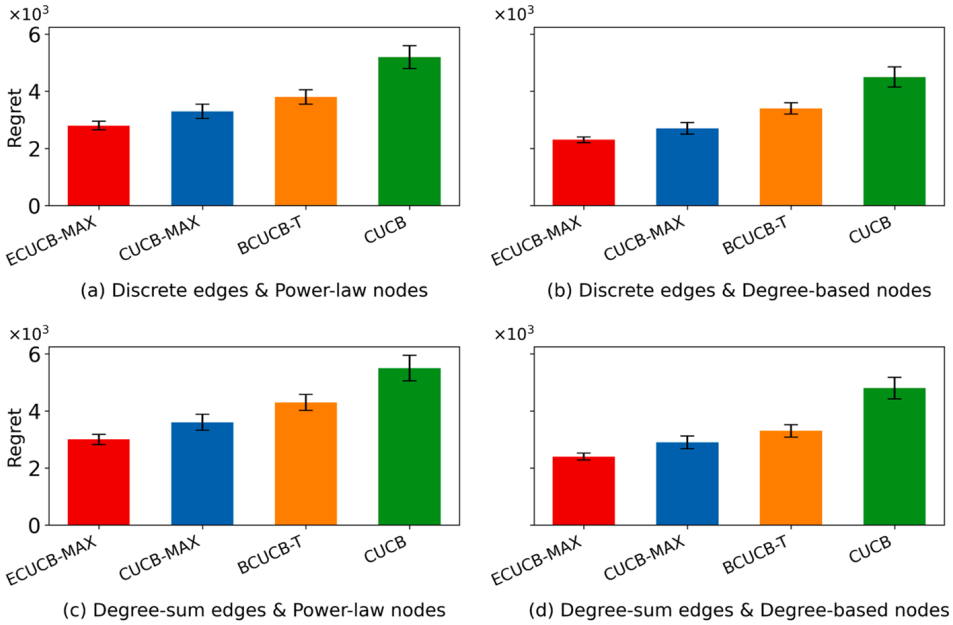


Fig. I.11. Regret@T = 1500 of algorithms under extended experimental settings.

under heavy-tailed weight distributions. In the degree-correlated cases (panels (b) and (d)), absolute regret decreases for all methods since importance aligns with structure, but the ranking remains consistent. Finally, randomizing the starting node introduces only minor additional variance and does not affect the relative ordering of algorithms.

References

[1] Q. Lv, P. Cao, E. Cohen, K. Li, S. Shenker, Search and replication in unstructured peer-to-peer networks, in: Proceedings of the 16th International Conference on Supercomputing, the 16th international conference on Supercomputing, 2002, pp. 84–95.
[2] D.F. Gleich, Pagerank beyond the web 57 (2015) 321–363.
[3] B. Wilder, N. Immorlica, E. Rice, M. Tambe, Maximizing influence in an unknown social network, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018.
[4] R. Wang, Y. Li, S. Lin, W. Wu, H. Xie, Y. Xu, J.C. Lui, Common neighbors matter: fast random walk sampling with common neighbor awareness, IEEE Trans. Knowl. Data Eng. 35 (5) (2022) 4570–4584.
[5] K. Lerman, L. Jones, Social browsing on flickr, arXiv preprint:cs/0612047.
[6] X. Dai, Z. Wang, J. Xie, X. Liu, J.C. Lui, Conversational recommendation with online learning and clustering on misspecified users, IEEE Trans. Knowl. Data Eng. 36 (12) (2024) 7825–7838.
[7] W. Chen, Y. Wang, Y. Yuan, Q. Wang, Combinatorial multi-armed bandit and its extension to probabilistically triggered arms, J. Mach. Learn. Res. 17 (1) (2016) 1746–1778.
[8] Y. Li, Z. Wu, S. Lin, H. Xie, M. Lv, Y. Xu, J.C. Lui, Walking with perception: efficient random walk sampling via common neighbor awareness, in: IEEE 35th International Conference on Data Engineering (ICDE), IEEE, 2019, pp. 962–973.
[9] D. Kempe, J. Kleinberg, E. Tardos, Maximizing the spread of influence through a social network, in: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, 2003, pp. 137–146.
[10] W. Chen, Y. Wang, S. Yang, Efficient influence maximization in social networks, in: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, 2009, pp. 199–208.
[11] C. Wang, W. Chen, Y. Wang, Scalable influence maximization for independent cascade model in large-scale social networks, Data Min. Knowl. Discov. 25 (3) (2012) 545–576.
[12] N. Alon, I. Gamzu, M. Tennenholtz, Optimizing budget allocation among channels and influencers, in: Proceedings of the 21st International Conference on World Wide Web, the 21st international conference on World Wide Web, ACM, 2012, pp. 381–388.
[13] T. Soma, N. Kakimura, K. Inaba, K.-I. Kawarabayashi, Optimal budget allocation: th eoretical guarantee and efficient algorithm, in: International Conference on Machine Learning, 2014, pp. 351–359.
[14] H. Robbins, Some aspects of the sequential design of experiments, Bull. Am. Math. Soc. 58 (5) (1952) 527–535.
[15] S. Bubeck, N. Cesa-Bianchi, Regret analysis of stochastic and nonstochastic multi-armed bandit problems, Found. Trend.®Mach. Learn. 5 (1) (2012) 1–122.
[16] Y. Gai, B. Krishnamachari, R. Jain, Combinatorial network optimization with unknown variables: multi-armed bandits with linear rewards and individual observations, IEEE/ACM Transact. Network.(TON) 20 (5) (2012) 1466–1478.
[17] Q. Wang, W. Chen, Improving regret bounds for combinatorial semi-bandits with probabilistically triggered arms and its applications, in: Advances in Neural Information Processing Systems, 2017, pp. 1161–1171.
[18] X. Chen, W. Huang, W. Chen, J.C. Lui, Community exploration: from offline optimization to online learning, in: Advances in Neural Information Processing Systems, 2018, pp. 5474–5483.
[19] X. Liu, J. Zuo, X. Chen, W. Chen, J.C. Lui, Multi-layered network exploration via random walks: from offline optimization to online learning, in: International Conference on Machine Learning, PMLR, 2021, pp. 7057–7066.
[20] X. Dai, X. Sun, J. Zuo, X. Liu, J.C. Lui, A unified online-offline framework for co-branding campaign recommendations, in: Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V, the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V, 2025, pp. 427–438.
[21] M. Kapralov, I. Post, J. Vondrák, Online submodular welfare maximization: greedy is optimal, in: Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms, 2013, pp. 1216–1225.

[22] T. Soma, Y. Yoshida, A generalization of submodular cover via the diminishing return property on the integer lattice, in: Advances in Neural Information Processing Systems, 2015, pp. 847–855.
[23] X. Dai, X. Liu, J. Zuo, H. Xie, C. Joe-Wong, J.C.S. Lui, Variance-aware bandit framework for dynamic probabilistic maximum coverage problem with triggered or self-reliant arms, IEEE Transact. Network. 33 (3) (2025) 982–993.
[24] X. Liu, X. Dai, X. Wang, M. Hajiesmaili, J.C. Lui, Combinatorial logistic bandits, in: abstracts of the 2025, in: ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, 2025, pp. 112–114.
[25] S. Khuller, A. Moss, J.S. Naor, The budgeted maximum coverage problem, Inf. Process. Lett. 70 (1) (1999) 39–45.
[26] X. Liu, X. Dai, J. Zuo, S. Wang, C. Joe-Wong, J.C. Lui, W. Chen, Offline learning for combinatorial multi-armed bandits, in: International Conference on Machine Learning, PMLR, 2025, pp. 38251–38289.
[27] N. Merlis, S. Mannor, Tight lower bounds for combinatorial multi-armed bandits, in: Conference on Learning Theory, PMLR, 2020, pp. 2830–2857.
[28] X. Liu, J. Zuo, S. Wang, C. Joe-Wong, J. Lui, W. Chen, Batch-size independent regret bounds for combinatorial semi-bandits with probabilistically triggered arms or independent arms, Adv. Neural Inf. Process. Syst. 35 (2022) 14904–14916.
[29] S. Wang, W. Chen, Thompson sampling for combinatorial semi-bandits, in: International Conference on Machine Learning, 2018, pp. 5114–5122.
[30] M.E. Dickison, M. Magnani, L. Rossi, Multilayer Social Networks, Cambridge University Press, 2016.
[31] N. Spring, R. Mahajan, D. Wetherall, Measuring isp topologies with rocketfuel„ ACM SIGCOMM Comput. Commun. Rev. 32 (4) (2002) 133–145.
[32] S. Dolev, Y. Elovici, R. Puzis, Routing betweenness centrality, J. ACM (JACM) 57 (4) (2010) 1–27.
[33] M. Kijima, Markov Processes for Stochastic Modeling, 6, CRC Press, 1997.
[34] R. Serfozo, Basics of Applied Stochastic Processes, Springer Science & Business Media, 2009.
[35] M. Raginsky, I. Sason, et al., Concentration of measure inequalities in information theory, communications, and coding, Foundat. Trends㸃 Commun. Inform. Theory, 10, 2013.
[36] A. Krause, J. Leskovec, C. Guestrin, J. Vanbriesen, C. Faloutsos, Efficient sensor placement optimization for securing large water distribution networks, J. Water Resour. Plann. Manage. 134 (6) (2008) 516–526.
[37] D.P. Dubhashi, A. Panconesi, Concentration of Measure for the Analysis of Randomized Algorithms, Cambridge University Press, 2009.
[38] J.-Y. Audibert, R. Munos, C. Szepesvári, Exploration-exploitation tradeoff using variance estimates in multi-armed bandits, Theor. Comput. Sci. 410 (19) (2009) 1876–1902.
[39] N. Merlis, S. Mannor, Batch-size independent regret bounds for the combinatorial multi-armed bandit problem, in: Conference on Learning Theory, PMLR, 2019, pp. 2465–2489.
[40] X. Liu, J. Zuo, S. Wang, J.C. Lui, M. Hajiesmaili, A. Wierman, W. Chen, Contextual combinatorial bandits with probabilistically triggered arms, in: International Conference on Machine Learning, PMLR, 2023, pp. 22559–22593.