arXiv:2601.14737v2 [cs.DB] 11 Aug 2026

Trajectory-Driven Multi-product Influence Maximization in Billboard Advertising

Dildar Ali[1], Suman Banerjee[1*], Rajibul Islam[2]

[1*]Department of Computer Science and Engineering, Indian Institute of Technology Jammu, Jagti, Jammu, 181221, Jammu and Kashmir, India.
[2]Department of Computer Science and Engineering, Gandhi Institute for Technological Advancement, Bhubaneshwar, 752054, Odisha, India.

*Corresponding author(s). E-mail(s): suman.banerjee@iitjammu.ac.in;
Contributing authors: 2021rcs2009@iitjammu.ac.in;
rajibulislam1604@gmail.com;

Abstract

Billboard Advertising has emerged as an effective out-of-home advertising technique, where the goal is to select a limited number of slots and play advertisement content there, with the hope that it will be observed by many people and, effectively, a significant number of them will be influenced towards the brand. Given a trajectory and a billboard database and a positive integer $\boldsymbol{k}$, how can we select $\boldsymbol{k}$ highly influential slots to maximize influence? In this paper, we study a variant of this problem where a commercial house wants to make a promotion of multiple products, and there is an influence demand for each product. We study two variants of the problem. In the first variant, our goal is to select $\boldsymbol{k}$ slots such that the respective influence demand of each product is satisfied. In the other variant of the problem, we are given with $\boldsymbol{\ell}$ integers $\boldsymbol{k_1, k_2, \ldots, k_\ell}$, the goal here is to search for $\boldsymbol{\ell}$ many set of slots $\boldsymbol{S_1, S_2, \ldots, S_\ell}$ such that for all $\boldsymbol{i \in [\ell]}$, $\boldsymbol{|S_i| \leq k_i}$ and for all $\boldsymbol{i \neq j}$, $\boldsymbol{S_i \cap S_j = \emptyset}$ and the influence demand of each of the products gets satisfied. We model the first variant of the problem as a multi-submodular cover problem and the second variant as its generalization. To address these challenges, we propose approximation algorithms grounded in submodular optimization. To solve the common-slot variant, we formulate the problem as a multi-submodular cover problem and design a bi-criteria approximation algorithm based on the continuous greedy framework and randomized rounding. For the disjoint-slot variant, we propose a sampling-based approximation approach along with an efficient primal–dual greedy algorithm that enforces disjointness naturally. We provide theoretical guarantees on solution quality and analyze the computational complexity of the proposed algorithms. Extensive experiments with real-world

trajectory and billboard datasets highlight the effectiveness and efficiency of the proposed solution approaches.

Keywords: Billboard Advertisement, Influence Provider, Products, Influence Maximization, Disjoint Slot Allocation

1 Introduction

Creating and maximizing influence among customers is one of the main goals for any e-commerce house. Most commercial houses spend around $7 - 10\%$ of total revenue on advertisements [1]. Determining how to utilize this budget effectively is a fundamental research challenge in computational advertising. Advertisements can be delivered through various channels, including social media, television, and outdoor media. Among these, out-of-home (OOH) advertising has recently gained substantial attention, particularly through digital billboards that display dynamic content, such as videos and animations. Due to their significantly higher return on investment (ROI), digital billboards have been widely adopted. Market surveys indicate that billboard advertising yields a 65% higher ROI than other advertising methods [2]. Furthermore, outdoor advertising has experienced remarkable growth in recent years and is projected to reach a market value of 410.2 billion by the end of 2025 [3]. In this advertising technique, a limited number of influential billboard slots are used, with the hope that displaying attractive advertising content will influence viewers. The key problem in this context is selecting a limited number of influential slots to maximize their influence. This problem has been referred to as the Influential Billboard Slot Selection Problem. Several studies have been conducted on this problem [4–7].

In this setting, it is assumed that the billboards are owned by an advertising company, while an e-commerce house seeks to select k billboard slots to maximize its advertising influence. The underlying premise is that placing advertisements in high-visibility locations accessible to diverse audiences can significantly enhance public influence, thereby increasing product sales and financial gains. However, due to budget constraints, an e-commerce house can afford only a limited number of billboard slots. Consequently, a critical problem arises: which $k \in \mathbb{Z}^+$ billboard slots should be selected to maximize influence? This problem has attracted considerable attention in recent research, with numerous solutions proposed [8]. For example, Zhang et al. [9] reported that more than 50% of the travelers are exposed to at least five billboards during a single trip. Previous studies in consumer behavior [10–14] further suggest that repeated exposure to advertisements reduces the probability of immediate action, while the cumulative influence of advertising content increases. These observations underscore the practical significance of studying the Influential Billboard Slot Selection Problem. Existing market-driven approaches to billboard selection primarily rely on traffic volume [15]. However, such simplistic methods often lead to inaccurate performance estimates and suboptimal placement strategies. In contrast, we leverage trajectory data to enable more effective selection of billboard slots. Advances in wireless communication and mobile internet technologies have made it

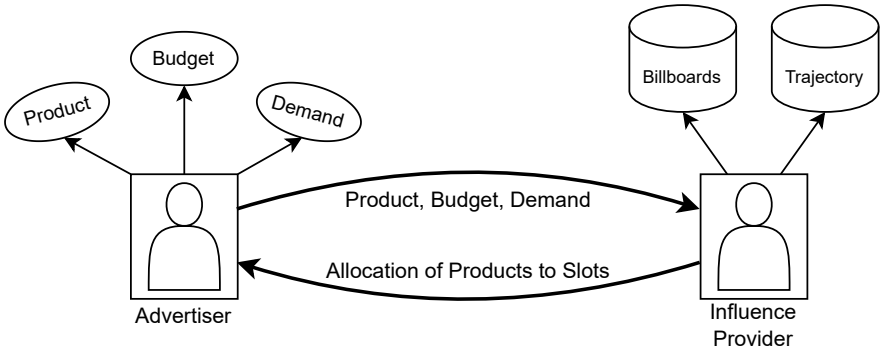


Fig. 1: Schematic Diagram of the Product to Slot Allocation

increasingly feasible to track user and vehicle movements, leading to the availability of large-scale trajectory datasets. These datasets have been widely used to address various real-world problems, including influential zone selection [16, 17], query optimization [18], route recommendation [13, 19–21], prediction of driving behavior [22], and various learning tasks [23, 24]. Recent studies have also utilized trajectory data to identify highly influential zones for billboard placement [7].

Assume the availability of a trajectory database for a city, which records the user locations along with the corresponding timestamps. Locations of interest include high-traffic areas such as shopping malls, street intersections, and metro stations, where digital billboards are typically installed. Information about these billboards, such as location, slot duration, and rental cost, is maintained in a billboard database. Influence is modeled using the well-established *triggering model* from the literature on influence maximization [25]. Intuitively, a user is influenced by a nearby billboard with a certain probability; however, when multiple billboards are encountered, the overall influence exhibits diminishing returns under the triggering model.

Most of the existing studies in billboard advertising scenarios share a common objective: (a) helping advertisers achieve maximum influence under budget constraints in a single or multi-advertiser setting [7, 26, 27], and (b) minimizing the regret of an influence provider by effective allocation of slots to advertisers [5, 28–30]. It is standard practice that multiple advertisers submit a campaign proposal to the influence provider, specifying the influence demand and the corresponding payment. A more challenging, yet unexplored, problem is how to allocate slots among multiple advertisers with multiple products within a given budget to maximize total influence from the perspective of the influence provider, as shown in the schematic diagram in Figure 1. In practice, advertisers promote multiple heterogeneous products, each targeting a different customer base. Billboard advertisements typically optimize slot selection to maximize the advertiser's influence or to minimize the regret of the influence provider. However, there is a practical need for commercial houses to meet different demand requirements across multiple products simultaneously. This requires a new formulation

to allocate billboard slots within a unified budget while maximizing overall influence across all products.

Our Contributions.

In practice, a commercial house often advertises multiple diverse products, each targeting different audiences. Thus, billboard slot selection must account for the needs of Multi-product promotions. To the best of our knowledge, no study has addressed this problem. To bridge this gap, we study the Multi-product Influential Billboard Slot Selection Problem in two variants in this paper. In the first variant, it asks for a subset of k billboard slots that satisfies the influence demand of each product and minimizes selection cost. Here, all selected slots are common, and all will be used to promote the products. We refer to the other variant of the problem as the Disjoint Multi-product Slot Selection Problem. In this problem, we are given a trajectory and billboard database and a set of ℓ integers. This problem asks to choose nonintersecting ℓ many subsets of slots bounded by their respective cardinality, such that the total selection cost is minimized and the influence demand of each product gets satisfied. We have developed approximation algorithms for these problems. In particular, we make the following contributions in this paper:

- We extend the influential billboard slot selection problem to the Multi-product setting and formally model both the common-slot and disjoint-slot variants using product-specific submodular influence functions, showing both problems are instances of the Multi-Submodular Cover Problem and hence NP-hard.
- For the common-slot variant, we design a bi-criteria approximation algorithm combining the continuous greedy framework with randomized rounding, with provable guarantees on both influence satisfaction and expected cost.
- For the disjoint-slot variant, we propose (i) a sampling-based randomized algorithm with a bounded estimation error via Hoeffding's inequality, and (ii) a primal–dual greedy algorithm (PDG) that enforces budget and disjointness constraints by construction and scales efficiently to large instances. For the cardinality-constrained case of this variant, we further establish a formal constant-factor approximation guarantee for PDG by reducing it to monotone submodular maximization over the intersection of two partition matroids.
- We conduct extensive experiments on real-world trajectory and billboard datasets from New York City and Los Angeles, comparing our methods against both naive baselines (Random, Top-k) and a stronger, theoretically-grounded lazy-greedy (CELF) baseline adapted to the Multi-product setting. The results show that PDG consistently achieves the best trade-off between influence satisfaction, cost efficiency, and runtime, while BCA achieves the highest influence satisfaction among common-slot methods at higher computational cost.

Organization of the Paper

The rest of the paper has been organized as follows. Section 2 describes recent studies from the literature. Section 3 describes the required preliminary concepts and defines the problem formally. The proposed solution approaches have been described in

Section 4. Section 5 describes the experimental evaluation. Finally, Section 6 presents the concluding remarks of our study.

2 Related Work

In this section, we will discuss the existing literature related to our work. We divide our literature study into four major categories: (a) Influential Zone selection, (b) Influential Billboard Slot Selection, (c) Trajectory Driven Influence Maximization, and (d) Tag-Based Influence Maximization. Next, we discuss existing literature on influential zone selection problems.

2.1 Influential Zone Selection

The selection of influential zones has been extensively studied in the context of spatial databases and facility placement. Most of the early work in this direction investigated proximity-based objectives, such as identifying locations that maximize the number of nearby trajectories or users. For instance, the MaxBRkNN problem studied by Zhou et al. [31] focuses on finding locations with the largest number of trajectory-based nearest neighbors. These approaches typically assume static user positions, which limits their applicability to dynamic mobility settings. The concept of location-aware influence maximization formulations was motivated by classical influence maximization problems in social networks [15, 32], where the goal is to maximize the total influence spread. Although both settings aim to maximize influence, the propagation mechanisms differ fundamentally. Social influence relies on peer-to-peer diffusion, whereas spatial influence occurs only when users enter the effective range of a location, such as a billboard. Several studies have incorporated user mobility into facility location and influence modeling. Probabilistic influence models for moving objects have been explored to capture uncertainty in user trajectories and exposure [33]. Competitive variants also consider the presence of existing facilities and model influence as relative quantities [34]. Alternative formulations define influence indirectly through user convenience, measured as additional travel distance, and aim to optimize either worst-case or aggregate inconvenience [35]. More recent work has emphasized efficiency through spatial pruning and geometric reasoning. Partition-based and region-intersection techniques have been proposed to accelerate influence evaluation over large spatial domains [36, 37]. These studies collectively highlight the importance of spatial structure, mobility awareness, and computational scalability in selecting influential zones.

2.2 Influential Billboard Slot Selection

The influential billboard slot selection problem represents a specialized instance of spatial influence maximization, where decisions are constrained by both cost and budget. Most existing work focuses on advertisers' influence maximization. Existing work primarily focuses on selecting a subset of billboard locations that maximizes trajectory-based influence under a fixed budget [38]. The problem is closely related to the budgeted maximum coverage problem, where each candidate incurs a cost

and contributes to the overall coverage [39]. In more recent work, the billboard selection problem has been formulated as a discrete submodular optimization task, enabling pruning-based strategies with approximation guarantees [40]. Spatial clustering techniques have further been introduced to reduce computational overhead while maintaining influence quality [6]. In a billboard advertisement, Ali et al. address the influential slot selection problem with a pruned submodularity graph-based solution approach [4, 6], the influential slot and tag selection problem using bisubmodular maximization approach [41], the billboard slot to tag allocation problem using bipartite graph-based solution approach [26], Multi-product influence maximization [42], balance popularity problem [43], fairness in slot allocation problem [44] etc.

In billboard advertising, there is very limited literature that considers regret minimization from the perspective of the influence provider. Aslay et al. [45] first studied the regret minimization problem in social network advertising from the perspective of an influence provider. Here, the advertiser's goal is to achieve maximum influence, and the influence provider's goal will be to minimize their total regret. Motivating this work, Zhang et al. [30] first introduce the regret minimization problem in the context of billboard advertising and, to address it, propose several heuristic solution methodologies. Further, in this direction, Ali et al. [5, 28, 29, 46] address this problem by considering the zone-specific influence demand in a multi-advertiser setting and proposed several greedy-based heuristic solutions.

2.3 Trajectory-Driven Influence Maximization

Trajectory-driven influence maximization has gained prominence as mobility data becomes increasingly available. Rather than focusing solely on static locations, these approaches optimize the influence along movement paths. Previous studies have examined the assignment of route-level advertisements to maximize exposure among targeted users [47]. Budget-aware billboard selection over trajectories has also been studied, with techniques that estimate upper bounds on influence to prune the search space [27]. Due to the effectiveness of influence maximization over trajectory databases, several researchers have examined the trajectory-driven influence maximization problem in the last few years. In this direction, Guo et al. [48] address the problem of finding the k best trajectories rather than the top-k most influential places. Their experimental results show how the best trajectories with an advertisement can maximize the overall influence. Zhang et al.[7] introduce trajectory-driven most influential places for billboard placement and influence maximization. Further, they studied the problem of billboard impression counts using trajectory information from a particular city [49]. Similarly, Wang et al.[47] studied targeted advertising and found that the outdoor advertising industry struggles to deliver the right content to the right users. To address this problem, they provide divide-and-conquer-based solution approaches. Beyond pure trajectory–billboard interactions, this work integrates additional contextual dimensions. Route capacity estimation queries leverage trajectory endpoints and movement patterns, while targeted advertising frameworks combine advertisement content, mobility transitions, and behavioral signals to improve audience matching [50]. These efforts reflect a broader trend toward multi-factor, mobility-aware influence modeling.

2.4 Tag-Based Influence Maximization

Tag-specific influence maximization has not yet been explored much in the context of billboard advertising. Nevertheless, related formulations exist in social network analysis, where influence is conditioned on semantic attributes. In particular, previous work has studied the joint selection of influential users and tags to maximize the spread of topic-based influence [51]. Here, they identify the top-k most influential users and the top-ℓ most influential tags using a bisubmodular maximization approach in the context of social network advertisements. Motivated by this work, Ali et al. [41] first studied the problem of influential slots and tag selection in billboard advertisements. Further, they studied the tag-specific influence maximization problem in billboard advertising and allocated slots to tags to maximize overall influence [26]. However, there is no existing literature that considers the product-specific influence maximization problem.

3 Background and Problem Definition

In this section, we introduce the required basic concepts and formally define our problem. Initially, we state the notion of Trajectory and Billboard Databases. $\mathbb{Z}^+$ denotes the set of positive integers. For any $\ell \in \mathbb{Z}^+$, $[\ell]$ denotes the set $\{1, 2, \ldots, \ell\}$.

3.1 Trajectory and Billboard Database

A trajectory database contains the location information of a set of people in a city across different time stamps. In the context to our problem, a trajectory database $\mathcal{D}$ contains tuples of the form $(u_i, \texttt{loc}, [t_a, t_b], \mathcal{P}(u_i))$ and this signifies that the person u_i is at the location `loc` for the duration t_a to t_b. Here, $\mathcal{P}(u_i)$ denotes the set of products in which user u_i will be interested in. For any person u_i, $loc_{[t_x,t_y]}(u_i)$ denotes the locations of u_i during the time interval $[t_x, t_y]$. Let $\mathcal{U}$ denote the set of people whose movement data is contained in $\mathcal{D}$. Let, $T_1 = \min\limits_{\tau \in \mathcal{D}} t_a$ and $T_2 = \max\limits_{t \in \mathcal{D}} t_b$ and we say that the trajectory database $\mathcal{D}$ contains the movement data from time stamp T_1 to T_2. A billboard database $\mathcal{B}$ contains the information about billboard slots. Typically, this contains the tuples of the following form $(b_{id}, s_{id}, \texttt{loc}, \texttt{duration})$ where b_{id} and s_{id} denote the billboard id and slot id. `loc` and `duration` signify the location of the billboard and the duration of the slot.

3.2 Billboard Advertisement

Let a set of m billboards $B = \{b_1, b_2, \ldots, b_m\}$ be placed in different locations of a city. For simplicity, we assume that all billboards can be leased for a multiple of a fixed duration Δ, called a 'slot' and defined in Definition 1.

Definition 1 (Billboard Slot) A billboard slot is defined by a tuple consisting of two entities, the billboard ID and the duration, that is, $(b_{id}, [t, t + \Delta])$.

The set of all billboard slots is denoted by $\mathcal{BS}$, i.e., $\mathcal{BS} = \{(b_i, [t, t + \Delta]) : i \in \{1, 2, \ldots, m\}$ and $t \in \{T_1, T_1 + \Delta, T_1 + 2\Delta, \ldots, T_2 - \Delta\}\}$. It can be observed that

$|\mathcal{BS}| = m \cdot \frac{T}{\Delta}$ where $T = T_2 - T_1$. For simplicity, we assume that T is perfectly divisible by Δ. For any slot $s_j \in \mathcal{BS}$, b_{s_j} denotes the corresponding billboard and $[t^s_{s_j}, t^f_{s_j}]$ denotes the time interval for the slot s_j where $t^f_{s_j} = t^s_{s_j} + \Delta$. Now, we state the notion of the influence probability of a billboard slot in Definition 2.

Definition 2 (Influence Probability of a Billboard slot) Given a slot $s_j \in \mathcal{BS}$ and a person $u_i \in \mathcal{U}$, the influence probability of s_j on u_i is denoted by $Pr(s_j, u_i)$ and can be computed using the following conditional equation.

$$Pr(s_j, u_i) = \begin{cases} \frac{size(s_j)}{\max\limits_{s_k \in \mathcal{BS}} size(b_{s_k})}, & \text{if } loc_{[t^s_{s_j}, t^f_{s_j}]}(u_i) = loc(s_j) \\ 0, & \text{otherwise} \end{cases}$$

Now, we define the influence of a subset of billboard slots in 3.

Definition 3 (Influence of Billboard Slots) Given a trajectory database $\mathcal{D}$, and a subset of billboard slots $\mathcal{S} \subseteq \mathcal{BS}$, the influence of $\mathcal{S}$ can be defined as the expected number of trajectories that are influenced, which can be computed using Equation 1.

$$\mathcal{I}^{'}(\mathcal{S}) = \sum_{u_i \in \mathcal{U}} [1 - \prod_{s_j \in \mathcal{S}} (1 - Pr(s_j, u_i))] \tag{1}$$

Here, $Pr(s_j, u_i)$ denotes the influence probability of the billboard slot s_j on the people u_i, and $\mathcal{I}^{'}(\mathcal{S})$ denotes the influence of the billboard slots of $\mathcal{S}$. $\mathcal{I}^{'}()$ is the influence function which maps each possible subset of billboard slots to its corresponding influence value, i.e., $\mathcal{I}^{'} : 2^{\mathcal{BS}} \longrightarrow \mathbb{R}^+_0$ where $\mathcal{I}^{'}(\emptyset) = 0$.

Now, it is important to observe that every product may not be relevant to every trajectory user. For effective advertisement, it is important to influence the relevant people for a particular product. Consider $\mathcal{P} = \{p_1, p_2, \ldots, p_\ell\}$ denotes the set of products. From the trajectory database, for every trajectory user, we have the information about the products for which he is relevant. For a given product, $j \in [\ell]$, we state the notion of Product Specific Influence for a given billboard slot in Definition 4.

Definition 4 (Product Specific Influence) Given a specific product and a subset of billboard slots $\mathcal{S} \subseteq \mathcal{BS}$, the product-specific influence of $\mathcal{S}$ for the product j is denoted by $\mathcal{I}_j(\mathcal{S})$ and can be computed using Equation 2.

$$\mathcal{I}_j(\mathcal{S}) = \sum_{u_i \in \mathcal{U}_j} [1 - \prod_{s_j \in \mathcal{S}} (1 - Pr(s_j, u_i))] \tag{2}$$

Here, $\mathcal{U}_j$ denotes the set of users relevant to the j-th product and $\mathcal{I}()$ is the product specific influence function, i.e., $\mathcal{I} : 2^{\mathcal{BS}} \times \mathcal{P} \longrightarrow \mathbb{R}^+_0$.

Example 1 Consider four billboard slots $\mathcal{BS} = \{b_1, b_2, b_3, b_4\}$ placed at different locations in a city, and four users $\mathcal{U} = \{u_1, u_2, u_3, u_4\}$ whose trajectories pass near these billboards during

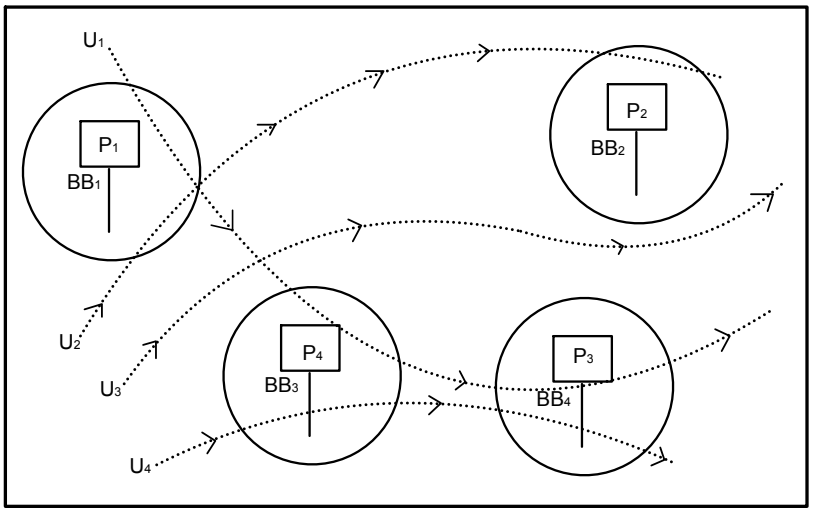


Fig. 2: A motivating example

the corresponding slot durations. Assume there are four products $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3, \mathcal{P}_4\}$. Each user is interested in exactly one product, as illustrated in Figure 2. The corresponding product–user relevance sets are: $\mathcal{U}_{\mathcal{P}_1} = \{u_1\}$,$\mathcal{U}_{\mathcal{P}_2} = \{u_2\}$,$\mathcal{U}_{\mathcal{P}_3} = \{u_3\}$,$\mathcal{U}_{\mathcal{P}_4} = \{u_4\}$. The influence probabilities of billboard slots on users are given as follows:

Billboard Slot	Influenced Users (Probability)
b_1	$u_1(0.6),\ u_2(0.2)$
b_2	$u_2(0.4)$
b_3	$u_1(0.4),\ u_4(0.3)$
b_4	$u_1(0.4),\ u_4(0.5)$

Let the selected slot set be $\mathcal{S} = \{b_1, b_2, b_3, b_4\}$. Using Definition 4, the product-specific influence for each product is as follows. For product $\mathcal{P}_1$, the relevant user is u_1. The slot b_1, b_3 and b_4 influence u_1. Thus, $\mathcal{I}_{\mathcal{P}_1}(\mathcal{S}) = 1 - (1 - 0.6)(1 - 0.4)(1 - 0.4) = 0.856$. For product $\mathcal{P}_2$, the relevant user is u_2. Both b_1 and b_2 influences u_2. Hence, $\mathcal{I}_{\mathcal{P}_2}(\mathcal{S}) = 1 - (1 - 0.2)(1 - 0.4) = 0.52$. For product $\mathcal{P}_3$, the relevant user is u_3. No billboard influences u_3. Therefore, $\mathcal{I}_{\mathcal{P}_3}(\mathcal{S}) = 0$. For product $\mathcal{P}_4$, the relevant user is u_4. Both b_3 and b_4 influences u_4. Hence, $\mathcal{I}_{\mathcal{P}_4}(\mathcal{S}) = 1 - (1 - 0.3)(1 - 0.5) = 0.65$. This example highlights that even when the same set of billboard slots is selected, the resulting influence values differ across products. The influence of a slot applies only to users relevant to that product, and the aggregation follows a diminishing-returns property.

3.3 Submodular Set Functions and Continuous Extensions

A function $f : 2^{\mathcal{N}} \to \mathbb{R}$ is submodular if for every $A \subseteq B \subseteq N$, $f(A \cup \{e\}) - f(A) \geq f(B \cup \{e\}) - f(B)$ such that $e \in \mathcal{N}$ and $e \notin B$. Such functions arise in many applications. The continuous extension of submodular set functions has played an important role in algorithmic aspects. The idea is to extend a discrete set function

$f : 2^{\mathcal{N}} \to \mathbb{R}$ to continuous space $[0,1]^{\mathcal{N}}$. Here, we are mainly concerned with multi-linear extensions motivated by maximization problems and refer the interested reader to Calinescu et al. [52] and Vondrak [53] for a detailed discussion.

The multi-linear extension is a real-valued set function $f : 2^{\mathcal{N}} \to \mathbb{R}$, denoted by F and defined as follows:

$$F(x) = \sum_{\mathcal{S} \subseteq N} f(\mathcal{S}) \prod_{i \in \mathcal{S}} x_i \prod_{j \notin \mathcal{S}} (1 - x_j) \tag{3}$$

where $x \in [0,1]^N$, and the random subset $\mathcal{S} \subseteq N$ is drawn by including each element $i \in N$ independently with probability x_i, i.e., $\Pr[i \in \mathcal{S}] = x_i$ independently for each i.

For any two vectors $p, q \in [0,1]^N$, we use $p \vee q$ and $p \wedge q$ to denote the coordinate-wise maximum and minimum, respectively of p and q. We also make use of the notation $w_e(t) \leftarrow F(y(t) \vee 1_e) - F(y(t))$, where 1_e is the characteristic vector of set $\{e\}$.

3.4 Problem Definition

In this section, we define our problem formally.

3.4.1 Common and Disjoint Multi-product Slot Selection Problem

As mentioned previously, we have studied the Multi-product Slot Selection Problem in two variants. First, we discuss the variant where a fixed number of slots will be selected for all the products. In this problem, we are given a trajectory and a billboard database (which includes the cost function $w : \mathcal{BS} \longrightarrow \mathbb{R}^+$), and ℓ positive integers $\sigma_1, \sigma_2, \ldots, \sigma_\ell \in \mathbb{Z}^+$. The task is to find a subset of the slots $\mathcal{S} \subseteq \mathcal{BS}$ such that the total cost is minimized and for each product $j \in [\ell]$, their influence demand constraint is satisfied, i.e., for all $j \in [\ell]$, $\mathcal{I}_j(\mathcal{S}) \geq \sigma_j$. From a computational point of view, this problem can be posed as follows:

Common Multi-product Slot Selection Problem
Input: Billboard $\mathcal{B}$ and Trajectory Database $\mathcal{D}$, Cost Function $w : \mathcal{BS} \longrightarrow \mathbb{R}^+$, ℓ integers $\sigma_1, \sigma_2, \ldots, \sigma_\ell \in \mathbb{Z}^+$ for influence demand constraint.
Problem: Select a subset of the slots $\mathcal{S} \subseteq \mathcal{BS}$ such that $\sum_{v \in \mathcal{S}} w(v)$ gets minimized and the constraints $\mathcal{I}_j(\mathcal{S}) \geq \sigma_j$ are satisfied.

In the Disjoint Multi-product Slot Selection Problem, along with the trajectory and billboard database, and the cost function, we are given with two sets of ℓ integers $k_1, k_2, \ldots, k_\ell \in \mathbb{Z}^+$ and $\sigma_1, \sigma_2, \ldots, \sigma_\ell \in \mathbb{Z}^+$. The task here is to select ℓ many subsets of slots $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_\ell \subseteq \mathcal{BS}$ such that for all $j \in [\ell]$, $|\mathcal{S}_j| \leq k_j$, for all $i \neq j$, $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset$, and for all $j \in [\ell]$, $\mathcal{I}_j(\mathcal{S}_j) \geq \sigma_j$. From a computational point of view, this problem can be posed as follows:

Disjoint Multi-product Slot Selection Problem
Input: Billboard $\mathcal{B}$ and Trajectory Database $\mathcal{D}$, Two sets of ℓ integers $k_1, k_2, \ldots, k_\ell \in \mathbb{Z}^+$ and $\sigma_1, \sigma_2, \ldots, \sigma_\ell \in \mathbb{Z}^+$.
Problem: Select ℓ many subsets of slots $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_\ell \subseteq \mathcal{BS}$ such that:

1. For all $j \in [\ell]$, $|\mathcal{S}_j| \leq k_j$
2. For all $i \neq j$, $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset$
3. For all $j \in [\ell]$, $\mathcal{I}_j(\mathcal{S}_j) \geq \sigma_j$

As our problem is closely related to the Multi-submodular Cover Problem, first, we describe and generalize it.

3.4.2 Multi-Submodular Cover Problem

In Multi-Submodular Cover Problem, we are given with a ground set $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$, a cost function $w : \mathcal{V} \longrightarrow \mathbb{R}^+$ which maps each ground set element to its corresponding cost, a set of ℓ submodular functions $\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_\ell$, and a set of ℓ integers $\sigma_1, \sigma_2, \ldots, \sigma_\ell \in \mathbb{Z}^+$. This problem asks to choose a subset $\mathcal{S}$ of ground set elements, i.e., $\mathcal{S} \subseteq \mathcal{V}$, such that the total cost of the chosen elements gets minimized and for any $i \in [\ell]$, $\mathcal{I}_i(S) \geq \sigma_i$, i.e., the constraints on the functional values of the submodular functions get satisfied. This is an optimization problem which can be posed as follows:

$$\mathcal{S}^* \longleftarrow \underset{\substack{\mathcal{S} \subseteq \mathcal{V} \\ \forall j \in [\ell], \mathcal{I}_j(\mathcal{S}) \geq \sigma_j}}{argmin} \sum_{v \in \mathcal{S}} w(v) \tag{4}$$

Here, $\mathcal{S}^*$ denotes the optimal solution for this problem. From the computational point of view, this problem can be posed as follows:

Multi-Submodular Cover Problem
Input: The ground set $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$, the cost function $w : \mathcal{V} \longrightarrow \mathbb{R}^+$, ℓ many submodular functions $\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_\ell$ defined on the ground set $\mathcal{V}$, ℓ many positive integers $\sigma_1, \sigma_2, \ldots, \sigma_\ell \in \mathbb{Z}^+$.
Problem: Select a subset of the ground set elements $\mathcal{S} \subseteq \mathcal{V}$ which minimizes $\sum_{x \in \mathcal{S}} w(x)$ but satisfies $\mathcal{I}_j(\mathcal{S}) \geq \sigma_j$ for all $j \in [\ell]$.

An instance of the Multi-Submodular Cover Problem is said to be r-sparse if each ground set element can be involved in a maximum of r objectives. This problem has been studied by Chekuri et al. [54], and they proposed a randomized bi-criteria approximation algorithm which runs in polynomial time. The formal result has been stated in Theorem 1.

Theorem 1 *There exists a randomized bi-criteria approximation algorithm that for a given instance of the Multi-Submodular Cover Problem it produces a set $\mathcal{S} \subseteq \mathcal{V}$ such that (i) For all $j \in [\ell]$, $\mathcal{I}_j(\mathcal{S}) \geq (1 - \frac{1}{e} - \epsilon)\sigma_j$ and (ii) $\mathbb{E}[w(\mathcal{S})] \geq \mathcal{O}(\frac{1}{\epsilon} lnr)$.*

GENERALIZED MULTI-SUBMODULAR COVER PROBLEM
Input: The ground set $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$, the cost function $w : \mathcal{V} \longrightarrow \mathbb{R}^+$, ℓ many submodular functions $\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_\ell$ defined on the ground set $\mathcal{V}$, two sets of ℓ positive real numbers each $k_1, k_2, \ldots, k_\ell \in \mathbb{Z}^+$ and $\sigma, \sigma_2, \ldots, \sigma_\ell \in \mathbb{R}^+$.
Problem: select ℓ many subsets of the ground set $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_\ell \subseteq \mathcal{V}$ such that:

- For all $i, j \in [\ell]$ and $i \neq j$, $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset$
- For all $i \in [\ell]$, $\sum_{x \in \mathcal{S}_i} w(x) \leq k_i$
- For all $i \in [\ell]$, $\mathcal{I}_i(\mathcal{S}_i) \geq \sigma_i$

3.4.3 Generalized Multi-Submodular Cover Problem

In this problem, we are given a ground set of n elements $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$, a cost function w that maps each ground set element to its corresponding cost; i.e., $w : \mathcal{V} \longrightarrow \mathbb{R}^+$, ℓ many submodular functions $\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_\ell$ defined on the ground set $\mathcal{V}$, and two sets of ℓ many real numbers $k_1, k_2, \ldots, k_\ell \in \mathbb{Z}^+$ and $\sigma, \sigma_2, \ldots, \sigma_\ell \in \mathbb{Z}^+$. The task is to select ℓ many subsets of the ground set $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_\ell \subseteq \mathcal{V}$ such that: (i) For all $i, j \in [\ell]$ and $i \neq j$, $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset$, (ii) For all $i \in [\ell]$, $\sum_{x \in \mathcal{S}_i} w(x) \leq k_i$, and (iii) For all $i \in [\ell]$, $\mathcal{I}_i(\mathcal{S}_i) \geq \sigma_i$.

4 Proposed Solution Approaches

In this section, we describe the proposed solution approaches. Section 4.1 and 4.2 contain the methodologies for the Common and Disjoint Multi-product Slot Selection Problem, respectively.

4.1 Common Multi-product Slot Selection

We analyze the continuous greedy algorithm for general monotone, submodular functions. At each step t, in Line No. 3 to 5, the marginal gain for each $e \in \mathcal{BS}$ is computed. Next, in Line No. 6, solve the linear program to get a direction vector, and for each slot in $\mathcal{BS}$, each coordinate of $y(t)$ is updated in Line No. 7 to 9. Finally, after time T, it returns the fractional solution $y(T)$. Parameter T is the stopping criterion of Algorithm 1 as mentioned in Theorem 2. It is important to note that in Line 8, we modify the direction y. The definition of δ implies that δ^{-1} is at least n^5 and is divisible by T^{-1}. This ensures that after $T\delta^{-1}$ iterations, t will be exactly equal to T.

Theorem 2 *[55] For any normalized monotone submodular function $f : 2^{\mathcal{N}} \to \mathbb{R}^+$, a solvable polytope $\mathcal{P} \in [0,1]^{\mathcal{N}}$ and stopping time $T > 0$, Algorithm 1 finds a point $x \in [0,1]^{\mathcal{N}}$, such that $F(x) \geq [1 - e^{-T} - \mathcal{O}(1)] \cdot f(OPT)$.*

We address the *Common Multi-product Slot Selection Problem* in Algorithm 2, where a single slot set must satisfy the influence demands of multiple products under budget. We propose a bi-criteria approximation algorithm combining continuous

Algorithm 1: Continuous Greedy Algorithm

Data: Trajectory Database $\mathcal{D}$, Billboard Database $\mathcal{B}$, Influence Functions $\mathcal{I}()$, Billboard Slots $\mathcal{BS}$
Result: A solution of vector $\mathcal{Y}(T)$

Initialize $t \leftarrow 0$, $\mathcal{Y}(0) \leftarrow 1_{\emptyset}$, $n \leftarrow |\mathcal{BS}|$, $\delta \leftarrow T(\lceil n^5 T \rceil)^{-1}$;
while $t < T$ **do**
  for *each* $e \in \mathcal{BS}$ **do**
    $w_e(t) \leftarrow F(y(t) \vee 1_e) - F(y(t))$
  end
  $\mathcal{I}(t) \leftarrow argmax\{x \cdot w(t) \mid x \in \mathcal{P}\}$;
  for *each* $e \in \mathcal{BS}$ **do**
    $y_e(t+\delta) \leftarrow y_e(t) + \delta \mathcal{I}_e(t) \cdot (1 - y_e(t))$
  end
  $t \leftarrow t + \delta$
end
return $y(T)$;

greedy and randomized rounding. First, in Line No. 1, we scale each product's influence function $\mathcal{I}_j$ such that $\mathcal{I}_j(\mathcal{BS}) = 1$, and set $\sigma_j \leftarrow 1$ and in Line No. 2, we define the polytope $\mathcal{P}$. In Line No. 3, using Algorithm 1, we solve the multilinear extension over the matroid polytope to obtain a fractional solution $x \in [0,1]^{|\mathcal{BS}|}$. In the rounding step, Line No. 5 to 7, sample $\ell = \lceil \log_{1/(1-\epsilon)}(r) \rceil$ random subsets from x to form $\mathcal{S}$. Next, in Line No. 8 to 14 for any product j where $\mathcal{I}_j(S)$ is below $(1 - \frac{1}{e} - 2\epsilon)\sigma_j$, add slots greedily by marginal gain until the bound $(1 - \frac{1}{e})\sigma_j$ is met.

Algorithm 2: Bi-criteria Approximation Algorithm for Multi-product Influence Maximization

Data: Trajectory Database $\mathcal{D}$, Billboard Slots $\mathcal{BS}$, Influence Functions $\{\mathcal{I}_j(\cdot)\}_{j=1}^{\ell}$, Cost Function $w : \mathcal{BS} \to \mathbb{R}^+$, Sparsity r, Approximation Parameter $\epsilon > 0$, Influence Thresholds $\{\sigma_j\}_{j=1}^{\ell}$
Result: A set of billboard slots $\mathcal{S} \subseteq \mathcal{BS}$ satisfying all influence constraints approximately

1 **Normalize:** For each $j \in [\ell]$, set $\mathcal{I}_j(\mathcal{BS}) = 1$ and scale $\sigma_j \leftarrow 1$;
2 Let $\mathcal{P} \subseteq [0,1]^{|\mathcal{BS}|}$ be the constraint polytope;
3 $x \leftarrow \text{ContinuousGreedy}(\mathcal{BS}, \mathcal{P}, T = 1)$;
4 $\alpha \leftarrow 1 - \epsilon$, $\ell \leftarrow \lceil \log_{1/\alpha} r \rceil$, $S \leftarrow \emptyset$;
5 **for** $t = 1$ *to* ℓ **do**
6 Sample $\mathcal{S}_t \subseteq \mathcal{BS}$ by including each $i \in \mathcal{BS}$ independently with probability $x(i)$;
7 $\mathcal{S} \leftarrow \mathcal{S} \cup \mathcal{S}_1 \cup \mathcal{S}_2 \cup \ldots \cup \mathcal{S}_t$;
8 **for** $j = 1$ *to* ℓ **do**
9 **if** $\mathcal{I}_j(\mathcal{S}) < (1 - \frac{1}{e} - 2\epsilon) \cdot \sigma_j$ **then**
10 $M_j \leftarrow \emptyset$;
11 **while** $\mathcal{I}_j(M_j) < (1 - \frac{1}{e}) \cdot \sigma_j$ **do**
12 $u^* \leftarrow \arg\max_{u \in \text{BS} \setminus M_j} \left[\mathcal{I}_j(M_j \cup \{u\}) - \mathcal{I}_j(M_j)\right]$;
13 $M_j \leftarrow M_j \cup \{u^*\}$;
14 $\mathcal{S} \leftarrow \mathcal{S} \cup M_j$;
15 **return** $\mathcal{S}$

Theorem 3 *[54] Let r be the maximum number of functions any slot contributes to (i.e., sparsity), and let $\epsilon > 0$. Then with high probability, the Algorithm 2 returns a set S such that for all $j \in [\ell]$, $\mathcal{I}_j(S) \geq \left(1 - \frac{1}{e} - \epsilon\right) \cdot \sigma_j$, and the expected cost satisfies $\mathbb{E}[w(S)] = O\left(\frac{1}{\epsilon} \log r\right) \cdot \mathrm{OPT}$.*

Complexity Analysis.

Let $n = |\mathcal{BS}|$ be the number of billboard slots and ℓ the number of products. The Continuous Greedy phase runs in $\mathcal{O}(n^2\ell/\epsilon)$ time due to gradient computation over a polytope of size n for each of ℓ influence functions. The rounding and union of $\ell = \mathcal{O}(\log r/\epsilon)$ random subsets take $\mathcal{O}(n \log r/\epsilon)$ time. In the worst case, the repair step examines all n slots per product, taking $\mathcal{O}(n\ell)$ time. So the overall time complexity is $\mathcal{O}\left(\frac{n^2\ell}{\epsilon} + n\ell\right)$. The additional space requirement for Algorithm 2 is $\mathcal{O}(n)$ for maintaining the fractional vector and final slot set.

4.2 Disjoint Multi-product Slot Selection

In this section, we describe the proposed solution approach for the Disjoint Multi-product Slot Selection Problem.

4.2.1 Randomized Algorithm for Multi-product Influence Maximization

In this problem, we are given a budget and the influence demand for every product. The goal here is to select billboard slots for each product within the budget such that the influence demand for every product is satisfied. Also, a slot can be used for only one product. In our proposed methodology, for every possible permutation of products and slots, we perform the following task. As long as the remaining budget of the current advertiser being processed is positive and his influence demand is unsatisfied, we select, in each iteration, a slot that maximizes the marginal gain given the existing set of slots chosen for the advertiser. This process terminates once the influence demand is satisfied or the advertiser's budget is exhausted. If the advertiser's budget is exhausted, it means that, for the current permutation of advertisers and slots, the influence demand for this advertiser cannot be satisfied. Hence, this does not yield a feasible solution, so we mark it as infeasible. This process is continued for all possible permutations of advertisers and slots. Among all the feasible solutions stored in $\mathcal{M}$, we return one with the least cost. Algorithm 3 describes the entire process in pseudocode.

One point to highlight here is that the number of possible permutations will be of $\mathcal{O}(\ell! \cdot n!)$. By Stirling's approximation, we can write it as $\mathcal{O}(\ell^\ell \cdot n^n)$. For moderate values of ℓ and n (in practice, the value of n could be excessively large), the number of possible permutations could be large. If we consider all of them in the computation, then the execution time requirement will not be practically feasible. To mitigate this problem, we sample a random subset of them in this paper. Now, it is an important question what the sample size should be so that the estimation error can be bounded with high probability. This leads to the sample complexity analysis, which is presented subsequently.

Sample Complexity Analysis.

We can use Hoeffding's Inequality to provide a sample bound that guarantees finding a close to optimal solution with high probability. Theorem 4 describes the statement.

Theorem 4 *Let $X_1, X_2, \ldots, X_n$ are independent and identically distributed random variables such that for any $i \in [n]$, $a_i \leq X_i \leq b_i$. Let $\bar{X} = \frac{1}{n} \sum_{i \in [n]} X_i$ and $\mu = \mathbb{E}[\bar{X}]$. For any $\epsilon > 0$, the following inequality holds*

$$Pr[|\bar{X} - \mu| \geq \epsilon] \leq 2 \cdot exp(-\frac{2n^2\epsilon^2}{\sum_{i \in [n]} (b_i - a_i)^2}) \tag{5}$$

So for estimation, the upper and lower bound of the solution cost of each feasible solution present in the sample is required. We prove the same in Proposition 5.

Proposition 5 *For any feasible solution $\mathcal{S} = (S_1, S_2, \ldots, S_\ell)$ contained in $\mathcal{M}$ the cost of solution will lie in between 0 and $\sum_{s \in \mathcal{BS}} w(s)$.*

Proof For any advertiser $a_i \in \mathcal{A}$, its influence demand $\sigma_i \geq 0$ and for any slot $s_j \in \mathcal{BS}$, its cost $w(s_j) \geq 0$. Now, it can be observed that the lower bound holds only when for all the advertisers, their influence demand is all 0. In that case, every advertiser will be assigned the empty set, i.e., for all $i \in [\ell]$, $S_i = \emptyset$, and hence, the solution cost will be 0. In any other case, the cost of the solution will be strictly greater than 0. The upper bound occurs when the allocated slots to the advertisers, i.e., $S_1, S_2, \ldots, S_\ell$, is a partition of $\mathcal{BS}$. This means (i) For all $i, j \in [\ell]$, $S_i \cap S_j = \emptyset$, (ii) $\bigcup_{i \in [\ell]} S_i = \mathcal{BS}$. In this case, the cost of the solution will be $\sum_{s \in \mathcal{BS}} w(s)$. In any other case, the cost of the solution will be strictly less than $\sum_{s \in \mathcal{BS}} w(s)$. Hence, for any solution $\mathcal{S} = (S_1, S_2, \ldots, S_\ell)$, the cost of the solution will be in the range in between 0 and $\sum_{s \in \mathcal{BS}} w(s)$. □

Now, we prove Theorem 6 which will prove the lower bound on the sample size for which Algorithm 3 will provide a close to optimal solution with high probability.

Theorem 6 *For any $\epsilon, \delta \in (0, 1)$ if the sample size is greater than equals to $\frac{ln(\frac{2}{\delta}) \cdot w(\mathcal{BS})^2}{2 \cdot \epsilon^2 \cdot (W^{\mathcal{A}})^2}$ then the probability error in computation will be strictly less than ϵ with probability at least $(1 - \delta)$.*

Proof Let, $\mathcal{Y}$ denote the set of sampled solutions. W^* and $W^{\mathcal{A}}$ denotes the optimal solution the best solution among the solutions in Sample, respectively. Now, by applying Hoeffding's Inequality and Proposition 5 we gave the following inequality:

$$Pr[|W^* - W^{\mathcal{A}}| \geq \epsilon \cdot W^*] \leq 2 \cdot exp(-\frac{2 \cdot \epsilon^2 \cdot (W^{\mathcal{A}})^2 \cdot |\mathcal{Y}|}{w(\mathcal{BS})^2}) \tag{6}$$

We want to establish the sample size such that $Pr[|W^* - W^{\mathcal{A}}| < \epsilon \cdot W^*] \geq 1 - \delta$. By simplifying we have the following:

$$2 \cdot exp(-\frac{2 \cdot \epsilon^2 \cdot (W^{\mathcal{A}})^2 \cdot |\mathcal{Y}|}{w(\mathcal{BS})^2}) \leq \delta$$

$$ln(\frac{2}{\delta}) \leq \frac{2 \cdot \epsilon^2 \cdot (W^{\mathcal{A}})^2 \cdot |\mathcal{Y}|}{w(\mathcal{BS})^2}$$

This leads to the lower bound on the sample size which is

$$|\mathcal{Y}| \geq \frac{ln(\frac{2}{\delta}) \cdot w(\mathcal{BS})^2}{2 \cdot \epsilon^2 \cdot (W^{\mathcal{A}})^2}$$

□

Remark 1 Theorem 6 is best interpreted as a posteriori guarantee. Once Algorithm 3 has been run on a sample of size $|Y|$ and has returned a best found solution of cost W^A, the theorem certifies that W^A is within a factor $(1 \pm \epsilon)$ of the true optimum W^* with probability at least $1 - \delta$. It does not serve as an a priori sample-sizing rule, since W^A is only observed after sampling. In practice, one can address this by (i) using a domain-specific lower bound on W^A when available (e.g., from the minimum per-product slot cost), or (ii) adopting a doubling strategy: start with a small sample, compute W^A, and increase the sample size until W^A stabilizes within tolerance ϵ across successive rounds.

Computational Complexity Analysis.

Consider $|\mathcal{X}|$ is the sample size used to implement Algorithm 3. Both the `for loops` from 2 to 4 and from 5 to 20 will execute $\mathcal{O}(|\mathcal{X}|)$. The *for loops* from 7 to 8 and 9 to 16 will execute for $\mathcal{O}(\ell)$ times. Now, it is important to analyze how many times the *while loop* from 10 to 16 will execute. Consider the minimum selection cost among all the billboards is denoted by w_{min}, i.e., $w_{min} = \min\limits_{bs_j \in \mathcal{BS}} w(bs_j)$. Let $\mathcal{Z} = \{z_1, z_2, \ldots, z_\ell\}$ are the budgets of the advertisers. Among these, let $\mathcal{Z}_{max}$ denote the maximum budget among all the advertisers, i.e., $\mathcal{Z}_{max} = \max\limits_{i \in [\ell]} z_i$. It can be observed that the maximum $\frac{\mathcal{Z}_{max}}{w_{min}}$ number of slots can be allocated to an advertiser. Hence, the *while loop* can be iterated at max $\mathcal{O}(\frac{\mathcal{Z}_{max}}{w_{min}})$ times. Within the *while loop*, the main computation is the marginal influence gain computation of a non-allocated slot, and this requires $\mathcal{O}(n \cdot t)$ time where n and t denote the number of billboard slots and the tuples in the trajectory database, respectively. Hence, the time requirement for the execution of the *while loop* will be of $\mathcal{O}(\frac{\mathcal{Z}_{max}}{w_{min}} \cdot n \cdot t)$. All the remaining statements within the *for loop* from 5 to 20 will take $\mathcal{O}(1)$ time. Hence, the time requirement till Line 20 will be of $\mathcal{O}(|\mathcal{X}| \cdot \ell \cdot \frac{B_{max}}{w_{min}} \cdot n \cdot t)$. In the worst case, it may happen that all sampled solutions are feasible. In Line 21, we find the best solution among all the feasible solutions in $\mathcal{X}$. It will take $\mathcal{O}(|\mathcal{X}|)$. Hence, the time requirement to execute of Algorithm 3 is of $\mathcal{O}(|\mathcal{X}| \cdot \ell \cdot \frac{B_{max}}{w_{min}} \cdot n \cdot t)$. Additional space consumed by Algorithm 3 is to store the feasible solutions, which will be of $\mathcal{O}(|\mathcal{X}| \cdot n)$, to store the sets $S_1, S_2, \ldots, S_\ell$ and $\mathbb{S}$ and both of them will take $\mathcal{O}(n)$ space. Hence, total extra space consumed by $\mathcal{O}(|\mathcal{X}| \cdot n + n)$ which is equal to $\mathcal{O}(|\mathcal{X}| \cdot n)$. Hence, Theorem 7 holds.

Algorithm 3: Randomized Algorithm for Multi-product Influence Maximization

Data: Billboard Slots $\mathcal{BS}$, Cost Function $w : \mathcal{BS} \longrightarrow \mathbb{R}^+$, The influence functions $\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_\ell$; $z_1, z_2, \ldots, z_\ell \in \mathbb{R}^+$, and $\sigma_1, \sigma_2, \ldots, \sigma_\ell \in \mathbb{R}^+$
Result: Non-intersecting sets of slots $S_1, S_2, \ldots, S_\ell \subseteq \mathcal{BS}$

$a \longleftarrow \ell!$, $b \longleftarrow |\mathcal{BS}|!$;
for *All* $p \in [a]$ *AND* $q \in [b]$ **do**
  $\mathcal{M}[p, q] \longleftarrow$ Empty Solution;
  $Flag[p, q] \longleftarrow 1$;
for *All* $p \in [a]$ *AND* $q \in [b]$ **do**
  $\mathbb{S} \leftarrow \emptyset$;
  for $i = 1$ *to* ℓ **do**
    $S_i \longleftarrow \emptyset$;
  for $i = 1$ *to* ℓ **do**
    while $z_i > 0$ *AND* $\mathcal{I}(S_i) < \sigma_i$ **do**
      $s^* \longleftarrow \underset{\substack{s \in \mathcal{BS} \setminus \mathbb{S} \\ w(s) \leq z_i}}{argmax} \mathcal{I}(S_i \cup \{s\}) - \mathcal{I}(S_i)$;
      $S_i \longleftarrow S_i \cup \{s^*\}$, $\mathbb{S} \longleftarrow \mathbb{S} \cup \{s^*\}$;
      $z_i \longleftarrow z_i - w(s^*)$;
    if $\mathcal{I}(S_i) < \sigma_i$ **then**
      $Flag[p, q] \longleftarrow 0$;
      break;
  if $Flag[p, q] = 0$ **then**
    $\mathcal{M}[p, q] \longleftarrow$ Not a Feasible Solution;
  else
    $\mathcal{M}[p, q] \longleftarrow [\{S_1, S_2, \ldots, S_\ell\}, \sum_{s \in S_1 \cup S_2 \cup, \ldots, S_\ell} w(s)]$;
$(S_1^*, S_2^*, \ldots, S_\ell^*) \longleftarrow \underset{(S_1, S_2, \ldots, S_\ell) \in \mathcal{M}}{argmin} \sum_{s \in S_1 \cup S_2 \cup, \ldots, S_\ell} w(s)$;
return $(S_1^*, S_2^*, \ldots, S_\ell^*)$;

Theorem 7 *The sampling-based approach will take* $\mathcal{O}(|\mathcal{X}| \cdot \ell \cdot \frac{\mathcal{Z}_{max}}{w_{min}} \cdot n \cdot t)$ *time and* $\mathcal{O}(|\mathcal{X}| \cdot n + n)$ *space to execute.*

4.3 Greedy Primal–Dual Algorithm for Multi-product Influence Maximization

We propose a primal–dual greedy algorithm to solve the disjoint Multi-product submodular cover problem under budget and disjointness constraints. The algorithm maintains a primal solution, consisting of disjoint slot sets $\mathcal{S}_1, \ldots, \mathcal{S}_\ell$, and a corresponding dual solution, represented by product-specific dual variables $\lambda_1, \ldots, \lambda_\ell$. Each dual variable λ_j reflects the urgency of satisfying the coverage demand σ_j of product j. Initially, all slot sets are empty, and the full budget is available for each product. At each iteration, the algorithm evaluates all feasible slot–product pairs and computes their marginal influence gain per unit cost, weighted by the corresponding dual variable. The pair with the maximum weighted marginal gain is selected, and the slot is

irrevocably assigned to the chosen product. This enforces the disjointness constraint naturally, as each slot can be selected at most once.

After assigning a slot, the remaining budget of the corresponding product is updated. Once a product's coverage requirement is met, its dual variable is set to zero, preventing further slot assignments to that product. This mechanism allows the algorithm to automatically shift its focus toward products whose demands are still unmet. By leveraging the monotonicity and submodularity of the influence functions, the greedy selection ensures diminishing returns, which is critical for both efficiency and approximation guarantees. The primal–dual framework enables a tight coupling between the cost incurred and the progress made toward satisfying coverage demands. We show that, when slot costs are uniform and demands are expressed as per-product cardinality caps, this greedy selection rule coincides with the classical greedy algorithm for submodular maximization over the intersection of two matroids, which yields a formal constant-factor approximation guarantee (Theorem 9). For the general knapsack-budget setting used in our experiments, Algorithm 4 remains a value-density greedy heuristic without a proven worst-case bound, though Remark 2 discusses a standard modification that restores a constant-factor guarantee in that setting as well.

Note that if the algorithm terminates with some unmet demands, this indicates that the instance is infeasible under the given budget and disjointness constraints. In such cases, the algorithm returns a partial allocation that maximizes total covered influence while respecting feasibility.

Algorithm 4: Primal–Dual Greedy for Disjoint Multi-product Slot Selection

Input: Billboard slots $\mathcal{BS}$, products $\mathcal{P} = \{1, \ldots, \ell\}$, influence functions $\{\mathcal{I}_j(\cdot)\}_{j\in\mathcal{P}}$, slot cost $w(\cdot)$, budgets $\{\mathcal{Z}_j\}$, demands $\{\sigma_j\}$

Output: Disjoint slot sets $\mathcal{S}_1, \ldots, \mathcal{S}_\ell$

Initialize $\mathcal{S}_j \leftarrow \emptyset$, remaining budget $R_j \leftarrow \mathcal{Z}_j$ for all j;
Initialize dual variables $\lambda_j \leftarrow \frac{1}{\sigma_j}$;
Initialize assigned slot set $\mathcal{A} \leftarrow \emptyset$;
while *there exists* j *such that* $\mathcal{I}_j(\mathcal{S}_j) < \sigma_j$ **do**
    foreach $s \in \mathcal{BS} \setminus \mathcal{A}$ **do**
        foreach $j \in \mathcal{P}$ *with* $w(s) \le R_j$ **do**
            $\Delta_{s,j} \leftarrow \mathcal{I}_j(\mathcal{S}_j \cup \{s\}) - \mathcal{I}_j(\mathcal{S}_j)$;
            $\rho_{s,j} \leftarrow \lambda_j \cdot \Delta_{s,j}/w(s)$;
    Choose $(s^*, j^*) = \arg\max_{s,j} \rho_{s,j}$;
    if $\Delta_{s^*,j^*} = 0$ **then**
        break;
    $\mathcal{S}_{j^*} \leftarrow \mathcal{S}_{j^*} \cup \{s^*\}$;
    $R_{j^*} \leftarrow R_{j^*} - w(s^*)$;
    $\mathcal{A} \leftarrow \mathcal{A} \cup \{s^*\}$;
    if $\mathcal{I}_{j^*}(\mathcal{S}_{j^*}) \ge \sigma_{j^*}$ **then**
        $\lambda_{j^*} \leftarrow 0$;
return $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_\ell$;

Complexity Analysis.

Let $n = |\mathcal{BS}|$ denote the number of billboard slots and $\ell = |\mathcal{P}|$ the number of products. The time required to evaluate the influence function for a billboard slot is $\mathcal{O}(n \cdot t)$ where t is the number of tuples in the trajectory database. In Line No. 1 to 3 initializing $\mathcal{S}_j \leftarrow \emptyset$ and $R_j \leftarrow \mathcal{Z}_j$ for all $j \in \mathcal{P}$, $\mathcal{A} \leftarrow \emptyset$ and setting $\lambda_j \leftarrow \frac{1}{\sigma_j}$ for all $j \in \mathcal{P}$ takes $\mathcal{O}(\ell)$ time. In Line No. 4, the `while loop` condition can be checked in $\mathcal{O}(\ell)$ time per iteration. The `for loop` at Line No. 5 will iterate for $\mathcal{O}(n)$ times, and the `for loop` at Line No. 6 for each slot s, the loop over products $j \in \mathcal{P}$ with $w(s) \leq R_j$ takes $\mathcal{O}(\ell)$ time in the worst case. Next, in Line No. 7 to 8, computing $\Delta_{s,j}$ requires two influence evaluations and thus costs $\mathcal{O}(n \cdot t \cdot \ell)$ time. Finding (s^*, j^*) over all evaluated pairs takes $O(n\ell)$ time per iteration at Line No. 9 and Line No. 10 to 11 will take $\mathcal{O}(1)$ time to execute. Now, in Line No. 12 to 14 will take $\mathcal{O}(1)$ time to execute. In Line 15 to 16, the coverage satisfaction check will take $\mathcal{O}(n \cdot t \cdot \ell)$ time. Therefore the total time Algorithm 4 will take $\mathcal{O}(n \cdot \ell^2 \cdot t)$. The space complexity is $\mathcal{O}(n + \ell)$.

Theorem 8 *The time and space requirements of Algorithm 4 will be $\mathcal{O}(n \cdot \ell^2 \cdot t)$ and $\mathcal{O}(n+\ell)$, respectively.*

Lemma 1 Algorithm 4 produces disjoint slot sets satisfying budget constraints for all products.

Proof We prove the claim by analyzing the update rules of Algorithm 4. The algorithm maintains a global set of assigned slots, $\mathcal{A}$. A slot s is added to $\mathcal{A}$ immediately after being assigned to some product j, and thereafter s is never considered again for any product. Hence, no slot can be assigned to more than one product, implying that the output sets $\mathcal{S}_1, \ldots, \mathcal{S}_\ell$ are pairwise disjoint. For each product j, the algorithm maintains a remaining budget R_j, initialized as $R_j = \mathcal{Z}_j$. A slot s is added to $\mathcal{S}_j$ only if $w(s) \leq R_j$, and upon assignment the budget is updated as $R_j \leftarrow R_j - w(s)$. Therefore, at any point $\sum_{s \in S_j} w(s) \leq \mathcal{Z}_j$, ensuring that the budget constraint is never violated. Since both disjointness and budget feasibility are preserved throughout the execution of the Algorithm 4, the lemma follows. □

We now show that, for the cardinality-constrained variant of the Disjoint Multi-product Slot Selection Problem introduced in Section 3.4.1 (i.e $|S_j| \leq k_j$ for each product j, together with pairwise disjointness), Algorithm 4 admits a formal worst-case approximation guarantee. The key observation is that this variant can be cast exactly as monotone submodular maximization over the intersection of two matroids, a setting for which the classical greedy algorithm has a well-known guarantee.

Reduction to matroid intersection.

Let $U = BS \times [\ell]$, i.e. one copy (s, j) of every slot $s \in BS$ for every product $j \in [\ell]$. We define two partition matroids on U as follows:

- M_1: partition classes $\{BS \times \{j\}\}_{j \in [\ell]}$, with capacity k_j on class j. Independence in M_1 enforces $|S_j| \leq k_j$ for every product j.

- M_2: partition classes $\{\{s\} \times [\ell]\}_{s \in BS}$, with capacity 1 on each class. Independence in M_2 enforces that every slot s is used by at most one product.

A subset $A \subseteq U$ is feasible for the disjoint problem if and only if A is independent in *both* M_1 and M_2 simultaneously, i.e. A is a common independent set of the two matroids M_1 and M_2. For fixed positive weights $\lambda_1, \ldots, \lambda_\ell$ (in particular $\lambda_j = 1/\sigma_j$, matching the initialization of the dual variables in Algorithm 4), define

$$G(A) \;=\; \sum_{j=1}^{\ell} \lambda_j \, \mathcal{I}_j\big(S_j(A)\big), \qquad S_j(A) := \{\, s \in BS : (s,j) \in A \,\}. \tag{7}$$

Lemma 2 G is a normalized, monotone, submodular function on 2^U.

Proof Normalization and monotonicity are immediate from those of each $\mathcal{I}_j$ and $\lambda_j > 0$. For submodularity, let $A \subseteq B \subseteq U$ and $(s,j) \notin B$. Since $S_j(A) \subseteq S_j(B)$ and $\mathcal{I}_j$ is submodular,

$$\mathcal{I}_j\big(S_j(A) \cup \{s\}\big) - \mathcal{I}_j\big(S_j(A)\big) \;\geq\; \mathcal{I}_j\big(S_j(B) \cup \{s\}\big) - \mathcal{I}_j\big(S_j(B)\big).$$

Multiplying by $\lambda_j \geq 0$ and noting that adding (s,j) to A (resp. B) changes only the j-th summand of G gives $G(A \cup \{(s,j)\}) - G(A) \geq G(B \cup \{(s,j)\}) - G(B)$, as required. □

Theorem 9 *Consider the cardinality-constrained Disjoint Multi-product Slot Selection Problem, and let $(S_1, \ldots, S_\ell)$ denote the output of Algorithm 4 run with unit slot cost and with the early-stopping rule (line 15–16, setting $\lambda_j \leftarrow 0$ once $\mathcal{I}_j(S_j) \geq \sigma_j$) disabled, so that the algorithm keeps adding slots to product j until $|S_j| = k_j$ or no positively-valued slot remains. Let $(S_1^*, \ldots, S_\ell^*)$ be any feasible solution respecting the cardinality and disjointness constraints. Then*

$$\sum_{j=1}^{\ell} \lambda_j \, \mathcal{I}_j(S_j) \;\geq\; \frac{1}{3} \sum_{j=1}^{\ell} \lambda_j \, \mathcal{I}_j(S_j^*). \tag{8}$$

Proof At unit slot cost, Algorithm 4's selection rule is pick the available pair (s,j) with s unused, $|S_j| < k_j$, maximizing marginal gain $\lambda_j \, \Delta_{s,j}$ which is precisely the classical greedy algorithm for maximizing a monotone submodular function (G, by Lemma 2) over the common independent sets of p matroids, here $p = 2$ (namely M_1 and M_2 defined above). By the result of Fisher, Nemhauser and Wolsey [56], this greedy procedure guarantees an objective value at least $\frac{1}{p+1} = \frac{1}{3}$ of the optimum over all common independent sets, which is exactly the right-hand side of Equation 8. □

Corollary 1 Suppose $\lambda_j = 1/\sigma_j$ for each j, and suppose the instance is feasible, i.e. there exists $(S_1^*, \ldots, S_\ell^*)$ satisfying the cardinality and disjointness constraints with $\mathcal{I}_j(S_j^*) \geq \sigma_j$ for every $j \in [\ell]$. Then Algorithm 4 (with early stopping disabled, as in Theorem 9) guarantees

$$\frac{1}{\ell} \sum_{j=1}^{\ell} \frac{\mathcal{I}_j(S_j)}{\sigma_j} \;\geq\; \frac{1}{3}.$$

Proof Since $\mathcal{I}_j(S_j^*) \geq \sigma_j$ for all j, we have $\sum_j \mathcal{I}_j(S_j^*)/\sigma_j \geq \ell$. Dividing both sides of Equation 8 by ℓ and substituting $\lambda_j = 1/\sigma_j$ gives the claim. □

Remark 2 Theorem 9 and Corollary 1 give an *average*, aggregate guarantee across products rather than a per-product worst-case guarantee; strengthening to the latter appears to require handling the adversarial interaction of products competing for the same high-value slots. The bound is established for the cardinality-budget variant ($|S_j| \leq k_j$), matching the original problem definition in Section 3.4.1. Here disjointness and the per-product cardinality caps together form the intersection of two partition matroids, which is what enables the reduction. For the cost/knapsack-budget variant used elsewhere in this section (arbitrary $w(s)$, budgets B_j), Algorithm 4 as stated performs *value-density* greedy (ranking by $\lambda_j \Delta_{s,j}/w(s)$) rather than raw marginal-gain greedy, and it is well known that value-density greedy alone can be made arbitrarily bad on a single knapsack constraint by a single large, high-value item that greedy's early, smaller picks render infeasible. The standard solution (Khuller, Moss and Naor [39]; Sviridenko [57]) is to also consider, for each product, the single best slot it could afford alone, and return the better of the two candidate solutions. This restores a constant-factor guarantee in the knapsack setting at the cost of a trivial $O(n\ell)$ additional check.

5 Experimental Evaluations

This section describes the experimental evaluation of the proposed solution approaches. Initially, we start by describing the datasets.

5.1 Dataset Descriptions.

We used two real-world datasets: New York City (NYC)[1] and Los Angeles (LA)[2], both adopted in previous studies [4, 6, 28, 41]. The NYC dataset contains 227,428 check-ins (Apr 2012–Feb 2013), while the LA dataset includes 74,170 check-ins across 15 streets. Billboard data for both cities is sourced from LAMAR[3], comprising 1,031,040 slots in NYC and 2,135,520 in LA. In Table 1 $|\mathcal{T}|$, $|\mathcal{U}|$, and Avg_{dist} denote the number of trajectories, the number of unique users, and the average distance between trajectories. Similarly, in Table 2 $|\mathcal{B}|$, $|\mathcal{BS}|$, and Avg_{dist} denote the number of billboards, the number of billboard slots, and the average distance between the billboards placed across the city. The descriptions of the datasets are summarized in Table 1 and Table 2.

Dataset	$\lvert\mathcal{T}\rvert$	$\lvert\mathcal{U}\rvert$	Avg_{dist}
NYC	227428	1083	3.12 km
LA	74170	2000	0.61 km

Table 1: Trajectory Datasets

Dataset	$\lvert\mathcal{B}\rvert$	$\lvert\mathcal{BS}\rvert$	Avg_{dist}
NYC	716	1031040	15.07 km
LA	1483	2135520	10.26 km

Table 2: Billboard Datasets

[1] https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page
[2] https://github.com/Ibtihal-Alablani
[3] http://www.lamar.com/InventoryBrowser

5.2 Key Parameters.

All key parameters used in our experiments are summarized in Table 3. The default parameter settings are highlighted in bold.

Table 3: Key Parameters

Parameter	Values
α	$40\%, 60\%, 80\%, \mathbf{100\%}, 120\%$
β	$1\%, 2\%, \mathbf{5\%}, 10\%, 20\%$
$\lvert\mathcal{P}\rvert$	$100, 50, \mathbf{20}, 10, 5$
ϵ	$0.01, 0.05, \mathbf{0.1}, 0.15, 0.2$
λ	$25\text{m}, 50\text{m}, \mathbf{100\text{m}}, 125\text{m}, 150\text{m}$

Advertiser

Given an advertiser $\mathcal{A}$, with a set of products $\mathcal{P} = \{p_1, p_2, \ldots, p_n\}$, the advertiser can be represented as $(\mathcal{A}, p_i, \sigma_i, \mathcal{B})$, where p_i is the i^{th} product, σ_i and $\mathcal{B}$ is the corresponding influence demand and budget.

Demand Supply Ratio.

The ratio $\alpha = \sigma^{\mathcal{P}}/\sigma^*$ represents the proportion of global influence demand ($\sigma^{\mathcal{P}} = \sum_{i=1}^{n} \sigma_i$) to total influence supply ($\sigma^* = \sum_{b \in \mathcal{BS}} \mathcal{I}(b)$). We evaluate five values of $\theta : 40\%, 60\%, 80\%, 100\%,$ *and* 120%.

Advertiser Individual Demand.

The ratio $\beta = \sigma^{\mathcal{P}''}/\sigma^*$ denotes the ratio of average individual demand to total influence supply, where $\sigma^{\mathcal{P}''} = \sigma^{\mathcal{P}}/|\mathcal{P}|$ is the average influence demand per product. This value β adjusts the demand of individual products.

Product Demand.

The demand for each product is generated as $\sigma = \lfloor \omega \cdot \sigma^* \cdot \beta \rfloor$, where ω is a random factor between 0.8 and 1.2 to simulate varying product payments.

Billboard Cost.

The outdoor advertising companies like LAMAR and JCDecaux do not disclose the actual cost of renting a billboard slot. In existing studies [4, 6, 26, 49, 58, 59], the cost of a billboard slot is proportional to its influence. Following this, we calculated the cost of a billboard slot 'bs' using $\lfloor \delta \cdot \frac{\mathcal{I}(bs)}{10} \rfloor$, where $\delta \in [0.8, 1.1]$ and $\mathcal{I}(bs)$ denotes the influence of billboard slot 'bs'.

Advertiser Budget.

Following existing studies [28–30, 58, 60], we set each product's payment proportional to its influence demand: $\mathcal{L}_i = \lfloor \eta \cdot \sigma_i \rfloor$, where η is a random factor in [0.9, 1.1] to model payment variations. The total advertiser budget is then $\mathcal{B} = \sum_{i=1}^{n} \mathcal{L}_i$.

Environment Setup.

All Python code is executed on an HP $Z4$ workstation with 64 GB RAM and a 3.50 GHz Intel Xeon(R) CPU.

5.3 Baseline Methodologies.

Random Allocation (RA)

In this approach, billboard slots are selected uniformly at random and assigned to products until both the influence demand and budget constraints are met.

Top-k Allocation

In this approach, billboard slots are first sorted by their influence values. The highest-influence slots are allotted to the products sequentially until the influence requirements and budget constraints are met.

CELF (common)

In this approach [61], a single priority queue of slots is maintained, ordered by marginal gain summed across all product's unsatisfied demands. At each step, the slot with the highest aggregate marginal gain is added to the shared slot set. The algorithm terminates when influence demand is satisfied or when all slots are exhausted.

CELF (disjoint)

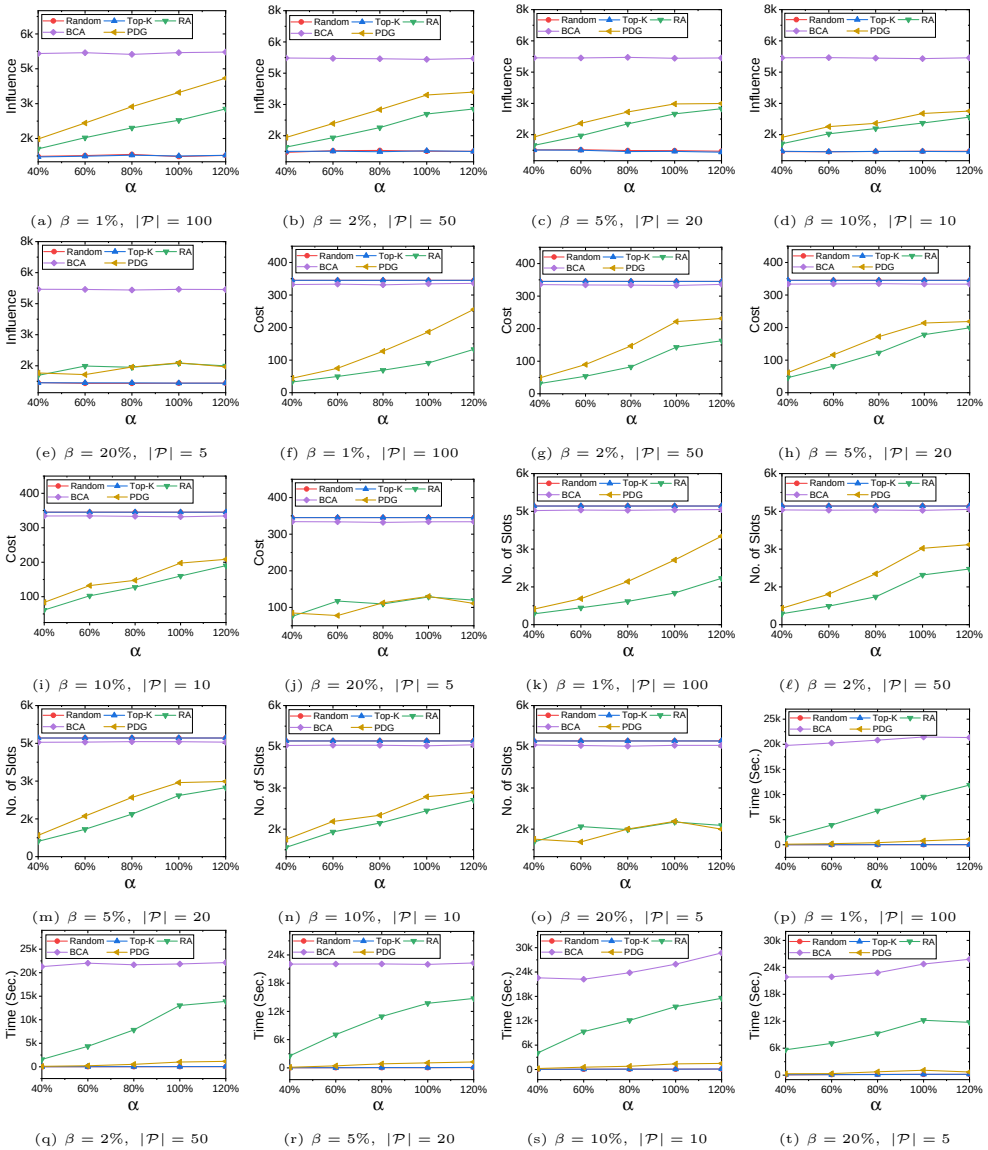
In this approach [61], products are processed with a per-product priority queue. At each global iteration, the slot–product pair with the highest lazily-verified marginal gain per unit cost is selected. The slot is removed from all other product's candidate pools and the corresponding product's remaining demand is updated. This terminates when all product's demands are satisfied or their budgets are exhausted.

5.4 Goals of Our Experimentation

- **RQ1**: Varying α, β, how do the satisfied product change?
- **RQ2**: Varying α, β, how do the total budget is utilized?
- **RQ3**: Varying α, β, how do the computational time change?
- **RQ4**: Varying ϵ and λ, how do the influence quality change?

5.5 Results and Discussions

In this section, we analyze the experimental results to show the effectiveness and efficiency of the proposed algorithms for Multi-product billboard influence maximization. The analysis focuses on four research questions (RQ1–RQ4) to examine how the demand-supply ratio α, the individual product demand ratio β, the approximation parameters ϵ, and the spatial influence range λ affect influence satisfaction, budget utilization, and computational performance. The results are reported for both the NYC and LA datasets and compared against baseline approaches: Random Allocation (RA) and Top-k Allocation, as well as the proposed Bi-Criteria Approximation (BCA), Randomized Allocation (RA), and Primal–Dual Greedy (PDG) algorithms.



(a) $\beta = 1\%,\ |\mathcal{P}| = 100$ (b) $\beta = 2\%,\ |\mathcal{P}| = 50$ (c) $\beta = 5\%,\ |\mathcal{P}| = 20$ (d) $\beta = 10\%,\ |\mathcal{P}| = 10$

(e) $\beta = 20\%,\ |\mathcal{P}| = 5$ (f) $\beta = 1\%,\ |\mathcal{P}| = 100$ (g) $\beta = 2\%,\ |\mathcal{P}| = 50$ (h) $\beta = 5\%,\ |\mathcal{P}| = 20$

(i) $\beta = 10\%,\ |\mathcal{P}| = 10$ (j) $\beta = 20\%,\ |\mathcal{P}| = 5$ (k) $\beta = 1\%,\ |\mathcal{P}| = 100$ (ℓ) $\beta = 2\%,\ |\mathcal{P}| = 50$

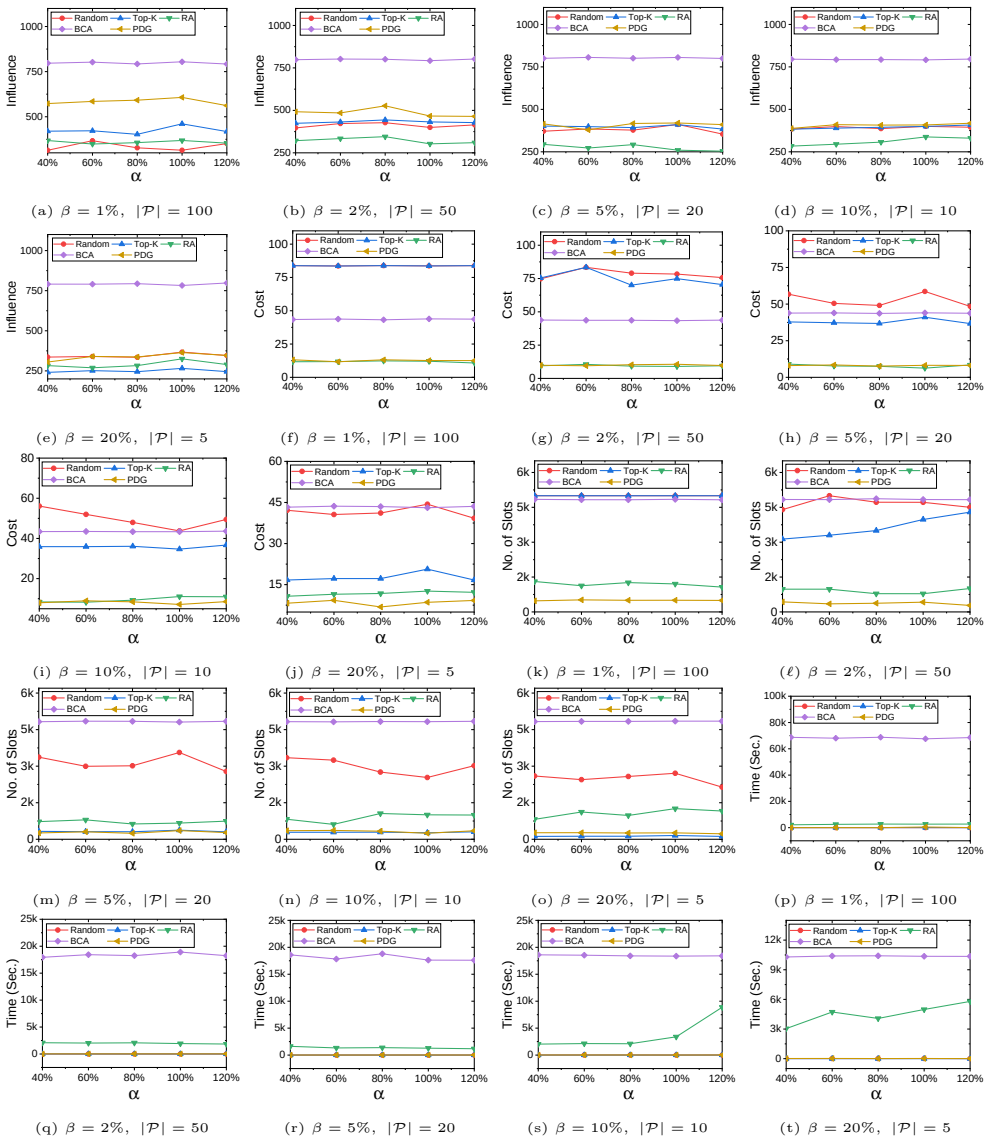
(m) $\beta = 5\%,\ |\mathcal{P}| = 20$ (n) $\beta = 10\%,\ |\mathcal{P}| = 10$ (o) $\beta = 20\%,\ |\mathcal{P}| = 5$ (p) $\beta = 1\%,\ |\mathcal{P}| = 100$

(q) $\beta = 2\%,\ |\mathcal{P}| = 50$ (r) $\beta = 5\%,\ |\mathcal{P}| = 20$ (s) $\beta = 10\%,\ |\mathcal{P}| = 10$ (t) $\beta = 20\%,\ |\mathcal{P}| = 5$

Fig. 3: Varying β and $|\mathcal{P}|$ value α vs. Influence (a, b, c, d, e), α vs. Cost (f, g, h, i, j). α vs. No. of Slots (k, ℓ, m, n, o), α vs. Time, and (p, q, r, s, t) for LA Dataset

Varying α and β Vs. Influence.

From Figure 3(a,b,c,d,e) and Figure 4(a,b,c,d,e), it is clear that with the increase of demand supply ratio (α), within the budget of each product, the influence value increases. When $\beta = 1\%, 2\%, 3\%$ and $\alpha = 40\%, 60\%, 80\%$, the 'PDG' approach satisfies all the demands of the products as shown in Figure 3(a,b,c) and Figure 4(a,b,c).



(a) $\beta = 1\%,\ |\mathcal{P}| = 100$ (b) $\beta = 2\%,\ |\mathcal{P}| = 50$ (c) $\beta = 5\%,\ |\mathcal{P}| = 20$ (d) $\beta = 10\%,\ |\mathcal{P}| = 10$

(e) $\beta = 20\%,\ |\mathcal{P}| = 5$ (f) $\beta = 1\%,\ |\mathcal{P}| = 100$ (g) $\beta = 2\%,\ |\mathcal{P}| = 50$ (h) $\beta = 5\%,\ |\mathcal{P}| = 20$

(i) $\beta = 10\%,\ |\mathcal{P}| = 10$ (j) $\beta = 20\%,\ |\mathcal{P}| = 5$ (k) $\beta = 1\%,\ |\mathcal{P}| = 100$ (ℓ) $\beta = 2\%,\ |\mathcal{P}| = 50$

(m) $\beta = 5\%,\ |\mathcal{P}| = 20$ (n) $\beta = 10\%,\ |\mathcal{P}| = 10$ (o) $\beta = 20\%,\ |\mathcal{P}| = 5$ (p) $\beta = 1\%,\ |\mathcal{P}| = 100$

(q) $\beta = 2\%,\ |\mathcal{P}| = 50$ (r) $\beta = 5\%,\ |\mathcal{P}| = 20$ (s) $\beta = 10\%,\ |\mathcal{P}| = 10$ (t) $\beta = 20\%,\ |\mathcal{P}| = 5$

Fig. 4: Varying β and $|\mathcal{P}|$ value α vs. Influence (a, b, c, d, e), α vs. Cost (f, g, h, i, j). α vs. No. of Slots (k, ℓ, m, n, o), α vs. Time, and (p, q, r, s, t) for NYC Dataset

However, the 'RA', 'Top-k', and 'Random' approaches satisfied fewer products. So the achieved influence of the 'PDG' approach is higher than the 'RA', 'Top-k', and 'Random'. When the α value is 100% or 120%, the influence demand is higher for each product and requires a larger number of slots to satisfy, as shown in Figure 3(d,e) and Figure 4(d,e). Hence, none of the algorithms satisfy all the products. However, in this case, the 'PDG' approach performs better than all other approaches. In the

case of the 'BCA' approach, the influence gain is higher than that of baseline methods such as 'Random' and 'Top-K', as well as the proposed 'RA' and 'PDG' approaches. This happens because in the 'BCA' approach, we find a common set of slots that can satisfy all the products. The influence difference of 'PDG' and baselines is 30% to 35% higher in the NYC dataset and 50% to 70% higher in the LA dataset. The influence difference between the 'PDG' and 'RA' approaches is 20% to 35% higher in the NYC dataset and 40% to 60% higher in the LA dataset. In the experimental setup, when $\alpha \geq 100\%$ and $\beta \geq 10\%$, the influence demand is greater than the influence supply in those cases, and the influence demand of all the products is not satisfied. In these cases, 'PDG' methods perform better than the baselines; however, the 'RA' approach fails to provide a feasible solution. When $\alpha \leq 80\%$ and $\beta \leq 5\%$, the PDG algorithm consistently satisfies the influence demand of almost all products. This indicates that when global demand is well within available supply, 'PDG' efficiently allocates disjoint slots while exploiting diminishing returns via submodularity.

Varying α and β Vs. No. of Slots.

With an increase in α and β, the number of slots allocated for each product increases due to an increase in the demand for influence, as shown in Figure 3(k, ℓ, m, n, o) and Figure 3(k, ℓ, m, n, o). The baseline uses more slots than the proposed 'RA' and 'PDG' approaches; however, the 'Top-k' approach allocates fewer slots than the 'Random' approach. As in the 'Top-k', influential slots are stored in descending order and allocated to each product. The 'BCA' approach allocates more slots than the 'PDG', 'RA', and baseline methods, as it generates a single set of slots to satisfy all the product's demand. However, 'PDG' allocates a reasonable number of slots to meet the influence demand of all products. The 'PDG' consistently selects fewer slots than BCA while achieving comparable influence for disjoint settings. This highlights PDG's ability to exploit high-impact slots and avoid redundant coverage. Next, the 'Top-k' tends to over-select popular slots, causing early saturation and reduced marginal returns. The 'RA' shows highly unstable slot usage due to randomness, further underscoring the importance of principled optimization. The 'RA' approach uses almost 2× fewer slots than 'PDG' and almost 8× fewer than baselines, and 7× fewer than the 'BCA'. So the 'PDG' takes 4× fewer slots than 'BCA' and 5× fewer than baselines.

Efficiency Test.

From Figure 3(p,q,r,s,t) and Figure 4(p,q,r,s,t), we have the five main observations. First, the running time of 'RA' is longer than that of the 'BCA', 'PDG', and baseline methods because 'RA' considers a larger number of slot-product permutations. However, with a fixed, smaller number of permutations (e.g., 500), the runtime is less than the 'BCA' approach. Second, as α and β increase, the computational time for all methods increases. Among the baseline approaches, 'Top-k' takes longer than 'Random'. Third, among the proposed approaches, the 'PDG' approach takes less time. In the case of the 'BCA' approach with an increase of α and β, the computation time change is less because the initially selected slots using continuous greedy are significant, such that they satisfy most of the product's influence demand. So, further allocation using greedy based on marginal gain requires less time to compute. Fourth, among all

methods, 'PDG' shows the best scalability, significantly outperforming 'BCA', whose continuous greedy and rounding steps incur substantial overhead. The 'PGD' is computationally cheaper, but delivers substantially inferior solution quality. Fifth, as α and β increase from 40% to 120% and 1% to 20%, respectively, the computational time for all algorithms increases. However, the runtime increase rate for the 'RA' approach is high due to randomness and slot-product permutations.

Product Vs. Budget.

From Figure 3(f,g,h,i,j) and Figure 4(f,g,h,i,j), we have five observations. First, with increasing demand supply ratio (α) and average individual demand (β), the influence demand for each product increases, and a larger number of slots are allocated. Hence, the product allocation cost increases. Second, in terms of cost, 'RA' outperforms 'Random', 'Top-k', 'PDG', and 'BCA' methods, and 'RA' uses the minimum cost to satisfy the influence demand for all the products. However, 'RA' does not outperform the 'PDG' in terms of influence. Among the proposed approaches, the 'BCA' incurs a higher cost than 'RA' and 'PDG'. Third, the baseline methods 'Random' and 'Top-k' incur a higher cost to satisfy all products because of the large number of slot selections. Fourth, the 'PDG' approach incurs higher costs than 'RA' and covers a larger number of product's influence demand. However, 'PDG' takes a lower cost than 'BCA' and the baseline methods to satisfy the highest number of products. Fifth, the 'RA' and 'Top-k' baselines exhibit inefficient budget utilization, often exhausting budgets early and failing to meet influence demands. This inefficiency arises from failing to account for marginal influence-to-cost ratios and the product's relevance. Sixth, in contrast, 'PDG' shows higher cost efficiency, achieving higher influence at comparable or lower cost. This is because 'PDG' explicitly optimizes the weighted marginal gain per unit cost and dynamically shifts focus toward unmet products using dual variables. 'BCA', while achieving high influence, often incurs a higher total cost due to selecting a larger shared slot set, making it less suitable when strict budget control is required.

Cost Vs. Objective Satisfaction.

We note that our formal objective (Section 3.4.1) is to minimize allocation cost subject to each product's influence demand being satisfied. Figures 3(a–e)/4(a–e) report influence satisfaction and Figures 3(f–j)/4(f–j) report the corresponding cost under identical parameter settings, allowing direct comparison. These results show that PDG achieves the lowest cost among methods that satisfy the largest number of product's demands under high-demand settings ($\alpha \geq 100\%$, $\beta \geq 10\%$). The PDG satisfies more products at lower cost than RA, BCA, and the baselines, directly reflecting effective cost-minimization under the demand-feasibility constraint. We reiterate this connection explicitly here to clarify that PDG's advantage is not merely in raw influence, but in cost-efficient satisfaction of the stated demand constraints.

Impact of Different Parameters.

We have experimented with different values of α, β, and the number of products to show that different cases occur due to the varying demand of the products. We have reported the experimental results of different settings: $\alpha = 40\%, 60\%, 80\%, 100\%, 120\%$

and $\beta = 1\%, 2\%, 5\%, 10\%, 20\%$, and the number of products ($|\ell|$) are $100, 50, 20, 10, 5$. The additional parameters used in our experiments are the distance parameter λ and the approximation parameter ϵ. The λ determines the distance from which a billboard slot can influence a number of trajectories. The ϵ controls the accuracy of approximation. The smaller $\epsilon \in (0, 1)$ provides a better approximation, but increases runtime. In our experiments, we set $\epsilon = 0.1$ and $\lambda = 100m$ as the default setting. Experiments varying *epsilon* and λ show predictable trends. Smaller ϵ values improve influence quality but increase runtime, consistent with theoretical expectations. Increasing λ expands the spatial influence region, leading to higher influence values but also increased overlap and cost. Importantly, 'PDG' remains stable across parameter variations, showing graceful degradation rather than abrupt performance loss. This robustness is critical in practice, where precise parameter tuning may not be feasible.

Additional Discussion.

Additionally, we compare our proposed methods with the CELF algorithm shown in Table 4. The results show that our proposed PDG algorithm pareto-dominates CELF (disjoint) across every metric on both datasets. On the NYC, PDG achieves 16.6% higher influence, 44.8% lower cost, 19.9% fewer slots, 13 percentage points higher product satisfaction (98% vs. 85%), and 2.2× faster runtime. In the case of LA, PDG achieves 6.8% higher influence, 15.4% lower cost, 15.7% fewer slots, full satisfaction (100% vs. 90%), and a 12× runtime improvement. For the common-slot variant, BCA achieves higher influence and full demand satisfaction than CELF (common) on both datasets, though at higher computational cost a trade-off directly explained by the complexity gap between the continuous-greedy-with-rounding procedure (Theorem 3, $O(n^2\ell/\epsilon)$) and lazy greedy's cheaper per iteration cost. We believe this addition substantively strengthens our empirical evaluation and directly demonstrates that our methods, particularly PDG, remain competitive against and often superior to the classical submodular optimization baselines, not just naive heuristics.

6 Concluding Remarks

In this paper, we study the problem of Multi-product influence maximization, motivated by practical advertising scenarios in which a commercial house plans to promote multiple products to its customers. We have studied the problem in two variants. In the first one, the same set of slots will be used for promoting multiple products. This problem asks to choose a given number of highly influential slots so that the influence demand of each product is satisfied. Whereas in the other case, for every product, there exists both a budget for the number of slots and influence demand. The task is to choose a disjoint subset of slots for each product such that their influence on demand is satisfied. Both variants were shown to be NP-hard via their connection to multi-submodular cover problems. For these problems, we have proposed several solution approaches. For the Common Multi-product Slot Selection Problem, we have proposed a continuous greedy algorithm and a bi-criteria approximation algorithm. For the Disjoint Multi-product Slot Selection Problem, we have proposed a sampling-based randomized algorithm with bounded estimation error and a primal-dual greedy

Table 4: Comparison with CELF baseline at default parameter settings ($\alpha = 100\%$, $\beta = 1\%$, $|\mathcal{P}| = 100$, $\epsilon = 0.1$, $\lambda = 100$m).

Dataset	Method	Influence	Cost	# Slots	Satisfied Product	Time (sec.)
NYC	Random	314.50	83.65	1568	74	4
	Top-k	460.61	83.65	1390	82	8
	RA	368.49	12.07	1204	64	2657
	CELF (common)	601.36	30.42	4473	95	4894
	BCA	804.49	43.87	4844	100	67588
	CELF (disjoint)	520.89	22.82	628	85	1652
	PDG	607.08	12.60	503	98	752
LA	Random	724.36	345.25	4712	19	13
	Top-k	749.07	345.25	3784	20	18
	RA	2283.38	91.35	1256	64	9540
	CELF (common)	4800.86	280.63	3800	95	14985
	BCA	5195.13	334.41	4560	100	101450
	CELF (disjoint)	3258.42	220.58	3045	90	9473
	PDG	3481.36	186.67	2567	100	783

algorithm that efficiently enforces budget and disjointness constraints and scales well to large datasets. We have derived the relationship between sample size and solution accuracy. We have analyzed all the algorithms for their time and space requirements and experimented with real-world billboard and trajectory datasets. The experimentation shows that the proposed methods significantly outperform baseline approaches. In particular, 'PDG' consistently achieves higher influence satisfaction and better budget utilization under high-demand, limited-supply conditions. In the future, this work can be extended to dynamic and online settings to incorporate multi-advertiser competition and multi-channel influence models.

References

[1] Lamar Advertising Company: Research - Lamar Advertising. Accessed: 2025-05-30 (2025). https://www.lamar.com/howtoadvertise/Research/

[2] Top Media Advertising: Billboard Advertising Statistics. Accessed: 2025-05-30 (2025). https://topmediadvertising.co.uk/billboard-advertising-statistics/

[3] The Business Research Company: Billboard and Outdoor Advertising Global Market Report. Accessed: 2025-05-30 (2025). https://www.thebusinessresearchcompany.com/report/billboard-and-outdoor-advertising-global-market-report

[4] Ali, D., Banerjee, S., Prasad, Y.: Influential billboard slot selection using pruned submodularity graph. In: International Conference on Advanced Data Mining and Applications, pp. 216–230 (2022). Springer

[5] Ali, D., Bhagat, A.K., Banerjee, S., Prasad, Y.: Efficient algorithms for regret minimization in billboard advertisement (student abstract). In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, pp. 16148–16149 (2023)

[6] Ali, D., Banerjee, S., Prasad, Y.: Influential billboard slot selection using spatial clustering and pruned submodularity graph. arXiv preprint arXiv:2305.08949 (2023)

[7] Zhang, P., Bao, Z., Li, Y., Li, G., Zhang, Y., Peng, Z.: Towards an optimal outdoor advertising placement: When a budget constraint meets moving trajectories. ACM Transactions on Knowledge Discovery from Data (TKDD) **14**(5), 1–32 (2020)

[8] Wang, L., Yu, Z., Guo, B., Yang, D., Ma, L., Liu, Z., Xiong, F.: Data-driven targeted advertising recommendation system for outdoor billboard. ACM Transactions on Intelligent Systems and Technology (TIST) **13**(2), 1–23 (2022)

[9] Zhang, Y., Li, Y., Bao, Z., Mo, S., Zhang, P.: Optimizing impression counts for outdoor advertising. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. KDD '19, pp. 1205–1215. Association for Computing Machinery, New York, NY, USA (2019)

[10] Feder, G., Just, R.E., Zilberman, D.: Adoption of agricultural innovations in developing countries: A survey. Economic Development and Cultural Change **33**(2), 255–298 (1985)

[11] Lee, S.: Examining the factors that influence early adopters' smartphone adoption: The case of college students. Telematics and Informatics **31**(2), 308–318 (2014)

[12] Sierzchula, W., Bakker, S., Maat, K., van Wee, B.: The influence of financial incentives and other socio-economic factors on electric vehicle adoption. Energy Policy **68**, 183–194 (2014)

[13] Zhou, F., Wu, H., Trajcevski, G., Khokhar, A., Zhang, K.: Semi-supervised trajectory understanding with poi attention for end-to-end trip recommendation. ACM Transactions on Spatial Algorithms and Systems (TSAS) **6**(2), 1–25 (2020)

[14] AlKheder, S.: Experimental road safety study of the actual driver reaction to the street ads using eye tracking, multiple linear regression and decision trees methods. Expert Systems with Applications **252**, 124222 (2024)

[15] Liu, D., Weng, D., Li, Y., Bao, J., Zheng, Y., Qu, H., Wu, Y.: Smartadp: Visual analytics of large-scale taxi trajectories for selecting billboard locations. IEEE Transactions on Visualization and Computer Graphics **23**(1), 1–10 (2017)

[16] Gudmundsson, J., Pfeifer, J., Seybold, M.P.: On practical nearest sub-trajectory

queries under the fréchet distance. ACM Transactions on Spatial Algorithms and Systems **9**(2), 1–24 (2023)

[17] Hung, H.-J., Yang, D.-N., Lee, W.-C.: Social influence-aware reverse nearest neighbor search. ACM Transactions on Spatial Algorithms and Systems (TSAS) **2**(3), 1–35 (2016)

[18] Uddin, R., Mahin, M.T., Rajan, P., Ravishankar, C.V., Tsotras, V.J.: Dwell regions: Generalized stay regions for streaming and archival trajectory data. ACM Transactions on Spatial Algorithms and Systems **9**(2), 1–35 (2023)

[19] Dai, J., Yang, B., Guo, C., Ding, Z.: Personalized route recommendation using big trajectory data. In: 2015 IEEE 31st International Conference on Data Engineering, pp. 543–554 (2015). IEEE

[20] Qu, B., Yang, W., Cui, G., Wang, X.: Profitable taxi travel route recommendation based on big taxi trajectory data. IEEE Transactions on Intelligent Transportation Systems **21**(2), 653–668 (2019)

[21] Huang, Y., Xiao, Z., Yu, X., Wang, D., Havyarimana, V., Bai, J.: Road network construction with complex intersections based on sparsely sampled private car trajectory data. ACM Transactions on Knowledge Discovery from Data (TKDD) **13**(3), 1–28 (2019)

[22] Xue, Q., Wang, K., Lu, J.J., Liu, Y.: Rapid driving style recognition in car-following using machine learning and vehicle trajectory data. Journal of advanced transportation **2019** (2019)

[23] Qin, K.K., Ren, Y., Shao, W., Lake, B., Privitera, F., Salim, F.D.: Multiple-level point embedding for solving human trajectory imputation with prediction. ACM Transactions on Spatial Algorithms and Systems **9**(2), 1–22 (2023)

[24] Musleh, M., Mokbel, M.F.: Let's speak trajectories: A vision to use nlp models for trajectory analysis tasks. ACM Transactions on Spatial Algorithms and Systems (2023)

[25] Li, Y., Fan, J., Wang, Y., Tan, K.-L.: Influence maximization on social graphs: A survey. IEEE Transactions on Knowledge and Data Engineering **30**(10), 1852–1872 (2018)

[26] Ali, D., Kumar, H., Banerjee, S., Prasad, Y.: An effective tag assignment approach for billboard advertisement. arXiv preprint arXiv:2409.02455 (2024)

[27] Kempe, D., Kleinberg, J., Tardos, É.: Maximizing the spread of influence through a social network. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 137–146 (2003)

[28] Ali, D., Banerjee, S., Prasad, Y.: Regret minimization in billboard advertisement under zonal influence constraint. In: Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing, pp. 329–336 (2024)

[29] Ali, D., Banerjee, S., Prasad, Y.: Minimizing regret in billboard advertisement under zonal influence constraint. arXiv preprint arXiv:2402.01294 (2024)

[30] Zhang, Y., Li, Y., Bao, Z., Zheng, B., Jagadish, H.: Minimizing the regret of an influence provider. In: Proceedings of the 2021 International Conference on Management of Data, pp. 2115–2127 (2021)

[31] Zhou, Z., Wu, W., Li, X., Lee, M.L., Hsu, W.: Maxfirst for maxbrknn. In: 2011 IEEE 27th International Conference on Data Engineering, pp. 828–839 (2011)

[32] Li, G., Chen, S., Feng, J., Tan, K.-l., Li, W.-s.: Efficient location-aware influence maximization. In: Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data. SIGMOD '14, pp. 87–98. Association for Computing Machinery, New York, NY, USA (2014)

[33] Wang, M., Li, H., Cui, J., Deng, K., Bhowmick, S.S., Dong, Z.: Pinocchio: Probabilistic influence-based location selection over moving objects. IEEE Transactions on Knowledge and Data Engineering **28**(11), 3068–3082 (2016)

[34] Liu, P., Wang, M., Cui, J., Li, H.: Top-k competitive location selection over moving objects. Data Science and Engineering **6**(4), 392–401 (2021)

[35] Mitra, S., Saraf, P., Bhattacharya, A.: Tips: mining top-k locations to minimize user-inconvenience for trajectory-aware services. IEEE Transactions on Knowledge and Data Engineering **33**(3), 1238–1250 (2019)

[36] Wong, R.C.-W., Özsu, M.T., Yu, P.S., Fu, A.W.-C., Liu, L.: Efficient method for maximizing bichromatic reverse nearest neighbor. Proc. VLDB Endow. **2**(1), 1126–1137 (2009)

[37] Xia, T., Zhang, D., Kanoulas, E., Du, Y.: On computing top-t most influential spatial sites. In: Proceedings of the 31st International Conference on Very Large Data Bases, pp. 946–957 (2005)

[38] Zhang, P., Bao, Z., Li, Y., Li, G., Zhang, Y., Peng, Z.: Towards an optimal outdoor advertising placement: When a budget constraint meets moving trajectories. ACM Trans. Knowl. Discov. Data **14**(5) (2020) https://doi.org/10.1145/3350488

[39] Khuller, S., Moss, A., Naor, J.S.: The budgeted maximum coverage problem. Information processing letters **70**(1), 39–45 (1999)

[40] Ali, D., Banerjee, S., Prasad, Y.: Influential billboard slot selection using pruned submodularity graph. In: Chen, W., Yao, L., Cai, T., Pan, S., Shen, T., Li, X.

(eds.) Advanced Data Mining and Applications, pp. 216–230. Springer, Cham (2022)

[41] Ali, D., Banerjee, S., Prasad, Y.: Influential slot and tag selection in billboard advertisement. In: International Conference on Database and Expert Systems Applications, pp. 276–290 (2025). Springer

[42] Ali, D., Islam, R., Banerjee, S.: Multi-product influence maximization in billboard advertisement. In: Proceedings of the 13th ACM IKDD International Conference on Data Science, pp. 10–18 (2025)

[43] Ali, D., Banerjee, S., Prasad, Y.: Balanced popularity in multi-product billboard advertisement. In: Australasian Database Conference, pp. 266–278 (2025). Springer

[44] Ali, D., Banerjee, S., Jain, S., Prasad, Y.: Fairness driven slot allocation problem in billboard advertisement. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 55–67 (2025). Springer

[45] Aslay, C., Lu, W., Bonchi, F., Goyal, A., Lakshmanan, L.V.: Viral marketing meets social advertising: Ad allocation with minimum regret. arXiv preprint arXiv:1412.1462 (2014)

[46] Ali, D., Banerjee, S., Prasad, Y.: Toward regret-free slot allocation in billboard advertisement. International Journal of Data Science and Analytics **20**(3), 1951–1975 (2025)

[47] Wang, L., Yu, Z., Yang, D., Ma, H., Sheng, H.: Efficiently targeted billboard advertising using crowdsensing vehicle trajectory data. IEEE Transactions on Industrial Informatics **16**(2), 1058–1066 (2019)

[48] Guo, L., Zhang, D., Cong, G., Wu, W., Tan, K.-L.: Influence maximization in trajectory databases. IEEE Transactions on Knowledge and Data Engineering **29**(3), 627–641 (2016)

[49] ZHANG, Y., LI, Y., BAO, Z., MO, S., ZHANG, P.: Optimizing impression counts for outdoor advertising.(2019). In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, Alaska, pp. 4–8 (2019)

[50] Wang, L., Yu, Z., Yang, D., Ma, H., Sheng, H.: Efficiently targeted billboard advertising using crowdsensing vehicle trajectory data. IEEE Transactions on Industrial Informatics **16**(2), 1058–1066 (2020)

[51] Ke, X., Khan, A., Cong, G.: Finding seeds and relevant tags jointly: For targeted influence maximization in social networks. In: Proceedings of the 2018 International Conference on Management of Data, pp. 1097–1111 (2018)

[52] Calinescu, G., Chekuri, C., Pál, M., Vondrák, J.: Maximizing a submodular set function subject to a matroid constraint. In: International Conference on Integer Programming and Combinatorial Optimization, pp. 182–196 (2007). Springer

[53] Vondrák, J.: Submodularity in combinatorial optimization (2007)

[54] Chekuri, C., Inamdar, T., Quanrud, K., Varadarajan, K., Zhang, Z.: Algorithms for covering multiple submodular constraints and applications. Journal of combinatorial optimization **44**(2), 979–1010 (2022)

[55] Feldman, M., Naor, J., Schwartz, R.: A unified continuous greedy algorithm for submodular maximization. In: 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science, pp. 570–579 (2011). IEEE

[56] Fisher, M.L., Nemhauser, G.L., Wolsey, L.A.: An analysis of approximations for maximizing submodular set functions—ii. In: Polyhedral Combinatorics: Dedicated to the Memory of D. R. Fulkerson, pp. 73–87. Springer, Berlin, Heidelberg (2009)

[57] Sviridenko, M.: A note on maximizing a submodular set function subject to a knapsack constraint. Operations Research Letters **32**(1), 41–43 (2004)

[58] Aslay, C., Bonchi, W.L.F., Goyal, A., Lakshmanan, L.V.: Viral marketing meets social advertising: Ad allocation with minimum regret. Proceedings of the VLDB Endowment **8**(7) (2015)

[59] Banerjee, S., Pal, B., Jenamani, M.: Budgeted influence maximization with tags in social networks. In: Web Information Systems Engineering–WISE 2020: 21st International Conference, Amsterdam, The Netherlands, October 20–24, 2020, Proceedings, Part I 21, pp. 141–152 (2020). Springer

[60] Aslay, C., Lakshmanan, F.B.L.V., Lu, W.: Revenue maximization in incentivized social advertising. Proceedings of the VLDB Endowment **10**(11) (2017)

[61] Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., Glance, N.: Cost-effective outbreak detection in networks. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 420–429 (2007)