



A Survey on Influence Maximization: From an ML-Based Combinatorial Optimization

YANDI LI, HAOBO GAO, and YUNXUAN GAO, Department of Computer Science, BNU-HKBU United International College, China
JIANXIONG GUO, Advanced Institute of Natural Sciences, Beijing Normal University, China and Guangdong Key Lab of AI and Multi-Modal Data Processing, BNU-HKBU United International College, China
WEILI WU, Department of Computer Science, The University of Texas at Dallas, USA

Influence Maximization (IM) is a classical combinatorial optimization problem, which can be widely used in mobile networks, social computing, and recommendation systems. It aims at selecting a small number of users such that maximizing the influence spread across the online social network. Because of its potential commercial and academic value, there are a lot of researchers focusing on studying the IM problem from different perspectives. The main challenge comes from the NP-hardness of the IM problem and #P-hardness of estimating the influence spread, thus traditional algorithms for overcoming them can be categorized into two classes: heuristic algorithms and approximation algorithms. However, there is no theoretical guarantee for heuristic algorithms, and the theoretical design is close to the limit. Therefore, it is almost impossible to further optimize and improve their performance.

With the rapid development of artificial intelligence, technologies based on Machine Learning (ML) have achieved remarkable achievements in many fields. In view of this, in recent years, a number of new methods have emerged to solve combinatorial optimization problems by using ML-based techniques. These methods have the advantages of fast solving speed and strong generalization ability to unknown graphs, which provide a brand-new direction for solving combinatorial optimization problems. Therefore, we abandon the traditional algorithms based on iterative search and review the recent development of ML-based methods, especially Deep Reinforcement Learning, to solve the IM problem and other variants in social networks. We focus on summarizing the relevant background knowledge, basic principles, common methods, and applied research. Finally, the challenges that need to be solved urgently in future IM research are pointed out.

CCS Concepts: • **Networks** → **Network algorithms**; • **Theory of computation** → **Design and analysis of algorithms**; • **Computing methodologies** → **Machine learning approaches**; **Knowledge representation and reasoning**;

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant No. 62202055, the Start-up Fund from Beijing Normal University under Grant No. 310432104, the Start-up Fund from BNU-HKBU United International College under Grant No. UICR0700018-22, and the Project of Young Innovative Talents of Guangdong Education Department under Grant No. 2022KQNCX102.

Authors' addresses: Y. Li, H. Gao, and Y. Gao, Department of Computer Science, BNU-HKBU United International College, Zhuhai, Guangdong 519087, China; emails: liyandi@uic.edu.cn, {p930026030, p930026032}@mail.uic.edu.cn; J. Guo (corresponding author), Advanced Institute of Natural Sciences, Beijing Normal University, Zhuhai, Guangdong 519087, China and Guangdong Key Lab of AI and Multi-Modal Data Processing, BNU-HKBU United International College, Zhuhai, Guangdong 519087, China; email: jianxiongguo@bnu.edu.cn; W. Wu, Department of Computer Science, The University of Texas at Dallas, Richardson, Texas 75080; email: weiliwu@utdallas.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](https://permissions.acm.org).

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.

1556-4681/2023/07-ART133 \$15.00

<https://doi.org/10.1145/3604559>

Additional Key Words and Phrases: Influence maximization, machine learning, social networks, combinatorial optimization, deep reinforcement learning, graph embedding

ACM Reference format:

Yandi Li, Haobo Gao, Yunxuan Gao, Jianxiong Guo, and Weili Wu. 2023. A Survey on Influence Maximization: From an ML-Based Combinatorial Optimization. *ACM Trans. Knowl. Discov. Data.* 17, 9, Article 133 (July 2023), 50 pages.

<https://doi.org/10.1145/3604559>

1 INTRODUCTION

With the rapid development of the mobile internet, social media platforms and content-sharing platforms are rapidly emerging. Compared with newspapers, radio, television, and other traditional media, mobile Internet-based media has faster speed, lower cost, and more diverse forms. Thus, the commercial social giants represented by Weibo, Facebook, TikTok, and Twitter were quickly born and strengthened in the last two decades. All these applications are based on the **Online Social Network (OSN)**, which is a structure composed of individuals and social interactions between them. It acts as a medium for spreading information, ideas, or opinions among users through the effect of “word-of-mouth”. For example, Facebook has over 1.52 billion daily active users, and Twitter has 321 million monthly active users. The phenomenon of information diffusion in social networks has attracted the attention of scholars, businessmen, and politicians because of its huge value in many aspects. These values can be used in many specific applications, such as virus marketing [60, 171], network monitoring [119], rumor blocking [41, 78, 87, 88], and social recommendation [232]. As a result, they jointly promote research on the **Influence Maximization (IM)** problem.

The IM problem is the most important algorithmic problem in the research of information diffusion in social networks, which has been extensively studied for 20 years. Kempe et al. [104] formally defined the IM problem as a **combinatorial optimization (CO)** problem in 2003, which aims to select a size- k seed set such that the expected number of activated users (influence spread) can be maximized given a diffusion model such as **Independent Cascade (IC)** model and **Linear Threshold (LT)** model. This problem persists because of its inherent complexity. First, it has been proven that the IM problem is NP-hard [104], thus there is no exact solution within polynomial running time. Second, due to the stochasticity of an information diffusion process, it is #P-hard to accurately compute the influence spread given a seed set under the commonly used diffusion models [45, 47]. Third, the IM problem depends on the given diffusion model and its corresponding parameters, but it is difficult to get it in practical applications. They have hindered the application of IM algorithms in large-scale social scenarios together.

In the last two decades, the above three challenges have motivated a great deal of research to develop efficient algorithms for the IM problem. The existing work is mainly devoted to studying how to estimate the influence spread for the IM problem and using the design techniques of heuristic algorithms or approximate algorithms to seek a balance between effectiveness, efficiency, and generalization ability. In this article, heuristic algorithms and approximate algorithms are collectively referred to as “traditional algorithms”. The heuristic algorithm is efficient and works well in some cases, but it has to be designed for a specific diffusion model, which is not universal, and there is no theoretical guarantee of the worst case. The time complexity of the approximation algorithms cannot be further reduced on the premise of taking approximation into account, and it is difficult to meet the demand for algorithm performance due to the growing network scale. In addition, traditional algorithms do not have good generalization ability. In practical applications, frequent users’ access leads to dynamic changes in social networks. When the network changes

greatly, the traditional algorithm generally needs to search and solve again, or adjust the heuristic rules through trial and error to achieve better results. Therefore, these traditional methods have high computational costs and are difficult to be extended to large-scale, online, and real-time optimization problems, especially for variant IM problems.

With the development of artificial intelligence technology, the method based on **Machine Learning (ML)** has surpassed traditional methods and achieved remarkable success in many fields. For example, **Deep Reinforcement Learning (DRL)** [148], as an important branch of ML, achieves function approximation through **Deep Neural Networks (DNNs)**, showing amazing learning and decision-making ability. It chooses an action according to the current state and adjusts its own strategy according to the return brought by the action, which is called sequential decision-making. Based on that, there are several studies that recently begin to use DRL to learn a generalized policy for a certain CO problem on graphs [19, 59, 107, 152]. The IM problem is a typical CO problem, that is, the optimal choice of decision variables in discrete space is naturally coupled with the action choice of **reinforcement learning (RL)**. DRL has the characteristics of “offline training and online decision-making”, which makes it possible to solve CO problems in online real-time manners.

Therefore, we can see that it is a feasible direction to use ML-based methods to solve the IM problem in social networks, so as to overcome the bottleneck of traditional algorithms. In this article, we aim to make a comprehensive overview of the IM problem from an ML-based CO perspective. Here, we will focus on the following three aspects for detailed display.

- **The IM problem and traditional algorithms (Sections 3 and 5).** We first review the basic diffusion models, the definition of the IM problem, its properties, and computational complexity. Then, we introduce the classical greedy algorithm to study the IM problem and review traditional algorithms, heuristic and approximation algorithms, according to the taxonomy of existing IM algorithms [127], including simulation-, proxy-, and sampling-based approaches. At last, we summarize the design bottlenecks and shortcomings of traditional algorithms and point out the necessity of adopting ML-based methods at the same time.
- **DRL framework for the IM problem (Sections 6 and 7).** We first review the background and fundamental methods of adopting DRL to solve CO problems. These methods are mainly based on two categories: **Pointer Networks (Ptr-Nets)** and **Graph Neural Networks (GNNs)**, each of which has different characteristics and applicability. Here, we mainly introduce the related concepts and methods that will be used in solving the IM problem later. Then, we review the current DRL methods to solve the IM problem, which relies on the techniques of graph embeddings and the DRL framework to solve CO problems. Finally, we also summarize some ML-based methods from other niche technology stacks to address the IM problem.
- **The variants of the IM problem and future research (Sections 8 and 9).** Based on the IM problem, there are a large number of variants that are adapted to different application scenarios, including the topic-aware IM problem [39, 194], the time-aware IM problem [80, 177, 199], the location-aware IM problem [29, 48, 122], the target-aware IM problem [30, 79], and so on. Thus, we review the latest ML-based works that are used to solve different kinds of variant IM problems. Finally, we summarize the advantages and research opportunities of adopting ML-based methods to address the IM problem and the variants of the IM problem in different social applications.

As a special CO problem, the IM problem has inherent complexity and vitality. To the best of our knowledge, we are the first to survey the IM problem from an ML-based CO perspective. The architecture of this article is shown in Figure 1. It involves a variety of emerging ML technologies such as RL, graph learning, and online learning. Compared with traditional algorithms, it has great

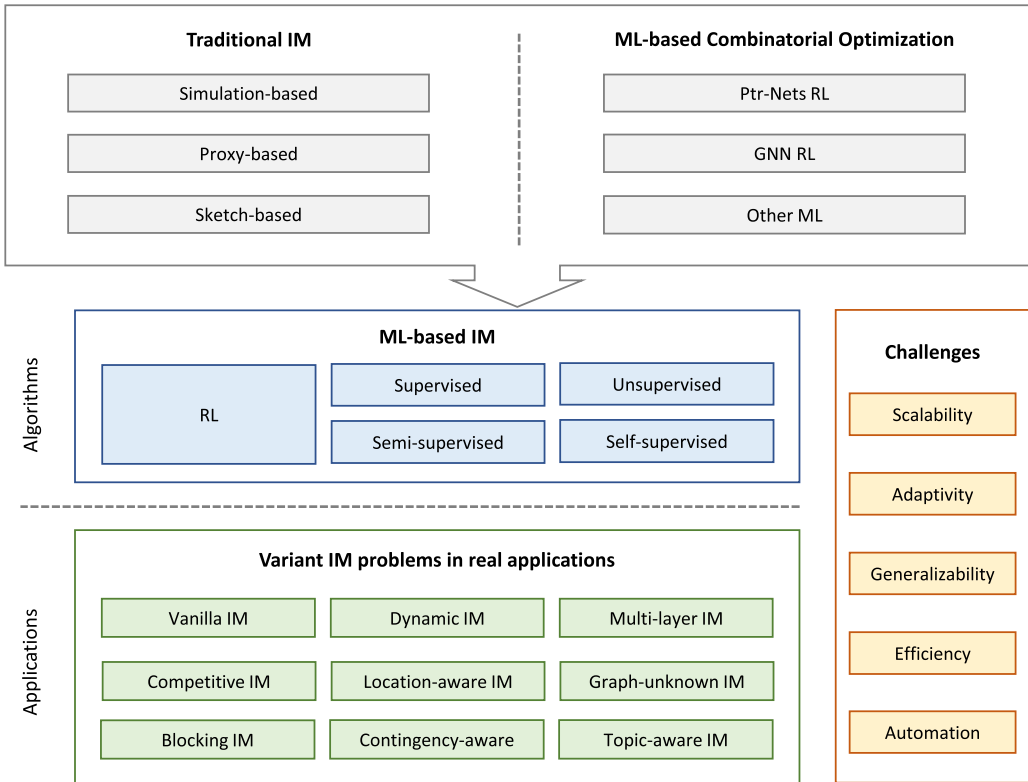


Fig. 1. The architecture of this survey.

advantages in time complexity, generalization ability, and real-time performance. This article not only overlooks the macro blueprint of this research field from a high-level perspective but also explains it in detail. This is our main contribution.

2 RELATED WORKS

The IM problem has been a hot research topic in the field of network science, social computing, and CO for the past 20 years since Kempe et al. [104] first defined the IM problem in 2003. There are a lot of existing surveys [10, 44, 76, 127, 183, 193] about the IM problem and social analysis. In [10, 44, 76, 183, 193], they summarized some works of early IM algorithms, including plenty of diffusion models, social influence analysis methods, and commonly used heuristic algorithms. Brogs et al. [23] proposed the technique of reverse influence sampling in 2014. Since then, the research of approximate algorithms for the IM problem has made breakthrough progress. In 2017, Arora et al. [10] first proposed a sound taxonomy of existing IM approaches with three categories: simulation-based, sampling-based, and approximate scoring based, according to the manners of seed selection, which was a milestone of this area. They comprehensively surveyed and benchmarked the 11 most promising approaches at that moment with systematically designed experiments from aspects of spread quality, time efficiency, main memory footprint, and robustness, which was a pioneering work that thoroughly evaluated and analyzed all the state-of-the-art IM algorithms on a common platform. Afterward, Li et al. [127] gave a comprehensive survey on the IM problem from an algorithmic perspective, which covered almost all the traditional algorithms of the IM problem before this, including heuristic and approximation algorithms. They

extended the taxonomy in [10] and divided traditional algorithms into three categories similarly: simulation-, proxy-, and sampling-based approaches. The introduction of traditional algorithms in this article will continue this classification method and supplement some of the latest works.

After 2018, there have been some new surveys [2, 12, 15, 31, 34, 95, 165, 179, 182, 188, 240] on the IM problem. In [2, 12, 15, 34, 95, 165, 179, 182, 188], they were all based on previous heuristic or approximation algorithms. The only difference here was that some new elements have been added, including the latest methods, parallel or distributed computing, theoretical analysis, and comparison. In [31, 240], they considered two kinds of typical variant IM problems, behavior- and location-aware IM problems. They summarized some updated research progress and challenges in this direction.

Compared with the existing surveys, it is not difficult to find that they are all based on traditional algorithms and diffusion models. In this article, we present a comprehensive review of the IM problem by introducing the latest techniques and approaches based on ML and give a fine-grained discussion and analysis of these new ML-based methods.

3 INFLUENCE MAXIMIZATION

In this section, we give some preliminary definitions of diffusion models, the IM problem and its hardness properties, and the greedy hill-climbing algorithm.

3.1 Diffusion Models

A social network can be denoted as a directed graph $G = (V, E)$, where V with $|V| = n$ is the set of nodes in G and E with $|E| = m$ is the set of edges between nodes in graph G . A node $u \in V$ represents an individual or user in a social network, while an edge $(u, v) \in E$ indicates a certain type of social connection between user u and v . For each edge $(u, v) \in E$, we say that u is the in-neighbor of v , and v is the out-neighbor of u . For each node $v \in V$, we denote by $N^-(v)$ the set of its in-neighbors and $N^+(v)$ the set of its out-neighbors. A node $u \in V$ is a seed node if it acts as the source of information diffusion at the beginning. The set of seed nodes is called seed set, denoted as S , $S \subseteq V$. The influence of the given seed set is based on the information diffusion process. Here, we consider that a user is active if she accepts (is activated by) the information cascade from her in-neighbors or she is selected as a seed.

Then, we first define two commonly used diffusion models, IC model, and LT model [104], as follows in order to quantify the information diffusion across the social network. It can be represented by a discrete-time stochastic process, where we denote by $S_i \subseteq V$ the set of all active nodes at timestep t_i . At timestep t_0 , all seed nodes in S are activated, $S_0 := S$.

Definition 1 (IC Model). There is a diffusion probability $p_{uv} \in (0, 1]$ associated with each edge $(u, v) \in E$. We set $S_i := S_{i-1}$ at timestep t_i ($t \geq 1$) first; then, for each node $u \in S_{i-1} \setminus S_{i-2}$ activated first at timestep t_{i-1} , it has one chance to activate each of its inactive out-neighbor v with probability p_{uv} . We add v into S_i if u activates v successfully at t_i . The influence diffusion terminates when no more inactive nodes can be activated.

Definition 2 (LT Model). Each edge $(u, v) \in E$ has a weight b_{uv} , and each node $v \in V$ has a threshold θ_v sampled uniformly in $[0, 1]$ and $\sum_{u \in N^-(v)} b_{uv} \leq 1$. We set $S_i := S_{i-1}$ at timestep t_i ($t \geq 1$) first; then, for each inactive node $v \in V \setminus S_{i-1}$, it can be activated if $\sum_{u \in S_{i-1} \cap N^-(v)} b_{uv} \geq \theta_v$. We add v into S_i if v is activated successfully at t_i . The influence diffusion terminates when no more inactive nodes can be activated.

The diffusion probability p_{uv} under the IC model can be regarded as the weight of edge directed from node u to node v , which would distinctly affect the diffusion process. As a model parameter,

the weight of edge should be learned from the data ideally [70, 114, 143, 190]. However, in practice, most of the existing works on the IM problem are still adopting pre-defined assignments for the sake of algorithm comparison [10]. A commonly used weight assignment mechanism is called **weighted cascade (WC)**, which assigns $1/N^-(v)$ to weight p_{uv} . The weight b_{uv} under the LT model is primarily assigned according to some heuristic mechanism, e.g., uniformly assignment as WC [47, 68] or random assigning a value from $[0, 1]$ before normalization [47, 191, 192].

Then, Kempe et al. [104] took the IC model and LT model as special cases, and summarized them to the Triggering model.

Definition 3 (Triggering Model). Each node v selects a triggering set T_v randomly and independently according to a distribution \mathcal{D}_v over the subsets of $N^-(v)$. We set $S_i := S_{i-1}$ at timestep t_i ($t \geq 1$) first; then, for each inactive node $v \in V \setminus S_{i-1}$, it can be activated if there is at least one node in T_v activated in t_{i-1} . We add v into S_i if v is activated successfully at t_i . The influence diffusion terminates when no more inactive nodes can be activated.

From above, a triggering model can be defined as $\Omega = (G, \mathcal{D})$, where $\mathcal{D} = \{\mathcal{D}_{v_1}, \mathcal{D}_{v_2}, \dots, \mathcal{D}_{v_n}\}$ is a set of distribution over the subsets of each $N^-(v_i)$. For each node $v \in V$, each node $u \in N^-(v)$ appears in v 's random triggering set T_v with probability p_{uv} independently under the IC model; and at most one node can appear in T_v , thus, for each node $u \in N^-(v)$, $T_v = \{u\}$ with probability b_{uv} exclusively and $T_v = \emptyset$ with probability $1 - \sum_{u \in N^-(v)} b_{uv}$ under the LT model.

3.2 The Definition of IM Problem

The IM problem is to find a seed set $S \subseteq V$ such that its influence spread $\sigma(S)$, can be maximized, which is the expected number of active nodes after the diffusion terminates. To mathematically define the influence spread, we first introduce a concept called “realization” or “possible world”. Given a triggering model $\Omega = (G, \mathcal{D})$, Guo et al. [81] defined the realization as follows.

Definition 4 (Realization). Given a directed graph $G = (V, E)$ and triggering model $\Omega = (G, \mathcal{D})$, a realization $g = \{T_{v_1}, T_{v_2}, \dots, T_{v_n}\}$ of G is a set of triggering set sampled from distribution \mathcal{D} , denoted by $g \sim \Omega$. For each node $v \in V$, we have $T_v \sim \mathcal{D}_v$, respectively.

If a node u appears in v 's triggering set, $u \in T_v$, we say edge (u, v) is live, or else edge (u, v) is blocked. Thus, realization g can be regarded as a subgraph of G , which is the remaining graph by removing these blocked edges. Let $\Pr[g|g \sim \Omega]$ be the probability of realization g of G sampled from distribution \mathcal{D} , that is,

$$\Pr[g|g \sim \Omega] = \prod_{i=1}^n \Pr[T_{v_i}|T_{v_i} \sim \mathcal{D}_{v_i}], \quad (1)$$

where $\Pr[T_{v_i}|T_{v_i} \sim \mathcal{D}_{v_i}]$ is the probability of T_{v_i} sampled from \mathcal{D}_{v_i} . Under the IC-model, $\Pr[T_v|T_v \sim \mathcal{D}_v] = \prod_{u \in T_v} p_{uv} \prod_{u \in N^-(v) \setminus T_v} (1 - p_{uv})$, and under the LT-model, $\Pr[T_v = \{u\}|T_v \sim \mathcal{D}_v] = b_{uv}$ for each $u \in N^-(v)$ and $\Pr[T_v = \emptyset|T_v \sim \mathcal{D}_v] = 1 - \sum_{u \in N^-(v)} b_{uv}$ deterministically.

The influence cascade on a realization becomes a deterministic process instead of a stochastic process. As a result, the influence spread across the network can be regarded as the expected spread on all possible realizations. Now, the IM problem can be written in an expectation form as follows:

Definition 5 (Influence Maximization). Given a social graph $G = (V, E)$, a triggering model $\Omega = (G, \mathcal{D})$, and a budget k , the IM problem asks to find a seed set $S^\circ \subseteq V$, with at most k nodes, that can maximize the expected influence spread across the graph, i.e.,

$$S^\circ \in \arg \max_{|S| \leq k} \sigma_\Omega(S) = \mathbb{E}_{g \sim \Omega}[|I_g(S)|] = \sum_g \Pr[g|g \sim \Omega] \cdot |I_g(S)|, \quad (2)$$

where $I_g(S)$ contains all nodes can be reached from a node in S by the live edges in the realization g .

ALGORITHM 1: Greedy (G, Ω, k)

```

1  $S \leftarrow \emptyset$ ;
2 for  $i = 1$  to  $k$  do
3    $u^* \leftarrow \arg \max_{u \in V \setminus S} \{\sigma_\Omega(S \cup \{u\}) - \sigma_\Omega(S)\}$ ;
4    $S \leftarrow S \cup \{u^*\}$ ;
5 return  $S$ 

```

The IM problem is NP-hard under the IC model, LT model, and Triggering model [104]. The core idea of proofs is to reduce from the NP-complete Set Cover problem to the IM problem under the IC model and the NP-complete Vertex Cover problem to the IM problem under the LT model. The Triggering model is a generalized form of the IC model and the LT model, thus it easily has the same hardness by simple induction.

3.3 Greedy Algorithm and Hardness

Due to the computational hardness of the IM problem, the optimal solution cannot be exactly calculated out within polynomial time. Fortunately, the influence function $\sigma_\Omega(S)$ satisfies two properties, monotonicity, and submodularity, which is helpful to design approximate algorithms. A lot of subsequent IM algorithms are based on these two properties and related optimization theory.

Definition 6 (Monotonicity and Submodularity). Given a set function $f : 2^V \rightarrow \mathbb{R}_+$ and any two subsets S and T with $S \subseteq T \subseteq V$, we say it is monotone if $f(S) \leq f(T)$ and submodular if $f(S \cup \{v\}) - f(S) \geq f(T \cup \{v\}) - f(T)$.

The IM problem is NP-hard and the influence function $\sigma_\Omega(S)$ is monotone and submodular with respect to S under the Triggering model.

Most of the traditional IM algorithms are derived from the greedy hill-climbing algorithm, which is shown in Algorithm 1. Shown as Algorithm 1, it initializes an empty seed set S , and iteratively select a node u^* with the maximum marginal gain $\sigma_\Omega(u|S) = \sigma_\Omega(S \cup \{u\}) - \sigma_\Omega(S)$ until there are k nodes in the seed set. Nemhauser et al. [66, 153] build a theoretical guarantee for maximizing a monotone and submodular function under different constraints in 1978. They point out that given a non-negative monotone and submodular function under the cardinality constraint, the greedy algorithm can guarantee a $(1 - 1/e)$ approximation under the value oracle. However, the value oracle cannot be satisfied with the IM problem because of the stochasticity of information diffusion. Chen et al. [45, 47] present that given a seed set S , it is #P-hard to compute the accurate influence spread $\sigma_\Omega(S)$ under the IC model and LT model. The most direct way to estimate the influence spread is to use **Monte Carlo (MC)** simulation, where we repeatedly simulate the diffusion process of a given seed set and take its average value as the final result. This estimation introduces an error term ε , and then the greedy algorithm can only return a $(1 - 1/e - \varepsilon)$ approximation within the $\Omega(kmn \cdot \text{poly}(1/\varepsilon))$ time complexity.

In the greedy algorithm, it uses MC simulation to estimate the marginal gain for all nodes in each iteration, leading to an unacceptable running time when applied in large-scale social networks. Therefore, a series of subsequent works focused on how to improve the running efficiency of IM algorithms. Their goals are to effectively estimate the influence spread, thus overcoming the #P-hardness. The representatives of these approaches will be reviewed in detail in Section 5.

4 TAXONOMY OF IM ALGORITHMS

The traditional IM algorithms face the challenges such as time efficiency and generalizability. Moreover, the advance in ML algorithms set off a research boom for CO problems. Since the IM problem

is a branch of CO problems, there comes the development of ML-based IM algorithms. We categorize the existing approaches into two big families: Traditional IM and ML-based IM. Within each family, the specific categories are provided according to different design strategies or model training methods, as shown in Figure 1. We expand this article following the architecture of Figure 1 and focus on the following five aspects.

Traditional IM. According to the theoretical guarantee of the worst case, the IM algorithms can be divided into heuristic algorithms and approximation algorithms. According to different design strategies for estimating the influence spread, the IM algorithms can be categorized into three classes [127], i.e., simulation-based, proxy-based, and sampling-based approaches. Among them, the simulation-based and sampling-based approaches are mainly based on approximation algorithms, but proxy-based approaches are mainly based on heuristic algorithms. The representative algorithms of these three classes are listed as follows, and we will introduce them in detail in Section 5.

- Simulation-based: CELF [119], CELF++ [71], UBLF [249], CGA [215], and SA [96].
- Proxy-based: SP [110], MIA/PMIA [45], LDAG [47], SIMPATH [72], and EASYIM [68].
- Sampling-based: TIM/TIM+ [192], IMM [191], SSA/D-SSA [157], OPIM-C [189], and HIST [82, 83].

ML-based CO. For the common properties between IM and CO, the ML-based CO algorithms can display values of reference for ML-based IM. The ML-based algorithms for CO are mainly divided into Ptr-Nets based [208] class and GNNs based [107] class. The works of both classes are reviewed in Section 6. Particularly, in the network embedding learning, structures for CO problems are prone to utilization in the IM problem. Therefore, different types of typical embedding approaches for CO problems are explained, including random walk based methods, recurrent GNNs, **Graph Convolutional Networks (GCNs)**, and **graph autoencoders (GAEs)**. The summary of methods for those CO problems similar to the IM problem is shown in Table 1.

ML-based IM. In Section 7, we categorize the existing ML-based IM algorithms according to the training methods as RL, unsupervised or self-supervised learning, and supervised or semi-supervised learning. Due to the characteristics of the IM problem, RL is preferable, thus we first review RL-based IM algorithms, then other ML-based algorithms. Here, we focus on the algorithms with innovation on the methodology itself for the vanilla IM problem and leave the context-aware variants of applications in the subsequent section. The corresponding representatives of these approaches are summarized in the “Vanilla IM” category of Table 2.

Variant IM Problems in Real Applications. For variant IM, we refer to the combination of vanilla IM frameworks and practical scenarios such as **competitive IM (CIM)**, Topic-Aware, and dynamic IM, which bring different settings and complex constraints to the specific formulation of problems. We review these problems in Section 8 from an ML-based view. These variants may also have different objective functions designed for various application scenarios. The variant IM problems usually have complex characteristics such as multi-objective, multi-constraints, and non-static, which cannot be effectively approximated by traditional and vanilla IM algorithms. Table 2 summarizes ML-based works for these scenarios, respectively.

Challenges. In Section 9, we discuss two main challenges in this area of IM. The first challenge is the difficulty of ML-based approaches to solving general CO problems, including scalability, adaptivity, generalizability, efficiency, and Automation. The second challenge is the difficulty, especially for ML-based IM, such as diversified diffusion models, training efficiency, reward estimation, and variant IM problems. These consider how to apply general ML methods to solve the IM problem and demand new creative ML-based approaches.

5 TRADITIONAL ALGORITHMS

In this section, we introduce three kinds of traditional algorithms, respectively, explain the difficulties faced by traditional algorithms, and supplement some of the latest works.

5.1 Simulation-Based Approach

The core idea of this category is to adopt MC simulation to estimate the influence spread $\sigma_\Omega(S)$. Specifically, with the stochastic traits of the diffusion process, the activation behavior between two neighboring nodes occurs with some probability, which leads to simulation being a straightforward approach for calculating $\sigma_\Omega(S)$. The seminal work on the simulation-based approach has been discussed in the last section. To estimate a marginal gain, it conducts r rounds of MC simulation to get r different realizations based on the distribution given by a diffusion model. Note that the estimation of the influence spread $\sigma_\Omega(S)$ becomes more and more accurate as the value of r increases. Then the node u^* with the highest estimated marginal gain will be added to the seed set S . To make the error small enough, r can be empirically set to a large number such as $r = 10000$ [104]. Chen et al. [44] quantify the relationship between r and error ε , and point out that the simulation-based greedy algorithm returns a $(1 - 1/e - \varepsilon)$ approximation if the number of MC simulation r is given by $\Theta(k^2 n \log(n^2 k) / \varepsilon^2)$ under the triggering model, then the algorithm runs in time $O(k^3 n^2 m \log n / \varepsilon^3)$.

The unacceptable running time of the simulation-based greedy algorithm leads to a lot of researches trying to reduce the number of MC simulations. Leskovec et al. [119] propose the CELF algorithm in 2007, which reduces the time complexity on the premise of ensuring the approximate ratio. They devise a mechanism with early stopping to largely improve the efficiency through identifying and pruning insignificant nodes to reduce the usage of MC simulation, but it does not improve the worst-case time complexity. In the greedy algorithm, there are k iterations, and in each iteration, it would choose one node with maximum marginal gain, which normally requires traversal over all nodes out of seed set S to compute the marginal gain. However, CELF introduces an upper bound attached to each node whose value is its marginal gain in some past iteration. Thus, we do not need to compute the marginal gain if its upper bound is already smaller than the maximum of those visited nodes' marginal gains that have been computed in the current iteration. Specifically, in the first iteration, the upper bound of each node $u \in V$ is assigned to the $\sigma_\Omega(\{u\})$. During the iteration, the nodes out of the seed set are traversed in descending order based on their upper bounds to compute the marginal gain. Here, it introduces an early stopping mechanism that the traversal within an iteration would be stopped if the maximum upper bound of all unvisited nodes is smaller than at least one marginal gain of the visited nodes. Once the early stopping is triggered, the current iteration would be terminated, and the upper bound of each visited node would be updated with the latest marginal gains. The idea behind this is that the marginal gain of a node in the current iteration cannot be larger than its marginal gain in the previous iteration, and hence, maintains it while selecting seeds in subsequent iterations. This early stopping mechanism can greatly accelerate the greedy algorithm and achieve up to 700 times faster speed. Goyal et al. [71] design the CELF++ algorithm to further reduce unnecessary MC simulation on the basis of CELF by including the marginal gain of the node that has the best marginal gain in the previous iterations. UBLF [249] utilizes a quick method to estimate the upper bound of $\sigma_\Omega(\{u\})$ for all $u \in V$ by using matrix analysis, and prove it is much faster than CELF/CELF++. However, it can only be used under the IC model and LT model.

In addition to reducing the number of MC simulations, there are some researches, such as CGA [215] and SA [96], aiming to reduce the MC complexity. Here, they design some heuristic rules to simplify the process of MC simulation, which cannot guarantee the approximation ratio. In summary, the simulation-based approach has the benefit of model generalization, which implies

that it does not depend on diffusion models since MC simulation is pervasively adaptive for arbitrary diffusion models. However, the time complexity of MC simulation is exponentially high due to the existence of parameter r even with acceleration techniques such as CELF/CELF++, which is inherently hard to be further improved and cannot be applied to large-scale social networks.

5.2 Proxy-Based Approach

The core idea of this category is to adopt proxy models to approximate the influence spread $\sigma_\Omega(S)$ given seed set S , thereby avoiding time-consuming MC simulation. The general method is to simplify the information diffusion process and replace the real diffusion model with the proxy model to estimate the influence spread, so as to overcome its #P-hardness and achieve the purpose of improving efficiency. Proxy models are usually designed according to some heuristic rules, and there are two typical strategies: (1) Simplifying the stochastic diffusion model to the deterministic model such that the influence spread can be exactly computed; and (2) Restricting the influenced scope of each node to the adjacent hops and ignoring the further hops.

Under the IC model, Kimura et al. [110] put forward a **Shortest-Path (SP)** model, which only considered the shortest path from node u to v when computing the influence spread. Chen et al. [45] propose a representative proxy model in 2010, MIA/PMIA model. The key idea of MIA is to reduce stochastic diffusion models into deterministic models by constructing special spanning trees of propagation for each node v to avoid MC simulation and speed up computing efficiency. Specifically, for a node v , MIA first builds up an in-arborescence subgraph ending with v and an out-arborescence subgraph starting from v where the paths between any node u and v , namely (u, v) or (v, u) , are selected through Dijkstra SP algorithm with designed length metric inversely proportional to the likelihood of a path under the IC model. Some paths in the subgraphs are then pruned if their likelihood is less than a given threshold θ , and the remaining subgraphs are kept for further computation of the influence spread. Thus, the nodes influenced by a node u can be directly targeted if there exist paths. PMIA is an extension of MIA for more accurate influence estimation, updating the in-arborescence subgraph after adding a node into the seed set so that existing seeds will not block the influence of future seeds. Since the influence function under the MIA/PMIA model is monotone and submodular, with the greedy framework, the $(1 - 1/e)$ approximation ratio can be guaranteed under these proxy models. Afterward, Kim et al. [109] continue to reduce the IC model similar to the MIA model by the independent path algorithm, which prunes all influence paths with a probability less than a given threshold.

Under the LT model, Chen et al. [47] propose a pioneering proxy model in 2010, the LDAG model. It uses **directed acyclic graphs (DAGs)** to reduce the stochastic diffusion models, which constructs DAG for each node by the Dijkstra SP algorithm and prunes the paths with the probability less than the threshold θ in a similar way to MIA/PMIA, then estimates the influence based on those DAGs. Because of the NP-hardness of computing the optimal DAG for a node, sub-optimal DAGs are used and extra error may be introduced. Goyal et al. [72] design a SIMPATH algorithm where the influence spread can be estimated by counting the total number of simple paths beginning from each node of the seed set. Unlike LDAG enumerating all simple paths, SIMPATH estimates the influence spread considering fewer paths by removing the paths of which the likelihood is less than the threshold θ . Thus, it is usually faster than LDAG. Besides, EASYIM [68] can be used for both the IC model and the LT model. It estimates the influence of a node u by computing the number of the paths starting from u with a length less than l and counts overlapping paths, respectively, to improve the quality of estimation.

In addition, Li et al. [127] point out that there is another proxy-based approach called “influence ranking proxy”, such as IRIE [99], GroupPR [135], SPIN [151], and IMRank [50]. Actually, they transform the IM problem to another problem and rank all nodes according to some index, which

has deviated from IM algorithms based on CO. Thus, we will not repeat them here. In summary, the proxy-based approach can significantly reduce the computation overhead of estimating the influence spread and show empirical efficiency in many IM-based applications. However, it lacks a theoretical guarantee (approximation ratio), which leads to extreme instability in the worst case. Furthermore, the proxy model has a drawback. It has to be designed relying on a specific diffusion model, which cannot be generalized to the other diffusion models.

5.3 Sampling-Based Approach

The core idea of this category is to sample a number of realizations from a graph given a diffusion model, and then estimate the influence spread of any seed set based on these sampled realizations. It avoids iteratively running MC simulation in the greedy algorithm, and can significantly improve the time efficiency and ensure the theoretical guarantee at the same time. Regarding the advance of effectiveness, we only focus on reviewing the algorithms based on **Reverse Influence Sampling (RIS)** rather than **Forward Influence Sampling (FIS)**. The representative algorithms based on FIS are NEWGREIC [46], STATICGREEDY [51], StaticGreedyDU [51], PRUNEDMC [160], and SKIM [53]. Even though FIS-based algorithms formed a pioneering sampling-based approach, they have been quickly replaced by RIS-based algorithms because of their limited improvement in time efficiency. Thus, there is no one cares about it now.

In 2014, Borgs et al. [23] first propose the technique of RIS, which is a milestone in the development history of traditional IM algorithms. This technique is based on the concept of random **reverse reachable (RR)** set. A random RR set can be generated in the following three steps:

- Uniformly select a node v from V .
- Randomly sample a realization g from G under a given diffusion model Ω .
- Generate RR set, denoted by R , by collecting the nodes that can reach v through a live path in realization g .

For each node $u \in V$, the probability that it is contained in R randomly generated by the v equals the probability that u can activate v . Given a seed set S and a random RR set R sampled by a diffusion model Ω , we have

$$\sigma_{\Omega}(S) = n \cdot \mathbb{E}_R[\mathbb{I}(S \cap R)], \quad (3)$$

where $\mathbb{I}(S \cap R) = 1$ if $S \cap R \neq \emptyset$, else $\mathbb{I}(S \cap R) = 0$. Thus, the general framework of RIS-based algorithms is to first generate a collection of random RR set $\mathcal{R} = \{R_1, R_2, \dots, R_{\theta}\}$. Here, it estimates $\sigma_{\Omega}(S)$ by $n \cdot \text{Cov}_{\mathcal{R}}(S)/\theta$ where $\text{Cov}_{\mathcal{R}}(S)$ is the number of RR sets covered by the seed set S . The IM problem has been transformed to the **maximum coverage problem (MCP)**, and then the greedy algorithm is used to select a seed set S^* that covers the maximum number of RR sets.

Obviously, the estimation will be more and more accurate with the increase of θ . In order to strike a balance between computation efficiency and estimation accuracy, the subsequent researches concentrate on how to determine the value of θ and how to bound the error and get a $(1 - 1/e - \epsilon)$ approximation with a high probability. Borgs et al. [23] design a threshold-based method that keeps generating RR sets until the total number of edges examined during the generation process reaches a pre-defined threshold τ . They prove that it returns a $(1 - 1/e - \epsilon)$ approximate solution with $1 - 1/n$ probability in polynomial time when setting τ to $O(k(m+n) \cdot \log^2 n / \epsilon^3)$. Tang et al. [192] propose a TIM/TIM+ that follows the RIS to improve efficiency. They were the first to directly build the relationship between the number of RR sets θ and approximation loss, whose expected time complexity is $O((m+n) \cdot \log n / \epsilon^2)$ when ensuring the same theoretical bound. Tang et al. [43, 191] further improve the TIM/TIM+ and proposed an IMM algorithm. It adopts a martingale analysis to better estimate the lower bound of optimal value and further scales down the number of RR sets while keeping the same worst-case guarantees. Nguyen et al. [157] propose

an SSA/D-SSA algorithm that iteratively doubles the number of RR sets and selects the seed set based on the current collection of RR sets until satisfying the stopping condition. They claim that they achieve much better performance than previous methods, but their theoretical analysis is doubted by other researchers [92, 156]. In 2018, Tang et al. [189] change this pattern by adopting an online sampling method. They created an OPIM-C, which sampled two collections of random RR sets, one for selecting a seed set and the other one for verifying whether the seed set is qualified until a $(1 - 1/e - \epsilon)$ approximation can be provided. It keeps the same expected time complexity as IMM but achieves faster in practice because of the much smaller number of RR sets needed to be generated.

The above-mentioned RIS-based algorithms may result in a large memory consumption because a large number of RR sets should be stored in the memory used for the seed set selection. To effectively reduce the memory consumption, Wang et al. [214] propose a lazy sampling technique (BKRIS) that adopts the bottom-K minHash technique and selects the seed set without fully implementing all θ RR sets. Guo et al. [82, 83] propose a SUBSIM framework to speed up the generation process of RR sets and create a **Hit-and-Stop (HIST)** algorithm to tackle the scalability issue in high influence networks by reducing the average size of random RR sets without losing the approximation guarantee, thus saving the memory consumption.

In summary, the sampling-based approach has advantages over the other two approaches. It has a rigorous approximation guarantee and relatively low time complexity, which achieves the balance between theoretical guarantee and algorithm efficiency. It represents the highest level of traditional algorithms. Nevertheless, the sampling should be aligned with the given diffusion model. Unlike the proxy-based approach, it does not need to be specially designed according to the diffusion model, but does not mean that any diffusion model can easily give an explicit distribution so that the realization can be sampled. This is a bad thing. Next, RIS-based algorithms have to store a large number of RR sets, which is a waste of memory and storage resources. The setting of parameters in these algorithms heavily affects the approximation and time complexity, which shows inadequate generalization ability and can barely adjust to dynamic social networks.

6 DRL FOR CO

As mentioned in the previous section, there is no traditional algorithm that can perfectly solve the IM problem. For heuristic algorithms, they have to be designed according to given diffusion models and have no theoretical guarantee. For approximation algorithms, their scalability has reached the limit, and time efficiency cannot be further optimized. No matter which traditional algorithm we use, it has a poor generalization ability. That is to say, when the network topology changes, the algorithm must be re-run to keep the validity of the seed set. In recent years, a new technique of solving CO problems by DRL has emerged, which can effectively address the drawbacks of traditional algorithms. The IM problem is a special kind of CO problem. Besides, since CO problems inherently have several traits such as lack of labeled data, dynamic environments, and delayed rewards, RL fits them more than other ML methods like supervised or unsupervised learning and dominates the existing solutions. Thus, before introducing ML-based IM algorithms, we first survey some latest development of DRL for CO. We will focus on the concepts, technologies, and methods related to ML-based IM algorithms.

The CO problem is a sub-field of the mathematical optimization problem, which is defined in the discrete space. The CO problem can be defined by a standard form, which is mathematically formulated as follows:

$$\min F(x) \tag{4}$$

$$\text{s.t. } G(x) \geq 0, x \in D, \tag{5}$$

where x is the decision variable, $F(x)$ is the objective function, $G(x)$ is the constraint condition, and D is a discrete space. Typical CO problems include **Traveling Salesman Problem (TSP)**, **Vehicle Routing Problem (VRP)**, Job Scheduling, Knapsack, **Minimum Vertex Cover (MVC)**, and **Minimum Dominating Problem (MDoP)**.

Similar to traditional algorithms for the IM problem, traditional algorithms for CO problems consist of exact, approximate, and heuristic approaches. The exact approach primarily adopts the idea of “divide and conquer”, to divide the problem into smaller sub-problems and solve them recursively to finally achieve the optimal solution, such as Branch-and-Bound [115] and Dynamic Programming [21]. The approximate approach aims at finding the local optimal solution for higher efficiency, such as the greedy algorithm and local search algorithm, which can provide theoretical guarantees for the quality of solutions. The heuristic algorithm, such as simulated annealing algorithm [203] and evolutionary algorithms [207], can largely accelerate the computation by using specialized domain knowledge and involving trial-and-error as well as tuning. However, all of these approaches suffer from computational inefficiency for large-scale problems. In addition, for more and more applications that require high real-time performance, once the situation of the problem changes, it is necessary to recalculate the solution, which would inevitably produce too expensive calculation costs. Thus, we need to find other efficient and effective solutions.

A natural question is whether an ML-based algorithm can have a generalization ability such that the algorithm of a CO problem learned from part of instances can be generalized to unseen instances with the same distribution. To answer this question, a lot of researchers have put effort into it. Using ML-based methods to solve CO problems can be traced back to the Hopfield network [89] proposed in 1985. It can be used to solve the TSP [180] but only for those small-scale single instances. With the development of hardware performance and deep learning, a lot of researchers began to study how to use ML-based methods to solve generalized NP-hard CO problems [18, 32, 77, 107, 111, 118, 130, 140, 141, 152, 167, 173, 208, 238]. They mainly rely on DRL because it can learn a policy to guide a series of decision-making actions under different circumstances in order to gain a maximum accumulated benefit, which aligns with the characteristics of CO problems like TSP, Knapsack, and IM.

According to the coding strategies of problem instances, the architecture can be roughly divided into two categories: one based on Ptr-Nets [208] and the other based on GNNs [107]. They form two kinds of end-to-end frameworks: “Ptr-Nets based RL” and “GNNs based RL”. An end-to-end framework takes a given problem instance as input, and can directly output the solution without searching. Its problem embedding and solving is integrated into training as a whole. Once the training is completed, the problem instances with the same distribution characteristic can be directly solved. Therefore, it has the advantages of fast solution speed and strong generalization ability. We will introduce them separately. Here, we assume that readers have basic ML knowledge [69, 149].

6.1 Ptr-Nets Based RL Framework

The Ptr-Net is a pioneering model, which learns from the **sequence-to-sequence (seq2seq)** model [187] in the machine translation to solve CO problems, that is, encode the input sequence of problem instance by an encoder, then computes the selection probability of each node by a decoder and attention mechanism [13], and gradually select the nodes in an auto-regressive way until a complete solution. Its architecture is shown in Figure 2, where the yellow network is the encoder and the blue network is the decoder. The encoder and decoder are implemented by a kind of Recurrent Neural Network, such as RNN, LSTM, and **Gated Recurrent Unit (GRU)**. It first converts the input sequence to a code and feeds it into the decoder, which generates an output according to the content-based attention mechanism over the input sequence.

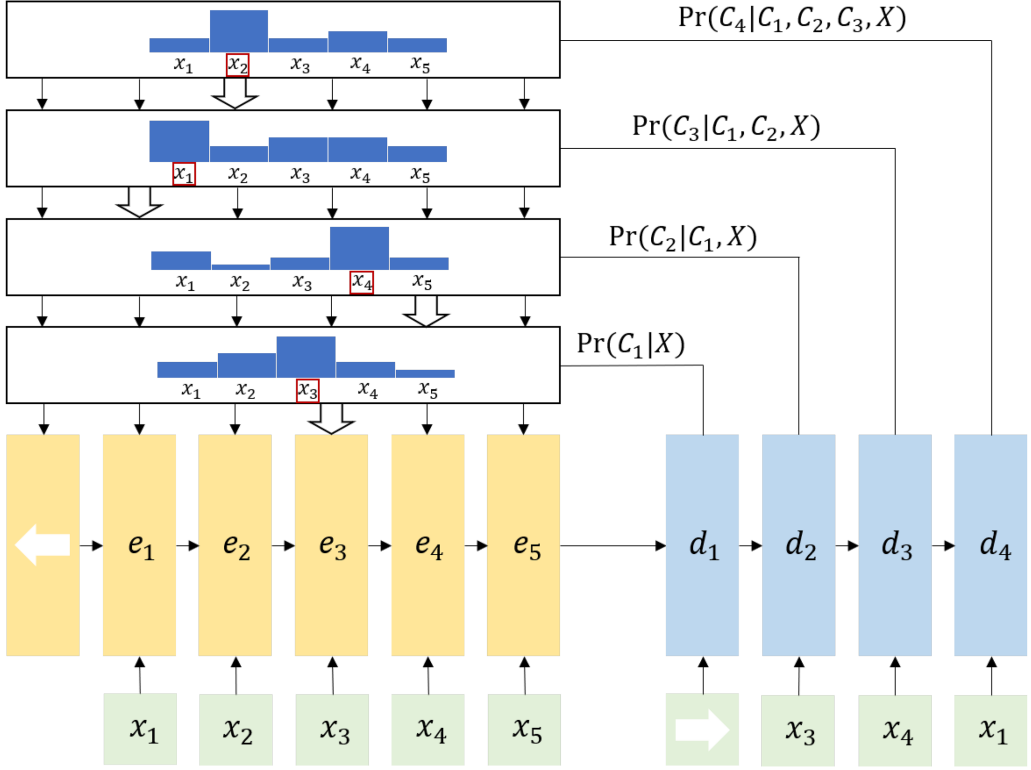


Fig. 2. The architecture of Ptr-Nets [208], where the yellow network is the encoder and the blue network is the decoder. They interact through the attention mechanism.

For convenience, we define the encoder and decoder hidden states as (e_1, \dots, e_n) and (d_1, \dots, d_m) , respectively. The input is denoted by X , and the i th outcome is denoted by C_i . In the i th decoding, the RNN reads the hidden state of the previous layer d_{t-1} and outputs its hidden layer d_t . Based on d_t and encoder hidden state, the posterior probability of the i th outcome can be modeled as follows:

$$u_j^i = v^T \tanh(W_1 e_j + W_2 d_i), \quad j \in (1, \dots, n), \quad (6)$$

$$\Pr(C_i|C_1, \dots, C_{i-1}, X) = \text{softmax}(u^i), \quad (7)$$

where softmax normalizes each component of the vector u^i (of length n) to be an output distribution over the dictionary of inputs. The v , W_1 , and W_2 are learnable (trainable) parameters of the decoder. Taking TSP for example, the input is the sequence of city coordinates, and the output is the solution of city indexes in order.

The Ptr-Net can be trained through a supervised approach or a DRL approach. But due to the lack of labeled data (i.e., optimal routes for TSP) in practice, the current studies tend to train the model parameters such as v , W_1 , and W_2 through DRL. The DRL formulates the optimizing objective as the reward and learns a policy to maximize the accumulated reward by sampling different events with corresponding instantaneous rewards. The Value- and policy-based algorithms are two mainstream approaches of DRL. Since the policy-based DRL approach models its stochastic policy considering state S and action A as $P_\theta(A|S) = \prod_{t=1}^n P_\theta(A_t|S, A_{1:t-1})$, which coincides with the mechanism given by Ptr-Nets. Policy-based DRL algorithms that take the total feedback as

the standard and adopt the round update model, especially REINFORCE algorithm, are usually exploited to train Ptr-Nets.

Vinyals et al. [208] propose the Ptr-Net in 2015 to solve three challenging geometric CO problems with supervised training, Planar Convex Hulls, Delaunay Triangulations, and Planar TSP. The supervised training requires a large number of labeled data, such as (*city coordinates, optimal route*) in TSP, to train the network parameters, which is hard to obtain. Also, supervised training inherently limits the quality of solutions to the training data, too dependent on the quality of labels. To address them, Bello et al. [18] employ a DRL framework to train the Ptr-Net on CO problems. They used REINFORCE algorithm to learn the parameters, taking each instance of the CO problem as a training sample and formulating the optimizing objective function as the reward function in DRL. Actor-critic algorithm is introduced to restrain variance during training. The results outperform the Ptr-Net with supervised training [208] and it achieves a near-optimal solution to TSP as well as the optimal solution to the knapsack problem. Motivated by Bello's work, Nazari et al. [152] extend the Ptr-Net based RL to more complicated VRP, which take into account the static elements (customers' coordinates) and dynamic elements (customers' demands). By ignoring the order information of customers and replacing the LSTM of the input layer in the encoder, it would not impact the solution and greatly reduce the computation overhead. The results show comparable performance with Bello's work on TSP and superior to classic heuristics on VRP. Similar to Bello et al. [18], Gu and Yang [75] adopt Ptr-Net with Actor-critic algorithm to iteratively construct a solution for the MAXCUT problem. The experiments are conducted with simulated graphs up to 300 nodes and show good approximation ratios. However, they do not incorporate existing benchmark methods for comparison.

With the development of the attention mechanism, Transformer [204] adopts a multi-head attention mechanism instead of RNN/LSTM and enables models to extract deeper characteristics of data. With the tremendous success of Transformer in Natural Language Processing, many studies have begun to integrate Transformer into the Ptr-Nets based RL to solve CO problems. Deudon et al. [59] upgrade the conventional Ptr-Net by using Transformer's multi-head attention structure to output the feature vector of nodes in the encoder part, which effectively reduces the model complexity. They use this upgraded model to compute a preliminary result and then employ a simple 2OPT local search on it to compute the eventual result, which shows improvement in the quality of the solution. Kool et al. [111] not only integrate the multi-head attention idea given by Transformer in the encoder part of Ptr-Nets but also adopt self-attention mechanism given by Transformer to compute the attention weights in the decoder part for better performance. Furthermore, they devise a Rollout baseline to substitute for the critic neural network in Actor-Critic architecture to accelerate the training efficiency and outperform the **state-of-the-art (SOTA)** end-to-end methods including those mentioned above. Ma et al. [140] propose a **Graph Pointer Network (GPN)** to solve TSP with a time window, which is trained via a kind of **hierarchical reinforcement learning (HRL)**. The GPN builds upon Ptr-Net by placing a graph embedding layer on the input, which captures relationships between nodes and generates graph embedding vectors as part of the input to the decoder. They train the GPN through HRL to learn a hierarchical policy to find an optimal city permutation under constraints for TSP. With the graph embedding, the vector context information is merged so that the model trained on TSP 50/100 can generalize well to larger-scale TSP 500/1000 problems.

However, none of the above methods reach the solution obtained by professional solvers in the TSP problem over 50 nodes. The scalability and generalization ability of this method has a big bottleneck needed to be solved. The Ptr-Net is mainly used to solve CO problems with sequential characteristics such as TSP and VRP, where the solution to this kind of problem depends on the order of nodes. Thus, it is suitable for solving sequential CO problems because of using the attention mechanism to construct solutions in an auto-regressive way.

6.2 GNNs Based RL Framework

As we all know, graphs have complex structures, which can contain rich information about potential relationships between nodes [16]. How to mine useful information from a large-scale graph has aroused people's attention. GNNs are a new branch of methods that utilizes deep learning techniques to effectively analyze data with graph structures, which can abstract deep information from graph data. The key idea of GNNs is to encode the graph structure using a neural network model.

Generalization ability is a great challenge for traditional algorithms. Network encoding techniques can improve generalization ability by embedding a node v 's attributes as well as its associated local and global information in a graph G into a p -dimensional representation vector μ_v . We first introduce several typical methods of graph network encoding commonly associated with RL for solving the CO problem, then briefly review the studies in this area.

6.2.1 Random Walk Based Embedding. *DeepWalk* [166] is proposed by Perozzi et al. to learn representation vectors for a graph such as social networks and biological networks. It adopts random walks to extract information from a network, which samples a number of short trajectories that can be produced by random walks as sequential data with connectivity information. Arriving at a node v in a walk, the possibility of choosing its neighboring node u is proportional to the normalized weight of edge (v, u) . Then they use the language model *SkipGram* [146] to process those trajectory sequences and learn embeddings of nodes, similar to learning the word embedding from a given language dataset. *node2vec* [73] extends *DeepWalk*'s sampling method by introducing a biased random walk. Given current node v during a walk, the possibility of visiting its neighboring node u is updated by multiplying a coefficient α and the weight of edge (v, u) on the basis of *DeepWalk*. The coefficient α is related to two super parameters p and q which help to strike a balance between homophily and structural equivalence in node representation. This has been proven to be more effective.

Both *DeepWalk* and *node2vec* are representation learning for homogeneous networks where only a singular type of nodes and relationships exist. However, ubiquitous heterogeneous networks in nature, especially social networks, bring unique challenges resulting from diverse types of nodes and edges. Dong et al. [61] develop two frameworks, *metapath2vec* and its extension *metapath2vec++*, to learn the representation for heterogeneous network. *metapath2vec* aims at retaining information on structures and semantics of heterogeneous networks by adopting metapath-based [185] random walks to retrieve heterogeneous neighborhood trajectories and then use *SkipGram* to generate node embeddings, which are expected to encode features of network structures and semantics. Moreover, *metapath2vec++* integrates a heterogeneous negative sampling method on the basis of *metapath2vec* to conduct simultaneous modeling of structures and semantics for efficient embedding. Extensive experiments show superior competence to the SOTA methods including *DeepWalk* and *node2vec* on heterogeneous networks.

6.2.2 Recurrent GNNs. Dai et al. [55] propose a *structure2vec* for embedding networks, which has been widely used in subsequent social applications. It is inspired by graphical model inference algorithms, where node-specific features or attributes x_v are aggregated recursively according to G 's graph topology. Specifically, given a partial solution S which is being updated at each step of recursion, in $(i + 1)$ th iteration, *structure2vec* calculates each node v 's temporary representation considering its features or attributes x_v , all its neighbors' representations in i th iteration $\{\mu_u^{(i)}\}_{u \in N(v)}$, and the connectivity information between v and all its neighbors in i th iteration $\{w(v, u)\}_{u \in N(v)}$. One variant of the *structure2vec* architecture can be formulated as follows:

$$\mu_v^{(i+1)} \leftarrow F \left(x_v + \left\{ \mu_u^{(i)} \right\}_{u \in N(v)}, \{w(v, u)\}_{u \in N(v)}; \Theta \right), \quad (8)$$

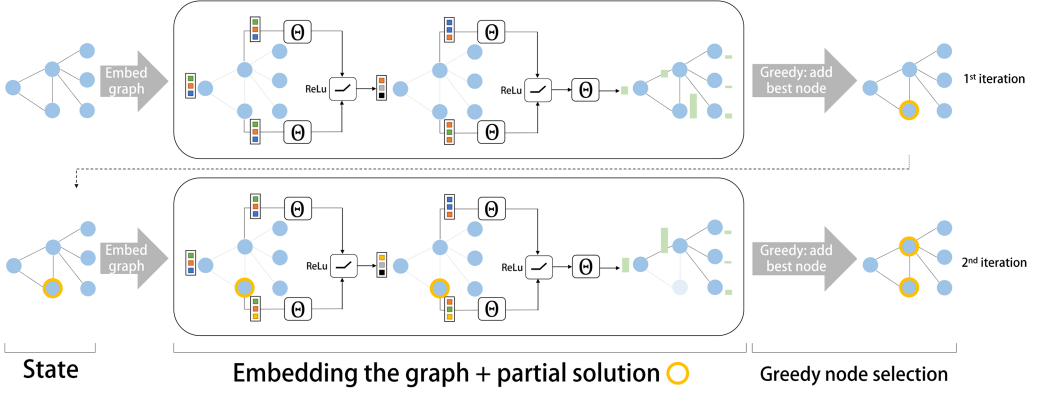


Fig. 3. The illustration of *structure2vec* framework as applied to an instance of MVC [55]. The middle part illustrates two iterations of the graph embedding, which results in node scores (green bars). Image extracted from [55].

where $\mathcal{N}(v)$ is the neighbor set of node v in graph G , F is a generic nonlinear mapping to the current embedding such as a neural network or kernel function, Θ is the set of function parameters of F , and $\mu_v^{(0)}$ is initialized as 0 for all $v \in V$. The update rule is shown as Equation (8) indicates that the embedding is carried out based on the graph topology. After I iterations, the embedding $\mu_v^{(I)}$ will definitely contain information about all neighbors within the I -hop neighborhood. The parameter set Θ can be learned through either supervised learning or RL, the latter of which fits cases where the training labels are lacking. The framework is shown in Figure 3.

6.2.3 Graph Convolutional Networks. The great success achieved by **convolutional neural networks (CNNs)** in structured data, such as feature extraction from pictures, motivates the appearance of GCNs dealing with unstructured graph data. Similar to CNNs, GCNs define filters to do convolution operations with a graph signal $\mathbf{x} \in \mathbb{R}^n$ (a scalar for every node) yet in the spectral domain in order to neglect irrespective features [27]. Given a graph G , the graph convolution operation, denoted by $*$, of two graph signals \mathbf{x}_1 and \mathbf{x}_2 is defined formally as follows:

$$\mathbf{x}_1 * \mathbf{x}_2 = \mathbf{U} \left(\left(\mathbf{U}^T \mathbf{x}_1 \right) \odot \left(\mathbf{U}^T \mathbf{x}_2 \right) \right), \quad (9)$$

where \mathbf{U} is the matrix of eigenvectors of Laplacian matrix \mathbf{L} of G and \odot denotes Hadamard product. The normalized Laplacian matrix of an undirected graph is defined as $\mathbf{L} = \mathbf{I}_n - \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}}$, where \mathbf{A} is the adjacent matrix of G and $\mathbf{D} \in \mathbb{R}^{n \times n}$ is a diagonal matrix with $D(i, i) = \sum_j A(i, j)$. With the semi-definite property, the normalized Laplacian matrix can be factored as $\mathbf{L} = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^T$, with a diagonal matrix of its eigenvalues $\mathbf{\Lambda}$. Multiplying \mathbf{U}^T transforms the graph signals \mathbf{x}_1 and \mathbf{x}_2 into the spectral domain (i.e., the graph Fourier transform), while multiplying \mathbf{U} performs the inverse transform [246]. Let us denote a filter $g_\theta = \text{diag}(\theta)$ with a parameter $\theta \in \mathbb{R}^n$ in the spectral domain. According to Equation (9), the output graph signal \mathbf{x}' after filter g_θ can be obtained as follows:

$$\mathbf{x}' = g_\theta * \mathbf{x} = \mathbf{U} g_\theta \mathbf{U}^T \mathbf{x}. \quad (10)$$

We can regard g_θ as a function of the eigenvalues of \mathbf{L} , i.e., $g_\theta(\mathbf{\Lambda})$. Due to the computational complexity of the eigenvector matrix \mathbf{U} which is $O(n^2)$, prohibitively expensive for large graphs, ChebNet [58] is proposed to simplify the computation process by adopting Chebyshev polynomials to approximate to $g_\theta(\mathbf{\Lambda})$. Kipf and Welling [217] propose an algorithm that introduces a

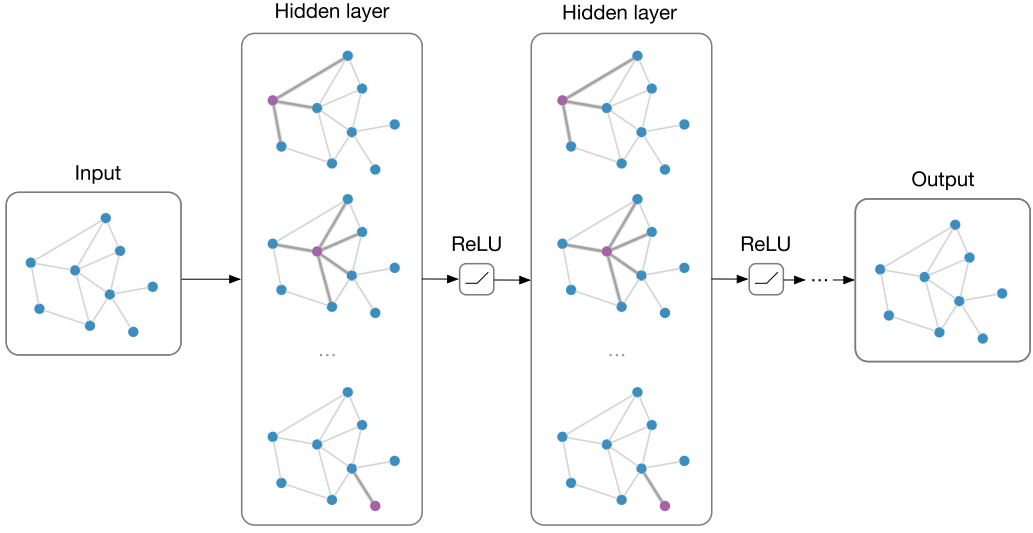


Fig. 4. A schema of GCN proposed by Kipf and Welling [217]. The graph structure is shared over convolutional layers and nodes are affected only by adjacent neighbors in each layer. Image extracted from [246].

first-order approximation of ChebNet and improves the Equation (10) as follows:

$$\mathbf{x}' = g_\theta * \mathbf{x} = \theta_0 \mathbf{x} - \theta_1 \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}} \mathbf{x}. \quad (11)$$

Several improvements are further imposed by Kipf's work [217]. To alleviate overfitting, they reduce the number of parameters by assuming $\theta_0 = \theta_1 = \theta$ as follows:

$$\mathbf{x}' = g_\theta * \mathbf{x} = \theta \left(\mathbf{I} - \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}} \right) \mathbf{x}. \quad (12)$$

They generalize the model to a multi-channel signal $\mathbf{X} \in \mathbb{R}^n \times C$ with C input channels (i.e., a C -dimensional feature vector for every node). To address the problem of numerical instability caused by using term $\mathbf{I} - \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}}$, they apply a normalization trick of using $\tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}}$ instead. This is the basic framework of GCNs, which is a prototype of many other GNNs based models. The schema is shown in Figure 4.

When processing graphs, algorithms should consider the neighbors of a node for aggregating information. As the number of neighbors of a node can be enormous in large graphs which are pervasive nowadays such as social networks, taking into account the full size of a node's neighborhood might lead to computational inefficiency. GraphSAGE [84] addresses this problem by sampling a fixed number of neighbors for each node in order to reduce the time complexity. The graph convolution operation is performed as follows:

$$\mathbf{h}_{S_{N(v)}}^k \leftarrow \text{AGGREGATE}_k \left(\left\{ \mathbf{h}_u^{k-1}, \forall u \in S_{N(v)} \right\} \right), \quad (13)$$

$$\mathbf{h}_v^k \leftarrow \sigma \left(\mathbf{W}^k \cdot \left[\mathbf{h}_v^{k-1}, \mathbf{h}_{S_{N(v)}}^k \right] \right), \quad (14)$$

where \mathbf{h}_v^k denotes the hidden feature vector of v in the k th layer, $S_{N(v)}$ denotes a random sample of the node v 's neighbors, $\text{AGGREGATE}(\cdot)$ denotes the aggregator function, $\sigma(\cdot)$ denotes a nonlinear activation function, and $[\cdot, \cdot]$ denotes the concatenation operation. The aggregation function should be invariant to the permutation of nodes, such as element-wise mean, sum, and max-pooling.

In the aforementioned GNNs, the models assume equal or manually defined weights for the neighbors of a node when aggregating them. The authors of **Graph Attention Network**

Table 1. Summary of Deep Algorithms for the IM-Like CO Problems

CO problem	Authors	Learning structure	Training method
MaxCut	Dai et al. [107]	structure2vec	DQN
	Abe et al. [1]	GNN	MCTS
	Barrett et al. [17]	MPNN	DQN
	Gu and Yang [75]	Ptr-Net	A3C
MCP	Manchanda et al. [141]	GraphSAGE	DQN
MVC	Dai et al. [107]	structure2vec	DQN
	Li et al. [130]	GCN	supervised
	Abe et al. [1]	GNN	MCTS
	Manchanda et al. [141]	GraphSAGE	DQN

(GAT) [206] claim that the weights of neighbors may vary largely in practice. For example, different neighbors have different influences on a user. Thus the weights should be learned from the data. GAT introduces the attention mechanism into convolution-based GNNs in order to learn the corresponding weights between every two nodes. The convolutional operation of GAT [224] is defined as follows:

$$\mathbf{h}_v^k = \sigma \left(\sum_{u \in \mathcal{N}(v) \cup \{v\}} \alpha_{vu}^k \mathbf{W}^k \mathbf{h}_u^{k-1} \right), \quad (15)$$

where $\mathbf{h}_v^0 = \mathbf{x}_v$, $\sigma(\cdot)$ denotes a nonlinear activation function, and α_{vu}^k denotes the attention weight between node v and its neighbor u which is computed by the self-attention mechanism of the model.

6.2.4 Graph Autoencoders. GAE is an unsupervised learning method that encodes the node or graph into a hidden vector space through reconstructing the graphs' structural information, such as adjacent matrices, according to the hidden vector representation. **Structural Deep Network Embedding (SDNE)** [210] proposes a stacked autoencoder architecture, which learns effective embedding by tailoring the loss function, so as to consider the first-order proximity and the second-order proximity with two terms. The first term is designed for the first-order proximity by giving a penalty for the distance of embedded vectors of two adjacent nodes, and the second term is designed for the second-order proximity by giving a penalty for the difference between the input and reconstructed output of the adjacent matrix, where it assigns different weights to zero and non-zero elements.

6.2.5 GNN-Based RL Frameworks. Here, we only give a simple review of GNNs that will be used in subsequent GNN-based RL frameworks and GNN-based IM algorithms. To learn more about GNNs, please read the latest review for deep learning on graphs [246]. Next, we begin to introduce the GNN-based RL framework in two parts. First, we review some representative algorithms which can also be applied to IM-like CO problems such as MVC, MAXCUT, and MCP. These referred algorithms are also summarized in Table 1. Then, some applications of GNN-based RL frameworks are reviewed.

Dai et al. [107] propose an S2V-DQN framework that explores solving the CO problem using GNNs and DRL. They utilize *structure2vec* [55] as the encoder to embed node information of graph into latent node representation (i.e., embedding), and the Q-network in **Deep Q-learning (DQN)** algorithm takes the partial solution with latent node representation to predict the Q-value of each new node. The Q-value can be designed as the marginal gain of a node, which is normally implemented by a DNN having a strong approximation ability. Then based on greedy policy, the model appends one node with the largest Q-value to the partial solution set iteratively. Once the partial

solution gets updated, the embedding of all nodes has to be re-generated. The model is trained on 50–100 nodes MVC, MAXCUT, and TSP, then gets tested on these problems with 1200 nodes, resulting in near-optimum and better generalization for graph out of distribution. Li et al. [130] propose a GCN-TREESEARCH framework that adopts GCNs to directly estimate the probability of all nodes being selected, and based on the probability estimation, they exploit the search tree to construct a feasible solution. This method is suitable for node selection problems, regardless of the order of node selection, such as **Maximal Independent Set (MIS)**, **Maximal Clique (MaxC)**, and Satisfiability. Abe et al. [1] adopt the architecture proposed by Dai et al. [107] but replace the Q-learning with a **Monte Carlo tree search (MCTS)** based approach for the MVC, MaxC, and MAXCUT problems. Manchanda et al. [141] propose a GCOMB framework, which employs a convolution-based GNN named GraphSAGE [84] to predict the quality of a node and utilizes Q-learning of DRL techniques with importance sampling. It uses GraphSAGE to prune poor nodes and learn the embedding of good nodes in a supervised manner, and in turn, Q-learning can only focus on good nodes for efficiency concerns. Both S2V-DQN [107] and GCOMB [141] build the solution set by adding one node at a time, while Barrett et al. [17] report such manner can cause suboptimal results for the difficulty in learning accurate Q-function. Instead, Barrett et al. [17] propose an exploration method where the solution set is explored by the agent at test time and can be revised in terms of the improper earlier decisions by re-evaluating and reversing the Q-values that are learned by a **message passing neural network (MPNN)**. All these methods [1, 130, 141] give a better performance than the original one [107].

Besides, there are a lot of researchers using the above-mentioned GNN-based RL frameworks to solve CO problems in realistic applications. Qiu et al. [169] propose a multi-agent DRL approach with convolution-based GNNs to address the road-tolling problem in a transportation network. They adopt edge-based convolution-based GNNs to extract the spatio-temporal correlations of the road network state features where both the feature matrix and adjacency matrix in this problem are defined on edges instead of nodes. The model is trained using multi-agent DRL by decomposing the state and action spaces into sub-spaces and solving the problem in a divide-and-conquer manner. Mao et al. [142] propose a model called *Decima* to address the scheduling problem in data processing clusters. The *Decima* introduces DAGs to represent the graph of scheduling jobs and their dependencies, and it combines convolution-based GNN and policy gradient algorithm REINFORCE to embed the nodes and train the whole model. Ou et al. [162] address recurrent disease prevention on a social network by using GCNs and a two-level hierarchical DQN framework. Other applications can be seen in **Virtual Network Function (VNF)** placement [184], adversarial attacks [186], connected autonomous vehicles [38], VNF forward graph placement [226], route optimization [7], and mobile crowdsourcing [228]. They are all cases of successful application of ML method on graph-based problems.

In summary, GNNs can learn the effective representation of graphs by their powerful learning ability, capturing the inner local and global features. The resulting embeddings can be utilized for future tasks such as classification or decision-making. In recent years, the GNN-based RL framework shows big potential in solving CO problems, especially for node selection problems such as MVC and MIS. On one hand, nodes can still be selected step by step by an auto-regressive method [107]. On the other hand, the selection probability of nodes can be directly estimated according to their features and graph topology, and it can use a searching technique to construct the feasible solution [98].

7 ML-BASED IM ALGORITHMS

As shown in Sections 3 and 5, the IM problem is a branch of CO problems, which is a typical node selection problem. Traditional algorithms for solving the IM problem usually require expert

knowledge and suffer from high computational costs. In addition, it lacks generalization ability for IM algorithms used in evolving graphs or other variant problems. ML-based methods using deep learning techniques have the potential for solving these drawbacks. There are pioneering studies [3, 132] combining RL with the IM problem. However, they do not model the influence estimation as an RL problem or node influence as a learning task. Thus, many subsequent pieces of research [124, 141, 209] take advantage of graph embedding methods to learn the representation of nodes or networks which captures the underlying association features for further use and contributes to generalization ability.

In this section, we begin to review the deep learning techniques related to the vanilla IM problem, which is strictly defined by Definition 5, including graph embedding, RL-based methods, and other ML-based methods. Here, we first introduce some interesting techniques of graph embedding related to the IM problem. Then, we mainly introduce the standard process of RL algorithm that takes the current embedding of nodes or networks as the state of the RL agent in order to choose the next seed node as action. Finally, we review some other latest ML-based methods to address it.

7.1 Graph Embeddings

The GNNs introduced in Section 6 are very suitable for embedding social networks, which exhibit considerable power in many applications. Also, plenty of variants are developed for diverse purposes. We describe a few of them which are applicable to the IM problem. Based on the classic *word2vec* method [146, 147], Feng et al. [64] propose a social influence embedding method *Inf2vec* against the sparsity of observed propagation data. Under the tasks of node activation and influence diffusion prediction, it outperforms the SOTA baseline algorithms, and it is applicable to the IM problem with unknown diffusion probabilities [164]. They try to get the representation of the node that encodes influence context, and then compute the similarity of every two node embeddings as an indicator of edge propagation probability between them, instead of directly estimating the edge influence probability, which is affected by the sparse edge propagation data. In order to encode the influence context, they devise *Inf2vec* which utilizes *word2vec* to learn node embedding from the network structure, observed influence diffusion history, and similarity of user interest. Specifically, the network structure and influence diffusion can be extracted from the results of a random walk strategy, and the similarity information of user interest is characterized by sets of users who perform the same action. Subsequently, Ying et al. [234] propose a deep GNN model named *DiffPool*, aiming at the problem of ignoring hierarchical structure in graph embedding. The main idea of *DiffPool* is to map the whole graph to a single embedding vector in an end-to-end differentiable manner with hierarchical structures encoded. It has an architecture of multiple layers of GCNs. In each layer, the nodes of the input network are first encoded into embedding vectors and then clustered according to their embedding vectors to form a coarser network, in which each cluster is regarded as a new node, and edges between clusters are preserved. The input network is coarsened iteratively to a single embedding vector of the entire network, and the hierarchical features are encoded. The model is first raised for graph classification and achieves 5–10% improvement in accuracy compared with the SOTA benchmarks. Then, Kamarathi et al. [100] adopt it to solve the IM problem under an unknown graph. Most existing ML-based solutions to graph optimization problems like IM utilize two-stage manners which train the representation learning model first for the downstream optimization tasks [14, 20], while recent preferable approaches take a pure end-to-end approach by implanting downstream optimization tasks into the training with embedding learning [107, 111, 208]. Between these two extreme manners, Wilder et al. [219] integrate a differentiable solver for a simpler problem of the true complex problem as a layer of the architecture and learn the mapping representation from the true problem to the simpler problem for

the solver. They design an end-to-end decision-focused learning framework with GCNs for embedding, which is suitable for the problem of selecting an optimal seed set of k nodes such as the IM problem. This method focuses on learning representation rather than optimization and shows better performance than two-stage or conventional end-to-end methods on a subset selection task.

In real applications of the IM problem, different scenarios focus on different context information of social networks, such as topic-aware and location-aware, which gives new challenges to embedding algorithms. Here, we briefly introduce some context-aware embedding methods. Wu et al. [222] take the evolving nature of social networks into account and propose a DNN-based embedding approach, *DiffNet*, modeled by a layer-wise influence diffusion structure. As a graph based model, *DiffNet* shows better performance than the non-graph based recommendation models. Wu et al. [221] then improve *DiffNet* structure and propose a *DiffNet++* model, which considers the social network and interest network of users to learn the user embedding in recommendation systems. It embeds both the user-item graph and the user-user graph for the social recommendation and is shown to be effective by the experiment results. Khajehnejad et al. [106] propose an adversarial embedding method, in which an auto-encoder is used for node embedding and a discriminator is used for discriminating sensitive attributes, so as to achieve similar distributions among sensitive attributes, thus protect sensitive attributes in fair IM scenarios. Nishad et al. [159] devise a position-aware GNN called *GraphReach* to improve embedding effectiveness by considering positional information. Yu et al. [235] devise an influence-aware GNN which can encode the influence of a node in a network, facilitating subsequent tasks such as the IM problem. Chen et al. [40] present a comprehensive review of recent works that leverages Graph Representation Learning for the popularity prediction problem, in which a lot of ML-based techniques are used to study information diffusion. It can be applied to the IM problem as well.

To sum up, graph embedding techniques in the IM problem can learn IM-related features, such as hierarchical features, community features, and various context-related information. For a specific application scenario, the existing embedding models can be tailored or advanced according to the specific objective or trait to adapt to the problem need.

7.2 RL-Based IM Algorithms

7.2.1 Single-Agent RL without Pruning. Some methods attempt fusing DRL into greedy the IM framework, estimating the marginal influence gain of each node by DNN and selecting the most influential one each time. Li et al. [124] extend the method in [107] for generic IM problem and propose the DISCO framework composed of GNNs and DQN, where the action is designed as selecting a node into the seed set and the reward is its marginal influence gain. Contrary to hand-crafted feature, DISCO embeds the nodes in the graph network as feature vectors through a recurrent GNN named *structure2vec* [55] which takes account of the connection information of nodes in the graph, as well as the partial solution seed set S , resulting in better generalization ability. The framework is depicted in Figure 5. Once the seed set S updated, DISCO would update the embedding vectors $\mu_v^{(i+1)}$ as follows:

$$\mu_v^{(i+1)} = \text{ReLU} \left(\alpha_1 \sum_{u \in \mathcal{N}(v)} \mu_u^{(i)} + \alpha_2 \sum_{u \in \mathcal{N}(v)} \text{ReLU}(\alpha_3 w(v, u)) + \alpha_4 a_v \right), \quad (16)$$

where ReLU refers to the Rectified Linear Unit of a neural network, $\mu_v^{(i)}$ is the embedding vector of node v during the i th iteration, $\mathcal{N}(v)$ is the set of neighbors of node v , $w(v, u)$ is the weight of edge (v, u) which can correspond to diffusion possibility under the IC model, a_v is a scalar of 1 or 0 to indicate whether node v is in partial seed set S currently, and α_1 , α_2 , α_3 , and α_4 are trainable parameters which can be learned from the given IM problem. DISCO defines a Q-function of DQN as follows, which is the approximate function mapping from the embedding vectors of v and all

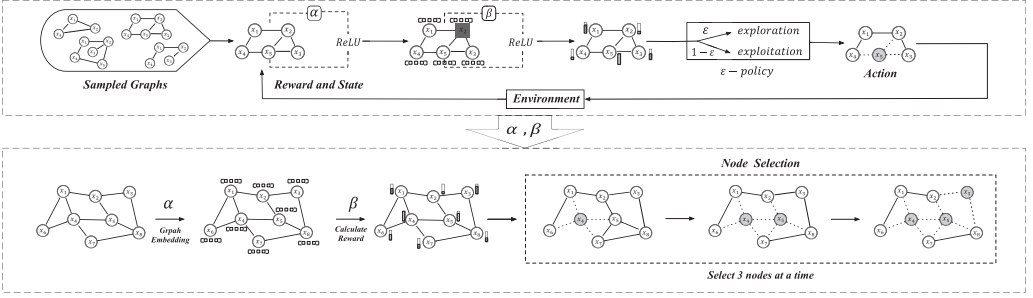


Fig. 5. An illustrative framework of DISCO that incorporates network embedding and DRL [124], where the upper part illustrates embedding learning and the lower part refers to seed selection. Image extracted from [124].

other nodes to the marginal influence of node v given seed set S . That can be defined as follows:

$$Q(v, S, \Theta) = \beta_1^T \text{ReLU} \left(\left[\beta_2 \sum_{u \in V} \mu_u^{(I)}, \beta_3 \mu_v^{(I)} \right] \right), \quad (17)$$

where $[\cdot, \cdot]$ denotes the concatenation operator, β_1, β_2 , and β_3 are learnable parameters, and Θ represents the set of all related learnable parameters, i.e., $\alpha_1, \dots, \alpha_4$ and β_1, \dots, β_3 . As we know, computing the marginal influence is #P-hard. DISCO trains a Q-function to predict the marginal influence gain of a node based on the generalized node representation, avoiding sampling the stochastic diffusion process of traditional approaches which is costly. For the application of the trained model, the authors propose a seed selection method that would select all k nodes for a seed set at once according to decreasing Q-values, given an empty partial solution set instead of picking only one node at a time by hill-climbing greedy strategy, which accelerates the process. They claim that the order of the nodes with respect to their Q-values remains almost unchanged whether or not they do a hill-climbing greedy strategy with multiple-round computing of network embedding and Q-values. DISCO outperforms traditional methods like IMM [191] and SSA [157] in terms of computational efficiency with comparable influence quality of the seed set. Even though this work basically follows the procedures of S2V-DQN [107] and its experiments are not robust and convincing, it is an important work to address the IM problem by using the latest RL algorithm.

Soon afterward, Li et al. [123] (same as the authors of DISCO) proposed the PIANO framework, which is an evolved version of DISCO. Its general idea is the same as DISCO's, but it perfects the algorithm design and theoretical analysis and supplements a large number of comparative experiments. Based on the framework of DISCO, Wang et al. [209] propose the IMGER for large-scale and dynamic social networks. The IMGER utilizes GAT [206] for network embedding, which can learn the weights of edges from data by attention mechanism instead of pre-defined weights related to the diffusion model in order to fit diverse realistic social networks. **Double DQN (DDQN)** [202] algorithm in DRL is used to learn a greedy policy and select the seed set to maximize social influence in a similar manner to DISCO. The results show that the IMGER demonstrates superior performance in both the social diffusion prediction and the quality of the IM solution set. To enhance the generalization ability, Chen et al. [42] propose an end-to-end framework called Three Coupled GNNs and Double DQN (ToupleGDD) to solve the IM problem by incorporating *DeepWalk* and three coupled GNNs for network embedding and DDQN for parameters learning. They claim that ToupleGDD can be trained on small randomly generated graphs and tested on totally and completely diversified networks. Their results have been validated in different real-world datasets, whose performance is very close to SOTA IM algorithms. Actually, this is a milestone work of pursuing a stronger generalization ability by applying DRL to the IM problem.

7.2.2 Single-Agent RL with Pruning. To enhance the efficiency of DRL-based IM algorithms, some methods integrate network pruning techniques to navigate the computing resources focusing on more important nodes or subnetworks. Shown as Section 6, Manchanda et al. [141] propose the GCOMB framework that can be used to solve the IM problem as well through introducing a node quality evaluation mechanism. Unlike the end-to-end architecture in DISCO, GCOMB has two separate components of network embedding and Q-learning using both supervised learning and RL to train the model. One component is a convolution-based GNN named GraphSAGE [84] employed to prune poor nodes and learn embedding of good nodes in a supervised manner, where the quality of a node v takes into account the influence overlapping between partial solution S and v . The other component is the Q-learning RL architecture which only considers the good nodes for the solution. They conduct a large number of comparative experiments with the SOTA IM algorithms, including S2V-DQN [107], GCN-TREESEARCH [130], Greedy (Algorithm 1), CELF [119], IMM [191], and OPIM-C [189]. It shows that the pruning method largely reduces the computational cost, making it scalable to billion-sized graphs with exponentially less time consumption than other algorithms. The GCOMB can be generalized to unseen graphs of significantly larger sizes and convincingly outperform the SOTA IM algorithms. Ireland et al. [93] propose a framework called LeNSE to prune graphs in order to further reduce the running time for CO problems. Specifically, it identifies qualified subgraphs from the original entire graph, which can represent the original one in terms of implicating the solution, and adopts heuristics in those subgraphs to get a solution within a reasonable time. The results show that the LeNSE outperforms the corresponding pruning methods in GCOMB.

7.2.3 Multi-Agent RL. There are also some efforts to model the IM problem as group dynamics of multiple learning agents. Khomami et al. [56] introduce a learning automata based algorithm for solving the minimum positive influence dominating set problem and apply it to address the IM problem. It assigns each node a learning agent to cooperate with each other for deciding the seed set. Singh et al. [178] utilize the two-hop influence spread to approximate the actual influence spread, and adopt learning automata based method to maximize influence. Liu et al. [137] propose a MAIM algorithm that uses multi-agent Memory Separated DQN to improve training efficiency without network embedding techniques. It designs the action as either accepting the current node and adding it to the seed set or dropping it without reconsideration by explicitly considering the influence overlapping. Ma et al. [139] combines the RL with evolutionary algorithms, using it to deal with a bunch of DQNs which make decisions on the selection of the next seed node. By using evolutionary algorithms, it can effectively compress the searching space of training in RL, thus achieving the purpose of improving efficiency.

7.3 Other ML-Based IM Algorithms

There are some studies using other ML-based methods (not RL-based methods), including using node or network embeddings as well as supervised or unsupervised deep learning methods to find influential seed nodes.

7.3.1 Unsupervised or Self-Supervised Manner. The complexity of graphs makes it difficult to obtain labeled training data in the IM problem. Thus the majority of ML-based IM algorithms resort to unsupervised or self-supervised learning approaches. The diffusion models with random or uniform influence parameters may suffer from the inaccurate estimation of realistic influence diffusion [9], which promotes estimating edge probabilities from historical data of diffusion cascades. Also, due to the number of parameters to estimate in this case being equal to a large number of edges, overfitting tends to happen. Bourigault et al. propose a practical method

that forms an edge probability dependently with the pair of embeddings corresponding to the connected influential and susceptible nodes [24]. Following this idea, Panagopoulos et al. [164] propose a model-independent method called CELFIE, introducing a node embedding method to form the activation probabilities between nodes for reconstructing a simpler bipartite network for seed set selection in the IM problem. The embedding model *L-Inf2vec* is proposed based on *Inf2vec* [64] which is introduced previously for influencer and susceptible node embeddings, where the influencers refer to nodes sending influence and the susceptible nodes refer to those receiving influence. Then, the two parts of a bipartite network are built up by connecting influencers to susceptible nodes where the weight of an edge is the diffusion probability computed with influencer and susceptible node embeddings. Then, the IM problem could be solved under this bipartite network by simple neighborhood sampling, with the simulation on the traditional diffusion model discarded, which accelerates the computation of influence spread.

Nevertheless, CELFIE has the drawbacks of relying on sampling-based influence spread and not considering conditional probabilities regarding the partial seed set, which causes weak spread increment as the size of the seed set increases during seed selection. To overcome this, they subsequently propose IMINFECTOR, an extension of CELFIE based solely on diffusion cascades [163]. IMINFECTOR also utilizes the influence and susceptibility node embeddings learned from cascades to form the diffusion probabilities between nodes by a multi-task neural network [33], but an expected influence spread is assigned to each node by taking into account the partial seed set. The experiment results show that IMINFECTOR outperforms several competitive methods including CELFIE in spread quality or time efficiency. Still, due to the chronological order in which IMINFECTOR retrieves the context of nodes, there exists the possibility that less influential nodes are actually selected as candidate nodes. To overcome this limitation, Liu et al. [136] propose LDRLIM that obtains the influence of the node according to node embeddings. LDRLIM uses **Discount-degree Descending (DED)** search strategy to generate the candidate nodes set and IC model based random Walk from candidate nodes to produce node propagation sequences for learning node embeddings. The IC model is taken and the diffusion probability of an edge depends on the 2-norm similarity between influencer and susceptible node embeddings, which is followed by the CELF method for the IM solution. Xie et al. [225] also assume unobservable diffusion probabilities under the IC model, and propose an approach to predicting probabilities by neural network and maximizing influence. They first compute the embeddings of nodes by using an **multi-layer perception (MLP)** model to take as input hierarchical community clusters, node attributes, and general network structure. The diffusion probability of an edge can then be calculated based on the 2-norm similarity between connected nodes. With the diffusion probabilities, the seed set is figured out via a greedy algorithm and outperforms the SOTA methods for the IM problem under attributed networks.

In addition, Li et al. [128] propose a deep learning algorithm MAHE-IM for the IM problem based on heterogeneous networks, designing an embedding method to extract high-order representations from trajectories by metapath-based random walk method *metapath2vec* [61]. By comparing two nodes' cosine similarity and setting a similarity threshold for relevance, they figure out a set of relevant nodes corresponding to each node and select top-*k* node with the highest occurrence frequencies among those sets. Extensive experiments are conducted against 18 SOTA baseline algorithms, showing the effectiveness of MAHE-IM on heterogeneous networks. To balance the scalability and high performance in the IM problem, Zhang et al. [242] modify a GCN with an adapted number of layers to different scales of networks and train it with a self-labeling mechanism by binning nodes regarding degree. Then, they use a distance-based heuristic IM method, taking as input the generated node embeddings, to compute the seed set and achieve better performance for seed set identification tasks with high efficiency than the baselines

such as work [103]. Ni et al. [158] reduce the IM problem to a budget-constrained multi-hop dominating set problem. The idea is to learn a scoring mechanism for all nodes such that each node's score can represent its topology-integrated influence in order to directly choose the top- k nodes as seeds. Specifically, they propose an unsupervised framework, FastCover, constituted of a multi-layer GNN called **graph reversed attention network (GRAT)** to generate the influence score of each node to range $[0, 1]$. The parameters of GRAT are trained via optimizing a differentiable loss function in an unsupervised manner. The authors claim FastCover achieves a near-optimal performance with quasi-linear time complexity.

7.3.2 Supervised or Semi-Supervised Manner. Despite the challenges of collecting labeled data for the IM problem, some studies make an effort to construct semi-supervised or even supervised deep learning forms for performance concerns. Shen et al. [174] propose a **Cross-Network Graph Sparsification (CNGS)** algorithm to prune edges of large-scale graphs for accelerating downstream IM algorithms. CNGS is trained in a semi-supervised on a source network and exploits the knowledge learned to guide the removal of edges in a target network with a minor impact on influence diffusion. By removing edges unlikely to be influential, the existing IM algorithms can be implemented quite faster due to a sparser network than before, while performance can barely suffer degradation. The results demonstrate at least 39% time saved with only less than 10% performance reduced on several real-world public datasets. Soon after this work, they further improve the method and introduce a fuzzy self-training approach to enhance the cross-network prediction performance [175]. Kumar et al. [113] convert the IM problem into a pseudo deep learning regression problem that trains a generalizable model to transfer to unseen networks. With a graph-based LSTM, they extract the centrality-based feature vectors of nodes from a training network as input and compute the influence spread of nodes under the **susceptible-infected-recovered (SIR)** and **IC** model as target labels, then transfer the trained LSTM to a new target network to predict the influence of each node for top- k selection of seed set. Similarly, Kumar et al. [112], the same team as the previous, propose another supervised IM algorithm, still converting to a regression problem but adopting the node embedding algorithm *struc2vec* [170] to extract node representation, and then feed the embedding vectors into a GNN to learn node's influence, outperforming some of classical and latest IM algorithms.

Generally speaking, the design of ML-based IM algorithms is much more flexible than the traditional IM algorithms, involving the integration of deep learning, RL, graph representation learning, evolutionary algorithms, CO, and other technologies. By making full use of the advantages of each technology, it gradually improves the quality of solutions, reduces time complexity, enhances generalization ability, and lands in real business scenarios. The research of IM algorithms and social applications also promotes the development of ML and CO.

8 ML-BASED VARIANTS OF IM

The variant IM problems combine the vanilla IM problem with diverse traits of practical scenarios and they are of research value for real-life applications. In this section, we survey the latest applications for the variant IM problems and focus more on those works adopting ML-based methods. In fact, these variant IM problems have been defined and studied in the era of traditional algorithms. If you want to know more about their definitions and are interested in traditional algorithms for solving variant IM problems, please check Li's survey [127]. It is very challenging to summarize this part of the work, which mainly involves two difficulties. First, how to classify these variant IM problems; second, how to accurately judge whether it belongs to the traditional algorithm or the ML-based algorithm, and summarize its adopted technologies. The taxonomy and its corresponding representatives are summarized in Table 2.

Table 2. Summary of Deep Algorithms for the IM and Its Variant Problems (CLT, CIC, and MCTS are Short for Competitive Linear Threshold, Competitive Independent Cascade, and Monte-Carlo Tree Search, Respectively)

Year	Author	Problem	Diffusion model	Embedding	Training method
2017	Shen et al. [174]	Vanilla IM	IC	N.A.	semi-supervised
2018	Khomami et al. [56]		LT	N.A.	learning automata
2019	Shen et al. [175]		IC	N.A.	semi-supervised
2019	Li et al. [124]		IC	<i>structure2vec</i>	DQN
2019	Liu et al. [134]		IC/LT	GCN	supervised
2019	Singh et al. [178]		IC/LT	N.A.	learning automata
2020	Manchanda et al. [141]		N.A.	GraphSAGE	Q-learning
2020	Panagopoulos et al. [164]		N.A.	<i>Inf2vec</i>	self-supervised
2020	Panagopoulos et al. [163]		N.A.	Multi-task NN	self-supervised
2021	Wang et al. [209]		N.A.	GAT	DDQN
2021	Liu et al. [137]		IC	N.A.	Multi-agent DQN
2021	Ni et al. [158]		N.A.	GAT	unsupervised
2022	Zhang et al. [242]		IC	GCN	self-supervised
2022	Li et al. [128]		IC/LT	<i>metapath2vec</i>	unsupervised
2022	Li et al. [123]		IC	<i>structure2vec</i>	DQN
2022	Kumar et al. [113]		SIR	hand-crafted features	supervised
2022	Kumar et al. [112]		IC	<i>struc2vec</i>	supervised
2022	Liu et al. [136]		IC	Multi-task NN	self-supervised
2022	Xie et al. [225]		IC	MLP	unsupervised
2022	Ma et al. [139]		IC	LINE	DQN
2022	Ireland et al. [93]		IC	GNN	Q-learning
2022	Chen et al. [42]		IC	GNN	DDQN
2021	Ou et al. [162]	Blocking IM	SIS	GCN	DQN
2022	He et al. [86]		LT variant	hand-crafted features	semi-supervised
2022	Yilmaz et al. [233]		SIS	N.A.	multi-agent DDPG
2015	Lin et al. [132]	CIM	CLT	hand-crafted features	Q-learning
2018	Yang et al. [231]		CLT	hand-crafted features	MCTS
2018	Ali et al. [3]		CLT variant	hand-crafted features	Q-learning
2020	Ali et al. [5]		CLT variant	hand-crafted features	DQN
2021	He et al. [85]		CIC	hand-crafted features	Q-learning
2021	Ali et al. [6]		CLT variant	<i>node2Vec</i>	DQN
2022	Ali et al. [4]		CLT	hand-crafted features	Q-learning
2018	Yadav et al. [230]	Conting.-aware IM	N.A.	N.A.	POMDP planner
2021	Chen et al. [35]		IC	<i>structure2vec</i>	DQN
2021	Meirom et al. [144]	Dynamic IM	IC/LT	GNN	PPO
2022	Mendonça et al. [145]		IC variant	<i>structure2vec</i>	DQN
2022	Sheng et al. [176]		IC/LT	<i>DynamicTriad</i>	semi-supervised
2019	Li et al. [126]	Location-aware IM	N.A.	hand-crafted features	supervised
2020	Zhang et al. [243]		self-defined	hand-crafted features	unsupervised
2022	Jin et al. [97]		IC	<i>All2Vec</i> [97]	unsupervised
2020	Keikha et al. [103]	Multilayer IM	IC/LT	CARE [102]	unsupervised
2020	Kamathi et al. [100]	Graph-unknown IM	IC	<i>DiffPool+DeepWalk</i>	DQN
2021	Tran et al. [201]		IC/WC	Siamese NN [26]	unsupervised
2021	Li et al. [121]		IC	<i>DiffPool+DeepWalk</i>	DQN
2022	Wu et al. [223]		IC	GraphSAGE+ <i>DiffPool</i>	DDPG
2019	Tian et al. [195]	Topic-aware IM	IC	<i>Diffusion2Vecv</i>	DDQN
2020	Tian et al. [194]		IC/LT	<i>Diffusion2Vecv</i> [194]	DDQN
2020	Huang et al. [90]		CIC variant	N.A.	supervised
2021	Huang et al. [91]		CIC variant	N.A.	supervised

8.1 Competitive Influence Maximization

The CIM problem assumes multiple parties in a social network compete against each other in the propagation of information, e.g., one mobile phone provider would make a suitable marketing policy to maximize the number of its customers when there are other providers giving policies to compete in the same social network. The CIM problem can be divided into two cases: Known opponent strategies and Unknown opponent strategies. Bharathi et al. [22] first define the CIM problem as a game problem. Thus, the research in traditional algorithms for this area mainly focused on algorithmic game theory and submodular optimization. Recently, there are a lot of ML-based methods used to address it as follows.

Lin et al. [132] propose solutions for both two CIM cases of known and unknown opponent strategies. If known, they adopt Q-learning techniques of RL considering the multi-round situation, training an agent to learn a strategy to defeat the opponent's strategy, or else a game theoretical solution is used to seek the Nash equilibrium as the best strategy which a rational agent should work out. To avoid extra computational overhead caused by the high dimensional state and action space of RL, they design hand-crafted features as the state representation to represent the current state of the social network and borrow the idea of meta-learning to choose a candidate strategy rather than a node as an action. The rewards are designed as the relative marginal influence gain compared to the opponents, i.e., the difference of activated nodes between parties in the last round. Since RL agents take a large number of random exploration actions to find the best policy, which is time-consuming, Ali et al. [3] develop based on Lin's work for the **Time-Constrained CIM (TC-CIM)** scenario where the objective of each party in TC-CIM scenario is to design a strategy that can maximize its overall influence spread within its very limited time budget. They use a similar framework with hand-crafted network features as well as Q-learning under the Time-Varying Competitive LT diffusion model and apply transfer learning with the help of other knowledge to reduce the training time. Yang et al. [231] extend the work in [132] to a general situation that the parties could select nodes and propagate by turns, and propose an RL-based framework to solve the multi-round multi-party IM problem. With a similar idea in [3], Ali et al. [5] model the CIM problem under topology-unknown network scenario as an RL problem, then apply DQN to learn when to explore the unknown network and how to select seed set, achieving better performance than heuristic methods. Next, He et al. [85] study competitive opinion maximization in a signed social network where negative weights of edges are introduced to model the relationship of dislike or distrust. They exploit Q-learning to select nodes against an unknown opponent strategy. Ali et al. [6] propose a DRL-EMB framework through adopting network embedding method *node2vec* [73] and DQN algorithm to formulate evolving social networks instead of static networks. Ali et al. [4] also improves the work in [132] by integrating transfer learning in the RL method on the competitive LT model. To facilitate the transfer of the model from the source to the target network, they normalize the two networks and propose a transfer learning method to accommodate the transferred agent in regard to heterogeneity between networks and agent settings. Better or similar performance is claimed for this transfer learning based model than the baseline model trained from scratch, with greatly less time consumption.

8.2 Topic-Aware IM

The topic-aware IM problem takes into account people's interests in topics when increasing the spread of certain topic-related information. It uses topic to represent item characteristics and users' interests, formulated to find the optimal seed set that maximizes the specifically defined topic-aware influence. The research in traditional algorithms in this area can be categorized into the IM for topic-relevant targets [129, 155] and for topic-dependent diffusion [11, 39]. The first one aims at maximizing the influence over users related to a specific topic, and the second one is to adjust

diffusion models such that the diffusion probability is topic-dependent. Recently, there are a lot of ML-based methods used to address it as follows.

Tian et al. [195] apply a similar method of DISCO [124] to the topic-aware IM problem where people have different probabilities to be activated in different topics. They also extend the work in [107], using network embedding and DDQN framework. The embedding model called *Diffusion2Vec* exploits the basic architecture of *structure2vec* [55], but modified it by incorporating a weighted vector, such as $\{< music, 0.7 >, < sport, 0.3 >, < politics, 0.0 >\}$ to represent the extent of interest in specific topics of a node, into the feature attribute vector x_v in Equation (8). In addition, they adopt a data-driven method to quantify the propagation parameters of the diffusion model rather than assigning random values from range $[0, 1]$ in order to approximate real-life situations. The IC model is adopted, and the diffusion parameters of edges can be calculated as follows:

$$p_{uv}^\tau = (\gamma_1 w_{uv} + \gamma_2 \text{sim}(u, v) + \gamma_3 B_v^\tau) / 3, \quad (18)$$

where p_{uv}^τ denotes the propagation probability from u to v in terms of topic τ , w_{uv} denotes the initial weights of edge from u to v , $\text{sim}(\cdot, \cdot)$ is the cosine similarity function between the representation of two nodes, B_v^τ represents the weight of interest toward topic τ , and $\gamma_1, \gamma_2, \gamma_3 \in (0, 1)$. Tian et al. [194] (the same authors of [195]) subsequently augment the theoretical analysis and extend the experiments to the problem under the LT model. The performance of this RL-based algorithm is shown superior to those manually designed topic-aware IM baselines.

Huang et al. [90] consider the diffusion spread of a certain entity where its competitive or complementary products have already been promoted across the same network, which can be regarded as a topic-aware variant where the activation probability of a complementary topic should be larger than that of a competitive topic. This is called a follower-based IM problem by the authors. To solve this, they propose an algorithm in which a recurrent neural variational model is adopted to update the evolving thresholds in the LT model by learning the dynamics of multiple-entity interactions and cascades, then the seed set can be computed via a greedy algorithm. On the basis of this work, Huang et al. [91] soon afterwards propose a similar method as [90] to solve the follower-based IM problem, but consider adjusted IC model and enhance the influence probability inference model, using modified LSTM model in place of recurrent neural variational model. Experiments show their method outperforms the SOTA methods in terms of prediction accuracy and efficiency.

8.3 Contingency-Aware IM

Yadav et al. [230] first propose the contingency-aware IM problem where it is uncertain whether a node is willing to accept the invitation when invited to become a seed node. This was inspired by spreading awareness of HIV prevention among homeless youth. In the past, this problem has been addressed by applying the Partially Observable **Markov Decision Process (MDP)** [229] and Greedy Algorithm [220]. However, the solution to the contingency-aware IM problem needs to be recomputed when the underlying network changes and the stakeholders have no high-performance computing resources. In other words, it requires a higher generalization ability of the algorithm.

Thus, Chen et al. [35] re-study the contingency-aware IM problem by using the DRL technique. To overcome the difficulties of how the state incorporates the uncertainty information and how the reward can be computed with the uncertainty in node status, they proposed a new MDP formulation, state-abstraction component, and reward-shaping component, which constitutes the RL4IM framework through network embedding model *structure2vec* and DQN. Besides, the authors assume historical data on the users' choices are available. The historical data can provide some prior knowledge of the willingness of nodes that can be used to build up the states and

actions of DQN. The RL4IM can conduct a seed selection strategy on a laptop within seconds after sufficient training.

8.4 Graph-Unknown IM

The fact that the social network connection is sometimes unknown triggers the IM problem in graph-unknown social networks, of which the algorithms normally integrate social network discovery methods. For example in public health, the network dataset is typically gathered by in-person intervention, which is not feasible for using to solve the IM problem. Thus, in some scenarios, obtaining the entire social graph is costly because it has to survey multiple people. To deal with it, Wilder et al. [220] propose a CHANGE algorithm that is based on friendship paradox by surveying a random node and one of its neighbors. The subgraph obtained by surveying a small number of nodes is used to select a seed set for the IM problem. Subsequently, Kamarthi et al. [100] formulate the same problem of network discovery by MDP, and developed an extension to DQN architecture named Geometric-DQN to learn policies for network discovery by extracting relevant graph properties. It can learn global state representation by using *DiffPool* [234] for the discovered graph and local action representation by using *DeepWalk* [166] for node embedding. They show that their RL-based algorithm has 7–13% improvement over the SOTA sampling algorithm, CHANGE.

However, Kamarthi's RL algorithm has to perform multiple interactions with the environment, which is very costly in real-world applications. To increase the sample efficiency, Li et al. [121] formulate the network discovery and IM problem as a goal directed learning problem [8, 150] with unknown goal states. Then, they propose a Curriculum Learning Policy for Influence Maximization (CLAIM) by using curriculum guided Hindsight Experience Replay [63] and goal directed Geometric-DQN to learn efficient policies, where the influence spread can be improved by 7.51% more than existing methods. To further study this problem, Tran et al. [201] introduce an end-to-end framework, IM-META, to deal with the IM problem in attributed networks with unknown topology, where they formulate it as an optimization problem to select both the seed set and the query set by three steps: network inference, reinforced weighted graph generation, and query node selection. They show that through query only 5% of nodes, it is enough to get an effective seed set compared with other benchmark IM algorithms. Wu et al. [223] consider the influence strength between users is unknown and proposed an end-to-end RL-based framework, **Geometric Actor-Critic (GAC)** to effectively find policies for incentive allocation, a variant IM problem, under limited budgets. They use GraphSAGE and *DiffPool* models to learn the representation of nodes and the entire graph, then feed the representation into **Deep Deterministic Policy Gradient (DDPG)** [131] to compute the action of which node to choose for the seed set. Even though the GAC outperforms other baselines, it needs two node matrices as input, which is too large to handle extremely large social networks.

8.5 Dynamic IM

In real applications, the social network is dynamic and keeps evolving. For example, new friendship forms, where a number of edges are added or removed at any time. It leads to additional challenges for the IM problem. Ohsaka et al. [161] first propose a dynamic mechanism under the IC model in evolving graphs, which can give a seed set in real-time when there is node or edge updating. In addition to adopting traditional sampling-based approaches, there are some ML-based methods emerging recently.

Meirom et al. [144] consider studying the problem of controlling partially-observed diffusion on an evolving graph, and based on it, present the partially observed IM problem. They formulate it as a partially observed MDP, then select a subset of nodes (seed set in the IM problem) for dynamic intervention as a ranking problem. In order to deal with underlying dynamics, they design an

actor-critic RL algorithm that contains two GNN modules, one updates the node representation based on dynamics, and the other cares about long-range information diffusion. In addition to the IM problem, this framework can be used in epidemic tests, traffic control, and active sensing. Mendonça et al. [145] design the **Spatio-Temporal Influence Maximization (STIM)** in evolving networks where the connectivity between nodes is constantly changing, which is a model trained with RL and graph embedding (*structure2vec*) aiming to find a suitable moment to begin information diffusion from a seed set such that its influence spread can be maximized. They claim that STIM outperforms the greedy agent and can learn complex patterns hidden in evolving networks without supervised data or handcrafted strategies. Sheng et al. [176] solve the dynamic IM problem by adopting embedding method *DynamicTriad* [251] as well as modified seed selecting algorithm in *DeepIM* [103]. Under the dynamic network, the embeddings would take into account the dynamics of edge-add/remove and weight-change with time, then an updating algorithm is designed to update the seed set by incorporating the change of node insertion and deletion. This is a pioneering work of applying ML-based methods to address dynamics in social networks, and it can be changed to those scenarios under different budget constraints [57, 200].

8.6 Multi-Layer IM

In addition to the spread in a single social network, in reality, information can also propagate across multiple heterogeneous networks via shared users [67, 154] at the same time, which means that users can be influenced by those users from other social networks [241]. This motivates the studies of the multi-layer IM problem. This problem has been studied with traditional algorithms, such as [78], and recently studied with ML-based methods.

Keikha et al. [103] propose a *DeepIM* method which adopts the community-aware method CARE [102] to embed the nodes for the IM problem under the interconnected networks scenario. By measuring the similarity of embeddings, the algorithm chooses r most similar nodes for each node as its similarity set, then figures out the top k nodes that are contained by the most sets as the seed set. Besides the improved performance compared with Greedy [104], CELF++ [71], SIMPATH [72], and LDAG [47], the experimental results indicate the importance of shared nodes between different networks, around 20% of which are selected as the seeds. Katukuri et al. [101] design an algorithm to identify the influential nodes in multi-layer networks by detecting nodes with strong connections in both inter- and intra-layers, and also employ node pruning technique to improve the computing efficiency.

8.7 Location-Aware IM

The location-aware IM problem is defined in location-based social networks, which aims at maximizing the influence spread of location-relevant users rather than any users in the vanilla IM problem. Li et al. [122] is the first to consider the location-aware IM problem under the IC model, selecting a seed set to activate the maximum number of users in a geographical region. Then, a series of research exploits traditional proxy-based [213] or sampling-based [181] algorithms to deal with the local-aware IM problem. Recently, there have been several ML-based frameworks to address similar issues defined in geo-networks.

Li et al. [126] study a vehicle selection problem by following the location-aware IM problem, which selects the most suitable candidate vehicles for mobile advertising in order to achieve maximum influence in vehicle networks where spatial and temporal centrality is considered as a key factor for information diffusion. They adopt MLP to predict the future centrality and trajectories of vehicles, and thus adopt preferable seed vehicles. Zhang et al. [243] define a geo-demographic IM problem, which is to find a set of locations within a budget that maximize expected reach given a distribution of population and point-to-point movement statistics. They

first give an approximation algorithm to address this problem but it is not efficient. Then, they propose an ML-based variant, NN-Sower, that uses randomization and deep learning to improve efficiency with a slight loss of quality. Jin et al. [97] want to incorporate user preferences and improve the efficiency in the location-based IM problem. They propose a new framework, *IM2Vec*, that includes a representation learning model, *All2Vec*, to capture user preferences and applies the technique of RIS to get an efficient approximation algorithm with $(1 - 1/e - \epsilon)$ approximation.

8.8 Blocking IM (Rumor Blocking)

OSNs have dramatically changed people's lifestyles, but also have caused widespread misinformation like rumors and fake news. Thus, the blocking IM problem has emerged, which aims to prevent or alleviate the spread of misinformation [239]. The blocking IM problem is usually of two categories according to employed strategies: spreading truth against misinformation (positive cascades) [28, 78] and removing key nodes or edges to suppress the propagation of misinformation [62, 108].

8.8.1 Positive Cascades. Instead of a known diffusion model, Tong [196] considers the situation where misinformation propagates under an unknown Triggering model and proposes a learning method called StratLearner to select k protector nodes given certain attacker nodes to diminish the spread of misinformation. Given a historical dataset of the target social network containing attacker-protector set pairs $\{(M, P)\}$, where the protector set M can be regarded as the label corresponding to the attacker set P for supervised learning, however, it cannot cover all the combinations of nodes in a network. Thus, StratLearner defines a scoring function of M and P to estimate the utility, with a parameterized form as $f(M, P) = \mathbf{w}^T G(M, P)$ where M denotes a set of attacker nodes, P denotes an optimal or sub-optimal set of protector nodes, $G(M, P) \in \mathbb{R}^l$ denotes a feature function extracted from m random sampled subgraphs, and $\mathbf{w} \in \mathbb{R}^l$ is a vector weight to learn through a structural **Support Vector Machine (SVM)**. The score of P given M is designed to indicate its prevention effect related to the number of nodes that are not influenced by M after cascade, and it can be used to compute the marginal gains of nodes in a greedy framework to find out the optimal seed set. StratLearner realizes set function learning with sets as input and output, and outperforms Naive Bayes, MLP, GCN, and **Deep Set Prediction Networks (DSPN)** [245]. With the same scenario of selecting k seeds to conduct a positive cascade diminishing the spread of misinformation, He et al. [86] extend the LT model of information diffusion considering the opinions and confidence of users, and modify a GCN for directed graphs to select the influential nodes for restrain the spread of a rumor. Yilmaz et al. [233] tackle the fake news propagation problem from a game-theoretic perspective as well as a multi-agent DRL perspective, the former of which models the nodes in the network as actual players while the latter of which adopts multi-agent DDPG to learn the strategy of selecting seed set.

8.8.2 Removal of Key Nodes or Edges. Active screening aims to find potentially infected individuals and provide testing and treatment in order to slow down the spread of disease. It can be regarded as targeting the seed set of people that might maximize the spread of disease, which is also a kind of variant IM problem. Ou et al. [162] apply the combination of network embedding and DRL to the active screening problem for recurrent disease prevention on a social network under the SIS model which is similar to the IC model, where the GCNs [217] are used for generating node embedding.

8.9 Diffusion Predicting IM

Most existing IM algorithms are applied to social networks with known graph structures and hand-crafted diffusion probabilities of edges. However, in reality, the diffusion probabilities are usually

implicit and costly to acquire, which issues the demand for learning the influence probabilities from network structure and historical cascades. It is referred to as the diffusion predicting IM problem [74, 172], which assumes that the diffusion model is known but the model parameters are unknown. Here, we review some ML-based diffusion predicting IM algorithms.

8.9.1 Edge Probability Prediction. Bourigault et al. [24] learn the parameters of the IC model by considering the similarity between node embedding vectors. They first embed the social network into a latent space to get node embedding vectors based on the community structures. Then, the influence diffusion probabilities of the IC model are computed by using the embeddings' distance information. Lee et al. [116] propose to construct brain networks by identifying edges contrasting between groups. They combine multiple CNNs in their model to learn a weighting function for each edge according to time series samples, which captures the correlations between brain regions and infers weighted undirected graphs for network construction.

8.9.2 Cascade Increment Prediction. Li et al. [120] learn an end-to-end model called *DeepCas* to predict the influence cascade growth. *DeepCas* first samples node sequences by using the random walk method from cascade subgraphs at each timestep. Then, feed those sequences into a GRU [52] model with the attention mechanism to learn the embeddings of the subgraphs. Based on the subgraph embeddings, an MLP is utilized for the prediction of cascade size. Liu et al. [134] propose a GCN-based model to predict the influence probabilities by taking into account both node and edge features as well as cascade information, which is trained with subgraphs for efficient concern and achieves 10–20% higher spread scores in the IM problem than robust baseline methods. Chen et al. [49] propose a Recurrent **Cascades Convolutional Networks (CasCN)** to predict the increment size of cascade next. CasCN samples sub-cascade graphs instead of random walk sequences for the preservation of cascade dynamics and local structures. GCNs and LSTMs are used by CasCN to extract topological and temporal features from the cascade subgraphs, and then produce an aggregated embedding vector which is fed into an MLP for the prediction of the cascade increment within a fixed time interval. It shows that CasCN outperforms the SOTA baselines such as *node2vec*, *DeepCas*, and Topo-LSTM in cascade prediction. Also, Zhao et al. [247] adopt a similar idea of [49] to build up a model named CasTCN for the prediction of increment size of a cascade. Recently, Ling et al. [133] propose a learning-based IM framework called DeepIM which combines deep graph representation learning for the latent representation of seed sets with optimization techniques to learn the diversified information diffusion pattern in a data-driven and end-to-end manner. The proposed method characterizes the complex nature of the seed set by directly searching for a more optimal seed set in continuous space and models the underlying diffusion pattern using two different learning-based diffusion models. It is claimed to achieve the best results in finding a seed set to maximize the influence compared with other SOTA. Chen et al. [36] propose a graph learning framework called CMINet to address **Content-aware Multi-channel Influence Diffusion (CMID)**. CMINet models the content-aware influence diffusion processes in a multi-channel network and predicts newly influenced users given a set of seed users with different multimedia contents. They propose a **Diffusion Graph Neural Network (DiffGNN)** to encode the influencing power of users and **Influence-aware Optimal Transport (IOT)** to align the embeddings to address the distribution shift across different diffusion channels. The experimental results show that CMINet outperforms 11 SOTA baselines on three real datasets.

8.9.3 Node Probability Prediction. Wang et al. [211] claim that *DeepCas* under-explores cascades by considering deficient traits of cascade structures. To overcome this, they propose an LSTM-based model named Topo-LSTM to predict the probability of a node being activated next given previous cascades. Topo-LSTM introduces diffusion topology, which is a DAG depicting

cascade till the current timestep, incorporating rich information of activation attempts and activation outcomes over timestep series. Topo-LSTM first learns the representation of nodes in the dynamic DAGs, which then are used for predicting activation possibilities by measuring the embedding similarity. Qiu et al. [168] focus on predicting the node-level influence probabilities according to both network structures and node-specific features. They propose an end-to-end supervised model, *DeepInf*, which integrates node embedding, GCN, and GAT techniques to predict the activation status of a node. By considering the activation status of the neighbors and the local subgraph structure of the target node, *DeepInf* can output the probability of the target node being activated, and outperforms the baselines with hand-crafted features on public social network datasets. Rather than assuming a known network structure like Topo-LSTM, Islam et al. [94] consider the problem of predicting the activation timing and the probability of the activation of a certain node only based on observed cascades, without knowing its neighborhood. The proposed model called *DeepDiffuse* is composed of two main LSTM-based deep network parts in the architecture: one generates cascade embeddings, and the other calculates the next activated node and timing, which has better performance than *DeepCas* and Topo-LSTM. To mitigate the drawbacks of only utilizing either network structural information or diffusion cascade information for diffusion prediction in aforementioned approaches such as Topo-LSTM and *DeepDiffuse*, Yuan et al. [237] propose a GCN model named DyHGCN which can incorporate both structural data and diffusion cascade data as well as users' dynamic preference in order to model the complexity of the diffusion process and predict user activation probability more precisely as their preferences vary with time. Also considering using both structural data and diffusion cascade data, Feng et al. [65] propose a model named H-Diffu to learn the representation of diffusion from encoded hyperbolic spaces instead of previous Euclidean spaces of structure and diffusion cascade. H-Diffu shows a better ability to capture hierarchical structures and outperforms several SOTA models such as DyHGCN.

8.10 IM-Based Problems

Recently, there emerge deep learning methods that solve optimization problems on social networks from an IM perspective. Corò et al. [54] integrate the IM problem into a link recommendation, which maximizes the social influence of a given targeted users' set by suggesting a fixed number of new connections. They propose a greedy-based algorithm that can give a constant approximation ratio and can be used in large-scale real-world social networks. Zhang et al. [244] advocate an IM-based method to enhance the efficiency of graph-structured data selection for GNN. Yu et al. [236] propose a CNN-based framework for recommendation dealing with hypergraphs that can model the high-order relations of users. Self-supervised learning can be integrated to compensate for the aggregating loss in the model by maximizing mutual information within the hypergraph. Zhu et al. [252] propose an approach motivated by the IM problem to facilitate the training of pre-trained transferable GNN, then produce guidance of transferability of GNNs between source and target graphs. Zhou et al. [250] study an IM-based experience replay mechanism that selects experience nodes based on influence function in order to alleviate the forgetting problem in generic GNNs on learning a series of graph-related tasks. Ma et al. [138] apply an IM-based method to help construct an effective attack strategy on several GNN architectures.

Thus, the IM problem can also be used as a tool to solve other ML problems or in the data science field, which is very interesting and give a new research direction.

9 RESEARCH CHALLENGES AND DIRECTIONS

In this section, we summarize some significant research challenges in this area, including the difficulties for ML-based approaches to solve CO problems and the difficulties for ML-based

approaches to solving the IM problem. Finally, we give several directions for ML-based IM algorithms in the future.

9.1 ML-Based Algorithms to Solve the CO Problem

Because the solution quality of a CO problem directly output by the DNN cannot be compared with that of professional solvers, and the search space is too large, some auxiliary measures are needed to further improve the performance, such as beam search and local search, thus we cannot completely get rid of the dependence on traditional algorithms. It indicates that the current deep models still have a lot of room for improvement. Besides, in the training of DRL algorithms, they usually design an end-to-end model which includes graph embedding and RL model. However, commonly used training algorithms, such as REINFORCE, DQN family, and Actor-Critic, have the defects of low sampling efficiency and slow convergence. Thus, how to improve the graph embedding and RL training algorithm according to the actual needs of a specific CO problem is a very important challenge.

As we know, the IM problem is a specific CO problem on graphs. Thus, the main challenges of ML-based CO algorithms also apply to the IM problem. We summarize them from five different perspectives.

- **Scalability:** It refers to the performance when running on large datasets with the same distribution. This property is very important for the IM problem because the social network is usually large-scale, including hundreds of millions of users. However, current ML-based algorithms cannot guarantee the quality of the solution when the graph is large. Compared to small-scale networks, the quality of the solution decreased obviously.
- **Adaptivity:** It refers to whether the algorithm can adapt to perturbation or adjustment in the problem. Back to the IM problem, there are a lot of related variants, for example, changing the objective function in a small range, and increasing or decreasing constraints. An ML-based framework can perfectly solve the vanilla IM problem but cannot effectively adapt to variant IM problems.
- **Generalizability:** It refers to the performance when running on unseen instances, which is the reason why we attempt to use ML-based algorithms to solve the IM problem. In the IM problem, this can avoid repeating the process of seed set selection when the graph changes dynamically. Thus, we want to let the ML-based algorithms be trained by using a number of small-scale graphs, and it can work well in large-scale new graphs. Chen et al. [42] have made efforts to this end, and they improve and validate the generalization ability of the DRL algorithm with multi-layer GNNs.
- **Efficiency:** It refers to the running time. In ML-based algorithms, its running time can be divided into training time and prediction time. The training time mainly depends on hardware conditions, dataset size, and convergence speed of the algorithm. As an algorithm designer, its main goal and difficulty lie in optimizing the convergence speed of algorithms, and the convergence speed of existing RL algorithms is also a big challenge, requiring multiple interactions with the environment. The prediction time is much shorter than the training time, but there is room for further improvement due to the large scale of social networks. It is the key to whether its algorithm can be applied to real-time environments.
- **Automation:** It refers to whether ML-based algorithms can automatically achieve scalability, adaptivity, and generalizability. For example, it can automatically revise parameters or structures to adapt to the change in network scale, disturbance of problems, and new unknown networks without the need for redesigning algorithms or retraining. This automatic framework not only meets all requirements of real applications, but also has intelligent adaptability to the environment. This is also the ultimate goal of artificial intelligence.

ML-based algorithms to solve CO problems have developed rapidly in recent years, but they are still in their infancy. Up to now, we can only deal with some simple and typical CO problems, whose scope of application is far less than the traditional optimization algorithm. Although the above goals have not been achieved at present: scalability, adaptivity, generalizability, efficiency, and automation, they point out a way for future research. Using this research framework to solve the IM problem can effectively alleviate the bottleneck faced by traditional algorithms, but it leads to a lot of new issues needed to be solved.

9.2 ML-Based Algorithms to Solve the IM Problem

The IM problem is a typical subset selection problem, where the construction of the solution has nothing to do with the order of nodes. Thus, we can adopt the GNN-based RL framework to effectively get a feasible solution. However, things are not easy because the IM problem has its unique difficulties, which leads to the inability to widely adopt ML-based algorithms. We discuss them from the following points.

- **Diversified Diffusion Models:** The current DRL framework to solve the IM problem adopts graph embedding to represent the states of nodes, which is shown as Equation (16). It relies on the weight $w(v, u)$ of each edge $(v, u) \in E$. It is perfect to be used in the classic IC model and LT model because there is a weight associated with each edge. However, these simple diffusion models cannot meet the needs of real applications. The real diffusion process is usually obtained by mining diffusion datasets, thus the diffusion model formed in this way cannot be simply represented by edge weights. Therefore, it brings difficulties to the training of node representation and RL. To address this challenge, we can consider studying how to combine DNNs with the attention mechanism. Through the attention mechanism, we can directly train and quantify the influence of a node's neighbors on it. Moreover, some studies have explored data-driven non-modeling methods for optimization problems [197, 198, 216], by abandoning the objective function and training a solver from data of input-solution pairs. We can borrow ideas from these methods to solve the IM problem without assuming the latent diffusion model. Note that we specifically discuss this model-agnostic problem toward the ordinary IM problem where the goal is normally to optimize the expected influence spread within a round rather than the cumulative gain in multiple rounds. Besides, there also exist works that study model-agnostic IM from an online perspective [117, 205], which are normally with bandit algorithms and beyond the scope of this article.
- **Training Efficiency:** The training process of the current DRL framework is shown in Equations (16) and (17). A serious defect here is that the embedding μ_v^{i+1} depends on a_v , whether or not the node v is in the current seed set S . Thus, when we update the seed set S in each iteration, we have to embed all nodes in the graph again. This is very time-consuming and inefficient. When the network scale is large, the training time will become unacceptable. Thus, this scheme is not perfect. A feasible idea is to study how to encode a set, that is, the seed set in the IM problem. Here, we can consider separating the partial solution S from the encoding of candidate nodes v , that is, the embedding of a node v does not depend on whether it is in the partial solution S , thereby reducing the update times of graph embedding and the time complexity of training. Then, it can make Q value satisfy

$$Q(enc(S), enc(v); \Theta) \propto \sigma_{\Omega}(v|S). \quad (19)$$

- **Reward Estimation:** In the process of designing RL algorithm, a key step is to define the reward function. For the IM problem, when adopting action v based on current state S , the reward function is usually defined as the marginal gain, $R(S, v) = \sigma_{\Omega}(v|S)$. However,

we have known that this value is very difficult to accurately compute. Now, this value can be estimated by using MC simulation [123, 209], which greatly increases the training time. Here, we can adopt some techniques used by traditional algorithms, such as proxy- and sampling-based approaches to accelerate this estimation process. The #P-hardness of computing the influence spread in the IM problem is also a huge challenge in ML-based algorithms, which leads to its inability to completely get rid of its dependence on traditional algorithms, and makes the scalability and generalization ability worse. Thus, studying how to overcome the #P-hardness of computing the influence spread in ML-based algorithms is a feasible research direction.

- **Variant Influence Maximization:** Shown as Section 7, there are a lot of variants based on the IM problem, which is mainly reflected by the following differences. Generally, they have different objective functions designed according to various application scenarios, which usually cannot guarantee to meet some good properties, such as monotonicity and submodularity. Perhaps, there have more complex constraints or a higher requirement for real-time and generalization ability. The variant IM problems usually have complex characteristics such as multi-objective, multi-constraints, and non-static, which cannot be effectively approximated by traditional algorithms. Thus, we can consider using ML-based algorithms to study these kinds of problems with bad properties and make full use of the advantages of ML-based algorithms to solve the problems that traditional algorithms are difficult to solve.

According to the above analysis, we can see that there are still many challenges in solving the IM problem in large-scale social networks by using existing ML-based approaches. Up to now, we can only utilize a simple DRL framework to deal with the vanilla IM problem under the IC model in medium-sized social networks, which has great limitations. It is mainly reflected in the fact that it cannot adapt to diversified diffusion models in real applications, the training efficiency based on GNNs is not acceptable, the calculation of reward functions cannot overcome the #P-hardness of computing the influence spread, and cannot adjust to more complicated variant problems. Because of this, it also brings new research opportunities and vitality to this field. To deal with these challenges, we need to make efforts from the basic theory of ML and the modeling of social network applications, and combine them effectively.

9.3 Summary and Future Directions

To deal with the IM problem and its variants by using ML-based methods, we need to make efforts in two aspects. One is the basic research of solving CO problems by ML, then it tries to design new frameworks or improve the existing ML-based frameworks to enhance scalability, adaptivity, generalizability, efficiency, and automation. The GNN-based RL will be the mainstream framework in the future. Compared with supervised learning, it does not need a lot of datasets. It is very difficult to produce a large number of datasets in CO problems. In addition to relying on traditional RL, one of the key factors affecting algorithm performance when dealing with graph problems is graph representation, which largely determines scalability and generalizability. Here, we should try to combine with the latest deep learning technologies, such as attention mechanism and Transformer. The combination of Transformers and RL is a promising trend. One potential direction is to explore how to combine RL and (self-)supervised learning in Transformer learning [25, 248]. Another direction is to investigate how to apply Transformers in more complex and challenging RL scenarios, such as multi-agent RL and hierarchical RL [218]. Finally, we should emphasize that adopting ML-based methods does not mean completely giving up traditional methods, where some ideas from approximation or heuristic algorithms are still worth learning. For an instance, using some simple heuristic rules to prune the search space will make

the subsequent operation more efficient. How to fully combine the two kinds of strategies and complement each other's advantages is also an important development direction in the future.

The other is how to apply the existing ML-based CO algorithms to solve the IM problem and its variants. The IM problem in social networks has its unique characteristics, which leads to the general solution framework for CO problems based on ML not fully adapting to it. It is mainly embodied in the following three points. First, social networks are very large. This is very demanding of scalability and efficiency. The quality of the solutions output by the current ML-based CO algorithms on large-scale networks is poor, and the training time of the scheme relying on the GNN model is too long, which leads to its applications limiting the network scale. Second, social applications at present require a high generalization ability of algorithms. This is due to the dynamic change of network topology caused by the number of users and the dynamic change of friendships among users. Third, computing the influence spread is not a valuable oracle. This is because computing the spread is $\#P$ -hard under classic diffusion models, which brings great difficulties to the training of ML-based algorithms, and it cannot completely get rid of dependence on traditional search algorithms. Fourth, variant IM problems totally inherit the above characteristics, and the objective functions and constraints are more complex. This is very demanding of adaptivity and automation. One possible direction is to combine **BERT (Bidirectional Encoder Representations from Transformers)** [105] and DRL to learn the problem's internal structure and generalize to similar or larger-scale problems, where general optimization problems can be treated as data points under an identified data distribution and the transformer encoder of BERT can be improved to embed the CO graph effectively [212]. There is also potential for using Transformers to bridge offline and online RL for IM [227], furthermore, training online Decision Transformer from scratch [37, 125], meeting challenges from the complex and dynamic environments in real-world IM applications.

Therefore, we can seek plenty of innovation and research opportunities based on the above two major aspects. Here, we can not only study the basic theory of solving CO problems by adopting ML-based approaches, but also study the transfer of existing theories to the IM problem and its variants. Our goal is to construct an ML-based solution framework for the IM problem to meet the practical application requirements, including scalability, adaptivity, generalizability, efficiency, and automation. Finally, they jointly promote the progress of theory and technology in this field.

10 CONCLUSION

In this article, we present a comprehensive survey on the IM problem from an ML-based CO perspective. In order to compare ML-based algorithms with traditional approaches, we first summarize three kinds of traditional approaches: simulation-, proxy-, and sampling-based approaches, and point out their respective defects. Then, we survey two kinds of commonly used framework that can be used to solve CO problems: Ptr-Nets Based RL framework and GNNs Based RL framework. We discuss their basic structures and characteristics in detail, and focus on summarizing the principles and technologies related to the IM problem. This paves the way for solving the IM problem by using ML-based methods. Furthermore, we cover almost all the ML-based IM algorithms up to now, focusing on several pioneering DRL solution frameworks, and summarizing the variant IM problems and their corresponding ML-based algorithms. Here, we elaborately explain their working principle and how to break through the bottleneck of traditional algorithms. Finally, we summarize the existing difficulties and future research directions in this field from two aspects.

In summary, ML, as a representative of emerging technology, is a hot topic at present. Its application in solving algorithm problems in social networks has broad application prospects and commercial value, and it is expected to solve the dilemma faced by traditional algorithms. Our survey will give relevant researchers a new perspective to look at the IM problem and can be used

as a learning material to understand the latest ML-based development and research trends in the field of social network and algorithm design, thus laying a good start for the research in this field.

REFERENCES

- [1] Kenshin Abe, Zijian Xu, Issei Sato, and Masashi Sugiyama. 2019. Solving NP-hard problems on graphs with extended AlphaGo Zero. arXiv:1905.11623. Retrieved from <https://arxiv.org/abs/1905.11623>.
- [2] Zahra Aghaee, Mohammad Mahdi Ghasemi, Hamid Ahmadi Beni, Asgarali Bouyer, and Afsaneh Fatemi. 2021. A survey on meta-heuristic algorithms for the influence maximization problem in the social networks. *Computing* 103, 11 (2021), 2437–2477.
- [3] Khurshed Ali, Chih-Yu Wang, and Yi-Shin Chen. 2018. Boosting reinforcement learning in competitive influence maximization with transfer learning. In *Proceedings of the 2018 IEEE/WIC/ACM International Conference on Web Intelligence*. IEEE, 395–400.
- [4] Khurshed Ali, Chih-Yu Wang, and Yi-Shin Chen. 2022. Leveraging transfer learning in reinforcement learning to tackle competitive influence maximization. *Knowledge and Information Systems* 64, 8 (2022), 1–32.
- [5] Khurshed Ali, Chih-Yu Wang, Mi-Yen Yeh, and Yi-Shin Chen. 2020. Addressing competitive influence maximization on unknown social network with deep reinforcement learning. In *Proceedings of the 2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. IEEE, 196–203.
- [6] Khurshed Ali, Chih-Yu Wang, Mi-Yen Yeh, Cheng-Te Li, and Yi-Shin Chen. 2021. NEDRL-CIM: Network embedding meets deep reinforcement learning to tackle competitive influence maximization on evolving social networks. In *Proceedings of the 2021 IEEE 8th International Conference on Data Science and Advanced Analytics*. IEEE, 1–9.
- [7] Paul Almasan, José Suárez-Varela, Krzysztof Rusek, Pere Barlet-Ros, and Albert Cabellos-Aparicio. 2022. Deep reinforcement learning meets graph neural networks: Exploring a routing optimization use case. *Computer Communications* 196, C (2022), 184–194.
- [8] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. 2017. Hindsight experience replay. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*.
- [9] Sinan Aral and Paramveer S. Dhillon. 2018. Social influence maximization under empirical influence models. *Nature Human Behaviour* 2, 6 (2018), 375–382.
- [10] Akhil Arora, Sainyam Galhotra, and Sayan Ranu. 2017. Debunking the myths of influence maximization: An in-depth benchmarking study. In *Proceedings of the 2017 ACM International Conference on Management of Data*. 651–666.
- [11] Cigdem Aslay, Nicola Barbieri, Francesco Bonchi, and Ricardo Baeza-Yates. 2014. Online topic-aware influence maximization queries. In *Proceedings of the International Conference on Extending Database Technology*. 295–306.
- [12] Mehdi Azaouzi, Wassim Mnasri, and Lotfi Ben Romdhane. 2021. New trends in influence maximization models. *Computer Science Review* 40, C (2021), 100393.
- [13] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations*.
- [14] Ashwin Bahulkar, Boleslaw K. Szymanski, N. Orkun Baycik, and Thomas C. Sharkey. 2018. Community detection with edge augmentation in criminal networks. In *Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. IEEE, 1168–1175.
- [15] Suman Banerjee, Mamata Jenamani, and Dilip Kumar Pratihar. 2020. A survey on influence maximization in a social network. *Knowledge and Information Systems* 62, 9 (2020), 3417–3455.
- [16] Albert-László Barabási. 2013. Network science. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 371, 1987 (2013), 20120375.
- [17] Thomas Barrett, William Clements, Jakob Foerster, and Alex Lvovsky. 2020. Exploratory combinatorial optimization with reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34, 3243–3250.
- [18] Irwan Bello, Hieu Pham, Quoc V. Le, Mohammad Norouzi, and Samy Bengio. 2017. Neural combinatorial optimization with reinforcement learning. In *Proceedings of the 5th International Conference on Learning Representations*.
- [19] Yoshua Bengio, Andrea Lodi, and Antoine Prouvost. 2021. Machine learning for combinatorial optimization: A methodological tour d'Horizon. *European Journal of Operational Research* 290, 2 (2021), 405–421.
- [20] Giulia Berlusconi, Francesco Calderoni, Nicola Parolini, Marco Verani, and Carlo Piccardi. 2016. Link prediction in criminal networks: A tool for criminal intelligence analysis. *PLoS One* 11, 4 (2016), e0154244.
- [21] Dimitri Bertsekas. 2012. Dynamic programming and optimal control: Volume I. Athena scientific.
- [22] Shishir Bharathi, David Kempe, and Mahyar Salek. 2007. Competitive influence maximization in social networks. In *Proceedings of the International Workshop on Web and Internet Economics*. Springer, 306–311.
- [23] Christian Borgs, Michael Brautbar, Jennifer Chayes, and Brendan Lucier. 2014. Maximizing social influence in nearly optimal time. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 946–957.

- [24] Simon Bourigault, Sylvain Lamprier, and Patrick Gallinari. 2016. Representation learning for information diffusion through social networks: An embedded cascade model. In *Proceedings of the 9th ACM International Conference on Web Search and Data Mining*. 573–582.
- [25] David Brandfonbrener, Alberto Bietti, Jacob Buckman, Romain Laroche, and Joan Bruna. 2022. When does return-conditioned supervised learning work for offline reinforcement learning? In *Proceedings of the Conference on Neural Information Processing Systems*.
- [26] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1993. Signature verification using a “siamese” time delay neural network. In *Proceedings of the 6th International Conference on Neural Information Processing Systems*.
- [27] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. 2014. Spectral networks and deep locally connected networks on graphs. In *Proceedings of the 2nd International Conference on Learning Representations*.
- [28] Ceren Budak, Divyakant Agrawal, and Amr El Abbadi. 2011. Limiting the spread of misinformation in social networks. In *Proceedings of the 20th International Conference on World Wide Web*. 665–674.
- [29] Ji Li Zhipeng Cai, Mingyuan Yan, and Yingshu Li. 2016. Using crowdsourced data in location-based social networks to explore influence maximization. In *Proceedings of the 35th Annual IEEE International Conference on Computer Communications*. IEEE, 1–9.
- [30] Taotao Cai, Jianxin Li, Ajmal Mian, Rong-Hua Li, Timos Sellis, and Jeffrey Xu Yu. 2020. Target-aware holistic influence maximization in spatial social networks. *IEEE Transactions on Knowledge and Data Engineering* 34, 4 (2020), 1993–2007.
- [31] Taotao Cai, Quan Z. Sheng, Xiangyu Song, Jian Yang, Wei Emma Zhang, Jia Wu, and Philip S. Yu. 2022. A survey on location-driven influence maximization. arXiv:2204.08005. Retrieved from <https://arxiv.org/abs/2204.08005>.
- [32] Quentin Cappart, Thierry Moisan, Louis-Martin Rousseau, Isabeau Prémont-Schwarz, and Andre A. Cire. 2021. Combining reinforcement learning and constraint programming for combinatorial optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35, 3677–3687.
- [33] Rich Caruana. 1997. Multitask learning. *Machine Learning* 28, 1 (1997), 41–75.
- [34] Bo-Lun Chen, Wen-Xin Jiang, Yi-Xin Chen, Ling Chen, Rui-Jie Wang, Shuai Han, Jian-Hong Lin, and Yi-Cheng Zhang. 2022. Influence blocking maximization on networks: Models, methods and applications. *Physics Reports* 976 (2022), 1–54.
- [35] Haipeng Chen, Wei Qiu, Han-Ching Ou, Bo An, and Milind Tambe. 2021. Contingency-aware influence maximization: A reinforcement learning approach. In *Proceedings of the 37th Conference on Uncertainty in Artificial Intelligence*. PMLR, 1535–1545.
- [36] Hsi-Wen Chen, De-Nian Yang, Wang-Chien Lee, Philip S. Yu, and Ming-Syan Chen. 2023. CMINet: A graph learning framework for content-aware multi-channel influence diffusion. In *Proceedings of the ACM Web Conference 2023*. 545–555.
- [37] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. 2021. Decision transformer: Reinforcement learning via sequence modeling. In *Proceedings of the 35th Conference on Neural Information Processing Systems*. 15084–15097.
- [38] Sikai Chen, Jiqian Dong, Paul Ha, Yujie Li, and Samuel Labi. 2021. Graph neural network and reinforcement learning for multi-agent cooperative control of connected autonomous vehicles. *Computer-Aided Civil and Infrastructure Engineering* 36, 7 (2021), 838–857.
- [39] Shuo Chen, Ju Fan, Guoliang Li, Jianhua Feng, Kian-lee Tan, and Jinhui Tang. 2015. Online topic-aware influence maximization. *Proceedings of the VLDB Endowment* 8, 6 (2015), 666–677.
- [40] Tiantian Chen, Jianxiong Guo, and Weili Wu. 2022. Graph representation learning for popularity prediction problem: A survey. *Discrete Mathematics, Algorithms and Applications* 14, 7 (2022), 2230003.
- [41] Tiantian Chen, Wenjing Liu, Qizhi Fang, Jianxiong Guo, and Ding-Zhu Du. 2019. Minimizing misinformation profit in social networks. *IEEE Transactions on Computational Social Systems* 6, 6 (2019), 1206–1218.
- [42] Tiantian Chen, Siwen Yan, Jianxiong Guo, and Weili Wu. 2023. ToupleGDD: A fine-designed solution of influence maximization by deep reinforcement learning. *IEEE Transactions on Computational Social Systems* (2023). DOI : <https://doi.org/10.1109/TCSS.2023.3272331>
- [43] Wei Chen. 2018. An issue in the martingale analysis of the influence maximization algorithm IMM. In *Proceedings of the International Conference on Computational Social Networks*. Springer, 286–297.
- [44] Wei Chen, Laks V. S. Lakshmanan, and Carlos Castillo. 2013. Information and influence propagation in social networks. *Synthesis Lectures on Data Management* 5, 4 (2013), 1–177.
- [45] Wei Chen, Chi Wang, and Yajun Wang. 2010. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1029–1038.

- [46] Wei Chen, Yajun Wang, and Siyu Yang. 2009. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 199–208.
- [47] Wei Chen, Yifei Yuan, and Li Zhang. 2010. Scalable influence maximization in social networks under the linear threshold model. In *Proceedings of the 2010 IEEE International Conference on Data Mining*. IEEE, 88–97.
- [48] Xuanhao Chen, Yan Zhao, Guanfeng Liu, Rui Sun, Xiaofang Zhou, and Kai Zheng. 2020. Efficient similarity-aware influence maximization in geo-social network. *IEEE Transactions on Knowledge and Data Engineering* 34, 10 (2020), 4767–4780.
- [49] Xueqin Chen, Fan Zhou, Kunpeng Zhang, Goce Trajcevski, Ting Zhong, and Fengli Zhang. 2019. Information diffusion prediction via recurrent cascades convolution. In *Proceedings of the 2019 IEEE 35th International Conference on Data Engineering*. IEEE, 770–781.
- [50] Suqi Cheng, Huawei Shen, Junming Huang, Wei Chen, and Xueqi Cheng. 2014. IMRank: Influence maximization via finding self-consistent ranking. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*. 475–484.
- [51] Suqi Cheng, Huawei Shen, Junming Huang, Guoqing Zhang, and Xueqi Cheng. 2013. StaticGreedy: Solving the scalability-accuracy dilemma in influence maximization. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*. 509–518.
- [52] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning*, December 2014.
- [53] Edith Cohen, Daniel Delling, Thomas Pajor, and Renato F. Werneck. 2014. Sketch-based influence maximization and computation: Scaling up with guarantees. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. 629–638.
- [54] Federico Corò, Gianlorenzo D'Angelo, and Yllka Velaj. 2019. Recommending links to maximize the influence in social networks. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. Vol. 4, AAAI Press/IJCAI, 2195–2201.
- [55] Hanjun Dai, Bo Dai, and Le Song. 2016. Discriminative embeddings of latent variable models for structured data. In *Proceedings of the International Conference on Machine Learning*. 2702–2711.
- [56] Mohammad Mehdi Daliri Khomami, Alireza Rezvani, Negin Bagherpour, and Mohammad Reza Meybodi. 2018. Minimum positive influence dominating set and its application in influence maximization: A learning automata approach. *Applied Intelligence* 48, 3 (2018), 570–593.
- [57] R. C. De Souza, Daniel R. Figueiredo, A. A. de A Rocha, and Artur Ziviani. 2020. Efficient network seeding under variable node cost and limited budget for social networks. *Information Sciences* 514, C (2020), 369–384.
- [58] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*.
- [59] Michel Deudon, Pierre Cournut, Alexandre Lacoste, Yossiri Adulyasak, and Louis-Martin Rousseau. 2018. Learning heuristics for the TSP by policy gradient. In *Proceedings of the International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research*. Springer, 170–181.
- [60] Pedro Domingos and Matt Richardson. 2001. Mining the network value of customers. In *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 57–66.
- [61] Yuxiao Dong, Nitesh V. Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 135–144.
- [62] Lidan Fan, Zaixin Lu, Weili Wu, Bhavani Thuraisingham, Huan Ma, and Yuanjun Bi. 2013. Least cost rumor blocking in social networks. In *Proceedings of the 2013 IEEE 33rd International Conference on Distributed Computing Systems*. IEEE, 540–549.
- [63] Meng Fang, Tianyi Zhou, Yali Du, Lei Han, and Zhengyou Zhang. 2019. Curriculum-guided hindsight experience replay. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*.
- [64] Shanshan Feng, Gao Cong, Arijit Khan, Xiucheng Li, Yong Liu, and Yeow Meng Chee. 2018. Inf2vec: Latent representation model for social influence embedding. In *Proceedings of the 2018 IEEE 34th International Conference on Data Engineering*. IEEE, 941–952.
- [65] Shanshan Feng, Kaiqi Zhao, Lanting Fang, Kaiyu Feng, Wei Wei, Xutao Li, and Ling Shao. 2022. H-Diffu: Hyperbolic representations for information diffusion prediction. *IEEE Transactions on Knowledge and Data Engineering* (2022). DOI: <https://doi.org/10.1109/TKDE.2022.3209067>
- [66] Marshall L. Fisher, George L. Nemhauser, and Laurence A. Wolsey. 1978. An analysis of approximations for maximizing submodular set functions. In *Polyhedral Combinatorics*. M. L. Balinski and A. J. Hoffman (Eds.), Springer, 73–87.

- [67] Rossano Gaeta. 2018. A model of information diffusion in interconnected online social networks. *ACM Transactions on the Web* 12, 2 (2018), 1–21.
- [68] Sainyam Galhotra, Akhil Arora, and Shourya Roy. 2016. Holistic influence maximization: Combining scalability and efficiency with opinion-aware models. In *Proceedings of the 2016 International Conference on Management of Data*. 743–758.
- [69] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press.
- [70] Amit Goyal, Francesco Bonchi, and Laks V. S. Lakshmanan. 2010. Learning influence probabilities in social networks. In *Proceedings of the 3rd ACM International Conference on Web Search and Data Mining*. 241–250.
- [71] Amit Goyal, Wei Lu, and Laks V. S. Lakshmanan. 2011. Celf++ optimizing the greedy algorithm for influence maximization in social networks. In *Proceedings of the 20th International Conference Companion on World Wide Web*. 47–48.
- [72] Amit Goyal, Wei Lu, and Laks V. S. Lakshmanan. 2011. SIMPATH: An efficient algorithm for influence maximization under the linear threshold model. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining*. IEEE, 211–220.
- [73] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 855–864.
- [74] Daniel Gruhl, Ramanathan Guha, David Liben-Nowell, and Andrew Tomkins. 2004. Information diffusion through blogspace. In *Proceedings of the 13th International Conference on World Wide Web*. 491–501.
- [75] Shenshen Gu and Yue Yang. 2020. A deep learning algorithm for the max-cut problem based on pointer network structure with supervised learning and reinforcement learning strategies. *Mathematics* 8, 2 (2020), 298.
- [76] Adrien Guille, Hakim Hacid, Cecile Favre, and Djamel A. Zighed. 2013. Information diffusion in online social networks: A survey. *ACM Sigmod Record* 42, 2 (2013), 17–28.
- [77] Udesch Gunarathna, Renata Borovica-Gajic, Shanika Karunasekara, and Egemen Tanin. 2022. Solving dynamic graph problems with multi-attention deep reinforcement learning. arXiv:2201.04895. Retrieved from <https://arxiv.org/abs/2201.04895>.
- [78] Jianxiong Guo, Tiantian Chen, and Weili Wu. 2020. A multi-feature diffusion model: Rumor blocking in social networks. *IEEE/ACM Transactions on Networking* 29, 1 (2020), 386–397.
- [79] Jianxiong Guo, Yi Li, and Weili Wu. 2019. Targeted protection maximization in social networks. *IEEE Transactions on Network Science and Engineering* 7, 3 (2019), 1645–1655.
- [80] Jianxiong Guo and Weili Wu. 2021. Adaptive influence maximization: If influential node unwilling to be the seed. *ACM Transactions on Knowledge Discovery from Data* 15, 5 (2021), 84:1–84:23.
- [81] Jianxiong Guo and Weili Wu. 2021. Continuous profit maximization: A study of unconstrained dr-submodular maximization. *IEEE Transactions on Computational Social Systems* 8, 3 (2021), 768–779.
- [82] Qintian Guo, Sibao Wang, Zhewei Wei, and Ming Chen. 2020. Influence maximization revisited: Efficient reverse reachable set generation with bound tightened. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. 2167–2181.
- [83] Qintian Guo, Sibao Wang, Zhewei Wei, Wenqing Lin, and Jing Tang. 2022. Influence maximization revisited: Efficient sampling with bound tightened. *ACM Transactions on Database Systems* 47, 3 (2022), 12:1–12:45.
- [84] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*.
- [85] Qiang He, Xingwei Wang, Yong Zhao, Bo Yi, Xijia Lu, Mingzhou Yang, and Min Huang. 2021. Reinforcement-learning-based competitive opinion maximization approach in signed social networks. *IEEE Transactions on Computational Social Systems* 9, 5 (2021), 1505–1514.
- [86] Qiang He, Dafeng Zhang, Xingwei Wang, Lianbo Ma, Yong Zhao, Fei Gao, and Min Huang. 2022. Graph convolutional network-based rumor blocking on social networks. *IEEE Transactions on Computational Social Systems* (2022). DOI: <https://doi.org/10.1109/TCSS.2022.3188701>
- [87] Zaobo He, Zhipeng Cai, and Xiaoming Wang. 2015. Modeling propagation dynamics and developing optimized countermeasures for rumor spreading in online social networks. In *Proceedings of the 2015 IEEE 35th International Conference on Distributed Computing Systems*. IEEE, 205–214.
- [88] Zaobo He, Zhipeng Cai, Jiguo Yu, Xiaoming Wang, Yunchuan Sun, and Yingshu Li. 2016. Cost-efficient strategies for restraining rumor spreading in mobile social networks. *IEEE Transactions on Vehicular Technology* 66, 3 (2016), 2789–2800.
- [89] John J. Hopfield and David W. Tank. 1985. “Neural” computation of decisions in optimization problems. *Biological Cybernetics* 52, 3 (1985), 141–152.
- [90] Huimin Huang, Zaiqiao Meng, and Shangsong Liang. 2020. Recurrent neural variational model for follower-based influence maximization. *Information Sciences* 528 (2020), 280–293.

- [91] Huimin Huang, Zaiqiao Meng, and Hong Shen. 2021. Competitive and complementary influence maximization in social network: A follower's perspective. *Knowledge-Based Systems* 213 (2021), 106600.
- [92] Keke Huang, Sibao Wang, Glenn Bevilacqua, Xiaokui Xiao, and Laks V. S. Lakshmanan. 2017. Revisiting the stop-and-stare algorithms for influence maximization. *Proceedings of the VLDB Endowment* 10, 9 (2017), 913–924.
- [93] David Ireland and Giovanni Montana. 2022. LeNSE: Learning to navigate subgraph embeddings for large-scale combinatorial optimisation. In *Proceedings of the International Conference on Machine Learning*. Vol. 162, 9622–9638.
- [94] Mohammad Raihanul Islam, Sathappan Muthiah, Bijaya Adhikari, B. Aditya Prakash, and Naren Ramakrishnan. 2018. DeepDiffuse: Predicting the 'who' and 'when' in cascades. In *Proceedings of the 2018 IEEE International Conference on Data Mining*. IEEE, 1055–1060.
- [95] Myriam Jaouadi and Lotfi Ben Romdhane. 2019. Influence maximization problem in social networks: An overview. In *Proceedings of the 2019 IEEE/ACS 16th International Conference on Computer Systems and Applications*. IEEE, 1–8.
- [96] Qingye Jiang, Guojie Song, Cong Gao, Yu Wang, Wenjun Si, and Kunqing Xie. 2011. Simulated annealing based influence maximization in social networks. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence*.
- [97] Ziwei Jin, Jiaxing Shang, Wancheng Ni, Liang Zhao, Dajiang Liu, Baohua Qiang, Wu Xie, and Geyong Min. 2022. IM2Vec: Representation learning-based preference maximization in geo-social networks. *Information Sciences* 604 (2022), 170–196.
- [98] Chaitanya K. Joshi, Thomas Laurent, and Xavier Bresson. 2019. An efficient graph convolutional network technique for the travelling salesman problem. arXiv:1906.01227. Retrieved from <https://arxiv.org/abs/1906.01227>.
- [99] Kyomin Jung, Wooram Heo, and Wei Chen. 2012. IRIE: Scalable and robust influence maximization in social networks. In *Proceedings of the 2012 IEEE 12th International Conference on Data Mining*. IEEE, 918–923.
- [100] Harshavardhan Kamarthi, Priyesh Vijayan, Bryan Wilder, Balaraman Ravindran, and Milind Tambe. 2020. Influence maximization in unknown social networks: Learning policies for effective graph sampling. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*. 575–583.
- [101] Venkatakrishna Rao K., Mahender Katukuri, and Maheswari Jagarapu. 2022. CIM: clique-based heuristic for finding influential nodes in multilayer networks. *Applied Intelligence* 52, 5 (2022), 5173–5184.
- [102] Mohammad Mehdi Keikha, Maseud Rahgozar, and Masoud Asadpour. 2018. Community aware random walk for network embedding. *Knowledge-Based Systems* 148 (2018), 47–54.
- [103] Mohammad Mehdi Keikha, Maseud Rahgozar, Masoud Asadpour, and Mohammad Faghih Abdollahi. 2020. Influence maximization across heterogeneous interconnected networks based on deep learning. *Expert Systems with Applications* 140 (2020), 112905.
- [104] David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 137–146.
- [105] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 4171–4186.
- [106] Moein Khajehnejad, Ahmad Asgharian Rezaei, Mahmoudreza Babaei, Jessica Hoffmann, Mahdi Jalili, and Adrian Weller. 2020. Adversarial graph embeddings for fair influence maximization over social networks. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*. 4306–4312.
- [107] Elias Khalil, Hanjun Dai, Yuyu Zhang, Bistra Dilkina, and Le Song. 2017. Learning combinatorial optimization algorithms over graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*.
- [108] Elias Boutros Khalil, Bistra Dilkina, and Le Song. 2014. Scalable diffusion-aware optimization of network topology. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1226–1235.
- [109] Jinha Kim, Seung-Keol Kim, and Hwanjo Yu. 2013. Scalable and parallelizable processing of influence maximization for large-scale social networks? In *Proceedings of the 2013 IEEE 29th International Conference on Data Engineering*. IEEE, 266–277.
- [110] Masahiro Kimura and Kazumi Saito. 2006. Tractable models for information diffusion in social networks. In *Proceedings of the European Conference on Principles of Data Mining and Knowledge Discovery*. Springer, 259–271.
- [111] Wouter Kool, Herke van Hoof, and Max Welling. 2019. Attention, learn to solve routing problems! In *Proceedings of the International Conference on Learning Representations*.
- [112] Sanjay Kumar, Abhishek Mallik, Anavi Khetarpal, and B. S. Panda. 2022. Influence maximization in social networks using graph embedding and graph neural network. *Information Sciences* 607 (2022), 1617–1636.
- [113] Sanjay Kumar, Abhishek Mallik, and B. S. Panda. 2023. Influence maximization in social networks using transfer learning via graph-based LSTM. *Expert Systems with Applications* 212 (2023), 118770.
- [114] Konstantin Kutzkov, Albert Bifet, Francesco Bonchi, and Aristides Gionis. 2013. STRIP: Stream learning of influence probabilities. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 275–283.

- [115] Eugene L. Lawler and David E. Wood. 1966. Branch-and-bound methods: A survey. *Operations Research* 14, 4 (1966), 699–719.
- [116] John Boaz Lee, Xiangnan Kong, Yihan Bao, and Constance Moore. 2017. Identifying deep contrasting networks from time series data: Application to brain network analysis. In *Proceedings of the 2017 SIAM International Conference on Data Mining*. SIAM, 543–551.
- [117] Siyu Lei, Silviu Maniu, Luyi Mo, Reynold Cheng, and Pierre Senellart. 2015. Online influence maximization. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 645–654.
- [118] Henrique Lemos, Marcelo Prates, Pedro Avelar, and Luis Lamb. 2019. Graph colouring meets deep learning: Effective graph neural network models for combinatorial problems. In *Proceedings of the 2019 IEEE 31st International Conference on Tools with Artificial Intelligence*. IEEE, 879–885.
- [119] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance. 2007. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 420–429.
- [120] Cheng Li, Jiaqi Ma, Xiaoxiao Guo, and Qiaozhu Mei. 2017. DeepCas: An end-to-end predictor of information cascades. In *Proceedings of the 26th International Conference on World Wide Web*. 577–586.
- [121] Dexun Li, Meghna Lowalekar, and Pradeep Varakantham. 2021. CLAIM: Curriculum learning policy for influence maximization in unknown social networks. In *Proceedings of the 37th Conference on Uncertainty in Artificial Intelligence*. PMLR, 1455–1465.
- [122] Guoliang Li, Shuo Chen, Jianhua Feng, Kian-lee Tan, and Wen-syan Li. 2014. Efficient location-aware influence maximization. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*. 87–98.
- [123] Hui Li, Mengting Xu, Sourav S. Bhowmick, Joty Shafiq Rayhan, Changsheng Sun, and Jiangtao Cui. 2022. PIANO: Influence maximization meets deep reinforcement learning. *IEEE Transactions on Computational Social Systems* 10, 3 (2022), 1288–1300.
- [124] Hui Li, Mengting Xu, Sourav S. Bhowmick, Changsheng Sun, Zhongyuan Jiang, and Jiangtao Cui. 2019. DISCO: Influence maximization meets network embedding and deep learning. arXiv:1906.07378. Retrieved from <https://arxiv.org/abs/1906.07378>.
- [125] Wenzhe Li, Hao Luo, Zichuan Lin, Chongjie Zhang, Zongqing Lu, and Deheng Ye. 2023. A survey on transformers in reinforcement learning. arXiv:2301.03044. Retrieved from <https://arxiv.org/abs/2301.03044>.
- [126] Xin Li, Liang Yang, Yi Li, and Chuan Zhou. 2019. Deep trajectory: A deep learning approach for mobile advertising in vehicular networks. *Neural Computing and Applications* 31, 7 (2019), 2813–2825.
- [127] Yuchen Li, Ju Fan, Yanhao Wang, and Kian-Lee Tan. 2018. Influence maximization on social graphs: A survey. *IEEE Transactions on Knowledge and Data Engineering* 30, 10 (2018), 1852–1872.
- [128] Ying Li, Linlin Li, Yijun Liu, and Qianqian Li. 2022. MAHE-IM: Multiple aggregation of heterogeneous relation embedding for influence maximization on heterogeneous information network. *Expert Systems with Applications* 202 (2022), 117289.
- [129] Yuchen Li, Dongxiang Zhang, and Kian-Lee Tan. 2015. Real-time targeted influence maximization for online advertisements. *Proceedings of the VLDB Endowment* 8, 10 (2015), 1070–1081.
- [130] Zhuwen Li, Qifeng Chen, and Vladlen Koltun. 2018. Combinatorial optimization with graph convolutional networks and guided tree search. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*.
- [131] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2016. Continuous control with deep reinforcement learning. In *Proceedings of the 4th International Conference on Learning Representations*. 1–14.
- [132] Su-Chen Lin, Shou-De Lin, and Ming-Syan Chen. 2015. A learning-based framework to handle multi-round multi-party influence maximization on social networks. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 695–704.
- [133] Chen Ling, Junji Jiang, Junxiang Wang, My Thai, Lukas Xue, James Song, Meikang Qiu, and Liang Zhao. 2023. Deep graph representation learning and optimization for influence maximization. In *International Conference on Machine Learning (ICML'23)*.
- [134] Jing Liu, Yudi Chen, Duanshun Li, Noseong Park, Kisung Lee, and Dongwon Lee. 2019. Predicting influence probabilities using graph convolutional networks. In *Proceedings of the 2019 IEEE International Conference on Big Data*. IEEE, 860–869.
- [135] Qi Liu, Biao Xiang, Enhong Chen, Hui Xiong, Fangshuang Tang, and Jeffrey Xu Yu. 2014. Influence maximization over large-scale social networks: A bounded linear approach. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. 171–180.
- [136] Yuening Liu, Liqing Qiu, and Chengai Sun. 2022. An influence maximization algorithm based on low-dimensional representation learning. *Applied Intelligence* 52, 14 (2022), 1–18.

- [137] Yiming Liu, Waichau Sze, Xiaofeng Gao, and Guihai Chen. 2021. Multiple agents reinforcement learning based influence maximization in social network services. In *Proceedings of the International Conference on Service-Oriented Computing*. Springer, 431–445.
- [138] Jiaqi Ma, Junwei Deng, and Qiaozhu Mei. 2022. Adversarial attack on graph neural networks as an influence maximization problem. In *Proceedings of the 15th ACM International Conference on Web Search and Data Mining*. 675–685.
- [139] Lijia Ma, Zengyang Shao, Xiaocong Li, Qiuzhen Lin, Jianqiang Li, Victor C. M. Leung, and Asoke K. Nandi. 2022. Influence maximization in complex networks by using evolutionary deep reinforcement learning. *IEEE Transactions on Emerging Topics in Computational Intelligence* (2022). DOI: <https://doi.org/10.1109/TETCI.2021.3136643>
- [140] Qiang Ma, Suwen Ge, Danyang He, Darshan Thaker, and Iddo Drori. 2019. Combinatorial optimization by graph pointer networks and hierarchical reinforcement learning. arXiv:1911.04936. Retrieved from <https://arxiv.org/abs/1911.04936>.
- [141] Sahil Manchanda, Akash Mittal, Anuj Dhawan, Sourav Medya, Sayan Ranu, and Ambuj Singh. 2020. GCOMB: Learning budget-constrained combinatorial algorithms over billion-sized graphs. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*. 20000–20011.
- [142] Hongzi Mao, Malte Schwarzkopf, Shaileshh Bojja Venkatakrishnan, Zili Meng, and Mohammad Alizadeh. 2019. Learning scheduling algorithms for data processing clusters. In *Proceedings of the ACM Special Interest Group on Data Communication*.
- [143] Michael Mathioudakis, Francesco Bonchi, Carlos Castillo, Aristides Gionis, and Antti Ukkonen. 2011. Sparsification of influence networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 529–537.
- [144] Eli Meir, Hagai Maron, Shie Mannor, and Gal Chechik. 2021. Controlling graph dynamics with reinforcement learning and graph neural networks. In *Proceedings of the International Conference on Machine Learning*. PMLR, 7565–7577.
- [145] Matheus R. F. Mendonça, André Barreto, and Artur Ziviani. 2022. Efficient information diffusion in time-varying graphs through deep reinforcement learning. *World Wide Web* 25, 6 (2022), 1–26.
- [146] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of the 1st International Conference on Learning Representations*.
- [147] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*.
- [148] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fiedjeland, Georg Ostrovski, Petersen Stig, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharmashan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (2015), 529–533.
- [149] Kevin P. Murphy. 2022. *Probabilistic Machine Learning: An Introduction*. MIT Press.
- [150] Ashvin V. Nair, Vitchyr Pong, Murtaza Dalal, Shikhar Bahl, Steven Lin, and Sergey Levine. 2018. Visual reinforcement learning with imagined goals. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*.
- [151] Ramasuri Narayanam and Yadati Narahari. 2010. A shapley value-based approach to discover influential nodes in social networks. *IEEE Transactions on Automation Science and Engineering* 8, 1 (2010), 130–147.
- [152] Mohammadreza Nazari, Afshin Oroojlooy, Lawrence Snyder, and Martin Takáč. 2018. Reinforcement learning for solving the vehicle routing problem. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*.
- [153] George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. 1978. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming* 14, 1 (1978), 265–294.
- [154] Dung T. Nguyen, Huiyuan Zhang, Soham Das, My T. Thai, and Thang N. Dinh. 2013. Least cost influence in multiplex social networks: Model representation and analysis. In *Proceedings of the 2013 IEEE 13th International Conference on Data Mining*. IEEE, 567–576.
- [155] Hung T. Nguyen, Thang N. Dinh, and My T. Thai. 2016. Cost-aware targeted viral marketing in billion-scale networks. In *Proceedings of the 35th Annual IEEE International Conference on Computer Communications*. IEEE, 1–9.
- [156] Hung T. Nguyen, Thang N. Dinh, and My T. Thai. 2018. Revisiting of the stop-and-stare algorithms for influence maximization. In *Proceedings of the International Conference on Computational Social Networks*. Springer, 273–285.
- [157] Hung T. Nguyen, My T. Thai, and Thang N. Dinh. 2016. Stop-and-stare: Optimal sampling algorithms for viral marketing in billion-scale networks. In *Proceedings of the 2016 International Conference on Management of Data*. 695–710.

- [158] Runbo Ni, Xueyan Li, Fangqi Li, Xiaofeng Gao, and Guihai Chen. 2021. FastCover: An unsupervised learning framework for multi-hop influence maximization in social networks. arXiv:2111.00463. Retrieved from <https://arxiv.org/abs/2111.00463>.
- [159] Sunil Nishad, Shubhangi Agarwal, Arnab Bhattacharya, and Sayan Ranu. 2020. GraphReach: Position-aware graph neural network using reachability estimations. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence*. 1527–1533.
- [160] Naoto Ohsaka, Takuya Akiba, Yuichi Yoshida, and Ken-ichi Kawarabayashi. 2014. Fast and accurate influence maximization on large networks with pruned Monte-Carlo simulations. In *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 28.
- [161] Naoto Ohsaka, Takuya Akiba, Yuichi Yoshida, and Ken-ichi Kawarabayashi. 2016. Dynamic influence analysis in evolving networks. *Proceedings of the VLDB Endowment* 9, 12 (2016), 1077–1088.
- [162] Han-Ching Ou, Haipeng Chen, Shahin Jabbari, and Milind Tambe. 2021. Active screening for recurrent diseases: A reinforcement learning approach. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*. 992–1000.
- [163] George Panagopoulos, Fragkiskos Malliaros, and Michalis Vazirgiannis. 2020. Multi-task learning for influence estimation and maximization. *IEEE Transactions on Knowledge and Data Engineering* 34, 9 (2020), 4398–4409.
- [164] George Panagopoulos, Fragkiskos D. Malliaros, and Michalis Vazirgianis. 2020. Influence maximization using influence and susceptibility embeddings. In *Proceedings of the International AAAI Conference on Web and Social Media*. Vol. 14, 511–521.
- [165] Sancheng Peng, Yongmei Zhou, Lihong Cao, Shui Yu, Jianwei Niu, and Weijia Jia. 2018. Influence analysis in social networks: A survey. *Journal of Network and Computer Applications* 106 (2018), 17–32.
- [166] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. DeepWalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 701–710.
- [167] Marcelo Prates, Pedro H. C. Avelar, Henrique Lemos, Luis C. Lamb, and Moshe Y. Vardi. 2019. Learning to solve NP-complete problems: A graph neural network for decision TSP. In *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33, 4731–4738.
- [168] Jiezhong Qiu, Jian Tang, Hao Ma, Yuxiao Dong, Kuansan Wang, and Jie Tang. 2018. DeepInf: Social influence prediction with deep learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2110–2119.
- [169] Wei Qiu, Haipeng Chen, and Bo An. 2019. Dynamic electronic toll collection via multi-agent deep reinforcement learning with edge-based graph convolutional networks. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 4568–4574.
- [170] Leonardo F. R. Ribeiro, Pedro H. P. Saverese, and Daniel R. Figueiredo. 2017. struc2vec: Learning node representations from structural identity. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 385–394.
- [171] Matthew Richardson and Pedro Domingos. 2002. Mining knowledge-sharing sites for viral marketing. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 61–70.
- [172] Kazumi Saito, Ryohei Nakano, and Masahiro Kimura. 2008. Prediction of information diffusion probabilities for independent cascade model. In *Proceedings of the 12th International Conference on Knowledge-Based Intelligent Information and Engineering Systems, Part III*. Springer, 67–75.
- [173] Daniel Selsam, Matthew Lamm, BÄijnz Benedikt, Percy Liang, Leonardo de Moura, and David L. Dill. 2019. Learning a SAT Solver from Single-Bit Supervision. In *7th International Conference on Learning Representations (ICLR'19)*. New Orleans, LA.
- [174] Xiao Shen, Fu-lai Chung, and Sitong Mao. 2017. Leveraging cross-network information for graph sparsification in influence maximization. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 801–804.
- [175] Xiao Shen, Sitong Mao, and Fu-lai Chung. 2019. Cross-network learning with fuzzy labels for seed selection and graph sparsification in influence maximization. *IEEE Transactions on Fuzzy Systems* 28, 9 (2019), 2195–2208.
- [176] Wei Sheng, Wenbo Song, Dong Li, Fei Yang, and Yatao Zhang. 2022. Dynamic influence maximization via network representation learning. *Frontiers in Physics* 9 (2022), 839.
- [177] Tuo Shi, Siyao Cheng, Zhipeng Cai, Yingshu Li, and Jianzhong Li. 2016. Retrieving the maximal time-bounded positive influence set from social networks. *Personal and Ubiquitous Computing* 20, 5 (2016), 717–730.
- [178] Shashank Sheshar Singh, Ajay Kumar, Kuldeep Singh, and Bhaskar Biswas. 2019. LAPSO-IM: A learning-based influence maximization approach for social networks. *Applied Soft Computing* 82 (2019), 105554.
- [179] Shashank Sheshar Singh, Divya Srivastva, Madhushi Verma, and Jagendra Singh. 2022. Influence maximization frameworks, performance, challenges and directions on social network: A theoretical study. *Journal of King Saud University-Computer and Information Sciences* 34, 9 (2022), 7570–7603.

- [180] Kate A. Smith. 1999. Neural networks for combinatorial optimization: A review of more than a decade of research. *INFORMS Journal on Computing* 11, 1 (1999), 15–34.
- [181] Chonggang Song, Wynne Hsu, and Mong Li Lee. 2016. Targeted influence maximization in social networks. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management*. 1683–1692.
- [182] N. Sumith, B. Annappa, and Swapna Bhattacharya. 2018. Influence maximization in large social networks: Heuristics, models and parameters. *Future Generation Computer Systems* 89 (2018), 777–790.
- [183] Jimeng Sun and Jie Tang. 2011. A survey of models and algorithms for social influence analysis. In *Social Network Data Analytics*. C. Aggarwal (Ed.), Springer, 177–214.
- [184] Penghao Sun, Julong Lan, Junfei Li, Zehua Guo, and Yuxiang Hu. 2020. Combining deep reinforcement learning with graph neural networks for optimal VNF placement. *IEEE Communications Letters* 25, 1 (2020), 176–180.
- [185] Yizhou Sun and Jiawei Han. 2012. Mining heterogeneous information networks: Principles and methodologies. *Synthesis Lectures on Data Mining and Knowledge Discovery* 3, 2 (2012), 1–159.
- [186] Yiwei Sun, Suhang Wang, Xianfeng Tang, Tsung-Yu Hsieh, and Vasant Honavar. 2020. Adversarial attacks on graph neural networks via node injections: A hierarchical reinforcement learning approach. In *Proceedings of the Web Conference 2020*. 673–683.
- [187] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*.
- [188] Kamal Taha. 2022. Identifying the Top-k influential spreaders in social networks: A survey and experimental evaluation. *IEEE Access* 10, (2022), 107809–107845.
- [189] Jing Tang, Xueyan Tang, Xiaokui Xiao, and Junsong Yuan. 2018. Online processing algorithms for influence maximization. In *Proceedings of the 2018 International Conference on Management of Data*. 991–1005.
- [190] Wenyi Tang, Guangchun Luo, Yubao Wu, Ling Tian, Xu Zheng, and Zhipeng Cai. 2019. A second-order diffusion model for influence maximization in social networks. *IEEE Transactions on Computational Social Systems* 6, 4 (2019), 702–714.
- [191] Youze Tang, Yanchen Shi, and Xiaokui Xiao. 2015. Influence maximization in near-linear time: A martingale approach. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. 1539–1554.
- [192] Youze Tang, Xiaokui Xiao, and Yanchen Shi. 2014. Influence maximization: Near-optimal time complexity meets practical efficiency. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*. 75–86.
- [193] V. Tejaswi, P. V. Bindu, and P. Santhi Thilagam. 2016. Diffusion models and approaches for influence maximization in social networks. In *Proceedings of the 2016 International Conference on Advances in Computing, Communications and Informatics*. IEEE, 1345–1351.
- [194] Shan Tian, Songsong Mo, Liwei Wang, and Zhiyong Peng. 2020. Deep reinforcement learning-based approach to tackle topic-aware influence maximization. *Data Science and Engineering* 5, 1 (2020), 1–11.
- [195] Shan Tian, Ping Zhang, Songsong Mo, Liwei Wang, and Zhiyong Peng. 2019. A learning approach for topic-aware influence maximization. In *Proceedings of the Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data*. Springer, 125–140.
- [196] Guangmo Tong. 2020. StratLearner: Learning a strategy for misinformation prevention in social networks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*. 15546–15555.
- [197] Guangmo Tong. 2021. USCO-Solver: Solving undetermined stochastic combinatorial optimization problems. In *Proceedings of the 35th Conference on Neural Information Processing Systems*. 1646–1659.
- [198] Guangmo Tong. 2022. Social-inverse: Inverse decision-making of social contagion management with task migrations. In *Proceedings of the 36th Conference on Neural Information Processing Systems*.
- [199] Guangmo Tong, Ruiqi Wang, Zheng Dong, and Xiang Li. 2020. Time-constrained adaptive influence maximization. *IEEE Transactions on Computational Social Systems* 8, 1 (2020), 33–44.
- [200] Guangmo Tong, Weiwei Wu, Shaojie Tang, and Ding-Zhu Du. 2016. Adaptive influence maximization in dynamic social networks. *IEEE/ACM Transactions on Networking* 25, 1 (2016), 112–125.
- [201] Cong Tran, Won-Yong Shin, and Andreas Spitz. 2021. IM-META: Influence maximization using node metadata in networks with unknown topology. arXiv:2106.02926. Retrieved from <https://arxiv.org/abs/2106.02926>.
- [202] Hado Van Hasselt, Arthur Guez, and David Silver. 2016. Deep reinforcement learning with double Q-learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 30.
- [203] Peter J. M. Van Laarhoven, Emile H. L. Aarts, and Jan Karel Lenstra. 1992. Job shop scheduling by simulated annealing. *Operations Research* 40, 1 (1992), 113–125.
- [204] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*.

- [205] Sharan Vaswani, Branislav Kveton, Zheng Wen, Mohammad Ghavamzadeh, Laks V. S. Lakshmanan, and Mark Schmidt. 2017. Model-independent online learning for influence maximization. In *Proceedings of the International Conference on Machine Learning*. PMLR, 3530–3539.
- [206] Petar Velićković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *Proceedings of the International Conference on Learning Representations*.
- [207] Pradnya A. Vikhar. 2016. Evolutionary algorithms: A critical review and its future prospects. In *Proceedings of the 2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication*. IEEE, 261–265.
- [208] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*.
- [209] Chao Wang, Yiming Liu, Xiaofeng Gao, and Guihai Chen. 2021. A reinforcement learning model for influence maximization in social networks. In *Proceedings of the International Conference on Database Systems for Advanced Applications*. Springer, 701–709.
- [210] Daixin Wang, Peng Cui, and Wenwu Zhu. 2016. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1225–1234.
- [211] Jia Wang, Vincent W. Zheng, Zemin Liu, and Kevin Chen-Chuan Chang. 2017. Topological recurrent neural network for diffusion prediction. In *Proceedings of the 2017 IEEE International Conference on Data Mining*. IEEE, 475–484.
- [212] Qi Wang, Kenneth H. Lai, and Chunlei Tang. 2023. Solving combinatorial optimization problems over graphs with BERT-Based Deep Reinforcement Learning. *Information Sciences* 619 (2023), 930–946.
- [213] Xiaoyang Wang, Ying Zhang, Wenjie Zhang, and Xuemin Lin. 2016. Distance-aware influence maximization in geo-social network. In *Proceedings of the 2016 IEEE 32nd International Conference on Data Engineering*. 1–12.
- [214] Xiaoyang Wang, Ying Zhang, Wenjie Zhang, Xuemin Lin, and Chen Chen. 2016. Bring order into the samples: A novel scalable method for influence maximization. *IEEE Transactions on Knowledge and Data Engineering* 29, 2 (2016), 243–256.
- [215] Yu Wang, Gao Cong, Guojie Song, and Kunqing Xie. 2010. Community-based greedy algorithm for mining top-k influential nodes in mobile social networks. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1039–1048.
- [216] Yifan Wang and Guangmo Tong. 2022. Learnability of competitive threshold models. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence*. 3985–3991.
- [217] Max Welling and Thomas N. Kipf. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of the 6th International Conference on Learning Representations*.
- [218] Muning Wen, Jakub Kuba, Runji Lin, Weinan Zhang, Ying Wen, Jun Wang, and Yaodong Yang. 2022. Multi-agent reinforcement learning is a sequence modeling problem. In *Proceedings of the 36th Conference on Neural Information Processing Systems*. 16509–16521.
- [219] Bryan Wilder, Eric Ewing, Bistra Dilkina, and Milind Tambe. 2019. End to end learning and optimization on graphs. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*.
- [220] Bryan Wilder, Laura Onasch-Vera, Juliana Hudson, Jose Luna, Nicole Wilson, Robin Petering, Darlene Woo, Milind Tambe, and Eric Rice. 2018. End-to-end influence maximization in the field. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. Vol. 18, 1414–1422.
- [221] Le Wu, Junwei Li, Peijie Sun, Richang Hong, Yong Ge, and Meng Wang. 2020. DiffNet++: A neural influence and interest diffusion network for social recommendation. *IEEE Transactions on Knowledge and Data Engineering* 34, 10 (2020), 4753–4766.
- [222] Le Wu, Peijie Sun, Yanjie Fu, Richang Hong, Xiting Wang, and Meng Wang. 2019. A neural influence diffusion model for social recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 235–244.
- [223] Shiqing Wu, Weihua Li, and Quan Bai. 2023. GAC: A deep reinforcement learning model toward user incentivization in unknown social networks. *Knowledge-Based Systems* 259 (2023), 110060.
- [224] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S. Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems* 32, 1 (2020), 4–24.
- [225] Luodi Xie, Huimin Huang, and Qing Du. 2022. A hierarchical generative embedding model for influence maximization in attributed social networks. *Applied Sciences* 12, 3 (2022), 1321.
- [226] Yanghao Xie, Lin Huang, Yuyang Kong, Sheng Wang, Shizhong Xu, Xiong Wang, and Jing Ren. 2021. Virtualized network function forwarding graph placing in SDN and NFV-enabled IoT networks: A graph neural network assisted deep reinforcement learning method. *IEEE Transactions on Network and Service Management* 19, 1 (2021), 524–537.
- [227] Zhihui Xie, Zichuan Lin, Junyou Li, Shuai Li, and Deheng Ye. 2022. Pretraining in deep reinforcement learning: A survey. arXiv:2211.03959. Retrieved from <https://arxiv.org/abs/2211.03959>.

- [228] Chenghao Xu and Wei Song. 2022. An adaptive data uploading scheme for mobile crowdsensing via deep reinforcement learning with graph neural network. *IEEE Internet of Things Journal* 9, 18 (2022), 18064–18078.
- [229] Amulya Yadav, Hau Chan, Albert Xin Jiang, Haifeng Xu, Eric Rice, and Milind Tambe. 2016. Using social networks to aid homeless shelters: Dynamic influence maximization under uncertainty. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*. Vol. 16, 740–748.
- [230] Amulya Yadav, Ritesh Noothigattu, Eric Rice, Laura Onasch-Vera, Leandro Soriano Marcolino, and Milind Tambe. 2018. Please be an influencer?: Contingency-aware influence maximization. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*.
- [231] Tzu-Hsin Yang, Hao-Shang Ma, and Jen-Wei Huang. 2018. DNA: General deterministic network adaptive framework for multi-round multi-party influence maximization. In *Proceedings of the 2018 IEEE 5th International Conference on Data Science and Advanced Analytics*. IEEE, 273–282.
- [232] Mao Ye, Xingjie Liu, and Wang-Chien Lee. 2012. Exploring social influence for recommendation: A generative model approach. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 671–680.
- [233] Tolga Yilmaz and Özgür Ulusoy. 2022. Misinformation propagation in online social networks: Game theoretic and Reinforcement Learning Approaches. *IEEE Transactions on Computational Social Systems* (2022). DOI : <https://doi.org/10.1109/TCSS.2022.3208793>
- [234] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. 2018. Hierarchical graph representation learning with differentiable pooling. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*.
- [235] Bin Yu, Yu Zhang, Yu Xie, Chen Zhang, and Ke Pan. 2021. Influence-aware graph neural networks. *Applied Soft Computing* 104 (2021), 107169.
- [236] Junliang Yu, Hongzhi Yin, Jundong Li, Qinyong Wang, Nguyen Quoc Viet Hung, and Xiangliang Zhang. 2021. Self-supervised multi-channel hypergraph convolutional network for social recommendation. In *Proceedings of the Web Conference 2021*. 413–424.
- [237] Chunyuan Yuan, Jiacheng Li, Wei Zhou, Yijun Lu, Xiaodan Zhang, and Songlin Hu. 2021. DyHGCN: A dynamic heterogeneous graph convolutional network to learn users' dynamic preferences for information diffusion prediction. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases, Part III*. Springer, 347–363.
- [238] Zhongju Yuan, Genghui Li, Zhenkun Wang, Jianyong Sun, and Ran Cheng. 2022. RL-CSL: A combinatorial optimization method using reinforcement learning and contrastive self-supervised learning. *IEEE Transactions on Emerging Topics in Computational Intelligence* (2022). DOI : <https://doi.org/10.1109/TETCI.2021.3139802>
- [239] Savvas Zannettou, Michael Sirivianos, Jeremy Blackburn, and Nicolas Kourtellis. 2019. The web of false information: Rumors, fake news, hoaxes, clickbait, and various other shenanigans. *Journal of Data and Information Quality* 11, 3 (2019), 1–37.
- [240] Ahmad Zareie and Rizos Sakellariou. 2023. Influence maximization in social networks: A survey of behaviour-aware methods. *Social Network Analysis and Mining* 13, 1 (2023), 78.
- [241] Qianyi Zhan, Jiawei Zhang, S. Yu Philip, Sherry Emery, and Junyuan Xie. 2016. Discover tipping users for cross network influencing. In *Proceedings of the 2016 IEEE 17th International Conference on Information Reuse and Integration*. IEEE, 67–76.
- [242] Cai Zhang, Weimin Li, Dingmei Wei, Yanxia Liu, and Zheng Li. 2022. Network dynamic GCN influence maximization algorithm with leader fake labeling mechanism. *IEEE Transactions on Computational Social Systems* (2022). DOI : <https://doi.org/10.1109/TCSS.2022.3193583>
- [243] Kaichen Zhang, Jingbo Zhou, Donglai Tao, Panagiotis Karras, Qing Li, and Hui Xiong. 2020. Geodemographic influence maximization. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2764–2774.
- [244] Wentao Zhang, Zhi Yang, Yexin Wang, Yu Shen, Yang Li, Liang Wang, and Bin Cui. 2021. GRAIN: Improving data efficiency of graph neural networks via diversified in fluence maximization. *Proceedings of the VLDB Endowment* 14, 11 (2021), 2473–2482.
- [245] Yan Zhang, Jonathon Hare, and Adam Prugel-Bennett. 2019. Deep set prediction networks. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*.
- [246] Ziwei Zhang, Peng Cui, and Wenwu Zhu. 2020. Deep learning on graphs: A survey. *IEEE Transactions on Knowledge and Data Engineering* 34, 1 (2020), 249–270.
- [247] Qihang Zhao, Yuzhe Zhang, and Xiaodong Feng. 2022. Predicting information diffusion via deep temporal convolutional networks. *Information Systems* 108 (2022), 102045.
- [248] Qinqing Zheng, Amy Zhang, and Aditya Grover. 2022. Online decision transformer. In *Proceedings of the International Conference on Machine Learning*. PMLR, 27042–27059.

- [249] Chuan Zhou, Peng Zhang, Wenyu Zang, and Li Guo. 2015. On the upper bounds of spread for greedy algorithms in social network influence maximization. *IEEE Transactions on Knowledge and Data Engineering* 27, 10 (2015), 2770–2783.
- [250] Fan Zhou and Chengtai Cao. 2021. Overcoming catastrophic forgetting in graph neural networks with experience replay. In *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35, 4714–4722.
- [251] Lekui Zhou, Yang Yang, Xiang Ren, Fei Wu, and Yueting Zhuang. 2018. Dynamic network embedding by modeling triadic closure process. In *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32.
- [252] Qi Zhu, Carl Yang, Yidan Xu, Haonan Wang, Chao Zhang, and Jiawei Han. 2021. Transfer learning of graph neural networks with ego-graph information maximization. In *Proceedings of the 35th Conference on Neural Information Processing Systems*. 1766–1779.

Received 6 November 2022; revised 23 March 2023; accepted 8 June 2023