

REM: A Scalable Reinforced Multi-Expert Framework for Multiplex Influence Maximization

Huyen Nguyen ^{*1}, Hieu Dam ^{*2}, Nguyen Hoang Khoi Do ³, Cong Tran ¹, Cuong Pham ¹

¹ Posts and Telecommunications Institute of Technology, Hanoi, Vietnam

² FPT University, Swinburne Vietnam Hanoi campus, Hanoi Vietnam

³ University of Florida, Gainesville, USA

huyenxam250400@gmail.com, hieudmswh00475@fpt.edu.vn, nguyen.do@ufl.edu, cuongpv@ptit.edu.vn, conggt@ptit.edu.vn

Abstract

In social online platforms, identifying influential seed users to maximize influence spread is a crucial as it can greatly diminish the cost and efforts required for information dissemination. While effective, traditional methods for Multiplex Influence Maximization (MIM) have reached their performance limits, prompting the emergence of learning-based approaches. These novel methods aim for better generalization and scalability for more sizable graphs but face significant challenges, such as (1) inability to handle unknown diffusion patterns and (2) reliance on high-quality training samples. To address these issues, we propose the Reinforced Expert Maximization framework (REM). REM leverages a Propagation Mixture of Experts technique to encode dynamic propagation of large multiplex networks effectively in order to generate enhanced influence propagation. Noticeably, REM treats a generative model as a policy to autonomously generate different seed sets and learn how to improve them from a Reinforcement Learning perspective. Extensive experiments on several real-world datasets demonstrate that REM surpasses state-of-the-art methods in terms of influence spread, scalability, and inference time in influence maximization tasks.

Introduction

Graph data has found a wide range of applications such as social networks and data mining (Lim et al. 2015; Liaghat, Rasekh, and Mahdavi 2013; Nettleton 2013; Bonchi 2011). One popular application is Influence Maximization (IM), which aims to identify a set of individuals that can maximize the spread of influence in a social network under a specific diffusion model. This problem is known to be NP-hard and has been extensively studied in various domains such as viral marketing (Domingos and Richardson 2001; Kempe, Kleinberg, and Tardos 2003). With the diversification of social platforms, many users on Online Social Networks (OSNs) like Facebook and Twitter are linking their accounts across multiple platforms. These interconnected OSNs with overlapping users are referred to as Multiplex Networks. The structure of multiplex networks allows users to post information across various OSNs simultaneously, presenting significant value for marketing campaigns (Vikatos, Gryllos, and Makris 2020; Zhang et al. 2022; Jalili et al. 2017).

^{*}These authors contributed equally.

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

The inner information propagation models on each OSN can vary, leading to differences in how information spreads and influences users across platforms. Consequently, it becomes crucial to customize influence maximization strategies that effectively exert influence over multiple platforms. This is known as Multiplex Influence Maximization (MIM). To date, Combinatorial Optimization (CO) algorithms for MIM (Zhan et al. 2015; Zhang et al. 2016; Kuhnle et al. 2018; Singh et al. 2019; Ling et al. 2023) have limitations compared to learning-based approaches. CO algorithms struggle to generalize to unseen graphs and handle diverse multiplex networks. They also face scalability issues when dealing with large-scale networks. Furthermore, CO algorithms rely on predefined rules or heuristics, limiting their ability to capture complex patterns and non-linear dependencies in multiplex networks. These shortcomings significantly undermine their effectiveness in optimizing the selection of influential seed nodes. In contrast, learning-based approaches (Do et al. 2024; Yuan, Shao, and Chen 2024; Chen et al. 2022; Li, Han, and Wu 2018) offer advantages in terms of generalization, scalability and capturing complex patterns. However, they still suffer critical weaknesses in MIM as follows:

- 1) *Inefficient optimization.* MIM, being a NP-hard problem with layers potentially scaling to billions, demands efficient training. RL methods, such as those presented in (Manchanda et al. 2020a; Chen et al. 2022; Yuan, Shao, and Chen 2024; Do et al. 2024), optimize seed sets in discrete spaces through exploration, iteratively improving solutions without an initial dataset. However, they rely on extensive random sampling, leading to long training time and risks of local optima. Data-driven approaches like (Ling et al. 2023) address these issues by leveraging generative models trained on diverse datasets, though their success is tied to dataset quality. Only with a sufficiently diverse training dataset can the model capture key features for optimization. Developing low-complexity models for efficient optimization remains a major challenge.
- 2) *Inaccurate propagation estimating models.* Accurately measuring propagation value is crucial for evaluating seed set effectiveness. Simulation-based methods (Manchanda et al. 2020a; Do et al. 2024; Yuan, Shao, and Chen 2024) rely on running propagation processes to compute spread which is computationally expensive and scales poorly for large graphs. GNN-based approaches (Chen et al. 2022;

Ling et al. 2023) predict spread more efficiently but face accuracy issues due to oversmoothing (Cai and Wang 2020). This challenge is exacerbated in multiplex networks, where each layer may use a different propagation model and scale to billions of nodes, complicating accurate predictions.

Our Contributions. We propose Reinforced Expert Maximization (REM), a novel framework for tackling challenges in Multiplex Influence Maximization (MIM). First, we introduce Seed2Vec, a VAE-based model that maps the discrete, noisy input space into a cleaner, continuous latent space following a Gaussian distribution. This allows us to optimize the seed set within this latent space. To address Seed2Vec’s reliance on training data quality, we frame it as a Reinforcement Learning (RL) policy, enabling efficient exploration of latent regions to generate novel seed sets with significant spread in multiplex networks. These samples are then used to iteratively retrain Seed2Vec, improving its performance. Finally, REM enhances spread estimation with the Propagation Mixture of Experts (PMoE), a method that employs multiple Graph Neural Network (GNN) models as experts to capture complex diffusion patterns. Experiments on real-world datasets show that REM outperforms state-of-the-art methods in influence spread, scalability, and inference time.

Related Work

Combinatorial optimization for IM. Influence Maximization (IM) is a simplified instance of Multiplex Influence Maximization (MIM), focusing on a single network rather than multiple interconnected ones. While IM has advanced significantly, MIM introduces unique challenges due to the interplay between interconnected layers. Early IM methods relied on simulation-based approaches (Leskovec et al. 2007), repeatedly simulating diffusion to estimate influence spread, but these are computationally expensive for large networks. Proxy-based methods (Kimura and Saito 2006; Chen, Wang, and Wang 2010; Chen, Yuan, and Zhang 2010) improved scalability by approximating influence spread with simpler metrics. Submodularity-based algorithms like CELF++ (Goyal, Lu, and Lakshmanan 2011) and UBLF (Zhou et al. 2015) achieve efficient seed selection with $(1 - 1/e)$ -approximation guarantees, while Tiptop (Li et al. 2019) recently provided near-exact solutions with $(1 - \epsilon)$ -optimal guarantees for any $\epsilon > 0$. However, MIM demands novel approaches to handle the complexity of interconnected networks. Combinatorial approximation algorithms (Zhang et al. 2016; Kuhnle et al. 2018) show promise, but MIM remains an active research area, with future efforts focusing on machine learning and exploiting multiplex network characteristics for greater efficiency and accuracy.

Machine Learning for IM. Learning-based methods, employing deep learning techniques, have emerged to overcome the limitations of traditional IM methods, particularly their lack of generalization ability. Integrating reinforcement learning (RL) with IM has shown potential (Lin, Lin, and Chen 2015; Ali, Wang, and Chen 2018), with recent advancements focusing on learning latent embeddings of nodes or networks for seed node selection (Manchanda et al. 2020a; Chen et al.

2022; Li et al. 2022). Graph neural networks (GNNs) have also been explored to encode social influence and guide node selection in IM (Ling et al. 2022b,a; Yuan, Shao, and Chen 2024). Specifically, DeepIM (Ling et al. 2023) leverages a generative approach for IM and achieves state-of-the-art performance. However, these methods are primarily designed for single-network IM problems and face challenges when extended to the more intricate MIM problem. This is primarily due to the scalability limitations in handling multiple networks and the difficulty in effectively capturing complex inter- and intra-propagation relationships within a multiplex network. Moreover, these methods are often restricted by their dependence on observation samples. While recent work (Do et al. 2024) utilizes probabilistic graphical models to represent the influence propagation process in multidimensional networks combined with reinforcement learning to find optimal seed sets, it faces scalability issues due to its reliance on traditional methods in the initial stages.

Our Method: REM enhances influence spread, scalability, and inference speed by addressing key limitations of existing MIM methods. It leverages Seed2Vec, a VAE-based model that maps seed selection into a continuous latent space, enabling smoother and faster optimization. To overcome the reliance on static datasets, REM incorporates reinforcement learning to dynamically explore and refine seed set embeddings. For accurate propagation estimation, it employs Propagation Mixture of Experts (PMoE), a set of GNN-based models specialized in capturing diverse diffusion dynamics across multiplex networks, mitigating oversmoothing. These innovations collectively improve REM’s performance in solving MIM problems.

Problem Formulation

A multiplex network with l layers is represented by $\mathcal{G} = \{G_1 = (V_1, E_1), G_2 = (V_2, E_2), \dots, G_l = (V_l, E_l)\}$, where each element consists of a directed graph $G_i = (V_i, E_i)$. If a node exists in more than one layer, then this node is added to set the overlapping users of the multiplex \mathcal{G} . Without loss of generality, we consider each layer of the multiplex has a same number of nodes. Therefore, if a node $v \in G_i$ does not belong to G_j ($i \neq j$) we add this node to G_j as an isolated node. Then for each node, interlayer edges are added to connect its adjacent interlayer copies across all the multiplex networks. Finally, we consider the set of all nodes of the multiplex network as $V = \bigcup_{i=1}^l V_i$. In this study, since we permit different layers of a multiplex to follow distinct models of influence propagation, it is essential to define a mathematical model for the propagation on network \mathcal{G} .

Definition 1 (Influence Spread). *Given a layer $G_i = (V, E_i)$, we define a seed set $S \subseteq V$. The function δ_i represents an influence propagation model within G_i , which maps from the power set of V to the non-negative real numbers, $\delta_i : 2^V \rightarrow \mathbb{R}_{\geq 0}$. The influence spread, defined as the expected number of nodes influenced by the seed set S , is denoted as $\delta_i(S)$ and is calculated as follows:*

$$\delta_i(S) = \lim_{m \rightarrow \infty} \frac{1}{m} \sum_{j=1}^m |T_j|, \quad (1)$$

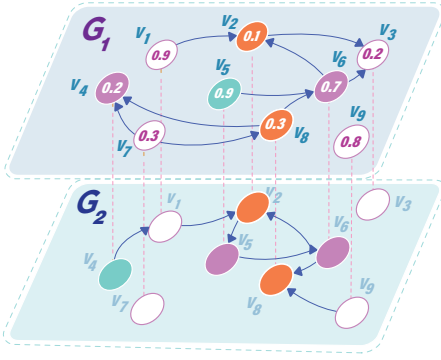


Figure 1: An example illustrating the unique "overlapping activation" property of influence propagation in a multiplex network. Two layers G_1 and G_2 has their own respective diffusion model, LT and IC. Orange nodes represent seed nodes, pink nodes are infected, and green nodes are nodes that are activated due to overlap. If propagation occurs independently within each layer, node v_6 of G_1 is inactive due to its high threshold of 0.7, meaning it requires at least 70% activated neighbors to become activated. However, in the multiplex, overlapping node v_5 of G_1 is also activated due to deterministic activation from G_2 , meeting activation requirement of node v_6 in G_1 . Therefore, total infected node set of the multiplex, not counting the same node in different layer, is now $(v_2, v_4, v_5, v_6, v_8)$.

where T_j represents the final activated sets $T_j \subset V$ given a seed set S at the j -th simulation step. The simulation continues until no more nodes are activated or until reaching the maximum number of Monte Carlo steps, m . Increasing m improves the accuracy of estimating influenced nodes. This method is applicable to most propagation models, including Independent Cascade (IC) and Linear Threshold (LT) models (Kempe, Kleinberg, and Tardos 2003).

Next, let us define the overall influence propagation model δ in the multiplex network \mathcal{G} . Firstly, when an overlapping node v is activated in one layer G_i , its copies in other layers also become activated in a deterministic manner. This phenomenon is known as "overlapping activation" (Kuhnle et al. 2018; Do et al. 2024), and is visualized in Figure 1. Secondly, when quantifying the expected number of influenced nodes in the multiplex network \mathcal{G} , overlapping nodes are counted as a single instance across layers, avoiding duplication. Thus, the overall influence $\delta(S)$ combines the independent influences from each layer while accounting for overlapping activations:

$$\delta(S) = \lim_{m \rightarrow \infty} \frac{1}{m} \sum_{j=1}^m \left| \bigcup_{i=1}^l T_{ij}(S) \right| \quad (2)$$

where $T_{ij} \subset V$ represents the final activated sets in layer i given a seed set S at the j -th simulation step. We are now ready to define our MIM problem as follows:

Definition 2 (Multiplex Influence Maximization (MIM)). Given a multiplex graph $\mathcal{G} =$

$(G_1 = (V, E_1), \delta_1), \dots, (G_l = (V, E_l), \delta_l)$ and a budget $b \in \mathbb{N}$. Specifically, seed set S is represented as a binary vector $\mathbf{x} \in \mathbb{R}^{1 \times |V|}$, where each element x_j corresponds to a node v_j in V . Specifically, $x_j = 1$ if v_j is included in the seed set, and $x_j = 0$ otherwise. Suppose we have a training dataset of seed set indicators pairs (\mathbf{x}, y) , where \mathbf{x} represents a seed set and $y = \delta(\mathbf{x})$ is the corresponding total number of infected nodes. The MIM problem asks us to find an optimal seed node set $\tilde{\mathbf{x}}$ of size at most b to maximize the overall influence spread $\delta(\mathbf{x})$ calculated in the multiplex. This problem is formulated as follows:

$$\tilde{\mathbf{x}} = \arg \max_{|\mathbf{x}| \leq b} \delta(\mathbf{x}) \quad (3)$$

For each layer $G_i \in \mathcal{G}$, many greedy based algorithms (Leskovec et al. 2007; Goyal, Lu, and Lakshmanan 2011; Tang, Xiao, and Shi 2014; Tang, Shi, and Xiao 2015) have obtained a performance guarantee bound of $(1 - 1/e)$, if δ_i is submodular and monotone increasing (Kempe, Kleinberg, and Tardos 2003). If all δ_i of all G_i satisfy the Generalized Deterministic Submodular (GDS) property, then δ is submodular (Kuhnle et al. 2018).

Our Framework: REM

The REM model addresses mentioned challenges by following concepts illustrated in Figure 2. First, instead of optimizing the seed set in a complex and discrete space, REM employs our proposed Seed2Vec, a Variational Autoencoder (VAE)-based model (Kingma 2013). VAE is a generative framework that encodes data into a continuous latent space while preserving meaningful structure, enabling the representation of complex seed sets in a less noisy form. This allows for optimization and the generation of new potential solutions within that space. Recognizing that Seed2Vec only captures and generates solutions within the feature distribution of the original training data, our framework treats Seed2Vec as an RL agent. This agent explores and exploits diverse latent representations during each training episode. For each latent sample generated by Seed2Vec, we apply our proposed Propagation Mixture of Experts (PMoE) to predict its propagation with very high accuracy, rank, and store it in a Priority Replay Memory (PRM) (Horgan et al. 2018), a structure designed to prioritize important samples based on their predictive value for enhanced learning efficiency. We then sample the top k samples from PRM and combine them with the original dataset to form a combined dataset. Finally, REM uses this combined dataset to retrain the Seed2Vec model, enhancing its seed set generation capability. More algorithmic details can be found in Appendix A.

Seed2Vec: Learning To Embed Complex Seed Set

To optimize and identify quality seed sets in a multiplex network, we propose characterizing the probability of a seed node set, denoted as $p_\phi(\mathbf{x})$, given the multiplex graph \mathcal{G} . Learning $p_\phi(\mathbf{x})$ can provide insights into the underlying nature of the seed set, facilitating effective exploration of seed sets in the search space. However, learning such a probability is challenging due to the interconnections between different

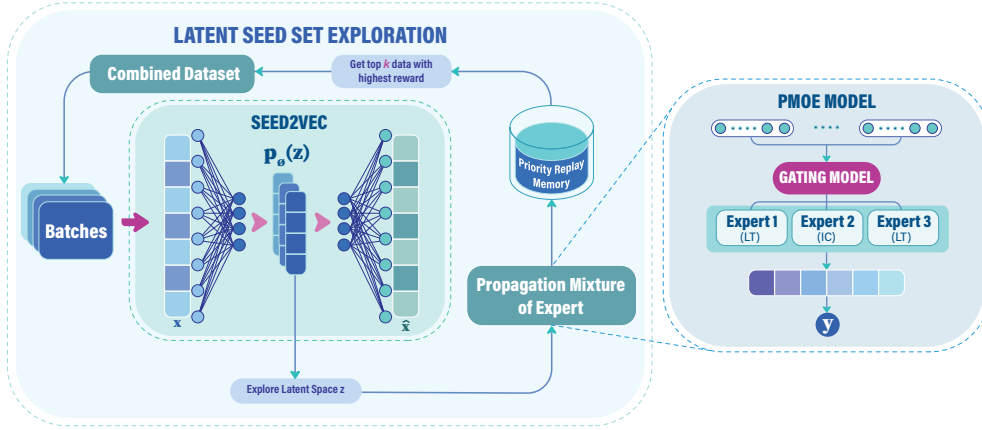


Figure 2: The diagram depicts REM’s process for addressing the MIM problem. Initially, REM utilizes Seed2Vec to embed complex seed set representations into a continuous latent space. Then, REM explores and generates various diverse seed sets from this latent space. REM controls the quality of seed set generation through the Propagation Mixture of Experts (PMoE), a model capable of accurately learning and predicting the propagation of a given seed set in a large-scale multiplex network. Once the synthetic sets are generated, they are stored in a priority replay memory. To prevent model collapse or catastrophic forgetting, the top k seed sets, which PMoE predicts to contribute the most to propagation in multiplex networks, are then combined with the original collected dataset to construct a new dataset for model retraining. This process reinforces the capability to produce higher-quality seed sets in future iterations.

nodes within each seed set and their high correlation based on the network topology of \mathcal{G} . These complex connections make the node relationships within seed sets difficult to decipher compared to other similar combinatorial problems. Therefore, instead of learning directly the complex representation of \mathbf{x} , we learn a latent presentation \mathbf{z} using Variational Auto Encoder (VAE) (Kingma and Welling 2013) denoted as \mathcal{F}_θ . For convenient, we further decompose the VAE model \mathcal{F}_θ into two models: the Encoder denoted as \mathcal{E}_ψ and the Decoder model denoted as \mathcal{D}_ϕ . Formally, we have:

$$\mathcal{F}_\theta = \mathcal{E}_\psi \circ \mathcal{D}_\phi, \quad \hat{\mathbf{x}} = \mathcal{F}_\theta(\mathbf{x}) = \mathcal{D}_\phi(\mathcal{E}_\psi(\mathbf{x})) = \mathcal{D}_\phi(\mathbf{z}), \quad (4)$$

where $\hat{\mathbf{x}} \in [0, 1]^{1 \times |V|}$ represents the reconstructed seed set generated. Specifically, to generate \mathbf{x} , \mathcal{F}_θ assumes the existence of a latent random variable $\mathbf{z} \in \mathbb{R}^{1 \times s}$, where s represents the dimension of the variables in \mathbf{z} . This latent variable captures the features of the original seed set and follows a latent distribution $p_\phi(\mathbf{z})$. The complete generative process can be described by the equation:

$$p_\phi(\mathbf{z} | \mathbf{x}) = \frac{p_\phi(\mathbf{x} | \mathbf{z})p_\phi(\mathbf{z})}{p_\phi(\mathbf{x})} \quad (5)$$

However, computing the exact value of $p_\phi(\mathbf{x}) = \int \dots \int p_\phi(\mathbf{x}, \mathbf{z}, d_{\mathbf{z}_0} \dots d_{\mathbf{z}_v})$ is intractable, making the equation computationally challenging. To address this problem, \mathcal{E}_ψ will learn q_ψ which is approximated posterior distribution of $p_\phi(\mathbf{z} | \mathbf{x})$. The goal is to approximate the intractable posterior distribution with a simpler distribution $q_\psi(\mathbf{z} | \mathbf{x})$ given the seed set \mathbf{x} . In other words, the objective is to have $p_\phi(\mathbf{z} | \mathbf{x}) \approx q_\psi(\mathbf{z} | \mathbf{x})$.

This is used to derive the following Evidence Lower Bound (ELBO) to train the model using the reparameterization trick

and SGD (Kingma and Welling 2013).

$$\begin{aligned} \mathcal{L}^{\text{ELBO}} &= \mathbb{E}_{q_\psi} [\log p_\phi(\mathbf{z}, \mathbf{x})] - \mathbb{E}_{q_\psi} [\log q_\psi(\mathbf{z} | \mathbf{x})] \\ &= \mathbb{E}_{q_\psi} [\log p_\phi(\mathbf{x} | \mathbf{z})] + \mathbb{E}_{q_\psi} [\log p_\phi(\mathbf{z})] \\ &\quad - \mathbb{E}_{q_\psi} [\log q_\psi(\mathbf{z} | \mathbf{x})] \\ &= \mathbb{E}_{q_\psi} [\log p_\phi(\mathbf{x} | \mathbf{z})] - \mathbb{E}_{q_\psi} \left[\log \frac{q_\psi(\mathbf{z} | \mathbf{x})}{p_\phi(\mathbf{z})} \right] \end{aligned} \quad (6)$$

Note that we model $p_\phi(\mathbf{z})$ as a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$, where μ and σ are defined hyperparameters. For a detailed decomposition of the ELBO and related details, refer to Appendix B.

Propagation Mixture of Expert

Applying Graph Neural Networks (GNNs) to predict propagation in billion-scale multiplex networks is challenging due to oversmoothing (Cai and Wang 2020). Additionally, a single GNN with h layers aggregates information from h -hop neighbors, potentially mixing data from different layers and causing inaccuracies. To address this, we propose the Propagation Mixture of Experts (PMoE), which employs multiple GNN models with varying layer depths to effectively capture propagation dynamics. Nodes are routed to the most suitable expert based on their characteristics and desired propagation depth, enabling the model to focus on relevant information while reducing noise, ensuring accurate and efficient predictions in large-scale networks.

Our PMoE framework captures the propagation process given a seed set \mathbf{x} and a multiplex graph \mathcal{G} . In this framework, we define a set of C "expert networks," denoted as e_1, e_2, \dots, e_C . Each expert e_i is implemented as a GNN with varying layer depths, outputting $e_i(\mathbf{x}, \mathcal{G}, \xi_i) \in [0, 1]^{1 \times |V|}$, a

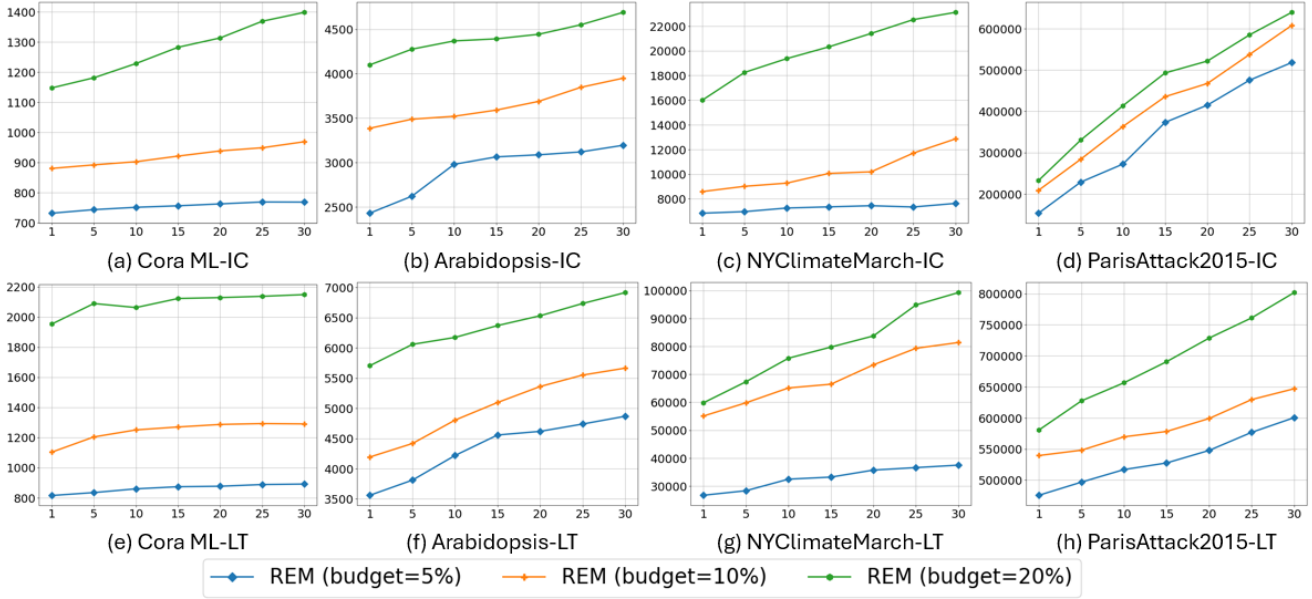


Figure 3: Difference in influence spread (y-axis) of REM output on different dataset and budget when increasing exploration steps(x axis). Fig. 3a - 3d and Fig. 3e - 3h are evaluated under the IC and LT model, respectively.

vector representing the estimated infection probability for each node in \mathcal{G} , where ξ_i is the parameter of the i -th expert. To effectively leverage the diverse knowledge of experts, we employ a routing network R , which outputs a probability distribution over experts $R(\mathbf{x}) \in \mathbb{R}^{1 \times C}$ based on the input seed set \mathbf{x} . Each element in this distribution corresponds to the relevance probability of a particular expert for the given input. Inspired by the noisy top- m routing mechanism proposed by (Shazeer et al. 2017), we select the m most relevant experts for each input. This mechanism operates as follows:

$$Q(\mathbf{x}) = \mathbf{x}\xi_g + \epsilon \cdot \text{Softplus}(\mathbf{x}\xi_n), \quad (7)$$

$$R(\mathbf{x}) = \text{Softmax}(\text{TopM}(Q(\mathbf{x}), m)), \quad (8)$$

In this equation, $\epsilon \sim \mathcal{N}(0, 1)$ represents standard Gaussian noise. The parameters ξ_g and ξ_n are learnable weights that control the contributions of the clean and noisy scores, respectively. The expected value $\mathcal{M}(\mathbf{x}, \mathcal{G}; \xi)$, where $\xi = [\xi_g, \xi_n, \xi_1, \dots, \xi_C]$ represents the parameters of the PMoE model \mathcal{M} , is calculated based on the outputs of all experts and can be formulated as follows:

$$\mathcal{M}(\mathbf{x}, \mathcal{G}; \xi) = \sum_{i=1}^C R_i(\mathbf{x}) e_i(\mathbf{x}, \mathcal{G}; \xi_i) \quad (9)$$

Here, $R_i(\mathbf{x})$ is the i -th element of routing network $R(\mathbf{x})$, representing the relevance probability of the i -th expert in predicting the influence of seed set \mathbf{x} . In this scenario, the total number of infected nodes, denoted as $\hat{y} \in \mathbb{R}_+$, is calculated as $\hat{y} = \mathcal{P}(\mathbf{x}, \mathcal{G}; \xi) = g(\mathcal{M}(\mathbf{x}, \mathcal{G}; \xi); \zeta)$. Here, $g(\cdot)$ is a normalization function (e.g., $l-1$ norm) and ζ is the threshold to transform the probability into discrete value.

Lemma 1 (Monotonicity of PMoE Models). *Assuming the PMoE model has been trained to convergence and during*

the inference phase, noisy scores ξ_n are not considered, for any GNN-based, \mathcal{P} is infection monotonic if the aggregation function and combine function in GNN are non-decreasing. (Proof in Appendix C1)

According to Lemma 1 the PMoE model $\mathcal{P}(\mathbf{x}, \mathcal{G}; \xi)$ has the theoretical guarantee to retain monotonicity, and the objective of learning the PMoE model $\mathcal{P}(\mathbf{x}, \mathcal{G}; \xi)$ is given as maximizing the following probability with a constraint:

$$\max_{\xi} \mathbb{E}[p_{\xi}(y|\mathbf{x}, \mathcal{G})], \quad (10)$$

Latent Seed Set Exploration

As a generative model, Seed2Vec can only produce quality seed sets if the original training data is feature-rich. If the data is biased toward dominant features or lacks diversity, Seed2Vec may miss important but less prevalent features. As the multiplex becomes more complex and the number of nodes increases, the model tends to favor dominant seed nodes in the dataset, often overlooking less frequent but potentially significant ones. REM overcomes this by treating Seed2Vec as an RL agent, actively exploring novel and potentially impactful seed sets that maximize propagation to retrain and reinforce itself by the following lemma:

Lemma 2 (Latent Entropy Maximization Equivalence). *Assume Seed2Vec model convergence, we have $\arg \max_{\mathbf{z}} \mathcal{H}(\mathcal{D}_{\phi}(\mathbf{z})) \propto \arg \max_{\mathbf{x}} \mathcal{H}(\mathbf{x})$. (Proof in Appendix C2)*

According to Lemma 2, exploration within the latent space \mathbf{z} , aimed at identifying the novel seed set \mathcal{S}_t , where $t = 1, 2, 3, \dots$ represents the training episode, is proportional to exploration within the discrete space \mathbf{x} . This correlation emerges because a well-trained Seed2Vec model,

using the original collected seed set \mathbf{X}_0 , ensures both continuity—where nearby points in the latent space decode into similar content—and completeness, meaning that any point sampled from the latent space’s chosen distribution generates ‘meaningful’ content. At this juncture, $p_\phi(\mathbf{z} | \mathbf{x}) \approx q_\psi(\mathbf{z} | \mathbf{x})$, with $q_\psi(\mathbf{z} | \mathbf{x})$ converging to a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ as indicated by the second term of Equation 6. Typically, an RL agent could explore various latent features by sampling $\mathbf{z} \sim \mathcal{N}(\mu, \sigma^2)$ and reconstructing the seed set $\hat{\mathbf{x}}$ using the Decoder (i.e., $\hat{\mathbf{x}} = \mathcal{D}_\phi(\mathbf{z})$). However, since $q_\psi(\mathbf{z} | \mathbf{x})$ converges to a continuous function that has a derivative with respect to \mathbf{z} . Instead of the RL agent exploring by random sampling, we use Gradient Descent directly on \mathbf{z} to minimize the following objective function:

$$\mathcal{L}^{\text{Explore}}(\mathbf{z}) = \mathbb{E}(c \cdot \mathcal{H}(\mathcal{D}_\phi(\mathbf{z})) + \exp(-\mathcal{P}(\mathcal{D}_\phi(\mathbf{z}))) \quad (11)$$

where c are coefficients. The term $\mathcal{H}(\mathcal{D}_\phi(\mathbf{z})) = -\sum_{i=1}^{|\hat{\mathbf{x}}|} p(\hat{\mathbf{x}}_i) \log p(\hat{\mathbf{x}}_i)$ denotes the entropy of the latent variable, which promotes exploration within new regions of the latent space. The function $\mathcal{P}(\mathcal{D}_\phi(\mathbf{z}))$ refers to the Propagation Mixture of Experts (PMoE), which is detailed in the following section, and is used for predicting the reconstructed seed set $\hat{\mathbf{x}} = \mathcal{D}_\phi(\mathbf{z})$. To align with the objective of minimizing the loss function, we employ the exponential function, $\exp(\cdot)$, to reduce the impact of $\mathcal{P}(\mathcal{D}_\phi(\mathbf{z}))$ as its value increases. With the novel synthetic seed set \mathcal{S}_t (store by using Priority Replay Memory (Schaul et al. 2015)) obtained by optimizing Equation 11, we sampling top k best samples and combine them with the original dataset \mathbf{X}_0 to create a Combined Dataset $\mathbf{X}_t = \mathcal{S}_t^{(<k)} \cup \mathbf{X}_0$. Therefore, as training episode t progresses, we can use \mathbf{X}_t to retrain the Seed2Vec model \mathcal{F}_θ . This approach allows \mathcal{F}_θ to generate improved seed sets in future iterations.

End-to-end Learning Objective. Finally, to bridge representation learning, latent seed set exploration, and diffusion model training, we minimize the following end-to-end objective function, which combines Eq. (6), (11), and (10):

$$\mathcal{L}_{\text{train}} = \mathbb{E}[\mathcal{L}^{\text{ELBO}}(\theta) + \mathcal{L}^{\text{PMoE}}(\xi) + \mathcal{L}^{\text{Explore}}(\mathbf{z})] \quad (12)$$

where $\mathcal{L}^{\text{PMoE}} = (\hat{y} - y)^2$.

Seed Node Set Inference. Finally, our method conclude with inferencing the seed node set from the continuous latent space. Specifically, gradient ascent is employed to find the latent representation $\tilde{\mathbf{z}}$ that maximizes the predicted influence spread, based on the estimation provided by the PMoE model. Representation $\tilde{\mathbf{z}}$ is decoded using the decoder network of Seed2Vec to obtain the optimal seed node set $\tilde{\mathbf{x}}$.

Theorem 3 (Influence Estimation Consistency). *Given two distinct seed sets $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$, with their corresponding latent representations $\mathbf{z}^{(i)}$ and $\mathbf{z}^{(j)}$ encoded by a Seed2Vec. If the reconstruction error is minimized during the training and $\mathcal{P}(p_\phi(\mathbf{z}^{(i)}), \mathcal{G}; \xi) > \mathcal{P}(p_\phi(\mathbf{z}^{(j)}), \mathcal{G}; \xi)$, then it follows that $\mathcal{P}(\mathbf{x}^{(i)}, \mathcal{G}; \xi) > \mathcal{P}(\mathbf{x}^{(j)}, \mathcal{G}; \xi)$. (Proof in Appendix C3)*

According to Theorem 3, the optimal seed set that maximizes influence can be found by optimizing \mathbf{z} .

Experiment

We conduct experiments to compare our proposed REM framework to 6 other state-of-the-art frameworks across 5 real world networks in various settings.

Experiment Setup

Our main objective is to evaluate the effect of influence spread across different scenarios in Influence Maximization (IM). Our experiments focus on two dominant propagation models within IM: the Linear Threshold (LT) and Independent Cascade (IC) models. To delve deeper into our experimental setup, we refer to Appendix D.

Dataset. Our experiments leverage multiple multiplex network datasets of diverse interaction types and systems. The Arabidopsis Multiplex Network from BioGRID (Stark et al. 2006) details genetic and protein interactions for *Arabidopsis thaliana*, comprising 7 layers, 6,980 nodes, and 18,654 edges. The NYClimateMarch2014 Twitter Network (Omodei, De Domenico, and Arenas 2015) captures tweets during the People’s Climate March, featuring 3 layers, 102,439 nodes, and 353,495 edges. The ParisAttack2015 Twitter Network (De Domenico and Altmann 2020) includes social interactions during the 2015 Paris Attacks, with 3 layers, 1,896,221 nodes, and 4,163,947 edges. We also use the Cora dataset (McCallum et al. 2000), a citation network of 2,708 publications and 7,981 edges, to analyze influence in academic.

Comparison to other Methods

We assess the performance of REM by comparing it against two categories of influence maximization techniques. 1) Traditional methods: *ISF (Influential Seed Finder)* (Kuhnle et al. 2018) is a greedy algorithm designed for multiplex influence maximization; *KSN (Knapsack Seeding of Networks)* (Kuhnle et al. 2018) utilizes a knapsack approach to find the best seed users in a multiplex network. 2) Deep learning methods: *ToupleGDD* (Chen et al. 2022), *GCOMB* (Manchanda et al. 2020b), *DeepIM* (Ling et al. 2023) are state-of-the-art single network influence maximization solutions. For multiplex network, the MIM-Reasoner (Do et al. 2024) method utilize probabilistic graphical models to capture the dynamics within the multiplex, then determine the best seed sets with a reinforcement learning solution. We also evaluate the performance of 2 different REM variants to demonstrate the effectiveness of our approach. One approach is REM-NonRL, which does not employ the exploration of seed sets and solely relies on an initial dataset to provide solution. This variant provides observation on the effectiveness of our proposed reinforcement learning set up. The other variant, REM-NonMixture, forego our Mixture of expert set up, capture the complicated multiplex propagation with one GNN model. This variant will underscore the advantages of our more complex configurations. The comparison is based on three metrics: total influence spread (activated nodes) and inference time (wall-clock time, in seconds).

Quantitative Analysis

We evaluate the performance of the REM method against other IM strategies by comparing their ability to optimize influence across various datasets. In each case, models identify

Methods	Cora-ML				Arabidopsis				NYClimateMarch2014				ParisAttack2015			
	1%	5%	10%	20%	1%	5%	10%	20%	1%	5%	10%	20%	1%	5%	10%	20%
ISF	398.34	778.62	979.87	1368.56	2415.04	3140.58	3871.10	4694.16	-	-	-	-	-	-	-	-
KSN	398.31	778.62	979.10	1366.03	2282.72	2941.54	3621.26	4641.34	-	-	-	-	-	-	-	-
GCOMB	347.11	766.02	976.14	1251.52	2315.44	3097.97	3622.08	4547.63	2093.32	7228.02	11780.29	16933.89	114672.50	180977.09	356187.81	587891.16
ToupleGDD	349.01	721.42	862.42	1132.66	2044.67	2856.37	3487.41	4483.27	1821.43	6714.98	10231.81	17822.21	102872.11	171992.43	335298.85	563387.05
DeepIM	311.52	606.82	826.41	1179.45	1993.37	2397.89	3328.33	4073.50	1893.02	6409.32	8064.77	14269.49	83972.39	149237.59	281298.85	480393.42
MIM-Reasoner	398.22	778.02	978.95	1363.35	2396.95	2989.15	3729.39	4621.54	2101.86	7387.91	11984.55	21062.87	129650.48	217291.02	379932.17	608192.57
REM-NonRL	321.24	732.84	880.51	1151.68	2076.31	2421.13	3399.34	4120.51	2057.82	6835.44	8564.63	16021.63	87109.27	152621.23	298923.41	515237.59
REM-NonMixture	343.16	736.42	921.24	1301.55	2317.39	2982.67	3712.35	4611.27	2215.82	7124.29	10034.71	18932.40	125098.55	207621.48	380274.03	605875.67
REM	347.34	765.48	965.04	1404.14	2430.84	3181.26	3964.45	4701.75	2162.83	7465.96	12834.45	23142.83	147193.96	229769.76	402372.32	637621.59

Table 1: Performance comparison under IC diffusion pattern. – indicates out-of-memory error. (Best is highlighted with bold.)

Methods	Cora-ML				Arabidopsis				NYClimateMarch2014				ParisAttack2015			
	1%	5%	10%	20%	1%	5%	10%	20%	1%	5%	10%	20%	1%	5%	10%	20%
ISF	381.0	907.0	1392.0	2145.0	2901.0	4571.0	5686.0	6855.0	-	-	-	-	-	-	-	-
KSN	381.0	907.0	1392.0	2145.0	2440.0	3969.0	4934.0	6147.0	-	-	-	-	-	-	-	-
GCOMB	379.0	1005.0	1297.0	2026.0	2501.0	4247.0	4993.0	6286.0	5281.0	29974.0	63187.0	74865.0	429978.0	498301.0	571193.0	729745.0
ToupleGDD	375.0	998.0	1192.0	1989.0	2479.0	4147.0	4725.0	5841.0	4866.0	26987.0	61827.0	71865.0	420086.0	487925.0	568872.0	739712.0
DeepIM	291.0	708.0	1023.0	1881.0	1772.0	3232.0	3211.0	3968.0	3582.0	21246.0	51721.0	57977.0	387129.0	460273.0	538160.0	578724.0
MIM-Reasoner	381.0	907.0	1392.0	2145.0	2554.0	4133.0	5027.0	6191.0	6872.0	35833.0	77925.0	96239.0	487871.0	550864.0	629830.0	786819.0
REM-NonRL	321.0	811.0	1101.0	1964.0	2176.0	3548.0	4185.0	5742.0	3976.0	26920.0	55315.0	59764.0	408149.0	468621.0	540824.0	581872.0
REM-NonMixture	361.0	1173.2	1212.0	2102.0	2632.0	4586.0	5089.0	6197.0	5352.0	30187.0	66872.0	77359.0	449862.0	529562.0	592055.0	692974.0
REM	376.0	883.0	1281.0	2141.0	2943.0	4894.0	5705.0	6890.0	7111.0	37417.0	81255.0	98976.0	503994.0	604890.0	651100.0	804469.0

Table 2: Performance comparison under LT diffusion pattern. – indicates out-of-memory error. (Best is highlighted with bold.)

seed nodes representing 1%, 5%, 10%, and 20% of all nodes. We simulate the diffusion process until completion and determine the average influence spread across 100 iterations. We report the final number infected nodes.

IM under IC Model. The methods are evaluated on five datasets under the IC diffusion model with budgets of 1%, 5%, 10%, and 20% of network nodes. As shown in Table 1, REM consistently outperforms other methods, particularly on large datasets such as NYClimateMarch2014 and ParisAttack2015. Traditional methods (ISF, KSN) perform well on smaller graphs but struggle to scale with larger graphs and higher budgets. Single-graph learning methods (GCOMB, TOUPLEGDD, DEEPIIM) fall behind due to their inability to adapt to multiplex networks. While MIM-Reasoner achieves strong results on larger multiplex networks, it is outperformed by REM. Notably, REM variants (REM-NonRL, REM-NonMixture) show significant performance drops, underscoring the importance of REM’s key components.

IM under LT Model. We evaluate the methods under the LT diffusion model, with the results in Table 2 showing that REM consistently outperforms other techniques in maximizing node infections. REM’s superiority is particularly evident on large networks and with a 20% seed set, achieving 10% and 15% higher influence spread than the best competing methods on the NYClimateMarch2014 and ParisAttack2015 datasets, respectively. This performance highlights REM’s superior generalization across diffusion models.

IM with explore step number. We compare the effectiveness of increasing exploration steps under the IC and LT models within a budget constraint. As shown in Figure 3, more exploration steps generally improve results across networks, especially for larger datasets. For smaller datasets like Cora-ML, the performance difference is minimal, while for larger datasets, the gap widens significantly with more steps.

Methods	10,000	20,000	30,000	50,000
GCCOMB	17.894s	30.831s	46.275s	73.983s
ToupleGDD	15.873s	25.321s	37.882s	58.985s
MIM-Reasoner	7.948s	12.532s	26.575s	36.437s
DeepIM	10.321s	19.325s	32.185s	44.871s
REM	8.873s	10.198s	23.404s	33.482s

Table 3: Average inference runtime (in seconds) as the number of nodes increases. We select 10% of nodes as seeds.

Scalability Analysis

We investigate the runtime of seed sets when increasing of graph size of REM verse other learning-based IM solutions. As can be seen in Table 3, REM demonstrates near-linear growth of runtime as the graph size increases. In addition, it achieves a generally shorter inference time (on average, it has a 10% faster inference time than the second-fastest MIM-Reasoner and a 20% faster inference time than the third-fastest DeepIM).

Conclusion

This paper has introduced REM, a framework designed to tackle the challenges of MIM. Through the integration of Mixture of Experts and a RL-based latent space exploration strategy, REM offers a robust solution to optimize influence spread across multiplex networks. Our approach demonstrates superior scalability and efficiency while also excels in handling the diversity of propagation mechanisms in these networks. The experimental results on real-world datasets validate REM’s effectiveness, showcasing its ability to outperform existing state-of-the-art methods in both influence spread and computational efficiency.

References

- Ali, K.; Wang, C.-Y.; and Chen, Y.-S. 2018. Boosting reinforcement learning in competitive influence maximization with transfer learning. In *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, 395–400. IEEE.
- Bonchi, F. 2011. Influence Propagation in Social Networks: A Data Mining Perspective. In *2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, volume 1, 2–2.
- Cai, C.; and Wang, Y. 2020. A Note on Over-Smoothing for Graph Neural Networks. *arXiv preprint arXiv: 2006.13318*.
- Chen, T.; Yan, S.; Guo, J.; and Wu, W. 2022. ToupleGDD: A Fine-Designed Solution of Influence Maximization by Deep Reinforcement Learning. *arXiv preprint arXiv:2210.07500*.
- Chen, W.; Wang, C.; and Wang, Y. 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, KDD '10*, 1029–1038. New York, NY, USA: Association for Computing Machinery. ISBN 9781450300551.
- Chen, W.; Yuan, Y.; and Zhang, L. 2010. Scalable Influence Maximization in Social Networks under the Linear Threshold Model. In *2010 IEEE International Conference on Data Mining*, 88–97.
- De Domenico, M.; and Altmann, E. G. 2020. Unraveling the origin of social bursts in collective attention. *Scientific reports*, 10(1): 4629.
- Do, N. H. K.; Chowdhury, T.; Ling, C.; Zhao, L.; and Thai, M. T. 2024. MIM-Reasoner: Learning with Theoretical Guarantees for Multiplex Influence Maximization. In Dasgupta, S.; Mandt, S.; and Li, Y., eds., *International Conference on Artificial Intelligence and Statistics, 2-4 May 2024, Palau de Congressos, Valencia, Spain*, volume 238 of *Proceedings of Machine Learning Research*, 2296–2304. PMLR.
- Domingos, P.; and Richardson, M. 2001. Mining the network value of customers. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '01*, 57–66. New York, NY, USA: Association for Computing Machinery. ISBN 158113391X.
- Goyal, A.; Lu, W.; and Lakshmanan, L. 2011. CELF++: Optimizing the greedy algorithm for influence maximization in social networks. volume 47-48, 47–48.
- Horgan, D.; Quan, J.; Budden, D.; Barth-Maron, G.; Hessel, M.; Van Hasselt, H.; and Silver, D. 2018. Distributed prioritized experience replay. *arXiv preprint arXiv:1803.00933*.
- Jalili, M.; Orouskhani, Y.; Asgari Mehrabadi, M.; Alipourfard, N.; and Perc, M. 2017. Link prediction in multiplex online social networks. *Royal Society Open Science*, 4: 160863.
- Kempe, D.; Kleinberg, J.; and Tardos, É. 2003. Maximizing the spread of influence through a social network. In *Proc. of the KDD*.
- Kimura, M.; and Saito, K. 2006. Tractable Models for Information Diffusion in Social Networks. In Fürnkranz, J.; Scheffer, T.; and Spiliopoulou, M., eds., *Knowledge Discovery in Databases: PKDD 2006*, 259–271. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 978-3-540-46048-0.
- Kingma, D. P. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.
- Kingma, D. P.; and Welling, M. 2013. Auto-Encoding Variational Bayes. *International Conference on Learning Representations*.
- Kuhnle, A.; Alim, M. A.; Li, X.; Zhang, H.; and Thai, M. T. 2018. Multiplex influence maximization in online social networks with heterogeneous diffusion models. *IEEE Transactions on Computational Social Systems*, 5(2): 418–429.
- Leskovec, J.; Krause, A.; Guestrin, C.; Faloutsos, C.; VanBriesen, J.; and Glance, N. 2007. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, 420–429.
- Li, H.; Xu, M.; Bhowmick, S. S.; Rayhan, J. S.; Sun, C.; and Cui, J. 2022. PIANO: Influence Maximization Meets Deep Reinforcement Learning. *IEEE Transactions on Computational Social Systems*.
- Li, Q.; Han, Z.; and Wu, X.-M. 2018. Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning. *AAAI Conference on Artificial Intelligence*.
- Li, X.; Smith, J. D.; Dinh, T. N.; and Thai, M. T. 2019. Tip-top:(almost) exact solutions for influence maximization in billion-scale networks. *IEEE/ACM Transactions on Networking*, 27(2): 649–661.
- Liaghat, Z.; Rasekh, A.; and Mahdavi, A. 2013. Application of data mining methods for link prediction in social networks. *Social Network Analysis and Mining*, 3.
- Lim, J. S.; Ri, S. Y.; Egan, B. D.; and Biocca, F. A. 2015. The cross-platform synergies of digital video advertising: Implications for cross-media campaigns in television, Internet and mobile TV. *Computers in Human Behavior*, 48: 463–472.
- Lin, S.-C.; Lin, S.-D.; and Chen, M.-S. 2015. A learning-based framework to handle multi-round multi-party influence maximization on social networks. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 695–704.
- Ling, C.; Chowdhury, T.; Jiang, J.; Wang, J.; Zhang, X.; Chen, H.; and Zhao, L. 2022a. DeepGAR: Deep Graph Learning for Analogical Reasoning. In *2022 IEEE International Conference on Data Mining (ICDM)*, 1065–1070. Los Alamitos, CA, USA.
- Ling, C.; Jiang, J.; Wang, J.; Thai, M. T.; Xue, R.; Song, J.; Qiu, M.; and Zhao, L. 2023. Deep graph representation learning and optimization for influence maximization. In *International Conference on Machine Learning*, 21350–21361. PMLR.
- Ling, C.; Jiang, J.; Wang, J.; and Zhao, L. 2022b. Source Localization of Graph Diffusion via Variational Autoencoders for Graph Inverse Problems. In *Proc. of the KDD*.
- Manchanda, S.; Mittal, A.; Dhawan, A.; Medya, S.; Ranu, S.; and Singh, A. 2020a. Gcomb: Learning budget-constrained combinatorial algorithms over billion-sized graphs. *Advances in Neural Information Processing Systems*, 33: 20000–20011.

- Manchanda, S.; MITTAL, A.; Dhawan, A.; Medya, S.; Ranu, S.; and Singh, A. 2020b. GCOMB: Learning Budget-constrained Combinatorial Algorithms over Billion-sized Graphs. In Larochele, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 20000–20011. Curran Associates, Inc.
- McCallum, A. K.; Nigam, K.; Rennie, J.; and Seymore, K. 2000. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3(2): 127–163.
- Nettleton, D. F. 2013. Data mining of social networks represented as graphs. *Computer Science Review*, 7: 1–34.
- Omodei, E.; De Domenico, M. D.; and Arenas, A. 2015. Characterizing interactions in online social networks during exceptional events. *Frontiers in Physics*, 3.
- Schaul, T.; Quan, J.; Antonoglou, I.; and Silver, D. 2015. Prioritized Experience Replay. *International Conference on Learning Representations*.
- Shazeer, N. M.; Mirhoseini, A.; Maziarz, K.; Davis, A.; Le, Q. V.; Hinton, G. E.; and Dean, J. 2017. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. *International Conference on Learning Representations*.
- Singh, S. S.; Singh, K.; Kumar, A.; and Biswas, B. 2019. MIM2: Multiple influence maximization across multiple social networks. *Physica A: Statistical Mechanics and its Applications*, 526: 120902.
- Stark, C.; Breitkreutz, B.-J.; Regul, T.; Boucher, L.; Breitkreutz, A.; and Tyers, M. 2006. BioGRID: A general repository for interaction datasets. *Nucleic acids research*, 34: D535–9.
- Tang, Y.; Shi, Y.; and Xiao, X. 2015. Influence maximization in near-linear time: A martingale approach. In *Proc. of the SIGMOD*.
- Tang, Y.; Xiao, X.; and Shi, Y. 2014. Influence maximization: Near-optimal time complexity meets practical efficiency. In *Proc. of the SIGMOD*, 75–86.
- Vikatos, P.; Gryllos, P.; and Makris, C. 2020. Marketing campaign targeting using bridge extraction in multiplex social network. *Artificial Intelligence Review*, 53.
- Yuan, Z.; Shao, M.; and Chen, Z. 2024. Graph Bayesian Optimization for Multiplex Influence Maximization. In Wooldridge, M. J.; Dy, J. G.; and Natarajan, S., eds., *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, 22475–22483. AAAI Press.
- Zhan, Q.; Zhang, J.; Wang, S.; Yu, P. S.; and Xie, J. 2015. Influence maximization across partially aligned heterogeneous social networks. In *Pacific-Asia conference on knowledge discovery and data mining*, 58–69. Springer.
- Zhang, H.; Chen, X.; Peng, Y.; Kou, G.; and Wang, R. 2022. The interaction of multiple information on multiplex social networks. *Information Sciences*, 605: 366–380.
- Zhang, H.; Nguyen, D. T.; Zhang, H.; and Thai, M. T. 2016. Least Cost Influence Maximization Across Multiple Social Networks. *IEEE/ACM Transactions on Networking*, 24(2): 929–939.
- Zhou, C.; Zhang, P.; Zang, W.; and Guo, L. 2015. On the Upper Bounds of Spread for Greedy Algorithms in Social Network Influence Maximization. *IEEE Transactions on Knowledge and Data Engineering*, 27(10): 2770–2783.