DeepSN: A Sheaf Neural Framework for Influence Maximization

Asela Hevapathige, Qing Wang, Ahad N. Zehmakan

Graph Research Lab, School of Computing, Australian National University
{asela.hevapathige, qing.wang, ahadn.zehmakan}@anu.edu.au

Abstract

Influence maximization is a key topic in data mining, with broad applications in social network analysis and viral marketing. In recent years, researchers have increasingly turned to machine learning techniques to address this problem. By learning the underlying diffusion processes from data, these methods improve the generalizability of solutions while optimizing objectives to identify the optimal seed set for maximizing influence. Nonetheless, two fundamental challenges remain unresolved: (1) While Graph Neural Networks (GNNs) are increasingly employed to learn diffusion models, their traditional architectures often fail to capture the complex dynamics of influence diffusion, (2) Designing optimization objectives is inherently difficult due to the combinatorial explosion associated with solving this problem. To address these challenges, we propose a novel framework, DeepSN. Our framework employs sheaf neural diffusion to learn diverse influence patterns in a data-driven, end-to-end manner, providing enhanced separability in capturing diffusion characteristics. We also propose an optimization technique that accounts for overlapping influence between vertices, significantly reducing the search space and facilitating the identification of the optimal seed set efficiently. Finally, we conduct extensive experiments on both synthetic and real-world datasets to demonstrate the effectiveness of our framework.

Code — https://github.com/Aselahp/DeepSN
Extended version — https://arxiv.org/abs/2412.12416

Introduction

Influence maximization (IM) is a challenging network science problem that involves identifying a set of vertices which, when activated, maximize the spread of influence across the network. IM has significant real-world applications including viral marketing (Li, Lai, and Lin 2009; Kempe, Kleinberg, and Tardos 2003), disease control (Marquetoux et al. 2016), social media content management (Hosseini-Pozveh, Zamanifar, and Naghsh-Nilchi 2017), and crisis communication (Fan, Jiang, and Mostafavi 2021). Despite decades of research, IM still remains challenging primarily due its exponentially large search space and the complex nature of the influence diffusion processes.

A plethora of traditional methods have been proposed to obtain optimal or near-optimal solutions for IM (Kempe, Kleinberg, and Tardos 2003; Leskovec et al. 2007; Wang et al. 2010; Tang, Shi, and Xiao 2015; Li et al. 2019). These methods vary in strategy and effectiveness: some offer theoretical performance guarantees, while others use heuristics to enhance scalability. A common trait among these traditional approaches is their reliance on the explicit specification of the influence diffusion model as input. Recently, researchers have turned to learning-based methods focusing on their ability to automatically identify the diffusion model from the ground truth and accommodate multiple diffusion models, thereby achieving broader applicability (Li et al. 2023).

Learning-based approaches for IM encounter two key challenges. *(1) Effectively modeling the underlying diffusion model from ground truth:* There is a growing interest in leveraging Graph Neural Networks (GNNs) for modeling influence propagation, due to their ability to capture structural insights in networks (Kumar et al. 2022; Ling et al. 2023; Panagopoulos et al. 2023). However, existing work relies on traditional GNN techniques like attention (Lee et al. 2019) and convolution (Zhang et al. 2019). While these traditional GNNs excel in tasks with a static nature such as vertex classification and graph regression, their inductive bias and inherent assumptions limit their ability to model the dynamic behaviors of influence diffusion. They also suffer from issues like over-smoothing (Chen et al. 2020) which hampers their capacity to capture global information and long-range dependencies crucial for influence diffusion (Xia et al. 2021). *(2) Identifying the optimal subset of vertices with maximal Influence:* Designing an optimization objective to select the optimal seed set is arduous due to the vast search space, especially in large networks. Existing learning-based approaches often use deep reinforcement learning (Li et al. 2022; Chen et al. 2023) and ranking-based methods (Kumar et al. 2022; Panagopoulos, Malliaros, and Vazirgianis 2020) to approximate the optimal seed set. However, these methods often incur significant computational costs and suffer from a lack of interpretability.

Our work addresses the limitations of existing approaches by leveraging sheaf theory (Tennison 1975; Bredon 2012), an algebraic-topological framework that captures the topological and geometric properties of complex networks, es-

sential for understanding dynamic processes. Current sheaf diffusion GNNs (Hansen and Gebhart 2020; Bodnar et al. 2022; Barbero et al. 2022) are limited by their homogeneous diffusion behaviors and inability to handle the evolving dynamics crucial for influence propagation. To tackle this, we propose *DeepSN*, a novel sheaf GNN framework designed to address the influence maximization problem. Our architecture effectively estimates diverse influence propagation processes by learning adaptive structural relationships between vertices, enabling effective identification of influence patterns in the network. Building on this, we design a seed set inference mechanism that uses subgraphs to efficiently reduce the search space in influence maximization. Our contributions are summarized as follows.

- **Influence Diffusion:** We redefine influence propagation as a sheaf diffusion-reaction process, effectively capturing the complex dynamics present in real-world information diffusion models.
- **GNN Architecture:** We propose a novel GNN architecture rooted in sheaf theory, specifically designed to capture the unique nuances of influence propagation.
- **Seed Selection:** We employ a subgraph-based strategy that leverages structural insights from our sheaf GNN to minimize overlapping influence among seed vertices, reducing the search space for influence maximization.
- **Experiments:** We conduct rigorous evaluations of our framework on real-world and synthetic datasets, showcasing its superior performance and generalizability compared to state-of-the-art methods.

Related Work

Influence Maximization Methods IM methods generally fall into two main categories: traditional and learning-based. Traditional approaches include simulation-based, proxy-based, and sketch-based methods (Li et al. 2018). Simulation-based methods rely on Monte Carlo simulations for stochastic evaluation with theoretical guarantees, while proxy-based methods use heuristics for efficient seed set approximation. Sketch-based methods combine both simulation and proxy methods, balancing theoretical guarantees and computational efficiency. For detailed discussions on these traditional methods, see the surveys by Li et al. (2018) and Banerjee, Jenamani, and Pratihar (2020).

Contrary to traditional methods that require a specific diffusion model as input, learning-based models excel in generalizability, adapting to multiple diffusion models. Several studies have leveraged deep reinforcement learning for this problem (Li et al. 2022; Ma et al. 2022; Wang et al. 2021; Chen et al. 2023), and some works explored the use of GNNs in this context (Xia et al. 2021; Kumar et al. 2022; Ling et al. 2023; Panagopoulos et al. 2023).However, reinforcement learning methods often face scalability challenges due to exploration complexity, limiting their application in large-scale networks. Meanwhile, GNNs typically rely on traditional architectures that struggle to capture dynamic diffusion phenomena and mainly focus on progressive models, limiting their adaptability to non-progressive scenarios. Our work breaks from these limitations by incorporating an inductive bias guided by diffusion-reaction processes, which is capable of modeling both information diffusion and intrinsic transformations that occur during the diffusion process. This allows us to effectively model complex propagation patterns in both progressive and non-progressive spread dynamics.

Diffusion GNNs A diffusion process typically refers to the spread of information across a structure over time. In graph representation learning, diffusion manifests in various areas, including graph generation (Liu et al. 2023) and information propagation (Khoshraftar and An 2024). Generative diffusion GNNs (Niu et al. 2020; Bao et al. 2022) use diffusion processes to learn graph distributions and generate new graphs through iterative, learned operations. On the other hand, information propagation-based diffusion GNNs (Gasteiger, Weißenberger, and Günnemann 2019; Chamberlain et al. 2021; Zhao et al. 2021) model the propagation of information in graphs by discretizing an underlying partial differential equation. Sheaf theory was integrated into diffusion GNNs by Bodnar et al. (2022). Subsequently, several studies have adopted sheaf-based GNNs for a variety of downstream tasks (Duta et al. 2024; Caralt et al. 2024; Nguyen et al. 2024). Reaction-diffusion models improve diffusion by adding regularization. Several diffusion-based GNNs (Wang et al. 2022; Choi et al. 2023; Eliasof, Haber, and Treister 2024) incorporate reaction terms as constraints to prevent oversmoothing and preserve the distinctiveness of vertex features.

Our work fundamentally differs from existing diffusion models. While generative diffusion models focus on global graph generation, we model influence propagation as a dynamic process within graphs. Traditional diffusion GNNs often treat propagation as a uniform process, focused on influence spread, overlooking individual vertex dynamics and neighboring transitions. In contrast, we incorporate reaction terms into sheaf structures to capture evolving dynamics and vertex transitions, setting our method apart from models that primarily target static tasks.

Problem Formulation

Let $G = (V, E)$ be a graph with the vertex set V, the edge set E, $|V| = n$ and $|E| = m$. We denote $A \in \{0, 1\}^{n \times n}$ to be the adjacency matrix of G. The set of neighboring vertices for vertex v is denoted as $N(v) = \{u \in V \mid (u, v) \in E\}$.

Influence Maximization Influence maximization is a graph optimization problem that aims to identify a subset of vertices, known as the *seed set* (i.e., initially activated vertices), in a graph to maximize influence spread according to a specified diffusion function. We formally define it below.

Definition 1 (Influence Maximization Problem)**.** *Given a graph $G = (V, E)$, the* influence maximization *(IM) problem is to find a subset $S^\star \subseteq V$ of up to k vertices that maximizes an expected influence diffusion function σ. More precisely,*

$$S^\star = \underset{|S| \leq k}{\operatorname{argmax}}\, \sigma(S, G; \theta).$$

Here, $S^\star$ is referred to as the optimal seed set and $\sigma(S, G; \theta)$ represents the expected influence diffusion of the seed set S

in the graph G (that is, the expected final number of activations) with model parameters θ.

The influence diffusion function $\sigma(\cdot;\theta)$ can be instantiated with various models to capture different dynamics of influence spread. In the Linear Threshold (LT) model (Granovetter 1978), θ denotes vertex thresholds and edge weights, with vertices becoming active when the weighted sum of neighbors' influences exceeds their thresholds. In the Independent Cascade (IC) model (Goldenberg, Libai, and Muller 2001), θ represents edge activation probabilities, where active vertices have a single chance to activate each neighbor. The Susceptible-Infected-Susceptible (SIS) model (d'Onofrio 2008) extends this by incorporating infection and recovery rates in θ, allowing vertices to transition between susceptible and infected states over time.

Our Work In this paper, we introduce a deep learning framework called *DeepSN* to address the influence maximization problem. Our framework comprises two phases: *learning to estimate influence* and *optimizing seed selection*. In the influence estimation phase, our goal is to measure the total influence exerted by a set of initially activated vertices in a learnable way. This estimated influence is then used in the seed selection optimization phase to identify the optimal seed set that maximizes overall influence. Fig. 1 provides an overview of our framework. Note that all eliminated proofs from the main content, due to space constraints, are provided in the appendix.

Learning to Estimate Influence

We introduce a novel sheaf GNN model based on sheaf theory, specifically designed to adapt to the complex diffusion patterns inherent in influence propagation.

Topological Sheaf Diffusion

We begin by introducing the concept of *cellular sheaf*, the foundational building block of our GNN.

Definition 2 (Cellular Sheaf)**.** *A* (cellular) sheaf (G, F) *on a graph $G = (V, E)$ consists of a vector space F_v for each vertex $v \in V$, a vector space F_e for each edge $e \in E$, a linear map $\mathcal{F}_{v \trianglelefteq e} : \mathcal{F}_v \to \mathcal{F}_e$ for each incident vertex-edge pair $v \trianglelefteq e$.*

$\mathcal{F}_v$ and $\mathcal{F}_e$ are the *vertex sheaf* and *edge sheaf*, respectively, while $\mathcal{F}_{v \trianglelefteq e}$ is the *transformation map*. Let $\bigoplus$ denote the direct sum of vector spaces. The space of *0-cochains* is the direct sum of vector spaces over the vertices: $C^0(G, F) = \bigoplus_{v \in V} F_v$, and the space of *1-cochains* is the direct sum of vector spaces over the edges: $C^1(G, F) = \bigoplus_{e \in E} F_e$.

We formulate influence propagation in a network as an opinion propagation process, where each vertex corresponds to a vertex sheaf, representing a private opinion. Edge sheaves represent public opinions related to influence propagation, while transformation maps translate information from vertex sheaves to edge sheaves, extracting public opinions from private opinions. Let $x_v \in F_v$ be a d-dimensional feature vector for each $v \in V$. The *coboundary map* $\delta : C^0(G, F) \to C^1(G, F)$ is a linear transformation, defined for an edge (v, u) as

$$\delta(x)_e = \mathcal{F}_{v \trianglelefteq e}.x_v - \mathcal{F}_{u \trianglelefteq e}.x_u \tag{1}$$

where $\mathcal{F}_{v \trianglelefteq e} \cdot x_v$ and $\mathcal{F}_{u \trianglelefteq e} \cdot x_u$ represent the public opinions associated with e, originating from vertices v and u, respectively.

The sheaf Laplacian of a vertex measures the disparity between its public opinion and that of its neighbors. To capture the fine-grained nuances of trust, a crucial factor in modeling influence propagation, we incorporate learnable sheaf coefficients to model each vertex's confidence in the opinions received from its neighbors.

Definition 3 (Non-linear Sheaf Laplacian)**.** *Let $\psi_{vu} \in [0, 1]$. The sheaf Laplacian $L_{\mathcal{F}} : C^0(G; F) \to C^0(G; F)$ on a sheaf (G, F) is defined vertex-wise as*

$$L_{\mathcal{F}}(x_v) = \sum_{v,u \trianglelefteq e} \mathcal{F}^T_{v \trianglelefteq e}.\psi_{vu}.\left(\mathcal{F}_{v \trianglelefteq e}.x_v - \mathcal{F}_{u \trianglelefteq e}.x_u\right). \tag{2}$$

Here, ψ_{vu} is the sheaf coefficient which is learnable. Note that the sheaf Laplacian employed by Bodnar et al. (2022) is a specific instance of our Laplacian, where $\psi_{u,v} = 1$ for all $(u, v) \in E$.

To ensure that the Laplacian matrix $\hat{L}_{\mathcal{F}}$ remains positive definite, a necessary condition for the convergence of the diffusion process, we apply the following modification:

$$\hat{L}_{\mathcal{F}} = L_{\mathcal{F}} + \epsilon \cdot I \tag{3}$$

where ϵ is a scalar, I is the identity matrix, and $\cdot$ denotes element-wise multiplication. The following lemma establishes the necessary and sufficient conditions for $\hat{L}_{\mathcal{F}}$.

Lemma 1. *Let λ_{min} denote the smallest eigenvalue of $L_{\mathcal{F}}$. $\hat{L}_{\mathcal{F}}$ is positive definite if and only if $\epsilon > -\lambda_{min}$.*

The sheaf diffusion operator (i.e., normalized sheaf Laplacian) models opinion diffusion by capturing discrepancies between vertex opinions and enabling smooth information propagation across the network, defined as

$$\Delta_{\mathcal{F}} = D^{-\frac{1}{2}} L_{\mathcal{F}} D^{-\frac{1}{2}}, \tag{4}$$

where D is the block-diagonal of $L_{\mathcal{F}}$.

Let $x \in C^0(G, F)$ denote an nd-dimensional vector constructed by column-stacking the individual vectors x_v. This vector x is then transformed into a vertex feature matrix $X \in \mathbb{R}^{(nd) \times f}$, where each column corresponds to a vector in $C^0(G; F)$ and f is the number of feature channels. The sheaf standard diffusion process is defined by the following Partial Differential Equation (PDE),

$$X(0) = X, \quad \frac{\partial X(t)}{\partial t} = -\Delta_{\mathcal{F}} X(t). \tag{5}$$

Given a diffusion coefficient $\alpha \in [0, 1]$, the discrete-time update rule for diffusion is defined as

$$X(t+1) = X(t) - \alpha \cdot \frac{\partial X(t)}{\partial t}; \alpha > 0. \tag{6}$$

When vertex features do not change between time steps, (i.e., $X(t+1) = X(t)$), a fixed point $X(t)$ is reached, which is also referred to as a *steady state*.

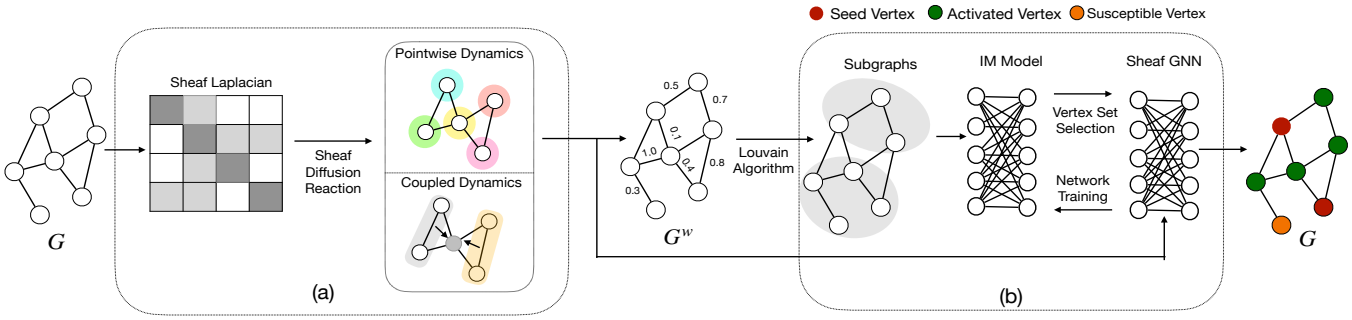

Figure 1: The DeepSN framework consists of two phases: (a) learning to estimate influence with sheaf GNN; b) optimizing seed selection using the subgraphs, IM model, and trained sheaf GNN.

Sheaf Reaction Diffusion

Traditional diffusion GNNs including sheaf diffusion in Eq. (5) (Bodnar et al. 2022; Hansen and Gebhart 2020) use a uniform diffusion process that primarily focuses on influence spreading but overlooks how individual vertex characteristics and neighboring state transitions shape the process, which is essential for modeling influence diffusion. To address this issue, we model influence dynamics as a diffusion-reaction process (Turing 1990; Choi et al. 2023). In this process, *diffusion* refers to the spread of information across the network, governed by its connectivity structure, and *reaction* involves altering this information based on interactions and inherent characteristics of vertices within the network.

Reaction Operators To capture complex dynamics related to influence diffusion, we introduce the *sheaf reaction diffusion* which extends the sheaf standard diffusion with two reaction operators for *pointwise dynamics* and *coupled dynamics*. That is,

$$\frac{\partial X_v(t)}{\partial t} = \underbrace{-\alpha \Delta_{\mathcal{F}} X_v(t)}_{\text{Diffusion}} + \underbrace{\beta A\left(X_v(t)\right)}_{\text{Pointwise Dynamics}} + \underbrace{\gamma R_v\left(X(t), A, S^t\right)}_{\text{Coupled Dynamics}} \tag{7}$$

Here, $X_v(t)$ represents the feature vector of vertex v at time t. α, β, and γ control the contribution of each term, and $S^t \in [0,1]^n$ is a vector representing the activation probabilities of vertices at time step t.

Pointwise Dynamics Pointwise dynamics refers to inherent characteristics that govern its activation or susceptibility. For instance, in models of opinion dynamics or epidemic propagation, a vertex may alter its activation state due to internal factors such as personal reassessment or spontaneous recovery, independent of its interactions with other vertices. Thus, we design a reaction operator to capture such vertex evolution as

$$A\left(X_v(t)\right) = \Phi_v^1 \odot \frac{X_v(t)}{\kappa_v^1 + |X_v(t)|}$$

where $\Phi_v^1, \kappa_v^1 \in \mathbb{R}^{d \times f}$ are coefficient vectors with $\kappa_v^1 > 0, \kappa_v^1 \neq -|X_v(t)|$, and $\odot$ is the Hadamard product.

Coupled Dynamics Coupled dynamics describe how neighboring interactions of vertices influence the activation or susceptibility of a vertex, creating a dynamic interplay between individual behaviors and network-wide interactions. We define a reaction operator for vertex $v \in V$ as

$$X_v^\dagger(t) = \sum_{u \in N(v)} \left(S_u^t \cdot X_u(t)\right) - \sum_{u \in N(v)} \left((1 - S_u^t) \cdot X_u(t)\right);$$

$$R_v\left(X(t), A, S^t\right) = \Phi_v^2 \odot \frac{X_v^\dagger(t)}{\kappa_v^2 + |X_v^\dagger(t)|}$$

where Φ_v^2 and κ_v^2 are coefficient vectors with $\kappa_v^2 > 0, \kappa_v^2 \neq -|X_v^\dagger(t)|$ for any $t > 0$, and S_u^t denotes the activation probability of vertex u at time t. This reaction operator accounts for the combined impact of both activated and susceptible neighbors by producing either positive or negative effects: a high activation level in the neighborhood yields a positive impact, whereas a high susceptibility leads to a negative impact. Thus, it can enhance the model's ability to capture complex dynamics in a network.

Remark 1. *Recent research has largely focused on progressive models (Chen, Castillo, and Lakshmanan 2022; Chen et al. 2022), where vertices remain indefinitely active once activated. However, this is often unrealistic, as many real-world propagation models involve vertices transitioning between active and inactive states (Lou et al. 2014). While diffusion terms typically follow fixed-point dynamics, our sheaf diffusion model uses reaction operators to introduce non-monotonic, adaptive, and oscillatory behaviors, capturing dynamic vertex state transitions.*

Stability of Reaction Diffusion Stability in a diffusion process refers to its ability to maintain a controlled and predictable behavior over time (Wang et al. 2022). A stable process maintains vertex features within a bounded region over time. This bounded behavior allows the diffusion process to converge to or remain near a fixed point. If vertex features become unbounded, it signals instability, indicating that the diffusion process can deviate significantly from a fixed point or expected behavior (Bellman 1947; Sastry 2013). Such instability can lead to unpredictable outcomes and make it difficult to control or interpret diffusion dynamics effectively.

In the following, we explain the bounded behavior of the reaction operators.

Lemma 2. *The reaction operators in Eq.* (7) *are bounded for all $t > 0$ and $v \in V$ in terms of norms $||\cdot||$. We have $\|A(X_v(t))\| < \|\Phi_v^1\|$ and $\|R_v(X(t), A, S^t)\| < \|\Phi_v^2\|$.*

The boundedness of the reaction operators ensures that the fixed-point of Eq. (7) is also bounded, as demonstrated in the lemma below.

Lemma 3. *Let $X^\star$ denote the fixed point of Eq.* (7)*. Given that $\Delta_{\mathcal{F}}$ is well-defined (i.e., positive definite) and operates on a finite-dimensional space, $X^\star$ is bounded.*

Sheaf GNN Training

Building on the sheaf Laplacian and reaction operators, our GNN applies the following diffusion propagation rule to update vertex features:

$$\begin{aligned} X_v(t+1) = X_v(t) - \alpha\Big(\Delta_{\mathcal{F}}\left(I_n \otimes W_1^t\right) X_v(t) W_2^t\Big) \\ + \beta\left(\Phi_v^1 \odot \frac{X_v(t)}{\kappa_v^1 + X_v(t)}\right) \\ + \gamma\left(\Phi_v^2 \odot \frac{X_v^\dagger(t)}{\kappa_v^2 + |X_v^\dagger(t)|}\right) \end{aligned} \tag{8}$$

Notably, $W_1^t \in \mathbb{R}^{d\times d}$, $W_2^t \in \mathbb{R}^{f\times f}$, and $\Phi_v^1, \Phi_v^2, \kappa_v^1, \kappa_v^2 \in \mathbb{R}^{d\times f}$ are learnable weight matrices, $\otimes$ denotes the Kronecker product, and $I_n \in \mathbb{R}^{n\times n}$ represents an identity matrix. Moreover, $X(0)$ is obtained by transforming the vertex features of the network through a multi-layer perceptron (MLP), followed by reshaping, resulting in a matrix of dimensions $(nd) \times f$.

After obtaining the updated vertex embeddings in each iteration, we utilize a non-linear neural function $f_\eta(\cdot)$, parameterized by η, to determine the activation state of each vertex $v \in V$ at the time step $t+1$ as

$$S_v^{t+1} = f_\eta\Big(X_v(t+1)\Big); S_v^{t+1} \in [0,1]. \tag{9}$$

We use the Mean Square Error (MSE) loss to measure the difference between the predicted activation probabilities $\hat{S}$ and the ground truth probabilities $Y \in [0,1]^n$ as:

$$\mathcal{L}_{train} = ||\hat{S} - Y||_2^2 \tag{10}$$

Optimizing Seed Selection

In this section, we present a learning-based approach to optimally select a seed set that maximizes influence spread. Selecting an optimal seed set for influence maximization faces a combinatorial explosion, with vertex combinations growing exponentially as the graph size increases.

We address this challenge leveraging two observations. First, instead of relying on the adjacency matrix for graph connectivity, which fails to capture network dynamics in diffusion models, we learn the connectivity through sheaf coefficients, enhancing model flexibility. This leads to a weighted graph G^w that is constructed from sheaf coefficients and the adjacency matrix of the input graph. Then, we employ a partitioning mechanism to identify subgraphs, which reduces the search space in the process of determining the optimal seed set. We apply the Louvain algorithm (Blondel et al. 2008; Dugué and Perez 2015; Traag, Waltman, and Van Eck 2019) to divide the graph G^w into r subgraphs $\{G_i\}_{i=1}^r$, minimizing the overlap of influence between vertices in different subgraphs. Then, allocating seed vertices across subgraphs can significantly reduce the search space by limiting the number of vertex combinations within each smaller subgraph, rather than across the entire graph.

We train a neural network $\mathcal{T}_\phi$, parameterized by ϕ, to select seed vertices within subgraphs in a learnable manner. More specifically, $\mathcal{T}_\phi$ learns to select $S_i \subseteq V_i$ seed vertices from each subgraph $G_i = (V_i, E_i)$ such that $S = \bigcup_{i=1}^k S_i$ can maximize the overall influence spread

$$S^* = \mathcal{T}_\phi\Big(\{G_i\}_{i=1}^r, G^w\Big) \text{ subject to } \bigwedge_{i\in[1,r]} |S_i| \le \frac{k}{n}|V_i|.$$

The condition in the above equation ensures that the number of seeds is chosen proportionally to the size of each subgraph. $\mathcal{T}_\phi$ is trained using a loss function based on the difference between the maximal influence and the predicted influence

$$\mathcal{L}_{train} = n - \sigma\Big(\bigcup_{i=1}^r S_i, G^w; \theta\Big). \tag{11}$$

Note that we use the sheaf GNN with trained parameters θ to approximate the influence diffusion function $\sigma(\cdot)$.

Complexity Analysis

We first analyze the layer-wise complexity of our GNN component. The diffusion operation within our GNN has a complexity of $O\left(n\left(f^2d^2 + d^3\right) + m\left(fd + d^3\right)\right)$, where m is the number of edges. The complexity of this operation is similar to that reported in Bodnar et al. (2022). Each reaction operator introduces an additional complexity of $O(ndf)$. Consequently, the total complexity of our GNN is $O\left(n\left(f^2d^2 + d^3\right) + m\left(fd + d^3\right)\right)$. Given that we employ $d = \{1, 2\}$ in our experiments, our GNN incurs only a constant overhead compared to traditional GNNs, such as GCN (Kipf and Welling 2016).

The Louvain algorithm, used as a preprocessing step, has a time complexity of $O(lm)$, where l is the number of iterations, and a space complexity of $O(n + m)$ (Lancichinetti and Fortunato 2009). In our experiments, we employ $\text{DeepSN}_{\text{SP}}$ with a sparsified graph structure, resulting in a time complexity of $O(l \times n \log n)$ and a space complexity of $O(n)$. We implement $\mathcal{T}_\phi$ using a multi-layer perceptron (MLP) with a time complexity of $O(ndh)$, where h represents the number of hidden neurons in the MLP.

Vertex Feature Separability

We investigate the separation power of our sheaf GNN for vertex features, which plays a crucial role in mitigating oversmoothing (Bodnar et al. 2022). The following proposition

Methods	Cora-ML (IC)				Network Science (IC)				Power Grid (IC)				Cora-ML (LT)				Network Science (LT)				Power Grid (LT)			
	1%	5%	10%	20%	1%	5%	10%	20%	1%	5%	10%	20%	1%	5%	10%	20%	1%	5%	10%	20%	1%	5%	10%	20%
IMM	8.1	26.2	37.3	50.2	5.2	16.8	27.0	45.7	5.6	17.4	31.5	51.1	1.7	34.8	52.2	66.4	2.5	11.8	18.1	33.6	4.6	19.9	31.7	56.9
OPIM	13.4	26.9	37.4	50.9	6.4	19.4	28.9	48.6	5.7	17.7	29.7	50.1	2.3	36.9	51.2	71.5	1.6	12.0	18.1	34.1	4.4	21.6	29.4	55.5
SubSIM	10.1	25.7	36.8	51.1	4.8	15.4	27.9	44.8	4.6	19.2	31.7	50.2	1.7	33.6	54.7	70.1	1.8	10.4	19.2	34.1	4.5	21.1	31.2	57.4
IMINfECTOR	9.6	26.8	37.7	50.6	5.4	17.9	27.8	47.6	5.4	18.2	31.6	50.9	2.1	33.9	51.3	70.6	2.1	11.8	18.7	34.5	4.2	21.3	31.6	56.2
PIANO	9.8	25.2	37.4	51.1	5.3	18.1	27.1	47.2	5.3	18.1	31.7	50.2	2.1	33.5	53.3	69.8	2.1	11.3	19.1	33.9	4.3	21.3	31.4	57.1
ToupleGDD	10.6	27.5	38.5	51.5	6.3	17.8	28.3	50.5	5.4	19.3	31.6	51.3	2.3	36.2	54.5	70.9	2.8	12.4	19.8	34.6	4.8	21.9	32.6	58.1
DeepIM	14.1	28.1	39.6	52.4	7.8	20.9	31.5	51.2	6.3	21.0	32.5	52.4	**13.4**	**69.2**	**83.5**	94.1	**4.1**	**16.6**	26.7	41.5	6.3	24.4	46.8	71.7
DeepSN	11.5	25.6	40.9	52.8	6.2	**22.0**	**32.0**	**52.4**	6.4	**24.0**	36.9	**61.0**	7.4	40.7	68.2	**95.3**	2.9	14.4	25.3	**52.0**	**6.3**	**24.6**	**47.2**	**73.2**
DeepSN$_{SP}$	**14.2**	**30.1**	**42.3**	**58.4**	**7.9**	18.6	30.0	51.2	**7.1**	22.4	**37.4**	57.4	7.9	46.4	72.8	93.8	3.6	15.5	**27.8**	51.8	5.2	23.8	40.3	68.1

Methods	Cora-ML (SIS)				Network Science (SIS)				Power Grid (SIS)				Jazz (SIS)				Random (SIS)				Digg (SIS)			
	1%	5%	10%	20%	1%	5%	10%	20%	1%	5%	10%	20%	1%	5%	10%	20%	1%	5%	10%	20%	1%	5%	10%	20%
IMM	2.0	9.5	15.4	27.6	1.3	5.6	12.2	22.1	1.1	5.6	11.0	22.9	7.6	37.8	55.6	67.1	2.7	12.6	20.9	37.7	2.5	9.4	16.3	32.6
OPIM	2.3	9.3	16.2	27.2	1.4	5.9	13.0	22.1	1.2	5.9	11.1	22.4	8.2	35.1	56.8	68.3	2.8	12.5	20.2	36.1	2.3	9.3	16.5	32.3
SubSIM	2.3	9.2	16.9	28.8	1.5	5.6	12.2	23.3	1.2	5.6	11.4	21.9	2.9	30.1	53.8	67.0	3.2	14.4	24.5	39.1	2.5	9.5	16.1	32.3
IMINfECTOR	2.1	9.4	16.1	27.9	1.7	5.8	12.4	22.3	1.3	5.5	10.2	23.1	8.8	35.4	54.8	66.2	2.5	12.4	20.5	36.6	2.3	9.1	16.4	32.4
DeepIM	7.1	16.1	21.9	30.8	2.7	8.7	15.1	25.1	1.9	7.6	13.3	23.8	27.1	57.1	68.1	74.1	3.2	14.4	24.5	39.1	5.6	11.4	18.8	**36.3**
DeepSN	12.8	24.7	34.2	46.5	2.0	9.6	16.1	28.1	**2.4**	**9.8**	**15.7**	25.3	**34.3**	**64.9**	**75.6**	**85.8**	**5.2**	**24.2**	**37.6**	**54.1**	**16.1**	20.2	**24.7**	33.2
DeepSN$_{SP}$	**16.8**	**29.9**	**37.9**	**46.9**	**2.7**	**9.9**	**17.4**	**29.0**	2.1	8.2	14.9	**26.3**	35.2	58.3	73.4	82.4	4.1	18.7	32.8	52.6	16.1	**20.3**	23.6	33.1

Table 1: Performance of DeepSN variants for influence maximization, compared to baseline methods, under IC, LT, and SIS models. The best results are highlighted in **bold**. Baseline results are sourced from Ling et al. (2023).

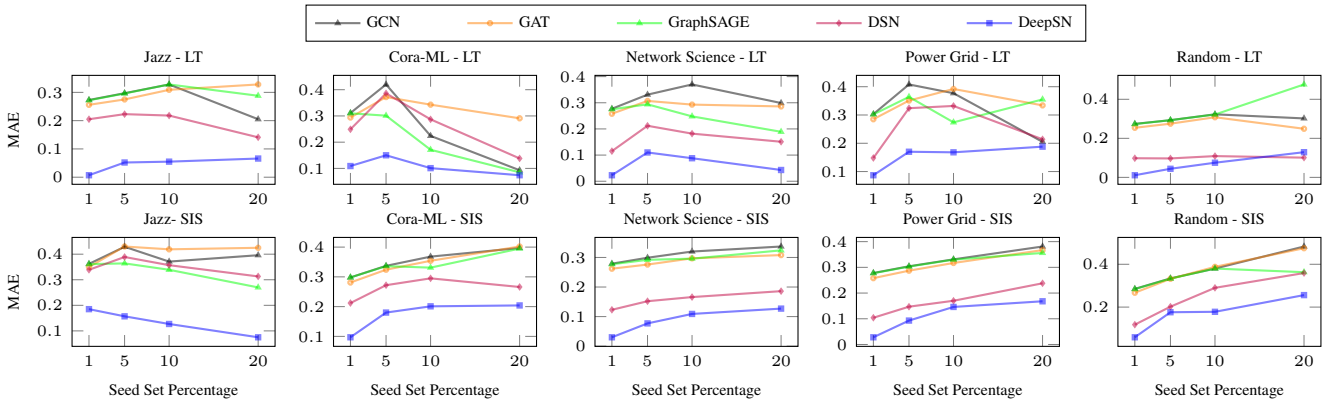

Figure 2: Performance of DeepSN for influence estimation in terms of MAE (Mean Absolute Error), compared to baseline methods. Results under IC model are provided in the appendix.

shows the existence of the fixed point in the sheaf diffusion process under Eq. (5) and the corresponding properties of transformation maps.

Proposition 1. *For any graph $G = (V, E)$ with sheaf Laplacian $L_{\mathcal{F}}$ and initial vertex features $X(0)$, there exists a unique fixed-point $X(t)$ in the sheaf diffusion process such that for any $(v, u) \in E$, $\mathcal{F}_{v \trianglelefteq e} x_v = \mathcal{F}_{u \trianglelefteq e} x_u$ holds.*

Due to the condition $\mathcal{F}_{v \trianglelefteq e} x_v = \mathcal{F}_{u \trianglelefteq e} x_u$ at the fixed point, the sheaf diffusion process has limited capacity to separate distinct vertex features between a vertex v and its neighbors u, as shown in Proposition 2.

Proposition 2. *There exists a graph $G = (V, E)$ with the fixed point $X(t)$ in the sheaf diffusion process which satisfies $x_u = x_v$ for at least one $(u, v) \in E$.*

Unlike traditional sheaf diffusion, the sheaf reaction diffusion (Eq. (7)) does not require the condition $\mathcal{F}_{v \trianglelefteq e} x_v = \mathcal{F}_{u \trianglelefteq e} x_u$ to be satisfied upon convergence. The following proposition demonstrates this.

Proposition 3. *For any graph $G = (V, E)$ with sheaf Laplacian $L_{\mathcal{F}}$ and initial vertex features $X(0)$, there exists a unique fixed-point $X(t)$ in the sheaf reaction diffusion process; however, $\mathcal{F}_{v \trianglelefteq e} x_v = \mathcal{F}_{u \trianglelefteq e} x_u$ does not necessarily hold for each $(v, u) \in E$.*

Remark 2. *Relaxation of condition $\mathcal{F}_{v \trianglelefteq e} x_v = \mathcal{F}_{u \trianglelefteq e} x_u$ in Proposition 3 significantly improves the separability of distinct vertex features. This allows sheaf GNN to learn vertex features distinguishable from their neighbors through more powerful transformation maps, offering an advantage over existing sheaf diffusion networks (Bodnar et al. 2022).*

Experiments

A set of experiments was conducted to evaluate the performance of DeepSN, considering two variants: *DeepSN* and *DeepSN$_{SP}$*, a computationally efficient variant that uses sheaf coefficients to sparsify the graph structure. We focus on three diffusion models: IC, LT, and SIS (Li et al. 2018). IC and LT are progressive, while SIS is non-progressive.

Datasets We evaluate DeepSN against other methods using a diverse set of datasets, including five real-world datasets (Jazz (Rossi and Ahmed 2015), Network Science

(Rossi and Ahmed 2015), Cora-ML (McCallum et al. 2000), Power Grid (Rossi and Ahmed 2015), and Digg (Lerman and Galstyan 2008)) and one synthetic dataset (Random (Ling et al. 2023)), which range from small graphs to those with over 250,000 vertices.

Experimental Setups and Baselines We follow the experimental setup of Ling et al. (2023) for our influence maximization task, comparing DeepSN with traditional baselines: IMM (Tang, Shi, and Xiao 2015), OPIM (Tang et al. 2018), SubSIM (Guo et al. 2020) and learning-based methods: IMINFECTOR (Panagopoulos, Malliaros, and Vazirgiannis 2020), ToupleGDD (Chen et al. 2023), PIANO (Li et al. 2022), DeepIM (Ling et al. 2023).

Additional details on dataset statistics, experimental setups, and model hyperparameters are in the appendix.

Exp–1. Performance of DeepSN We evaluate DeepSN's performance in selecting an optimal seed set for the IM task across various budget constraints $\{1\%, 5\%, 10\%, 20\%\}$ (i.e. seed set size as a percentage of total number of vertices). The results are demonstrated in Table 1. We observe that both DeepSN and DeepSN$_{SP}$ outperform or deliver comparable performance across all diffusion models. Notably, DeepSN achieves a substantial improvement over all baseline methods for the SIS diffusion model. This enhanced performance is attributed to DeepSN's capability to effectively capture complex diffusion dynamics inherent to non-progressive diffusion models. The experimental results for more datasets are provided in the appendix.

Exp–2. Ablation Study: Sheaf GNN In Figure 2, we compare the influence estimation performance of DeepSN with several widely used GNNs. The results show that DeepSN significantly outperforms traditional GNNs, such as GCN (Kipf and Welling 2016), GAT (Veličković et al. 2018), and GraphSAGE (Hamilton, Ying, and Leskovec 2017), across all diffusion models and datasets, highlighting its superiority in influence estimation. These traditional GNNs often struggle to capture long-range dependencies due to oversmoothing, leading to lower performance. Furthermore, while existing sheaf neural networks like DSN (Bodnar et al. 2022) address some issues, they still struggle to capture the intricate dynamics needed for influence estimation, leading to suboptimal performance.

Exp–3. Ablation Study: IM Model We evaluate the effectiveness of our IM model $\mathcal{T}_\phi$ by replacing it with different IM variants. These variants include: **DeepSN-CELF**, which incorporates the CELF algorithm (Leskovec et al. 2007) as the IM component; **DeepSN-WC**, where seed vertices are optimized across the entire network as a single subgraph; and **DeepSN-WSA**, which uses the adjacency matrix for dividing the graph into subgraphs, instead of sheaf coefficients. The results in Table 2 show that our IM model outperforms other algorithms. Particularly, DeepSN and DeepSN$_{SP}$ consistently exceed the performance of DeepSN-WC and DeepSN-WSA. This is largely due to subgraph-based seed optimization and sheaf coefficients.

	IC		LT		SIS	
Methods	10%	20%	10%	20%	10%	20%
DeepSN-CELF	37.2	52.8	**76.0**	88.6	30.6	38.7
DeepSN-WC	39.2	50.4	46.3	88.6	25.3	35.9
DeepSN-WSA	40.0	50.6	48.0	89.6	26.0	36.0
DeepSN	40.9	52.8	68.2	**95.3**	34.2	46.5
DeepSN$_{SP}$	**42.3**	**58.4**	72.8	93.8	**37.9**	**46.9**

Table 2: Comparison of DeepSN with various influence maximization approaches for Cora-ML dataset.

Exp–4. Impact of Layer Depth and Feature Dimension We explore the effect of layer depth and feature dimension on the performance of DeepSN. Figure 3 illustrates how these factors influence the performance of DeepSN in selecting optimal seed vertices for Cora ML dataset, evaluated under the IC diffusion model with a 10% seed set.

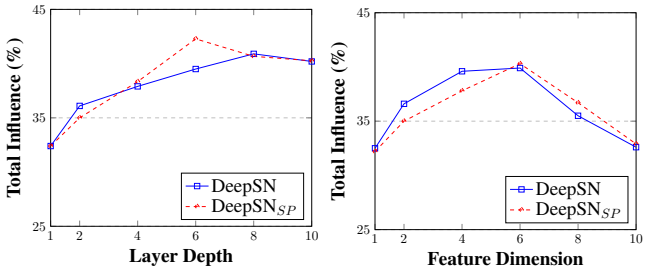

Figure 3: Impact of layer depth and feature dimension on DeepSN's performance

As expected, DeepSN's performance improves with greater layer depth, effectively capturing long-range interactions and demonstrating robustness against oversmoothing. However, performance only improves with increasing feature dimension up to a certain point. Beyond this, the added complexity, such as more learnable parameters, begins to outweigh the benefits of enhanced representational power, leading to a decline in performance.

Conclusion, Limitations and Future Work

In this work, we proposed a novel learning framework for the IM problem. Our approach integrates a GNN that harnesses sheaf theory to learn the underlying influence diffusion model in a data-driven manner, while effectively addressing the topological and dynamic complexities of propagation phenomena. Additionally, we proposed a subgraph-based maximization objective to identify the optimal seed set, thereby reducing the combinatorial search space inherent to the IM problem. The empirical results demonstrated the effectiveness of the proposed framework.

Currently, our framework only supports diffusion models with two states. A potential avenue for future work is to extend the framework to handle more complex diffusion models with more than two states, including multi-state threshold models and epidemic models with multiple stages.

Acknowledgments

This research was supported partially by the Australian Government through the Australian Research Council's Discovery Projects funding scheme (project DP210102273).

References

Banerjee, S.; Jenamani, M.; and Pratihar, D. K. 2020. A survey on influence maximization in a social network. *Knowledge and Information Systems*, 62.

Bao, F.; Zhao, M.; Hao, Z.; Li, P.; Li, C.; and Zhu, J. 2022. Equivariant energy-guided sde for inverse molecular design. In *international conference on learning representations*.

Barbero, F.; Bodnar, C.; de Ocáriz Borde, H. S.; Bronstein, M.; Veličković, P.; and Liò, P. 2022. Sheaf neural networks with connection laplacians. In *Topological, Algebraic and Geometric Learning Workshops*.

Bellman, R. 1947. On the boundedness of solutions of nonlinear differential and difference equations. *Transactions of the American Mathematical Society*, 62(3).

Blondel, V. D.; Guillaume, J.-L.; Lambiotte, R.; and Lefebvre, E. 2008. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10).

Bodnar, C.; Di Giovanni, F.; Chamberlain, B.; Lio, P.; and Bronstein, M. 2022. Neural sheaf diffusion: A topological perspective on heterophily and oversmoothing in gnns. *Advances in Neural Information Processing Systems*, 35.

Bredon, G. E. 2012. *Sheaf theory*, volume 170. Springer Science & Business Media.

Caralt, F. H.; Bernardez, G.; Duta, I.; Alarcon, E.; and Lio, P. 2024. Complex Diffusion Processes as an Inductive Bias in Sheaf Neural Networks. In *ICML Workshop on Geometry-grounded Representation Learning and Generative Modeling*.

Chamberlain, B.; Rowbottom, J.; Gorinova, M. I.; Bronstein, M.; Webb, S.; and Rossi, E. 2021. Grand: Graph neural diffusion. In *International conference on machine learning*.

Chen, D.; Lin, Y.; Li, W.; Li, P.; Zhou, J.; and Sun, X. 2020. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *AAAI conference on artificial intelligence*, volume 34.

Chen, T.; Yan, S.; Guo, J.; and Wu, W. 2023. ToupleGDD: A fine-designed solution of influence maximization by deep reinforcement learning. *IEEE Transactions on Computational Social Systems*, 11(2).

Chen, W.; Castillo, C.; and Lakshmanan, L. V. 2022. *Information and influence propagation in social networks*. Springer Nature.

Chen, W.; Peng, B.; Schoenebeck, G.; and Tao, B. 2022. Adaptive greedy versus non-adaptive greedy for influence maximization. *Journal of Artificial Intelligence Research*.

Choi, J.; Hong, S.; Park, N.; and Cho, S.-B. 2023. Gread: Graph neural reaction-diffusion networks. In *International Conference on Machine Learning*.

Dugué, N.; and Perez, A. 2015. *Directed Louvain: maximizing modularity in directed networks*. Ph.D. thesis, Université d'Orléans.

Duta, I.; Cassarà, G.; Silvestri, F.; and Liò, P. 2024. Sheaf hypergraph networks. *Advances in Neural Information Processing Systems*, 36.

d'Onofrio, A. 2008. A note on the global behaviour of the network-based SIS epidemic model. *Nonlinear Analysis: Real World Applications*, 9(4).

Eliasof, M.; Haber, E.; and Treister, E. 2024. Feature transportation improves graph neural networks. In *AAAI Conference on Artificial Intelligence*, volume 38.

Fan, C.; Jiang, Y.; and Mostafavi, . A. 2021. The role of local influential users in spread of situational crisis information. *Journal of Computer-Mediated Communication*, 26(2).

Gasteiger, J.; Weißenberger, S.; and Günnemann, S. 2019. Diffusion improves graph learning. *Advances in neural information processing systems*, 32.

Goldenberg, J.; Libai, B.; and Muller, E. 2001. Using complex systems analysis to advance marketing theory development: Modeling heterogeneity effects on new product growth through stochastic cellular automata. *Academy of Marketing Science Review*, 9(3).

Granovetter, M. 1978. Threshold models of collective behavior. *American journal of sociology*, 83(6).

Guo, Q.; Wang, S.; Wei, Z.; and Chen, M. 2020. Influence maximization revisited: Efficient reverse reachable set generation with bound tightened. In *ACM SIGMOD international conference on management of data*.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

Hansen, J.; and Gebhart, T. 2020. Sheaf neural networks. *arXiv preprint arXiv:2012.06333*.

Hosseini-Pozveh, M.; Zamanifar, K.; and Naghsh-Nilchi, A. R. 2017. A community-based approach to identify the most influential nodes in social networks. *Journal of Information Science*, 43(2).

Kempe, D.; Kleinberg, J.; and Tardos, É. 2003. Maximizing the spread of influence through a social network. In *Ninth ACM SIGKDD international conference on Knowledge discovery and data mining*.

Khoshraftar, S.; and An, A. 2024. A survey on graph representation learning methods. *ACM Transactions on Intelligent Systems and Technology*, 15(1).

Kipf, T. N.; and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Kumar, S.; Mallik, A.; Khetarpal, A.; and Panda, B. S. 2022. Influence maximization in social networks using graph embedding and graph neural network. *Information Sciences*.

Lancichinetti, A.; and Fortunato, S. 2009. Community detection algorithms: a comparative analysis. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 80(5).

Lee, J. B.; Rossi, R. A.; Kim, S.; Ahmed, N. K.; and Koh, E. 2019. Attention models in graphs: A survey. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 13(6).

Lerman, K.; and Galstyan, A. 2008. Analysis of social voting patterns on digg. In *workshop on Online social networks*.

Leskovec, J.; Krause, A.; Guestrin, C.; Faloutsos, C.; VanBriesen, J.; and Glance, N. 2007. Cost-effective outbreak detection in networks. In *13th ACM SIGKDD international conference on Knowledge discovery and data mining*.

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*, 10(3).

Li, X.; Smith, J. D.; Dinh, T. N.; and Thai, M. T. 2019. Tiptop:(almost) exact solutions for influence maximization in billion-scale networks. *IEEE/ACM Transactions on Networking*, 27(2).

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

Li, Y.; Gao, H.; Gao, Y.; Guo, J.; and Wu, W. 2023. A survey on influence maximization: From an ml-based combinatorial optimization. *ACM Transactions on Knowledge Discovery from Data*, 17(9).

Li, Y.-M.; Lai, C.-Y.; and Lin, C.-H. 2009. Discovering influential nodes for viral marketing. In *2009 42nd Hawaii International Conference on System Sciences*. IEEE.

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*.

Liu, C.; Fan, W.; Liu, Y.; Li, J.; Li, H.; Liu, H.; Tang, J.; and Li, Q. 2023. Generative diffusion models on graphs: methods and applications. In *Thirty-Second International Joint Conference on Artificial Intelligence*.

Lou, V. Y.; Bhagat, S.; Lakshmanan, L. V.; and Vaswani, S. 2014. Modeling non-progressive phenomena for influence propagation. In *ACM conference on Online social networks*.

Ma, L.; Shao, Z.; Li, X.; Lin, Q.; Li, J.; Leung, V. C.; and Nandi, A. K. 2022. Influence maximization in complex networks by using evolutionary deep reinforcement learning. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 7(4).

Marquetoux, N.; Stevenson, M. A.; Wilson, P.; Ridler, A.; and Heuer, C. 2016. Using social network analysis to inform disease control interventions. *Preventive Veterinary Medicine*, 126.

McCallum, A. K.; Nigam, K.; Rennie, J.; and Seymore, K. 2000. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3.

Nguyen, B.; Sani, L.; Qiu, X.; Liò, P.; and Lane, N. D. 2024. Sheaf HyperNetworks for Personalized Federated Learning. *arXiv preprint arXiv:2405.20882*.

Niu, C.; Song, Y.; Song, J.; Zhao, S.; Grover, A.; and Ermon, S. 2020. Permutation invariant graph generation via score-based generative modeling. In *International Conference on Artificial Intelligence and Statistics*.

Panagopoulos, G.; Malliaros, F. D.; and Vazirgianis, M. 2020. Influence maximization using influence and susceptibility embeddings. In *International AAAI Conference on Web and Social Media*, volume 14.

Panagopoulos, G.; Malliaros, F. D.; and Vazirgiannis, M. 2020. Multi-task learning for influence estimation and maximization. *IEEE Transactions on Knowledge and Data Engineering*, 34(9).

Panagopoulos, G.; Tziortziotis, N.; Vazirgiannis, M.; and Malliaros, F. 2023. Maximizing influence with graph neural networks. In *International Conference on Advances in Social Networks Analysis and Mining*.

Rossi, R.; and Ahmed, N. 2015. The network data repository with interactive graph analytics and visualization. In *AAAI conference on artificial intelligence*, volume 29.

Sastry, S. 2013. *Nonlinear systems: analysis, stability, and control*, volume 10. Springer Science & Business Media.

Tang, J.; Tang, X.; Xiao, X.; and Yuan, J. 2018. Online processing algorithms for influence maximization. In *2018 international conference on management of data*.

Tang, Y.; Shi, Y.; and Xiao, X. 2015. Influence maximization in near-linear time: A martingale approach. In *2015 ACM SIGMOD international conference on management of data*.

Tennison, B. R. 1975. *Sheaf theory*, volume 21. Cambridge University Press.

Traag, V. A.; Waltman, L.; and Van Eck, N. J. 2019. From Louvain to Leiden: guaranteeing well-connected communities. *Scientific reports*, 9(1).

Turing, A. M. 1990. The chemical basis of morphogenesis. *Bulletin of mathematical biology*, 52.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.

Wang, C.; Liu, Y.; Gao, X.; and Chen, G. 2021. A reinforcement learning model for influence maximization in social networks. In *International Conference on Database Systems for Advanced Applications*. Springer.

Wang, Y.; Cong, G.; Song, G.; and Xie, K. 2010. Community-based greedy algorithm for mining top-k influential nodes in mobile social networks. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Wang, Y.; Yi, K.; Liu, X.; Wang, Y. G.; and Jin, S. 2022. ACMP: Allen-cahn message passing with attractive and repulsive forces for graph neural networks. In *International Conference on Learning Representations*.

Xia, W.; Li, Y.; Wu, J.; and Li, S. 2021. Deepis: Susceptibility estimation on social networks. In *14th ACM International Conference on Web Search and Data Mining*.

Zhang, S.; Tong, H.; Xu, J.; and Maciejewski, R. 2019. Graph convolutional networks: a comprehensive review. *Computational Social Networks*, 6(1).

Zhao, J.; Dong, Y.; Ding, M.; Kharlamov, E.; and Tang, J. 2021. Adaptive diffusion in graph neural networks. *Advances in neural information processing systems*, 34.