Deep Graph Representation Learning and Optimization for Influence Maximization

Chen Ling ▪ Junji Jiang ▪ Junxiang Wang ▪ My Thai ▪ Lukas Xue ▪ James Song ▪ Meikang Qiu ▪ Liang Zhao

2023

Presented by
Daniel Stulberg Huf
Mir Wise Khan

Paper Review
Massive Graph Management and Analytics (MGMA)

29/02/2024

Overview

Overview

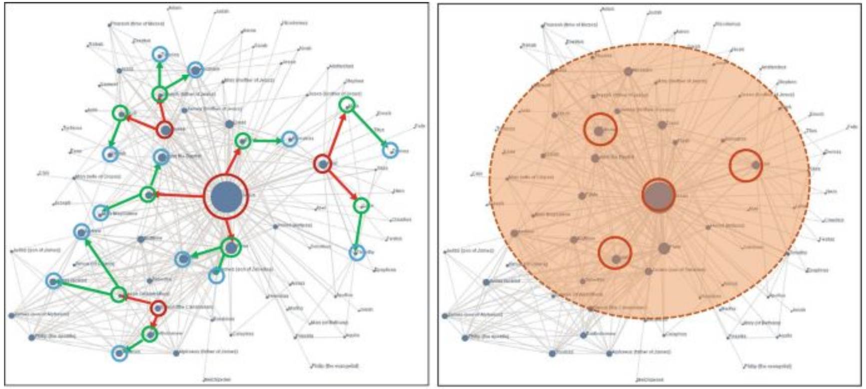
1. **Problem statement**

Problem Statement

↪ **Influence Maximization** (IM) problem: “finding a subset of nodes in a social network that could maximize the spread of influence”. [1]

↪ Traditional IM models [2]:

- **Simulation-based**
- **Proxy-based**
- **Heuristic-based**
- **Sketch-based**
- **Community-based**



(a) Influence Diffusion. (b) Influence Maximization.

Qi Liu “Influence Maximization Model”

[1] Chen et al. (2009) “Efficient Influence Maximization in Social Networks”; [2] Bouyer et al. (2022) “Influence maximization problem by leveraging the local traveling and node labeling method for discovering most influential nodes in social networks”

Problem Statement

↪ **Learning-based IM models**: stronger generalization with unknown graphs.

↪ Fundamental obstacles for learning-based approaches.

What is missing?

↪ Characterization of the latent representation of seed sets.

↪ New objective function to infer optimal seed sets under flexible node-centrality-based budget constraints.

Overview

1. Problem statement
2. Introduction

Introduction

Influence Maximization

↪ **Objective**: find a set of seed nodes that maximizes the spread of influence in a social network.

↪ **Value:** IM has been extensively studied in recent years thanks to its large commercial value:

▪ **Viral marketing for promoting a commercial product:** Chen, W., Wang, C., and Wang, Y. (2010) "Scalable influence maximization for prevalent viral marketing in large-scale social networks".

▪ **Network monitoring:** Wang, Y., Fan, Q., Li, Y., and Tan, K.-L. (2017) "Real-time influence maximization on dynamic social streams".

▪ **Misinformation containment:** Yang, L., Li, Z., and Giua, A. (2020) "Containment of rumor spread in complex social networks".

▪ **Friend recommendation:** Ye, M., Liu, X., and Lee, W.-C. (2012) "Exploring social influence for recommendation: a generative model approach".

Introduction

↪ **Challenge 1**: Stochastic nature of information diffusion and hardness of the optimization problem.

Introduction

↪ **Challenge 1**: Stochastic nature of information diffusion and hardness of the optimization problem.

$\Downarrow$

Traditional methods of IM

(explicit requirement of the diffusion model as an input)

Introduction

↪ **Challenge 1**: Stochastic nature of information diffusion and hardness of the optimization problem.

⬇

Traditional methods of IM

(explicit requirement of the diffusion model as an input)

⬇

↪ **Challenge 2**: Real-world information diffusion process is complex and cannot be modeled by prescribed diffusion models.

Introduction

↪ **Challenge 1**: Stochastic nature of information diffusion and hardness of the optimization problem.

⇓

Traditional methods of IM

(explicit requirement of the diffusion model as an input)

⇓

↪ **Challenge 2**: Real-world information diffusion process is complex and cannot be modeled by prescribed diffusion models.

⇓

Learning-based methods of IM

(characterize the underlying diffusion process)

Introduction

↪ Fundamental obstacles for learning-based approaches:

1. **Difficulty in solving the objective function:**

Solving the discrete problem in continuous space with deep network and reinforcement learning raises scalability issues.

Introduction

↪ Fundamental obstacles for learning-based approaches:

1. **Difficulty in solving the objective function:**

Solving the discrete problem in continuous space with deep network and reinforcement learning raises scalability issues.

2. **Difficulty in characterizing the underlying diffusion patterns:**

Need to rely on heuristics by leveraging pre-defined diffusion models (LT, IC, etc).

Introduction

↪ Fundamental obstacles for learning-based approaches:

1. **Difficulty in solving the objective function:**

Solving the discrete problem in continuous space with deep network and reinforcement learning raises scalability issues.

2. **Difficulty in characterizing the underlying diffusion patterns:**

Need to rely on heuristics by leveraging pre-defined diffusion models (LT, IC, etc).

3. **Difficulty in adapting the solution to different node-centrality-constrained IM variants:**

No well-defined paradigm for solving constraints on the number of seed nodes, total degree of seed nodes, etc.

Overview

1. Problem statement
2. Introduction
3. <u>Background</u>

[1] Kempe et al. (2003) “Maximizing the spread of influence through a social network”

Background: Traditional IM Models

↪ Kempe et al. (2003) proposed an analysis framework based on sub-modular properties of spread functions to solve the IM problem [1].

↪ **Greedy algorithm**: finds node with biggest spread, adds it to seed set, then finds node with next biggest marginal spread over original spread until k seed nodes are found.

Algorithm 1 The Greedy-based Influence Maximization Algorithm

Input: Graph $G(V, E)$, number of seed nodes k
Output: Seed node set S

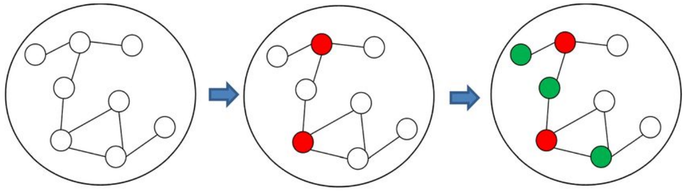
```
Let S = {}
While |S| < k do
    maxInf=0
    For each v in V − S do
        calculate σ(S ∪ v) using Monte Carlo method
        If σ(S ∪ v) − σ(S) > maxInf
            maxNode = v
            maxInf = σ(S ∪ v) − σ(S)
        End If
    End For
    S ← maxNode
End While
return S
```

Liu et al. “Influence maximization based on maximum inner product search”

↪ The greedy strategy guarantees that the spread of the chosen seed set is at least 63% of the spread of the optimal seed set. **Approximate ratio = 1 - 1/e - Ɛ**

↪ **Drawbacks:**

1. Calculates the spread over many nodes;
2. Gets too slow for realistic-sized networks.



Network modeling — Spare all the budget at time step t=0 — Return the marginal gain of the seed nodes

Background: Traditional IM Models

↪ Leskovec et al. (2007) proposed the CELF algorithm based on sub-modular properties of the IM problem under the IC and LT models to reduce the amount of unnecessary gain calculations [1].

↪ The marginal spread of a node in one iteration of the Greedy cannot be any larger than its marginal spread in the previous iteration.

Function: LazyForward($\mathcal{G} = (\mathcal{V}, \mathcal{E})$,$R$,$c$,$B$,$type$)

$\mathcal{A} \leftarrow \emptyset$; **foreach** $s \in \mathcal{V}$ **do** $\delta_s \leftarrow +\infty$;
while $\exists s \in \mathcal{V} \setminus \mathcal{A} : c(\mathcal{A} \cup \{s\}) \leq B$ **do**
  foreach $s \in \mathcal{V} \setminus \mathcal{A}$ **do** $cur_s \leftarrow$ **false**;
  while *true* **do**
    if $type$=UC **then** $s^* \leftarrow \underset{s \in \mathcal{V} \setminus \mathcal{A}, c(\mathcal{A} \cup \{s\}) \leq B}{\operatorname{argmax}} \delta_s$;
    if $type$=CB **then** $s^* \leftarrow \underset{s \in \mathcal{V} \setminus \mathcal{A}, c(\mathcal{A} \cup \{s\}) \leq B}{\operatorname{argmax}} \frac{\delta_s}{c(s)}$;
    if cur_s **then** $\mathcal{A} \leftarrow \mathcal{A} \cup s^*$; **break** ;
    else $\delta_s \leftarrow R(\mathcal{A} \cup \{s\}) - R(\mathcal{A})$; $cur_s \leftarrow$ **true**;
return $\mathcal{A}$;

Algorithm:CELF($\mathcal{G} = (\mathcal{V}, \mathcal{E})$,$R$,$c$,$B$)

$\mathcal{A}_{UC} \leftarrow$LazyForward($\mathcal{G}, R, c, B, UC$);
$\mathcal{A}_{CB} \leftarrow$LazyForward($\mathcal{G}, R, c, B, CB$);
return $\operatorname{argmax}\{R(\mathcal{A}_{UC}), R(\mathcal{A}_{CB})\}$

Algorithm 1: The CELF algorithm.

[1] Leskovec et al. (2007) “Cost-effective Outbreak Detection in Networks”

Background: Learning-based IM Models

↪ Use deep learning to address the drawbacks of traditional IM methods.

↪ The two phases of the learning-based IM problem:

LEARNING PHASE	INFERENCE PHASE
▪ Homogeneous networks G = {G1, G2, .., Gl);	▪ Target network G;
▪ Diffusion model Ψ;	▪ Integer k;
▪ Set of parameters Θ is trained such that y can approximate the objective function as accurately as possible.	▪ Function y that approximately calculates the marginal influence of nodes w.r.t partial solution S;
	▪ IM problem is solved in G w.r.t. Budget k and diffusion model Ψ.

↪ The choice of the diffusion model Ψ is vital for the selection of seeds.

Background: Learning-based IM Models

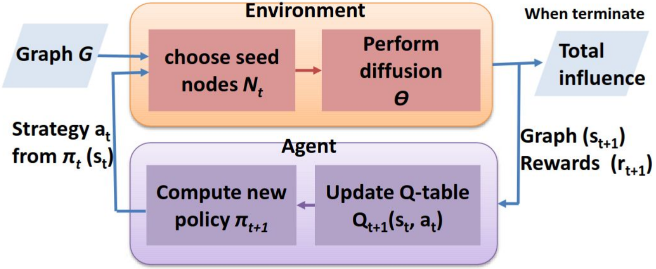
↪ Lin et al. (2015) proposed a framework capable of learning multi-parity IM strategies by using single-player IM strategies as its actions and finding the best policy to select given the observed conditions. The rewards are the gain of the influenced nodes compared to its opponents. The optimal strategy is learned by interacting with the network environments and the opponents. [1]

↪ 3 scenarios considered:

1. Opponent's strategy is known for every round;
2. Opponent's strategy is unknown but available for training;
3. Opponent's strategy is unknown and unavailable for training.

$$V^\pi(s) = E_\pi\{R_t | s_t = s\} = E_\pi\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s\}, \quad (1)$$

$$\begin{aligned} Q^\pi(s,a) &= E_\pi\{R_t | s_t = s, a_t = a\} \\ &= E_\pi\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s, a_t = a\}. \quad (2) \end{aligned}$$



STORM framework in training phase

[1] Lin et al. (2015) "A Learning-based Framework to Handle Multi-round Multi-party Influence Maximization on Social Networks"

Background: Learning-based IM Models

↪ Model complexity and adaptivity are still not comparable to traditional methods.

↪ Current ML-based algorithms cannot:

1. Handle diversified diffusion patterns;
2. Guarantee the quality of the solution;
3. Scale the model.

Background: Graph Neural Networks (GNNs)

↪ Class of deep learning methods designed to perform inference data described by graphs.

↪ GNNs alternates between node feature transformation and neighbor nodes' information aggregation.

↪ Generation of nodes' representations that depend on the structure of the graph and other feature information.

↪ This process captures both the structural and feature information of the graph at multiple layers, enabling the learning of complex node representations that incorporate local neighbourhood information up to K layers deep.

↪ Various applications:

- **Information diffusion estimation**
- **Graph source localization**
- **Deep graph generation**
- **Graph analogical reasoning**

↪ Our case: characterizing the underlying diffusion pattern to build an end-to-end estimation model.

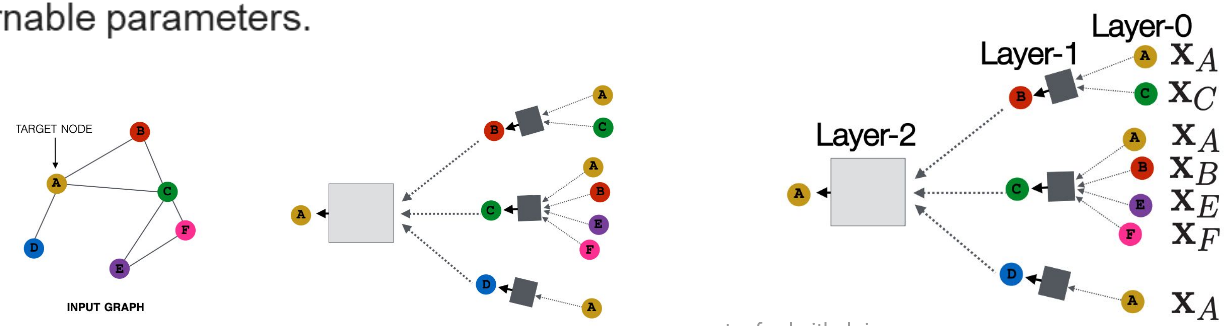
Background: Graph Neural Networks (GNNs)

$$a_k = A_k(h_{k-1}; \theta_k), \quad h_k = C_k(a_k; \theta_k) \qquad \forall\, 1 \leq k \leq K$$

a_k: Aggregation function from the neighbours of a node at layer k.

h_k: Feature representation of a node at layer k, obtained after combining a_k with the node's previous features h_{k-1} using parameters θ_k.

A_k and C_k : Aggregation and combination functions, respectively, parameterized by θ_k , which are learnable parameters.



snap-stanford.github.io

Overview

IM - Problem Formulation

$G = \{V, K\}$

The IM problem seeks to identify a subset of nodes (seed set) that maximizes the spread of influence through the network. It's formulated as:

$$x^* = \arg\max_{|x| \leq k} M(x, G; \theta) \qquad x \subseteq V$$

where θ can be:

1. the set of infection probability on each node if $M(.)$ is an independent cascade model
2. the set of parameters in the aggregation/combine functions if $M(.)$ is GNN-based.

- x^*: The optimal seed set.
- $M(x, G; \theta)$: The influence spread function, parameterized by θ, which measures the total influence exerted by seed set x on graph G.
- The objective is to find up to k nodes whose activation maximizes the influence spread across the network.

IM - Problem Formulation

$$\mathbf{x} \in \{0, 1\}^{|V|}$$

$$x_i = \begin{cases} 1, & \text{if } v_i \in x \\ 0, & \text{otherwise} \end{cases}$$

$$y \in \mathbb{R}_+$$

1. Most existing learning-based IM frameworks calculate the latent node embedding for selecting high-influential ones. However, their objective functions require iteratively updating the latent embeddings for each node at each action/optimization step no matter whether they are included in the current x. This poses a severe scalability problem if we are dealing with million-scale networks.
2. Existing frameworks are still tailored for specific diffusion models (e.g., they model M() as explicit IC and LT model). However, these simple diffusion models cannot meet the needs of real applications.

IM - Existing Limitations/Challenges Revised

Previous GNN + RL strategies

The expressiveness and generalization capability of existing learning-based IM frameworks is still limited due to the following challenges

1. The objective functions require iteratively updating the latent embeddings for each node at each action/optimization step no matter whether they are included in the current x. **Scalability PROBLEM**
2. Existing frameworks are still tailored for specific diffusion models (e.g., they model M() as explicit IC and LT model). However, these simple diffusion models cannot meet the needs of real applications.

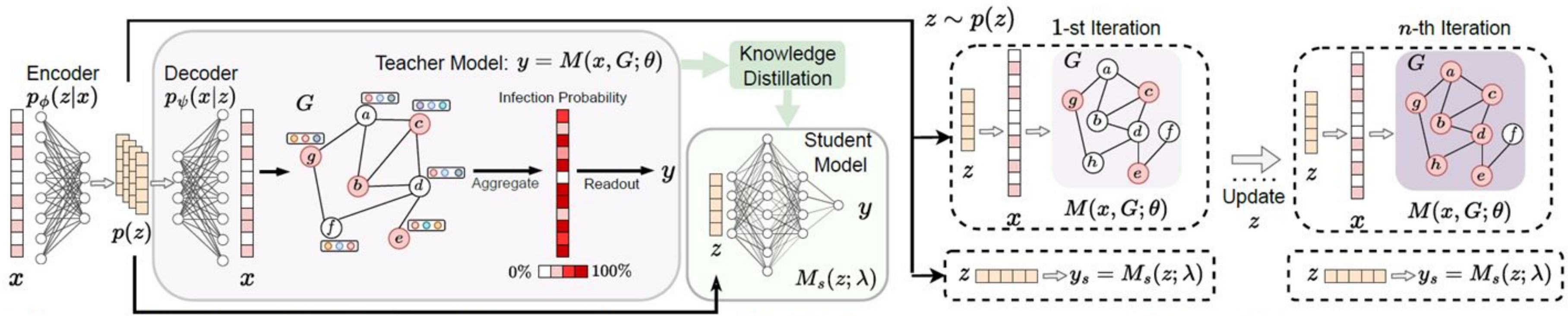
DeepIM Framework Overview

DeepIM framework **eases the computational overhead** of the learning-based IM methods and **automatically identify the underlying diffusion patterns**.

The framework can be divided into two phases:

1. **Learning phase**
   Characterize the probability of the observed seed set and model the underlying information propagation distribution,
2. **Inference phase**
   Optimize the selection of seeds in continuous space to maximize the influence spread.

End-to-End Pipeline



(a) Learning the Diffusion Model $M(\cdot)$ and Representation over Seed Node Set x

(b) Seed Set Inference

Learning Representation

Challenges with Direct Modelling: Directly modelling the probability $p(x)$ of a seed node set x is complex because the nodes within a seed set are interconnected and highly correlated based on the graph's topology.

Latent Variable Introduction: DeepIM introduces:

- a latent variable z to represent the seed set x
- a conditional probability $p(x \mid z)$

Obtaining the Probability Distribution: The probability of observing any particular seed set $p(x)$ is then derived by marginalizing over z

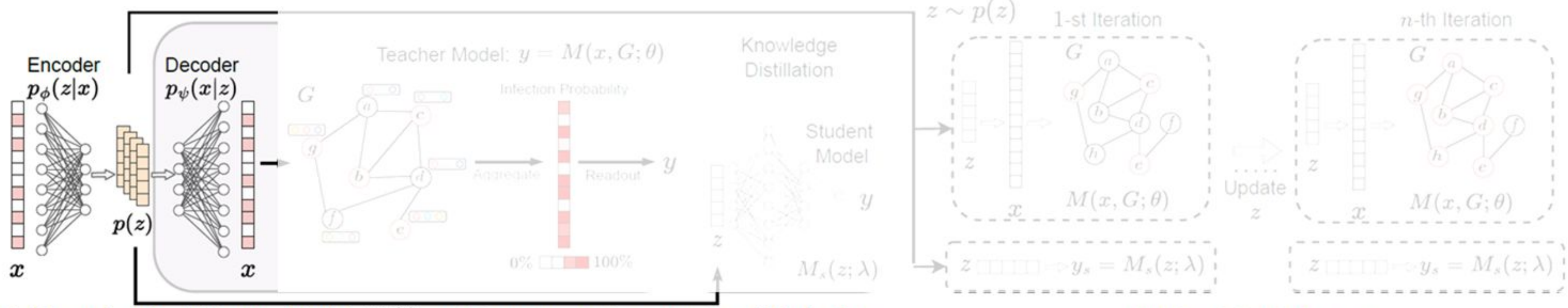
$$p(x) = \int p(x \mid z)p(z)dz$$

Learning Representation

Autoencoder for Learning: An autoencoder, with an encoder f_ϕ and a decoder f_ψ, learns to infer this distribution. The encoder maps seed sets to the latent space, and the decoder reconstructs the seed sets from the latent representations.

Objective: The objective of the autoencoder is to maximize the joint likelihood:

$$\max_{\phi,\psi} \mathbb{E}[p_\psi(\mathbf{x}|z) \cdot p_\phi(z|\mathbf{x})]$$



(a) Learning the Diffusion Model $M(\cdot)$ and Representation over Seed Node Set x (b) Seed Set Inference

End-to-End Diffusion Model (WHY?)

Objective: Update the seed node set x in order to increase the marginal gain of the influence spread $M(\mathbf{x}, G; \theta)$

Diffusion limitation: Current learning-based IM solutions still assume the computation of the influence spread relies on prescribed mathematical models (LT & IC).

- A chosen diffusion model may be mis-specified compared to real-world data and lead to large model bias.
- The diffusion network structure can also be hidden from us, so we need to learn not only the parameters in the diffusion model but also the diffusion network structure

End-to-End Diffusion Model (GNN-based)

GNN-based Diffusion Model: $M(.)$ for accurate modelling of the relationship between x and y by taking graph topology into account.

The output is composed of two functions: $$M = g_r \circ g_u(\mathbf{x}, G; \theta)$$

1. **Aggregation Function**: $$\tau = g_u(\mathbf{x}, G; \theta) \quad \tau \in [0,1]^{|V|}$$

   Using GNN, aggregates multi-hop neighbourhood information to produce an intermediate output τ(infection probability). Allows capturing the graph structure and node features to compute the likelihood of each node being influenced.

2. **Normalization Function**: $$y = g_r(\tau; \xi), y \in \mathbb{R}_+$$

   The final information spread y is determined by applying a normalization function $gr(\cdot)$ to the aggregated infection probabilities τ. This function, which can be an $l-1$ norm or another normalization technique, transforms the continuous probabilities into discrete values based on a threshold ξ. The output y represents the expected number of nodes influenced in the diffusion process.

Monotonicity (GNN Models)

The GNN-based diffusion model $M(x, G; \theta)$, which is composed of the aggregation function $gu(x, G; \theta)$ and the normalization function $gr(\tau; \xi)$, is said to be score and infection monotonic under certain conditions. Specifically, for any GNN-based model M, monotonicity is ensured if the following are true:

- The coefficients A_k and C_k in the aggregation and normalization functions are non-decreasing.
- The normalization function $gr(\cdot)$ is also non-decreasing.

Monotonicity (GNN Models)

The GNN-based diffusion model $M(x, G; \theta)$, which is composed of the aggregation function $gu(x, G; \theta)$ and the normalization function $gr(\tau; \xi)$, is said to be score and infection monotonic under certain conditions. Specifically, for any GNN-based model M, monotonicity is ensured if the following are true:

- The coefficients A_k and C_k in the aggregation and normalization functions are non-decreasing.
- The normalization function $gr(\cdot)$ is also non-decreasing.

Monotonicity (Graph Attention Network - GAT)

GAT can also satisfy score and infection monotonicity under conditions:

- The attention mechanism weights θ_k are non-negative.
- The normalization function $gr(\cdot)$ remains non-decreasing.

This implies that, in GAT, as long as the attention coefficients are non-negative and the post-processing of the attention scores (i.e., the normalization function) does not decrease with increasing input values, the influence spread will not decrease as more nodes are added to the seed set.

Knowledge Distillation for Diffusion Estimation Efficiency

Addressing computational overhead issue (**Teacher + Student Models**). Inspired by recent research on knowledge distillation.

Learning Teacher Model:

A GNN-based diffusion model $M(x, G; \theta)$ is first trained to accurately predict the influence spread from a given set of seed nodes x across the graph G, parameterized by θ.

This model performs three key steps:

1. Decoding a node vector x from the learned posterior $p(x \mid z)$
2. Executing the GNN-based diffusion model under the graph G
3. Normalizing the probabilistic output τ from $M(x, G; \theta)$ to the actual influence spread y

Knowledge Distillation for Diffusion Estimation Efficiency

Addressing computational overhead issue (**Teacher + Student Models**). Inspired by recent research on knowledge distillation.

Learning Student Model:

A student model $M_s\ (z; \lambda)$, which is a lightweight and more computationally efficient neural network parameterized by λ, is then trained under the supervision of the teacher model. The student model directly takes a latent variable z, sampled from the learned distribution p(z), as input and aims to predict the influence spread y_s directly.

The knowledge distillation process involves minimizing the distillation loss between the teacher model's output $y = M(x, G; \theta)$ and the student model's output $y_s\ = M_s\ (z; \lambda)$. This loss can be quantified simply as the squared difference $||\ y - y\ s\ \ ||^2$

End-to-End learning Objective

To integrate the representation learning of seed sets with the learning of the diffusion model, a unified objective function is proposed

$$L_{\text{train}} = \max_{\theta,\lambda,\psi,\phi} E[p_\theta(y|x,G) \cdot p_\lambda(y_s|z) \cdot p_\psi(x|z) \cdot p_\phi(z|x)] \quad \text{s.t.} \quad \theta \geq 0$$

This objective maximizes the expected joint probabilities, linking the efficiency of the student model's influence prediction with the accuracy of the teacher model's diffusion estimation.

The knowledge distillation approach allows for retaining the predictive accuracy of the **complex GNN-based teacher model** while significantly reducing the computational burden during **inference** through the **lightweight student model**. This strategy enables scalable and efficient influence spread estimation across large networks.

DeepIM with constraints

Algorithm 1 DeepIM Prediction Framework

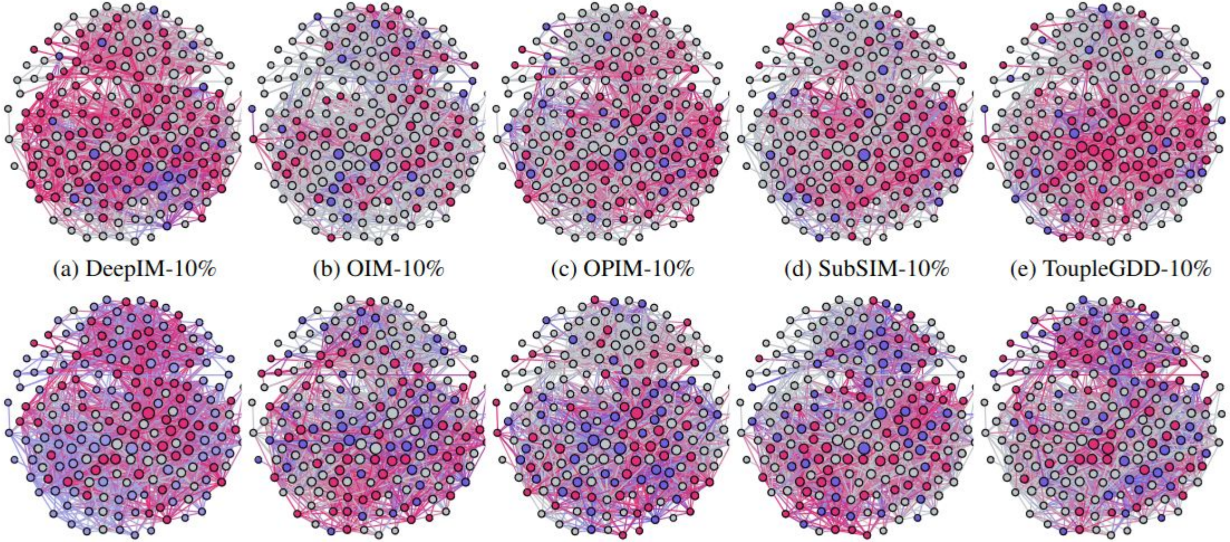
Input: $\mathcal{L}_{\text{pred}}$; $f_\psi(\cdot)$; $\Phi(\cdot)$; number of training instances N; the number of iteration η; learning rate α.

1: $z = 1/N \sum_{i=0}^{N} f_\psi(\mathbf{x})$ {$\mathbf{x}$ sampled from training set.}
2: **for** $i = 0, \ldots, \eta$ **do**
3: $\quad \mathbf{x} \leftarrow f_\psi(z)$ {seed set $\mathbf{x}$.}
4: $\quad \mathbf{x} \leftarrow \Phi(\mathbf{x})$ {Regularize $\mathbf{x}$ into valid regions.}
5: $\quad z \leftarrow z - \alpha \cdot \nabla \mathcal{L}_{\text{pred}}(\mathbf{x}, z)$
6: **end for**
7: $\tilde{\mathbf{x}} \leftarrow \Phi(f_\psi(z))$

Overview

Experiment: Graph Diffusion Visualization

↪ Visualization of influence spread in Jazz dataset:



(a) DeepIM-10% (b) OIM-10% (c) OPIM-10% (d) SubSIM-10% (e) ToupleGDD-10%

(f) DeepIM-20% (g) OIM-20% (h) OPIM-20% (i) SubSIM-20% (j) ToupleGDD-20%

- Initial seed nodes
- Infected nodes during spread
- Uninfected nodes

↪ Overall increased performance of DeepIM in influence spreading.

Experiment: Results

	Cora-ML				Network Science				Power Grid				Jazz				Synthetic				Digg				Weibo			
Methods	1%	5%	10%	20%	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	4.3	17.4	31.5	51.1	2.6	20.1	31.4	42.8	9.2	26.2	36.3	51.6	7.4	18.4	32.8	49.6	9.5	23.8	36.4	50.5
OPIM	13.4	26.9	37.4	50.9	6.6	19.4	28.9	48.6	5.7	17.7	29.7	50.1	2.4	20.1	34.4	46.8	9.6	25.3	36.6	51.7	7.6	18.5	32.9	48.9	9.7	23.7	36.6	50.3
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	3.6	18.8	37.6	44.7	9.5	26.7	36.5	51.5	7.5	18.9	33.3	49.4	9.3	23.1	36.5	50.6
OIM	8.9	27.6	38.0	51.3	4.2	16.7	26.5	48.2	5.7	17.5	31.9	50.8	2.0	18.5	36.3	42.2	9.6	26.2	36.7	51.3	7.8	18.2	33.1	49.6	-	-	-	-
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	3.6	19.7	37.5	45.9	9.1	26.2	36.1	51.5	7.9	18.6	33.5	49.8	9.4	23.5	36.9	50.3
PIANO	9.8	25.2	37.4	51.1	4.7	16.3	27.1	47.2	5.3	18.1	31.7	50.2	2.2	19.2	36.6	43.2	9.1	26.4	36.2	51.6	-	-	-	-	-	-	-	-
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	3.3	20.4	37.2	45.7	9.5	26.8	37.1	51.4	-	-	-	-	-	-	-	-
DeepIM$_s$	13.6	27.7	38.5	51.8	6.9	19.1	29.3	50.5	5.9	20.2	31.7	51.5	3.8	21.4	38.9	47.1	10.2	26.8	37.5	51.8	7.9	18.8	33.7	50.3	10.1	24.7	36.8	50.8
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**	**4.9**	**23.3**	**41.5**	**49.9**	**11.6**	**27.4**	**38.7**	**52.1**	**8.4**	**19.3**	**34.2**	**51.3**	**11.2**	**26.5**	**37.9**	**51.8**

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

	Cora-ML				Network Science				Power Grid				Jazz				Synthetic				Digg				Weibo			
Methods	1%	5%	10%	20%	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	1.7	34.8	52.2	66.4	2.5	11.9	18.1	33.6	4.6	19.9	31.7	56.9	1.4	5.7	13.4	24.5	1.1	5.2	13.1	66.9	2.4	10.8	37.4	55.6	1.6	6.7	19.3	45.2
OPIM	2.3	36.9	51.2	71.5	1.6	12.0	18.8	34.1	4.4	21.6	29.4	55.5	1.4	6.9	12.6	20.9	1.3	5.2	12.6	62.1	2.1	11.3	38.2	57.1	1.8	6.1	18.7	46.6
SubSIM	1.7	33.6	54.7	70.1	1.8	10.4	19.2	34.1	4.5	21.1	31.2	57.4	1.4	5.9	11.4	21.2	1.4	5.5	13.1	69.6	2.4	11.3	37.9	56.9	1.7	6.7	19.2	46.8
IMINFECTOR	2.1	33.9	51.3	70.6	2.1	11.8	18.7	34.5	4.2	21.3	31.6	56.2	1.4	6.2	13.5	22.8	1.3	5.2	12.9	67.4	2.2	11.1	38.9	58.7	1.8	6.4	18.6	47.5
PIANO	2.1	33.5	53.3	69.8	2.1	11.3	19.1	33,9	4.3	21.3	31.4	57.1	1.1	6.2	12.1	22.4	1.2	5.2	12.9	67.4	-	-	-	-	-	-	-	-
ToubleGDD	2.3	36.2	54.5	70.9	2.8	12.4	19.8	34.6	4.8	21.9	32.6	58.1	1.4	6.5	12.9	23.6	1.3	5.5	13.4	70.2	-	-	-	-	-	-	-	-
DeepIM$_s$	10.7	65.6	75.1	85.2	3.5	14.6	23.8	37.8	5.1	22.9	40.3	65.1	1.4	6.5	14.2	85.3	1.5	6.0	14.2	90.3	3.1	13.3	39.2	67.9	2.5	7.1	32.6	68.4
DeepIM	**13.4**	**69.2**	**83.5**	**94.1**	**4.1**	**16.6**	**26.7**	**41.5**	**6.3**	**24.4**	**46.8**	**71.7**	**1.9**	**6.5**	**16.4**	**99.1**	**1.5**	**6.5**	**15.5**	**99.9**	**3.5**	**15.9**	**41.3**	**76.2**	**3.1**	**7.6**	**39.3**	**72.4**

Table 3. Performance comparison under LT diffusion pattern. — indicates out-of-memory error. (Best is highlighted with bold.)

Experiment: Results

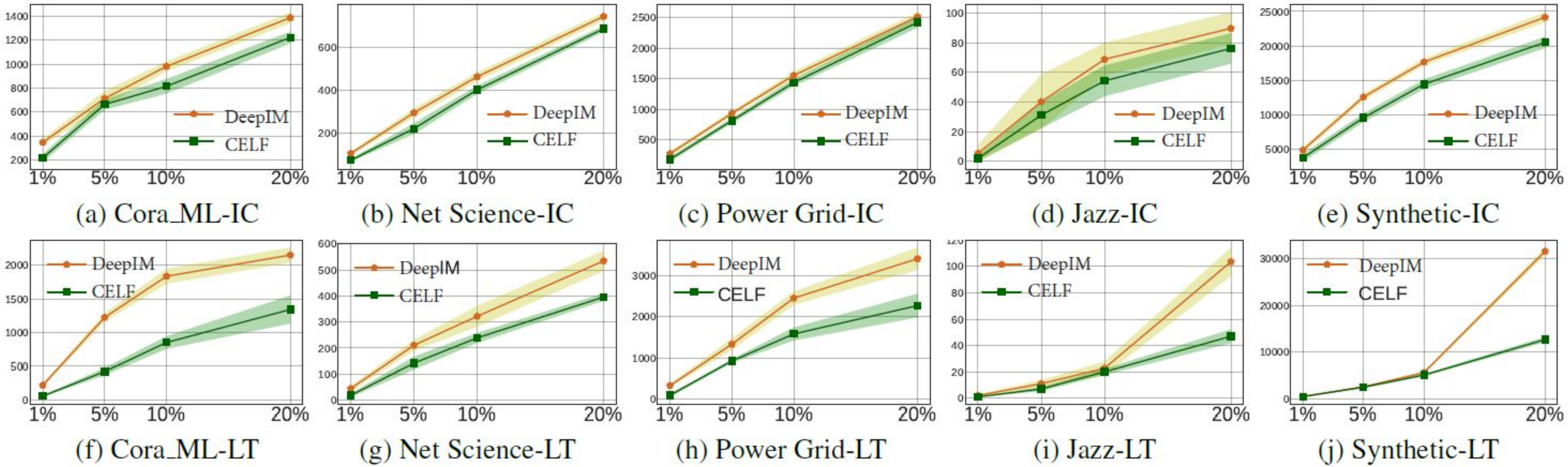


Figure 2. The influence spread (total infected nodes) in the y-axis under the constraint of the budget with the node size growth (x-axis: 1%, 5%, 10%, and 20%). Fig. 2a - 2e and Fig. 2f - 2j are evaluated under the IC and LT model, respectively.

Overview

Conclusion

↪ A novel framework was proposed to tackle the IM problem in a robust and generalized way.

↪ Outcomes:

1. Characterizing the probability of the seed set and directly search for a optimal seed set in continuous space;
2. Two learning-based diffusion models (GNN-based and student model) for characterizing different diffusion dynamics.
3. New objective function to couple with with multiple constraints for seed node set inference.

↪ Ubiquitous advantages of DeepIM over state-of-the-art (traditional and learning-based) IM methods on:

1. Synthetic and real-world datasets;
2. IC and LT diffusion patterns;
3. Budget constraints of 1%, 5%, 10%, and 20%.

Questions?

Thank you!