



CMINet: a Graph Learning Framework for Content-aware Multi-channel Influence Diffusion

Hsi-Wen Chen
National Taiwan University
Taipei, Taiwan
Academia Sinica
Taipei, Taiwan
hwchen@arbor.ee.ntu.edu.tw

De-Nian Yang
Academia Sinica
Taipei, Taiwan
dnyang@iis.sinica.edu.tw

Wang-Chien Lee
Pennsylvania State University
State College, United States
wul2@psu.edu

Philip S. Yu
University of Illinois
Chicago, United States
psyu@uic.edu

Ming-Syan Chen
National Taiwan University
Taipei, Taiwan
mschen@ntu.edu.tw

Abstract

The phenomena of influence diffusion on social networks have received tremendous research interests in the past decade. While most prior works mainly focus on predicting the total influence spread on a single network, a marketing campaign that exploits influence diffusion often involves multiple channels with various information disseminated on different media. In this paper, we introduce a new influence estimation problem, namely *Content-aware Multi-channel Influence Diffusion (CMID)*, and accordingly propose *CMINet* to predict newly influenced users, given a set of seed users with different multimedia contents. In *CMINet*, we first introduce *DiffGNN* to encode the influencing power of users (nodes) and *Influence-aware Optimal Transport (IOT)* to align the embeddings to address the *distribution shift* across different diffusion channels. Then, we transform *CMID* into a node classification problem and propose *Social-based Multimedia Feature Extractor (SMFE)* and *Content-aware Multi-channel Influence Propagation (CMIP)* to jointly learn the user preferences on multimedia contents and predict the susceptibility of users. Furthermore, we prove that *CMINet* preserves *monotonicity* and *submodularity*, thus enabling $(1 - 1/e)$ -approximate solutions for influence maximization. Experimental results manifest that *CMINet* outperforms eleven baselines on three public datasets.

CCS Concepts

• Information systems → Social networks; Web applications.

Keywords

Social Influence; Multi-channel Diffusion; Graph Neural Networks

ACM Reference Format:

Hsi-Wen Chen, De-Nian Yang, Wang-Chien Lee, Philip S. Yu, and Ming-Syan Chen. 2023. CMINet: a Graph Learning Framework for Content-aware Multi-channel Influence Diffusion. In *Proceedings of the ACM Web Conference 2023 (WWW '23)*, April 30–May 04, 2023, Austin, TX, USA. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3543507.3583465>

1 Introduction

Online social networks are an integral part of many people's lives nowadays. By sharing experiences and opinions in the form of text, audio, and videos on social media platforms, e.g., Twitter [43], Facebook [4], and Tiktok [27], users' social influence may implicitly (or even explicitly) affect others' emotions, opinions, or behaviors. Over the past decade, many research works have been proposed to exploit social influence in various applications, e.g., recommendation [50], advertising [38], and social campaigns [37]. For those applications, modeling the process of influence diffusion is essential and thus attracts significant interests from the research community. Early efforts on applications exploiting social influence are mainly based on the Independent Cascades (IC) model [20] and the Linear Threshold (LT) model [13]. The diffusion process of these models, starting with a set of pre-selected *seed* nodes for activation, iteratively activates the neighbors of active nodes based on some activation criteria, which generally assume fixed-yet-unknown interpersonal influence. Since then, several variants of the influence estimation problem have been formulated to incorporate various contextual features, e.g., topic [5, 29], location [6, 30], and time [1, 52], to improve the effectiveness of the diffusion models. However, these methods usually rely on oversimplified assumptions that do not match the real-world requirements, i.e., the influence only propagates to the direct neighborhood with a predefined weight, thereby failing to model users' diverse online behaviors, especially under the multimedia contents [2]. Thus, the accuracy of the spread estimation is undermined [48].

Generally speaking, prior influence diffusion models rarely consider the following aspects that are prevalent in real-world scenarios. **1) Multi-channel Scenario.** An online social networking service, e.g., Facebook [14] or TikTok [47], typically facilitates various interactive actions, e.g., like, share and comment, and thus provides different channels for influence diffusion. However, conventional spread estimation methods [36] only focus on one specific

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

WWW '23, April 30–May 04, 2023, Austin, TX, USA

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-9416-1/23/04...\$15.00
<https://doi.org/10.1145/3543507.3583465>

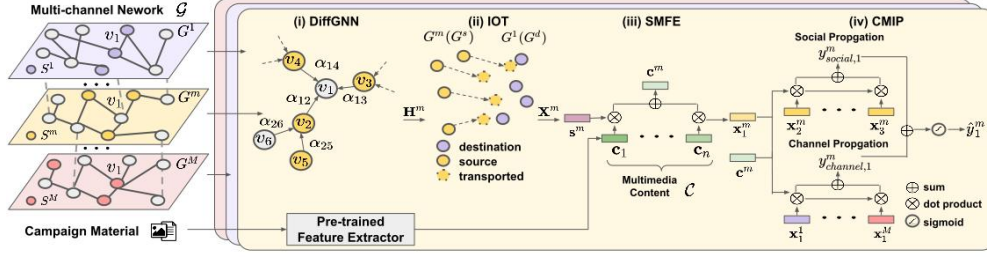


Figure 1: The illustration of CMINet. i) First, *Diffusion Graph Neural Network (DiffGNN)* extracts the embeddings of user nodes $\{H^m\}_{m \in [1, M]}$ for each channel of the multi-channel network $\mathcal{G} = \{G^m\}_{m \in [1, M]}$. ii) Then, *Influence-aware Optimal Transport (IOT)* aligns the user embeddings in each channel of $G^2, \dots, G^M$ to that of the destination G^1 to obtain the transported embeddings $\{X^m\}_{m \in [1, M]}$. iii) Meanwhile, *Social-aware Multimedia Feature Extractor (SMFE)* aggregates diverse types of multimedia contents $\mathcal{C} = \{c_1, \dots, c_n\}$ to a content embedding c^m by considering the seed set s^m in the m^{th} diffusion channel. iv) *Content-aware Multi-channel Influence Propagation (CMIP)* predicts the probability y_i^m of node v_i on the m^{th} channel by social propagation and channel propagation.

action on a single-channel network or assume different actions incur independent influence diffusion processes and thus use separate networks to model each type of influential activity. Realistically, the different interactive actions may all influence the user and thus likely affect each other in the diffusion process. While most prior influence diffusion models [5, 11] progressively estimate the diffusion probability on individual social connections, they do not adequately incorporate other influence factors, e.g., the correlation between different actions. Thus, we explore a multi-channel network to capture correlated actions jointly. **2) Multimedia Content.** In addition to the topology of the underlying social network, the user-to-user influence diffusion usually depends on the diverse multimedia content of campaign materials [34]. For example, a sports fan may be inclined to retweet a poster of shoes endorsed by basketball stars, while a fashion lover may be attracted to some videos of boutique events. While several prior works on topic-aware models [5, 11] explore tagged contents, e.g., advertorial and campaign, to obtain the probabilistic models for diffusion, they are usually constrained by the limited number of topic/themes and thus can not process multimedia to support a fine-grained susceptibility prediction.

In this paper, we introduce a new influence estimation problem, *Content-aware Multi-channel Influence Diffusion (CMID)*, which predicts the influenced users given the following two inputs: i) **Multi-channel Seed Set** (i.e., the initially activated nodes of a multi-channel network), and ii) **Multimedia Contents** (i.e., the initial representations of contents, extracted from the campaign materials, corresponding to various types of multimedia). CMID is a challenging problem since users have diverse personal preferences on multimedia contents and varied influences on friends in a multi-channel network. For social influence diffusion, Graph Neural Network (GNN) [22, 41] is a natural tool that aggregates multi-hop neighbors' information by stacking multiple layers as the influence diffuses across many users. Each layer may model the users' interactions with the multimedia contents of various campaigns to estimate the spreading process on the multi-channel network. Exploiting the strength of GNN, we propose a new learning framework, namely CMINet. As illustrated in Fig. 1, CMINet consists of four components: i) *Diffusion Graph Neural Network (DiffGNN)*, ii) *Influence-Aware Optimal Transport (IOT)*, iii) *Social-based Multimedia Feature Extractor (SMFE)*, and iv) *Content-Aware Influence Multi-channel Propagation (CMIP)*.

We first propose DiffGNN to encode the user influence in separate embedding spaces of different diffusion channels into user embeddings for other components of CMINet. Unlike conventional GNNs, which only aggregate the 1-hop information using uniform weights [22] or an attention mechanism [40], a number of DiffGNNs collectively capture the user behaviors in the diffusion process to measure the global influence of each node (user). After encoding the user influence as the node embedding by DiffGNNs, one for each diffusion channel, an issue is how to aggregate node embeddings across the multi-channel network. While a straightforward approach is to concatenate the embeddings obtained for the same user from DiffGNNs, the embeddings generated for different diffusion channels may exhibit *distribution shifts* due to the inherent characteristics of different diffusion channels. Thus, the learned embeddings are located in different spaces and thus degrade the model performance, especially in estimating the influence in the multi-channel network [10]. Therefore, we address the issue by designing IOT to stage various embeddings in the same latent space.

As users may act differently upon various multimedia contents [34], SMFE is introduced to aggregate various types of contents by considering the social structure of the multi-channel network. The generated content, in the form of an embedding, is fed along with the user embeddings from IOT to capture the user preference on contents in influence diffusion. Since the impact of multimedia contents and user preferences on influence diffusion are inherently intertwined, we propose CMIP model the influence upon a user by enabling a fine-grained estimation. CMIP transforms CMID to a node classification problem (i.e., influenced or not) and models the process of influence propagation in two aspects: i) **social propagation** (i.e., the influence from the neighbors in the same diffusion channel), and ii) **channel propagation** (i.e., the impact from different channels of the same user).

In summary, CMINet embeds all factors, including multi-channel seed nodes and multimedia contents, into a shared embedding space and inductively predicts the susceptibility, i.e., the activation probability of each user, based on aggregated influences from various channels. Compared to previous learning-based frameworks [48], which do not analyze the correctness of the susceptibility, we theoretically prove that GNN is a promising tool for modeling the influence spread on the social network. More importantly, we show that CMINet preserves *monotonicity* and *submodularity*, thus enabling

$(1 - 1/e)$ -approximate solutions in influence maximization [20]. The contributions made in this work are summarized as follows.

- We formulate a new problem, namely *Content-aware Multi-channel Influence Diffusion (CMID)*, and introduce a novel learning framework, CMINet, for CMID by modeling the content-aware influence diffusion processes in a multi-channel network.
- The novelty of CMINet lies in the design of its components: i) DiffGNN utilizes a new attention mechanism to encode both influential power and structural proximity in the user embeddings; ii) IOT aligns the node embeddings from different channels via the proposed influence-aware sampling to alleviate the impact of distribution shift, and iii) SMFE and CMIP together realize the new idea of capturing complex interactions between multimedia content and user actions to predict the user susceptibility.
- We theoretically prove that GNN (and thus CMINet) is a promising tool to model the influence spread on the social network. Moreover, we prove that CMINet preserves *monotonicity* and *submodularity*, enabling $(1 - 1/e)$ -approximate solutions in influence maximization.
- Extensive experiments on three benchmark datasets manifest that CMINet outperforms eleven state-of-the-art baselines.

2 Problem Formulation

First, a multi-channel network is formally defined as follows.

Definition 2.1. *Multi-channel Network.* A multi-channel network is an M -layer graph, defined as $\mathcal{G} = \{G^m\}_{m \in [1, M]}$, where each channel is represented by a subgraph $G^m = \{V, E^m\}$. V is the set of nodes shared in all channels, and E^m represents the m^{th} type edges.

For example, Tiktok [27], a popular online platform, provides three actions (channels), i.e., like, comment, and share. Thus, a multi-channel network of Tiktok users has three types of edges corresponding to these actions. While previous influence estimation models [5, 11] mainly focus on one specific action only (i.e., a single-channel network), behaviors in different channels may affect each other through cross-channel diffusion, e.g., users are likely to *comment* on information *shared* by their friends.

Definition 2.2. *Multi-channel Seed Set.* A multi-channel seed set $\mathcal{S} = \{S^m\}_{m \in [1, M]}$ is a multi-channel node set, where $S^m \subseteq V$ represents the initially activated nodes on the m^{th} -channel. The initially activated nodes S^m for different channels are not required to be the same and usually much smaller than V .

Consider a social marketing event. The campaign material usually contains rich multimedia contents, e.g., text, image, and video. With multimedia contents, users may show varied behavior and susceptibility in different channels, e.g., a fashion lover tends to *share* photos and videos from a boutique show than simply *likes* them.

Definition 2.3. *Multimedia Content.* A multimedia content is a set $C = \{c_n\}_{n \in [1, N]}$, where c_n represents the n^{th} -type of multimedia content.

One may employ various pre-trained feature extractors to obtain the features in terms of embeddings for each type of content [34], which constitutes the above-defined multimedia content. Due to the diverse interests of users in various types of media and content, they may spread differently on different diffusion channels. In the

following, we introduce the *Content-aware Multi-channel Influence Diffusion* problem.

Definition 2.4. *Content-aware Multi-channel Influence Diffusion (CMID).* Given a multi-channel seed set $\mathcal{S}$ and a multimedia content set C , we denote the state of node v_i on the m^{th} channel as $y_i^m = \theta(v_i | \mathcal{S}, C)$, where θ is the influence diffusion model to be learned. CMID aims to predict the susceptibility of every node under the setting, i.e., if v_i is influenced by the multi-channel seed set $\mathcal{S}$ with the content C on the m^{th} -channel, $y_i^m = 1$ and 0 otherwise. The obtained susceptibility of nodes may in turn be used to derive the influence spread of seeds on m^{th} -channel as $\sigma^m(\mathcal{S}, C) = \sum_{v_i \in V} y_i^m$.

Instead of only predicting the influence spread in a channel [26], i.e., the number of influenced nodes $\sigma(\mathcal{S})$, or only predicting the influenced probability [29, 36] for each node on a single channel, i.e., y_i , we model the diffusion process in a fine-grained manner to predict the probability for each node v_i to be influenced on every channel m , i.e., y_i^m . For example, given the multimedia contents extracted from photos and texts in a marketing campaign and multi-channel seed nodes (e.g., some users have commented, shared, or liked the campaign post), CMID aims to identify not only *who* are influenced but also *what* actions are taken to influence *which* neighbors according to the given content. As illustrated in Fig. 1, our goal is to predict whether the inactivated nodes (gray nodes) in each channel would be influenced by the activated nodes (purple nodes S^1 , yellow nodes S^m , and pink nodes S^M) via intertwined channels with the content C . This prediction is very appealing to real-world applications, e.g., precision marketing [38] on different channels.

It's challenging to solve CMID by conventional diffusion models [5, 11], because they do not model the complex interactions between users and multimedia content in different channels. Moreover, there are too many combinations to simulate with Monte-Carlo methods [5]. By contrast, we follow the GNN paradigm in our proposed framework to model the complex correlation between high-dimensional multimedia contents and various user actions in a shared embedding space to learn content and user embeddings which in turn are used to learn a user's fine-grained influence diffusion behavior in an end-to-end training manner. Furthermore, we show the advantages of employing the influence diffusion model derived by CMID as a core component for *Influence Maximization (IM)* [20], as we prove that the IM problem of CMID is *NP-hard*. The proposed CMINet is able to preserve *monotonicity* and *submodularity* that enable $(1 - 1/e)$ -approximate solutions in IM problem.

3 Related Work

Influence Estimation on Social Network. Influence spread estimation aims to approximate the expected number of influenced nodes given a set of seed nodes without considering the multimedia content. While computing the influence spread under the independent cascade [20] or linear threshold models [13] is a $\#P$ -hard problem, the Monte Carlo simulation with various speed enhancements are to estimate the influence spread. Several variants have been proposed by modeling contextual features, e.g., topic [5, 29, 31], location [6, 30], and time [1, 52], to improve the effectiveness of the spread estimation model. For example, Chen et al. [11] analyze the user interest and estimate the influence spread under the given topic. Later, Aslay et al. [3] model the edges with different propagation probabilities according to a set of topic variables. Chen et al. [12] adopt the Maximum Influence Arborescence (MIA) model

and develop a best-effort framework to ensure the approximation ratio of IM. However, all these methods are based on a probabilistic model, which may suffer from the combinatorial explosion [36] considering the high-dimensional multimedia contents. By discretizing the semantic features to derive the probabilistic model, these works cannot accurately estimate the influence spread between node pairs if their interaction did not happen. Another line of research explores network embedding [17, 34, 42, 44] to analyze the influence spread. Cao et al. [7] and Li et al. [29] encode the cascade logs via RNN to predict the popularity of social content. Qui et al. [36] utilize GNN to learn the central node's activation based on the neighbor nodes. However, these works consider the diffusion models on a single-channel network instead of a multi-channel network, thereby not considering the complex correlation between users' various actions on the social platform. In contrast, our model aims to learn the diffusion process on a multi-channel network and carefully analyze the personal preferences for spreading contents to find the susceptibility of each node on various channels in a fine-grained manner. Moreover, these methods do not retain monotonicity and submodularity, two fundamental properties of diffusion models [20].

Graph Neural Network. Recently, GNNs [22] have emerged as a powerful approach for solving many network mining tasks [9, 36, 41, 46], which derive the embedding for a node by aggregating its own and neighboring node features for node classification. For example, Hamilton et al. [19], and Velivckovic et al. [41] leverage different neural networks to aggregate neighbors' features. While most GNNs do not scale well to large graphs due to the recursive neighborhood aggregation, several approaches have been proposed to improve the efficiency of GNNs [45]. Klicpera et al. [23] utilize this propagation procedure to approximate the personalized PageRank via an iterative GNN. Wu et al. [45] propose an equivalent simple graph convolution (SGC) model and show that it scales to large graphs while achieving performance comparable to less scalable state-of-the-art GNNs. Moreover, Xie et al. [49] and Wang et al. [46] introduce a hierarchical aggregation scheme to classify the nodes on a multi-channel network. However, these models are designed for different missions (e.g., classify neighboring nodes into the same class) and thereby aggregate the neighbor node's features without considering the characteristics of diffusion models. Besides, they do not carefully analyze the effect of distribution shift in different channels of a multi-channel network, leading to a less accurate influence estimation [10].

4 Content-aware Multi-channel Influence Network

We propose *Content-aware Multi-channel Influence Network (CMINet)* to solve the CMID problem. We first introduce *Diffusion Graph Neural Network (DiffGNN)* to encode both the global influence power of a user and information of his network proximity on a single-channel network into an embedding. Next, to address the issue of distribution shift, we first introduce optimal transport (OT) as a preliminary solution and then propose *Influence-aware Optimal Transport (IOT)* to alleviate intensive computation by sampling highly influential users. Finally, we propose *Social-based Multimedia Feature Extractor (SMFE)* and *Content-aware Multi-channel Influence Propagation (CMIP)* to learn the diverse user preferences on the spreading multimedia content to predict the susceptibility

of each node in a fine grain manner. Fig. 1 illustrates the overall architecture of CMINet.

4.1 Diffusion Graph Neural Network

Previous diffusion methods [13, 20] discretize the input data, i.e., post topics, to estimate the diffusion probability and solve them linearly. As the multimedia content is usually high-dimensional, the above discretization is not able to properly capture the individual preference for sharing the content of the campaign materials in a multi-channel scenario. Recently, GNNs [22, 41] have proven their efficacy in capturing the content and the graph topology, inspiring us to explore GNN to model the interplays between the users and multimedia contents in order to estimate the influence spread. While conventional GNNs [22] iteratively aggregate the neighborhoods' information to compute the node embedding to preserve local proximity, they cannot correctly capture the global influence power of each node during the diffusion process [24]. To address the aforementioned limitations, we introduce DiffGNN to learn the user (node) embeddings for each channel of the multi-channel network, which consists of two primary components: 1) stacked GNN layers to derive the node embeddings, and 2) an influence score to estimate the influence power of each node in a self-supervised manner, which is used in the cross-channel alignment in Section 4.2. We also theoretically prove that GNN is a promising tool to model the diffusion process in Theorem 5.1.

For a single-channel network G^m with $m \in [1, M]$, DiffGNN updates the $(l + 1)^{th}$ latent vector of node $v_i \in V$, denoted as $\mathbf{h}_i^{l+1} \in \mathbf{H}^{l+1}$, by aggregating the l^{th} -layer hidden features, i.e., $\mathbf{H}^l$, of v_i 's neighboring nodes $\mathcal{N}(v_i)$, which can be written as follows.

$$\mathbf{h}_i^{l+1} = r\mathbf{h}_i^l + (1 - r) \sum_{v_j \in \mathcal{N}(v_i)} \alpha_{ij}^l \mathbf{h}_j^l, \quad (1)$$

where $r = 0.5$ is the hyperparameter for the residual connection [40] to preserve the central node v_i information, and α_{ij}^l is the attention weight between v_i and v_j , which can be defined as follows.

$$\alpha_{ij}^l = \frac{\exp(\gamma_{ij}^l)}{\sum_{v_k \in \mathcal{N}(v_i)} \exp(\gamma_{ik}^l)}, \text{ and } \gamma_{jk}^l = \text{LeakyRelu}(\mathbf{a}^{lT} [\mathbf{h}_j^l, \mathbf{h}_k^l]), \quad (2)$$

where $\mathbf{a}^l$ is the attention head to measure the correlation in l^{th} -layer, $\text{LeakyRelu}(\cdot)$ is a non-linear activation function [22], and $[\cdot]$ is the concatenation operation. Inspired by PageRank [23], DiffGNN normalizes the weight γ_{ij}^l between v_i and v_j by the neighborhood of node v_j (i.e., $\mathcal{N}(v_j)$), instead of that of the central node v_i (i.e., $\mathcal{N}(v_i)$), since the value of the endorsement, e.g., opinion leader, depreciates proportionally to the number of edges given out by the endorsing node. In Fig. 1, the embedding of node v_1 is updated by the embedding of the neighborhood nodes, i.e., v_2 , v_3 , and v_4 , with the corresponding weights α_{12} , α_{13} , and α_{14} , respectively. It's worth noting that α_{12} is normalized by v_5 (α_{25}) and v_6 (α_{26}), instead of v_3 (α_{13}), and v_4 (α_{14}).

After stacking the L layer, we summarize the embedding vectors at all depths of the DiffGNN into a single feature vector to capture patch information at different scales, i.e., $\mathbf{h}_i = \mathbf{W}^r [\mathbf{h}_i^0, \dots, \mathbf{h}_i^L]$, where $\mathbf{W}^r$ is a shared projection matrix for all nodes. Then, we adopt the skip-gram objective [35] as the proximity loss L_{prox} to the structural information on graph G , i.e.,

$$L_{prox} = - \sum_{v_i \in V} \left(\sum_{v_j \in \mathcal{N}(v_i)} \text{sigmoid}(\mathbf{h}_i^T \mathbf{h}_j) + \mathbb{E}_{v_j \sim P_{\mathcal{N}(v_i)}} (\text{sigmoid}(-\mathbf{h}_i^T \mathbf{h}_j)) \right), \quad (3)$$

where sigmoid is the activation function, and $P_N(\cdot)$ is the distribution for the negative sampling of users. After deriving the node embedding, the next step is to predict the influence score $\xi_i \in \xi$, which represents the global influence power of each node v_i , i.e., $\xi_i = \text{sigmoid}(\rho^\top \mathbf{h}_i)$. ρ is a vector to calculate the importance of each node in the social network. In contrast to traditional GNNs, which usually require a great effort in collecting labels (e.g., the importance of each node) [22, 41], we introduce influence loss L_{inf} to train DiffGNN in a self-supervised manner.

$$L_{inf} = \sum_{v_i \in V} (\|\xi_i - \xi'_i\|_2 - \mathbb{E}_{v_j \sim P_{N(v_i)}} (\|\xi_i - \xi_j\|_2)). \quad (4)$$

The first part of Eq. (4) minimizes the error between the self score and the estimated score, and the second part distinguishes the influence between each node by negative sampling. The estimated score ξ'_i is derived from v_i 's neighborhoods, i.e.,

$$\xi'_i = \sum_{v_j \in N(v_i)} \frac{\exp(\mathbf{a}_s^\top [\mathbf{h}_i, \mathbf{h}_j]) \xi_j}{\sum_{v_k \in N(v_i)} \exp(\mathbf{a}_s^\top [\mathbf{h}_i, \mathbf{h}_k])}, \quad (5)$$

where $\mathbf{a}_s$ is a vector to measure the influence between nodes. While previous influence diffusion methods [13, 20] need to identify the influence by some fixed hyperparameters to exploit a specific property of the graph, e.g., degree, DiffGNN derives the importance of each node according to Eq. (2) and thereby is more general to incorporate high-order proximity. The overall objective of DiffGNN is

$$L_{gnn} = L_{prox} + \lambda_1 L_{inf} + \lambda_2 \|\xi\|_1 + \lambda_3 \|\mathbf{H}\|_2, \quad (6)$$

where $\|\xi\|_1$ is the l_1 regularization to discretize the output distribution to force the model to concentrate the influence on a few nodes, and $\|\mathbf{H}\|_2$ is the l_2 regularization of node embedding. λ_1 , λ_2 and λ_3 are the hyperparameters to determine the trade-off between the proximity and the influence score. While DiffGNN leverages the characteristics of the underlying diffusion model, convolutional GNNs capture multi-hop neighbors' information by stacking multiple layers with the uniform weight [22] or the attention mechanism [41], undermining the performance of the downstream tasks.

4.2 Influence-aware Optimal Transport

After pretraining the embedding in the single-channel network, e.g., G^m , through DiffGNN, we aim to align the embeddings for each channel into the same latent space to support users with different actions, and thus the user embeddings learned in different channels may lie in different latent spaces. For example, some users like to share the posts while others only comment on them. Moreover, some users may never engage in certain interactions on the social network, and thus we cannot learn the proper embeddings from these channels. The unaligned (or even incorrect) embeddings usually lead to a wrong influence diffusion because the influence measurements, e.g., the inner product of the embeddings, are inaccurate [10]. Therefore, we introduce optimal transport (OT) [15], which is a promising technique for domain adaption to make channel distributions *aligned*, i.e., closer to each other, for a better estimation of influence diffusion.

Given the source network G^s (i.e., $G^2, \dots, G^M$) and the destination network G^d (i.e., G^1), along with their embedding $\mathbf{H}^s$ and $\mathbf{H}^d$, the objective of optimal transport π is formally written as follows.

$$\pi^* = \arg \min_{\pi} \sum_{v_i^s \in \hat{G}^s, v_j^d \in \hat{G}^d} w(v_i^s, v_j^d) \pi(v_i^s, v_j^d), \quad (7)$$

where $\pi(v_i^s, v_j^d)$ is the transportation probability between v_i^s and v_j^d , and $w(v_i^s, v_j^d)$ is the distance function, i.e., $\|\mathbf{h}_i^s - \mathbf{h}_j^d\|_2^2$. While computing the transporting plans of all nodes in G^s and G^d is computationally intensive, especially on a large graph, we propose the *Influence-aware Optimal Transport (IOT)* by sampling the top- K high-influence nodes according to Eq. (5), i.e., $\hat{G}^s$ and $\hat{G}^d$, because they usually contain more information (e.g., active users on the social platform) [20]. In addition, these high-influence nodes usually show more authoritativeness to the given content when the diffusion process converges [5]. By minimizing the cost of transporting $\hat{G}^s$ to $\hat{G}^d$, IOT is able to bridge the distribution shift of the multi-channel network via these nodes. Sinkhorn's algorithm [25] is adopted to solve the optimal transport plan π^* .

Equipped with the optimal plan π^* , we transport the embedding by barycenter mapping [15], defined as follows.

$$\mathbf{x}_i^s = \frac{\sum_{v_j^d \in \hat{G}^d} \pi^*(v_i^s, v_j^d) \mathbf{h}_j^d}{\sum_{v_j^d \in \hat{G}^d} \pi^*(v_i^s, v_j^d)}, \quad (8)$$

where $\mathbf{x}_i^s \in \mathbf{X}^s$ denotes the transported embedding of source embedding $\mathbf{h}_i^s \in \mathbf{H}^s$ by taking the destination embedding $\mathbf{h}_j^d \in \mathbf{H}^d$ as the reference. Therefore, we are able to align the embedding from G^s to G^d inductively by the transportation plan π^* .

Note that we select the network with the highest density as the destination graph, e.g., G^1 , and align other channels, e.g., $G^2, \dots, G^M$ on it because the highest density graph usually contains more historical data to learn more robust embeddings.

$$\mathbf{X}^1 = \mathbf{H}^1, \text{ and } \mathbf{X}^m = \text{IOT}(\mathbf{H}^m, \mathbf{H}^1), \forall m \in [2, M]. \quad (9)$$

Thus, the embeddings obtained by DiffGNNs (for different channels) all lie in the same embedding space that refers to $\mathbf{H}^1$. Fig. 1 illustrates an example of IOT, which alleviates the distribution shift between the source $G^m(G^s)$ to the destination $G^1(G^d)$. The source embedding $\mathbf{H}^m$ (yellow) is aligned to the destination embedding $\mathbf{H}^1$ (purple) to obtain the transported embedding $\mathbf{X}^m$ (light yellow).

4.3 Social-based Multimedia Feature Extractor

Here, we introduce *Social-based Multimedia Feature Extractor (SMFE)*. Consider a multimedia content set C of various media types, extracted from the campaign materials using a suite of pre-trained feature extractors, e.g., Par2Vec [28] for text and CNN [33] for images. Due to the heterogeneity of the multimedia content, SMFE aggregates the extracted contents into an embedding $\mathbf{c}^m$ from the given multimedia content C for each channel. Unlike the previous models [19] that concatenate different contextual features into a unified vector, we adopt the bilinear attention network [21] to aggregate different types of content by capturing *deep* feature interactions among two groups of input features, i.e., the multimedia contents and the multi-channel social network,

$$\mathbf{c}^m = \frac{\sum_{c_n \in C} \exp(\mathbf{s}^{m\top} \mathbf{W} \mathbf{c}_n) \mathbf{c}_n}{\sum_{c_n \in C} \exp(\mathbf{s}^{m\top} \mathbf{W} \mathbf{c}_n)}, \mathbf{s}^m = \frac{1}{|S^m|} \sum_{v_i \in S^m} \mathbf{x}_i^m, \quad (10)$$

where $\mathbf{W}$ is a trainable parameter to find bilinear attention distributions to utilize the given social-contextual information seamlessly. $\mathbf{c}_n$ denotes the embedding vector of the n^{th} types of content in C , and $\mathbf{s}^m$ represents the embedding of the m^{th} channel seed set S^m . If a content embedding $\mathbf{c}_n$ is located closer to the nodes in m^{th} -channel seed set S^m in the embedding space, it contributes

more in the final content embedding $\mathbf{c}^m$ in the m^{th} channel. For example, teenagers may show more interests in sharing a video, while the elders tend to leave comments on a post. Therefore, SMFE is able to aggregate different types of contextual features guided by the social structure to derive the content embedding $\mathbf{c}^m$ of the diffusion channel m .

4.4 Content-aware Multi-channel Influence Propagation

By deriving the embeddings from the multi-channel network, we transform the CMID problem to a node classification task to predict in *Content-aware Multi-channel Influence Propagation (CMIP)*. Conventional diffusion models [5, 11] are difficult to accurately predict the influence probabilities (i.e., susceptibility) for all nodes [36], when the interaction between two nodes does not happen previously, especially under a multi-channel scenario. CMIP embeds all implicit (or even explicit) factors, including multi-channel seed nodes and the multimedia content, in a shared embedding space and carefully investigates their correlation to predict the susceptibility of each user in a fine-grained manner. To correctly estimate the activated probability y_i^m of user v_i on m^{th} c, we model the influence on v_i in two aspects: i) *social propagation* and ii) *channel propagation*.

Given a seed set S^m , the social propagation aims to model the influence spread of the same channel m ,

$$y_{s,i}^m = \sum_{v_j \in S^m} p_{ij}^m, \text{ and } p_{ij}^m = \text{Relu}(\mathbf{x}_i^{m\top} \text{diag}(\mathbf{c}^m) \mathbf{x}_j^m), \quad (11)$$

where $\mathbf{x}_i^m$ and $\mathbf{x}_j^m$ denote the m^{th} -channel embedding of node v_i and v_j from IOT, respectively. p_{ij}^m can be regarded as the influence from node v_j to v_i on m^{th} -channel with the corresponding content embedding $\mathbf{c}^m$ from SMFE. *diag* converts the embedding vector into a diagonal matrix. It is worth noting that previous learning-based methods [26, 36] only model the node dependencies without considering the contextual information, i.e., $\mathbf{x}_i^{m\top} \mathbf{x}_j^m$. We adopt *Relu* activation to ensure the influence $p_{ij}^m \geq 0$. Besides, we use summation instead of other aggregation functions, e.g., average [22] or attention [40], to ensure the *monotonicity* of the function, because the influence would be accumulated from multiple sources.¹

Since influence may spread across the multi-channel network of an individual, channel propagation aims to capture the correlation between different behaviors of an individual as follows.

$$y_{c,i}^m = \sum_{t \in [1, M] \wedge v_i \in S^t} p_i^{mt}, \text{ and } p_i^{mt} = \text{Relu}(\mathbf{x}_i^{m\top} \text{diag}(\mathbf{c}^t) \mathbf{x}_i^t), \quad (12)$$

where p_i^{mt} measures the influence of node v_i on the m^{th} -channel that is affected from the activation behavior of the t^{th} -channel with the corresponding content embedding $\mathbf{c}^t$. Conventional influence estimation models [5, 11] discretize the contextual features to derive the probabilistic model of diffusion and thus are difficult to support the high-dimensional multimedia contents. In contrast, CMIP carefully derives the influence by modeling the correlations

between the seeds and the contents from the historical events. Combining Eq.(11) and Eq.(12), the predicted probability of node v_i being activated on the m^{th} -channel of the social network becomes

$$\hat{y}_i^m = 2 * \text{sigmoid}(y_{s,i}^m + y_{c,i}^m) - 1, \quad (13)$$

where *sigmoid* serves as the activation function to transform the influence to the probability, which ensures the *submodularity* of the function. While users may share different but related contents on the multi-channel network, our framework can also support multiple contents by encoding the personalized social content C of each user v_i (Eq. (10)), and calculating the content-dependent activation probability (Eq. (11) and (12)).

By ensuring the monotonicity and submodularity of $\hat{y}_i^m$, CMINet is able to provide $(1 - 1/e)$ -approximate solutions for influence maximization [20] (detailed in Theorem 5.2). This is a unique strength of CMINet, as previous learning-based methods [29, 36] cannot retain the theoretical guarantees of many greedy algorithms [20]. It's worth noting that $\text{sigmoid}(y_{s,i}^m + y_{c,i}^m) \in [0.5, 1]$, since $y_{s,i}^m$ and $y_{c,i}^m$ are greater or equal to 0 through *Relu* activation. Therefore, we adopt simple linear transformation to map $\hat{y}_i^m \in [0, 1]$. Following [26], we progressively derive the activation probability of each node. Specifically, we first classify the 1-hop neighborhood of the seed nodes and add the activated nodes into the seed set in the subsequent prediction. Afterward, we iteratively predict the higher-order neighbors (e.g., 2-hop, and then 3-hop).

As illustrated in Fig. 1, social propagation $y_{s,1}^m$ analyzes the influence from the activated nodes with the same actions, e.g., $\mathbf{x}_2^m$ and $\mathbf{x}_3^m$, and channel propagation $y_{c,1}^m$ represents the influence from other activated actions of node v_1 , e.g., $\mathbf{x}_1^1$ and $\mathbf{x}_1^M$. Then, we sum up $y_{s,1}^m$ and $y_{c,1}^m$ with the sigmoid activation to predict probability $\hat{y}_1^m$ of node v_1 on the m^{th} -channel with the content C .

4.5 Overall Objective

Since CMINet is differentiable, we integrate the propagation scheme into model training and utilize the final output $\hat{y}_i^m$ to compute the loss on the m^{th} channel. We adopt the cross-entropy loss to incorporate the probability estimation error on each node,

$$L_{prop} = -\frac{1}{|V|} \sum_{m \in [1, M]} \sum_{v_i \in V} y_i^m \log(\hat{y}_i^m) + \lambda_4 \|\theta\|_2, \quad (14)$$

where y_i^m is the ground truth probability, and $\hat{y}_i^m$ is the predicted probability. The l_2 regularization of model parameters θ aims to mitigate the over-fitting and facilitate the convergence process. Finally, we train our model in two stages. First, we warm up the model to pretrain DiffGNN by L_{gnn} . Then, we train the model by optimizing a mixed objective function combining L_{gnn} and L_{prop} . The time complexity of DiffGNN is $O(MdLD_{max}^2)$, IOT is $O(MDK|V|)$, SFME is $O(Md^2N)$, and CMIP is $O(MdnD_{max})$. The overall time complexity of CMINet is $O(Md(LD_{max}^2 + dN + K|V| + nD_{max}))$.

5 Theoretical Analysis

CMINet can be employed in various influence maximization applications, such as precision marketing. Many approximation algorithms designed for influence maximization achieve certain approximation ratios. Unlike previous learning-based frameworks [17, 29, 36],

¹Since social decisions are influenced only by their near neighbors within the network, following [36, 48], we only consider the seed nodes in the n -hop neighborhoods for each classified node v_i on m^{th} -c, i.e., $N_n^m(v_i) \cap S^m$, instead of processing the whole seed set, i.e., S^m in Eq. (11) to reduce the computational cost.

which are unable to retain the theoretical guarantees, CMINet carefully evaluates the influence of each node on the approximation ratio for influence maximization algorithms. First, we show how powerful GNN is for estimating the diffusion process.

Theorem 5.1. (Proof in Appendix A.3)

The influence probabilities $\hat{\mathbf{y}}_{t+1} \in \mathbb{R}^{|V|}$ of each node can be estimated by GNN, which is an upper bound of the actual influence probabilities $\mathbf{y}_{t+1}$ at time step $t + 1$. As $|V| \gg |S|$, the estimation error $\|\hat{\mathbf{y}}_{t+1} - \mathbf{y}_{t+1}\|_1 \rightarrow 0$.

Compared to previous learning-based frameworks [48], which do not analyze the correctness of the susceptibility, we prove that GNN is a promising tool to model the influence spread on the social network since the network size is typically much larger than the number of seed nodes [36]. Then, we prove that CMINet can satisfy two fundamentally important properties: *Monotonicity* and *Submodularity*, which are cornerstones for many existing works [5, 20, 52] to achieve their approximation ratios in influence maximization. Here, we define the monotonicity and submodularity of a neural network.

Proposition 5.1. (Proof in Appendix A.4)

Let S denote a set of nodes. Given a node $v \notin S$, a set-to-value function $f : \mathbb{R}^* \rightarrow \mathbb{R}$ is monotonic iff $f(S \cup \{v\}) \geq f(S)$. With the sum pooling over any non-negative parametric function $g(\cdot)$, i.e., $\sum_{u \in S} g(u)$, the function f is monotonic.

Proposition 5.2. (Proof in Appendix A.5)

Let S^1 and S^2 denote two sets of nodes, where $S^1 \subseteq S^2$. Given a node $v \notin S^2$, a set-to-value function $f : \mathbb{R}^* \rightarrow \mathbb{R}$ is submodular iff $f(S^1 \cup \{v\}) - f(S^1) \geq f(S^2 \cup \{v\}) - f(S^2)$. With the sigmoid activation over any non-negative and monotonic function h , the function f is submodular.

Equipped with these two properties, the following theorem proves that our model is able to guarantee the approximation ratio.

Theorem 5.2. (Proof in Appendix A.6)

CMINet allows $(1 - 1/e)$ -approximate solutions for influence maximization.

6 Experiments

6.1 Setup

Datasets. We employ three real datasets from different social media. 1) *Digg* [36] (279,630 nodes, 1,548,146 edges, 2 channels, and 24,428 instances) is a news platform that allows people to vote stories up or down. Each story contains a list of users who have voted for the story and the time stamp. The voters' friendship links are also collected. 2) *Twitter* [43] (456,626 nodes, 12,508,415 edges, 3 channels, and 499,160 instances) is one of the most popular social network platforms, which contains three types of actions, i.e., like, comment, and retweet. Besides, each post also has rich texts. 3) *Flickr* [8] (162,663 nodes, 10,226,532 edges, 2 channels, and 14,002 instances) contains a friendship graph and a list of marking records of photos. We pre-train feature extractors of the multimedia contents, w.r.t. the types of content c_n . We utilize Par2Vec [28] to embed text contents and employ CNNs [33] to embed image contents.

Table 1: Activation Prediction

method	Digg		Twitter		Flickr	
	AUC	F1	AUC	F1	AUC	F1
<i>Probabilistic Methods</i>						
TIC [5]	0.7677	0.6333	0.7650	0.4978	0.6621	0.3011
TLT [5]	0.7822	0.6500	0.7691	0.5005	0.6654	0.3029
INFLEX [3]	0.7902	0.6391	0.7704	0.5020	0.6533	0.3152
TMIA [11]	0.7930	0.6587	0.7800	0.5035	0.6791	0.3081
<i>Learning-based Methods</i>						
GCN [22]	0.7830	0.6832	0.7724	0.5111	0.7491	0.3215
GAT [41]	0.7984	0.6788	0.7771	0.5235	0.7418	0.3376
SGC [45]	0.8198	0.6864	0.7766	0.5294	0.7330	0.3411
DeepInf [36]	0.8270	0.7044	0.7924	0.5412	0.7212	0.3520
Inf2Vec [17]	0.8125	0.7012	0.7899	0.5433	0.7308	0.3569
MONSTOR [26]	0.7801	0.7101	0.7799	0.5288	0.7401	0.3552
DIEM [39]	0.8030	0.7156	0.7825	0.5293	0.7581	0.3590
CMINet	0.9191	0.7633	0.8308	0.5782	0.8268	0.4121

Baselines and Evaluation. We compare CMINet with four topic-aware probabilistic models, which estimate the propagation probability between nodes by different sampling schemes, including i) *TIC* [5], ii) *TLT* [5], iii) *INFLEX* [3], and iv) *TMIA* [11]. We also compare five representative learning-based methods, including v) *GCN* [22], vi) *GAT* [41], vii) *SGC* [45], viii) *DeepInf* [36], and ix) *Inf2Vec* [17]. While some of the representative learning-based models are not designed for influence estimation, following [36], we pretrain the embedding of each node and then employ embeddings to train a binary classifier to predict the susceptibility. We further compare two reinforcement learning models, which utilize an agent network [39] to estimate the influence spread, including x) *MONSTOR* [26] and xi) *DIEM* [39]. Note that all these baselines focus on single-channel diffusion, thus independently estimating propagation probabilities on each channel. We evaluate all baselines in terms of *Area Under Curve (AUC)* and *F1-Score (F1)*. The average results from 100 runs are reported.

Implementation Details. For the embedding layer, a 128-dimension network embedding is pre-trained using DeepWalk [35]. For all GNN models, we adopt a two-layer structure, as reported in their original papers. Both the first and second GNN layers contain 128 hidden units. In addition, the top- K influenced nodes in IOT and n -hop proportions in CMIP are set to 256 and 3 [36], respectively. The regularization hyperparameters λ_1 , λ_2 , λ_3 , and λ_4 are set to 1, 0.1, 0.1, and 0.1 by grid search [32]. All parameters are initialized with Glorot initialization [18] and trained using the Adagrad [16] optimizer with the learning rate 10^{-2} , weight decay 10^{-4} , and dropout rate 0.2. The mini-batch size is set to 32 across all datasets.

6.2 Quantitative Analysis

We consider the following two tasks, including the activation prediction [36] and the diffusion prediction [17]. The former predicts the susceptibility of each user, i.e., the activation probability that is influenced by friends, which aims to evaluate the fine-grained computation. The latter identifies the influence spread by given seed nodes, validating CMINet for high-order propagation.

Activation Prediction. Following [36], we aim to predict whether a target user v_i is activated by neighboring nodes. We use 80%, 10%, 10% instances for training, validation, and test, respectively. Table 1

Table 2: Diffusion Prediction

method	Digg		Twitter		Flickr	
	AUC	F1	AUC	F1	AUC	F1
<i>Probabilistic Methods</i>						
TIC [5]	0.7358	0.6112	0.7242	0.4677	0.6304	0.2872
TLT [5]	0.7461	0.6123	0.7111	0.4421	0.6231	0.2809
INFLEX [3]	0.7485	0.6200	0.7159	0.4478	0.6239	0.2845
TMIA [11]	0.7477	0.6232	0.7201	0.4566	0.6315	0.2793
<i>Learning-based Methods</i>						
GCN [22]	0.7582	0.6310	0.7280	0.4212	0.6982	0.2980
GAT [41]	0.7535	0.6412	0.7091	0.4481	0.7131	0.2817
SGC [45]	0.7728	0.6556	0.7098	0.4481	0.7215	0.3170
DeepInf [36]	0.8207	0.6720	0.7111	0.4554	0.7122	0.3208
Inf2Vec [17]	0.7921	0.6611	0.7228	0.4523	0.7129	0.3138
MONSTOR [26]	0.7934	0.6677	0.7184	0.4565	0.7040	0.3022
DIEM [39]	0.7966	0.6790	0.7204	0.4582	0.7120	0.3009
CMINet	0.8848	0.7301	0.7991	0.5511	0.8127	0.3744

shows that the proposed CMINet outperforms all baselines by 9.4% (Digg), 10.2% (Twitter), and 14.7% (Flickr) in terms of F1-score on average. Compared with the probabilistic models, CMINet significantly outperforms all baselines by a large margin of F1-score (at least 32.9%). In contrast to the probabilistic methods, which usually oversimplify the cascade in some linear ways and lead to less accurate estimations, the proposed CMINet learns the diffusion process by DiffGNN and captures the individuals' preference under the multimedia content by CMIP from the historical data. Generally, previous models on the Flickr dataset have relatively weak performance because the image features are high-dimensional compared with the text features in Digg and Twitter datasets. However, CMINet achieves much better improvements on the Flickr dataset because previous probabilistic models discretize the contextual features to derive the diffusion probability, and thus cannot correctly model the multimedia content. CMINet also outperforms all representative learning-based methods. While all these baselines extract the embedding on each single-channel network and predict the activation probability independently, CMINet employs IOT to align the learned embeddings for alleviating the distribution shift between individual channels. Moreover, CMINet achieves a better performance than the reinforcement learning approaches, i.e., MONSTOR and DIEM, because these methods aim at predicting the total influence spread, i.e., the number of influenced nodes.

Diffusion Prediction. Different from activation prediction, diffusion prediction additionally considers the high-order propagation [26]. For each test episode, we exploit the first 10% of activated users as the seed users and the rest 90% as the ground truth for each channel of the multi-channel network. Table 2 shows that CMINet achieves great improvement over all baselines in all datasets i.e., 10.2% (Digg), 10.9% (Twitter), and 16.6% (Flickr) in terms of AUC score, indicating that CMINet captures the high-order influence propagation effectively. Specifically, CMINet significantly outperforms all probabilistic models by at least 31.5% in terms of F1-score, because they do not accurately predict influence probabilities for all nodes due to the data sparsity problem. In contrast, CMINet embeds all factors, including multi-channel seed nodes and the multimedia content, in a shared embedding space and exploits their correlation from historical data.

Table 3: Ablation Studies

method	Digg		Twitter		Flickr	
	AUC	F1	AUC	F1	AUC	F1
<i>Effect of Influence-aware Optimal Transport (IOT)</i>						
OT	0.8810	0.7566	0.8154	0.5623	0.8129	0.4029
random	0.8228	0.7266	0.8024	0.5411	0.7889	0.3988
degree	0.8320	0.7316	0.8052	0.5429	0.7901	0.3990
pagerank	0.8341	0.7322	0.8049	0.5444	0.7905	0.4001
w/o IOT	0.8410	0.7399	0.8091	0.5410	0.7795	0.3879
<i>Effect of Content-aware Multi-channel Influence Propagation (CMIP)</i>						
social	0.8828	0.7455	0.8001	0.5494	0.8012	0.3802
channel	0.6012	0.3006	0.7021	0.4219	0.6901	0.2915
CMINet	0.9191	0.7633	0.8308	0.5782	0.8268	0.4121

6.3 Ablation Studies

Here, we analyze the importance of different modules in CMINet on the activation prediction (in Table 3). First, we evaluate IOT by comparing CMINet (i.e., with IOT) against five variants, including **i) OT**, which employs the original optimal transport [15] without influence-aware sampling, **ii) random**, which randomly samples the nodes on the destination network as anchors, **iii) degree** and **iv) pagerank**, which sample the anchors by the degree and pagerank centrality, respectively, and **v) w/o IOT**, which does not align the embeddings of different channels. Since optimal transport can align the distribution shift between each channel of the multi-channel network, the OT variant improves over the model w/o IOT by 5.4% in terms of F1-score. On the other hand, thanks to the idea of sampling highly influential nodes which usually show more authoritative-ness to the given content [5], CMINet outperforms the model w/o IOT by 8.2%. Compared to other centrality-based sampling strategies, e.g., degree and pagerank, CMINet models the global diffusion process via stacked GNN layers to carefully estimate the influence power of each node and thus outperform them by at least 5.1% in terms of F1-score on average. We also compare the components of CMIP as follows: **vi) social-propagation**, which only models the influence spread between the users with the same channel, and **vii) channel-propagation**, which only models the impact caused by other actions of an individual. CMINet outperforms the model with only the social propagation by 6.1%, indicating that modeling the influence cross multi-channel network can improve the accuracy. The performance of the model with only the channel propagation drops significantly by at least 35.1% in terms of F1-score, indicating that capturing the user-to-user social interactions is obviously critical.

7 Conclusion

In this paper, to better capture the real world scenarios on influence diffusion, we propose CMINet to effectively derive the social influence diffused on a multi-channel network with the multimedia content. Moreover, we theoretically prove that GNN is able to effectively estimate the influence diffusion. We show that CMINet can retain monotonicity and submodularity of influence diffusion models, enabling approximate solutions of influence maximization. Experimental results reveal that CMINet outperforms eleven state-of-the-art baselines on three real datasets. For future works, an interesting direction to explore is to predict influence spread under a multisource scenario in dynamic networks.

Acknowledgments

This work is supported by MOST Project No. 111-2223-E-002-006, Taiwan. Further, this work is supported in part by NSF under grants III-1763325, III-1909323, III-2106758, and SaTC-1930941.

References

- [1] Bijaya Adhikari, Yao Zhang, Aditya Bharadwaj, and B Aditya Prakash. 2017. Condensing temporal networks using propagation. In *Proceedings of the 2017 SIAM International Conference on Data Mining*. SIAM, 417–425.
- [2] Sinan Aral and Dylan Walker. 2012. Identifying influential and susceptible members of social networks. *Science* 337, 6092 (2012), 337–341.
- [3] Cigdem Aslay, Nicola Barbieri, Francesco Bonchi, and Ricardo Baeza-Yates. 2014. Online Topic-aware Influence Maximization Queries. In *EDBT*.
- [4] Eytan Bakshy, Itamar Rosenn, Cameron Marlow, and Lada Adamic. 2012. The role of social networks in information diffusion. In *WebConf*.
- [5] Nicola Barbieri, Francesco Bonchi, and Giuseppe Manco. 2013. Topic-aware social influence propagation models. *Knowledge and information systems* 37, 3 (2013), 555–584.
- [6] Taotao Cai, Jianxin Li, Ajmal S Mian, Timos Sellis, Jeffrey Xu Yu, et al. 2020. Target-aware holistic influence maximization in spatial social networks. *IEEE Transactions on Knowledge and Data Engineering* (2020).
- [7] Qi Cao, Huawei Shen, Keting Cen, Wentao Ouyang, and Xueqi Cheng. 2017. Deep-hawkes: Bridging the gap between prediction and understanding of information cascades. In *CKIM*. 1149–1158.
- [8] Meeyoung Cha, Alan Mislove, and Krishna P Gummadi. 2009. A measurement-driven analysis of information propagation in the flickr social network. In *WebConf*.
- [9] Hsi-Wen Chen, Hong-Han Shuai, De-Nian Yang, Wang-Chien Lee, Chuan Shi, S Yu Philip, and Ming-Syan Chen. 2021. Structure-Aware Parameter-Free Group Query via Heterogeneous Information Network Transformer. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 2075–2080.
- [10] Liquan Chen, Zhe Gan, Yu Cheng, Linjie Li, Lawrence Carin, and Jingjing Liu. 2020. Graph optimal transport for cross-domain alignment. In *International Conference on Machine Learning*. PMLR, 1542–1553.
- [11] Shuo Chen, Ju Fan, Guoliang Li, Jianhua Feng, Kian-lee Tan, and Jinhui Tang. 2015. Online topic-aware influence maximization. *Proceedings of the VLDB Endowment* (2015).
- [12] Wei Chen, Tian Lin, and Cheng Yang. 2016. Real-time topic-aware influence maximization using preprocessing. *Computational social networks* 3, 1 (2016), 1–19.
- [13] Wei Chen, Yifei Yuan, and Li Zhang. 2010. Scalable influence maximization in social networks under the linear threshold model. In *ICDM*.
- [14] Yi-Cheng Chen, Wen-Yuan Zhu, Wen-Chih Peng, Wang-Chien Lee, and Suh-Yin Lee. 2014. CIM: community-based influence maximization in social networks. *ACM Transactions on Intelligent Systems and Technology (TIST)* 5, 2 (2014), 1–31.
- [15] Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. 2017. Joint distribution optimal transportation for domain adaptation. In *NeurIPS*.
- [16] John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research* 12, 7 (2011).
- [17] Shanshan Feng, Gao Cong, Arijit Khan, Xiucheng Li, Yong Liu, and Yeow Meng Chee. 2018. Inf2vec: Latent representation model for social influence embedding. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*. IEEE, 941–952.
- [18] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 249–256.
- [19] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *NeurIPS*.
- [20] David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. 137–146.
- [21] Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. 2018. Bilinear attention networks. *Advances in Neural Information Processing Systems* 31 (2018).
- [22] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.
- [23] Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. 2018. Predict then Propagate: Graph Neural Networks meet Personalized PageRank. In *International Conference on Learning Representations*.
- [24] Johannes Klicpera, Stefan Weisberger, and Stephan Günnemann. 2019. Diffusion improves graph learning. In *NeurIPS*.
- [25] Philip A Knight. 2008. The Sinkhorn–Knopp algorithm: convergence and applications. *SIAM J. Matrix Anal. Appl.* (2008).
- [26] Jihoon Ko, Kyuhan Lee, Kijung Shin, and Noseong Park. 2020. MONSTOR: an inductive approach for estimating and maximizing influence over unseen networks. In *2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE, 204–211.
- [27] Hsu-Chao Lai, Jui-Yi Tsai, Hong-Han Shuai, Jiun-Long Huang, Wang-Chien Lee, and De-Nian Yang. 2020. Live Multi-Streaming and Donation Recommendations via Coupled Donation-Response Tensor Factorization. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 665–674.
- [28] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*. PMLR, 1188–1196.
- [29] Cheng Li, Jiaqi Ma, Xiaoxiao Guo, and Qiaozhu Mei. 2017. Deepcas: An end-to-end predictor of information cascades. In *WebConf*.
- [30] Jianxin Li, Taotao Cai, Ajmal Mian, Rong-Hua Li, Timos Sellis, and Jeffrey Xu Yu. 2018. Holistic influence maximization for targeted advertisements in spatial social networks. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*. IEEE, 1340–1343.
- [31] Yuchen Li, Ju Fan, George Ovchinnikov, and Panagiotis Karras. 2019. Maximizing multifaceted network influence. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE, 446–457.
- [32] Petro Liaschchynskiy and Pavlo Liaschchynskiy. 2019. Grid search, random search, genetic algorithm: A big comparison for NAS. *arXiv preprint arXiv:1912.06059* (2019).
- [33] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3431–3440.
- [34] Mayank Meghawat, Satyendra Yadav, Debanjan Mahata, Yifang Yin, Rajiv Ratn Shah, and Roger Zimmermann. 2018. A multimodal approach to predict social media popularity. In *2018 IEEE conference on multimedia information processing and retrieval (MIPR)*. IEEE, 190–195.
- [35] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 701–710.
- [36] Jiezhong Qiu, Jian Tang, Hao Ma, Yuxiao Dong, Kuansan Wang, and Jie Tang. 2018. Deepinf: Social influence prediction with deep learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2110–2119.
- [37] Ajitesh Srivastava, Charalampos Chelmiss, and Viktor K Prasanna. 2015. Social influence computation and maximization in signed networks with competing cascades. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*. 41–48.
- [38] Fangshuang Tang, Qi Liu, Hengshu Zhu, Enhong Chen, and Feida Zhu. 2014. Diversified social influence maximization. In *IEEE ASONAM*.
- [39] Shan Tian, Songsong Mo, Liwei Wang, and Zhiyong Peng. 2020. Deep reinforcement learning-based approach to tackle topic-aware influence maximization. *Data Science and Engineering* 5, 1 (2020), 1–11.
- [40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).
- [41] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.
- [42] Hao Wang, Cheng Yang, and Chuan Shi. 2021. Neural Information Diffusion Prediction with Topic-Aware Attention Network. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 1899–1908.
- [43] Jianshu Weng, Ee-Peng Lim, Jing Jiang, and Qi He. 2010. TwitterRank: finding topic-sensitive influential twitterers. In *Proceedings of the third ACM international conference on Web search and data mining*. 261–270.
- [44] Bo Wu, Wen-Huang Cheng, Yongdong Zhang, Qiushi Huang, Jintao Li, and Tao Mei. 2017. Sequential Prediction of Social Media Popularity with Deep Temporal Context Networks. In *IJCAI*.
- [45] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. 2019. Simplifying graph convolutional networks. In *International conference on machine learning*. PMLR, 6861–6871.
- [46] Qitian Wu, Hengrui Zhang, Xiaofeng Gao, Peng He, Paul Weng, Han Gao, and Guihai Chen. 2019. Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems. In *WebConf*. 2091–2102.
- [47] Xudong Wu, Luoyi Fu, Shuaiqi Wang, Bo Jiang, Xinbing Wang, and Guihai Chen. 2021. Collective Influence Maximization in Mobile Social Networks. *IEEE Transactions on Mobile Computing* (2021).
- [48] Wenwen Xia, Yuchen Li, Jun Wu, and Shenghong Li. 2021. DeepIS: Susceptibility Estimation on Social Networks. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 761–769.
- [49] Yu Xie, Yuanqiao Zhang, Maoguo Gong, Zedong Tang, and Chao Han. 2020. Mgat: Multi-view graph attention networks. *Neural Networks* 132 (2020), 180–189.
- [50] Bingbing Xu, Junjie Huang, Liang Hou, Huawei Shen, Jinhua Gao, and Xueqi Cheng. 2020. Label-consistency based graph neural networks for semi-supervised node classification. In *ACM SIGIR*.

- [51] Chuan Zhou, Peng Zhang, Wenyu Zang, and Li Guo. 2015. On the upper bounds of spread for greedy algorithms in social network influence maximization. *IEEE Transactions on Knowledge and Data Engineering* 27, 10 (2015), 2770–2783.
- [52] Honglei Zhuang, Yihan Sun, Jie Tang, Jialin Zhang, and Xiaoming Sun. 2013. Influence maximization in dynamic social networks. In *2013 IEEE 13th International Conference on Data Mining*. IEEE, 1313–1318.

A Detailed Proof

A.1 Proof of Theorem 2.1

By Definition 2.4, we first formulate the *Content-aware Multi-channel Influence Maximization (CMIM)* problem.

Definition 2.5. *Content-aware Multi-channel Influence Maximization (CMIM).* Let $\{w^m\}_{m \in [1, M]}$ denote the influence weight w.r.t. each channel G^m . The content-aware multi-channel influence maximization problem aims to select the multi-channel seed sets S^* , such that the weighted sum over the number of influenced nodes $\sigma^m(S, C)$, on network G^m , is maximized as followed.

$$S^* = \arg \max_{S, C} \sum_{m \in [1, M]} w^m \sigma^m(S, C),$$

where $S^* = \{S^m\}_{m \in [1, M]}$, and each channel m has a corresponding k^m -node budget, i.e., $|S^m| = k^m$.

For example, different advertisers may have their individual social goals on the influence diffusion, i.e., require more users to share the posts, with varying budgets on each channel [5]. Then, we introduce the following theorem. Note that, by Definition 2.2, even if the seed size k^m is the same for every channel, we do not have to activate the same seed nodes S^m for each channel m .

Theorem 2.1. *It is NP-hard to approximate the Content-aware Multi-channel Influence Maximization (CMIM) problem.*

PROOF. Given an instance of the conventional influence maximization problem, we can construct an instance of our problem with a single-layer network, i.e., $M = 1$ and content $C = \phi$, and the solution of our problem embeds a solution of the conventional problem, which is NP-hard. The theorem follows. $\square$

A.2 Proof of Lemma 5.1

Lemma 5.1. *The influence probabilities $\hat{y}_{t+1}$ can be estimated by a simple GNN model.*

PROOF. Given the influenced nodes in step t with activation probabilities $\hat{y}_t \in \mathbb{R}^{|V|}$, and the propagation probability matrix P , the cascade model recomputes the probabilities of step $t + 1$ as follows.

$$\hat{y}_{t+1} = P\hat{y}_t.$$

Intuitively, the message passing in GNN can compute inherently an approximation of influence diffusion by stacking multiple layers. We can parameterize it to learn a function that tightens this approximation based on supervision. In the t^{th} layer of GNN,

$$H^{t+1} = \text{Relu}(WPH^t),$$

where $H^t, H^{t+1} \in R^{V \times d}$ denote the node feature matrices. By removing the projection matrix W and activation Relu , we can train a simple linear classifier to predict the influence probability by the node features, i.e., $y_t = \text{Linear}(H^t)$. Then, we have

$$\hat{y}_{t+1} = \text{Linear}(H^{t+1}) = \text{Linear}(PH^t) = P\text{Linear}(H^t) = P\hat{y}_t.$$

The lemma follows. $\square$

A.3 Proof of Theorem 5.1

Theorem 5.1. *The influence probabilities $\hat{y}_{t+1} \in \mathbb{R}^{|V|}$ of each node can be estimated by GNN, which is an upper bound of the actual influence probabilities y_{t+1} at time step $t + 1$. As $|V| \gg |S|$, the estimation error $\|\hat{y}_{t+1} - y_{t+1}\|_1 \rightarrow 0$.*

PROOF. According to [51], we have

$$\begin{aligned} \hat{y}_{i,t+1} &= \sum_{v_j \in N(v_i) \cap S} p_{ij} \hat{y}_{j,t} \geq \sum_{v_j \in N(v_i) \cap S} p_{ij} y_{j,t} \\ &\geq 1 - \prod_{v_j \in N(v_i) \cap S} (1 - p_{ij} y_{j,t}) = y_{i,t+1}, \end{aligned}$$

where $\hat{y}_{i,t+1}$ and $y_{i,t+1}$ denote the estimated and actual activation probabilities of node v_i at time step $t + 1$, respectively. $p_{ij} \in P$ represents the propagation probability from v_i to v_j . Combining Lemma 5.1, $\hat{y}_{t+1}$ is the upper bound of y_{t+1} .

Assume that $\|P\|_1 \leq 1$, we have

$$\begin{aligned} &\|\hat{y}_{t+1} - y_{t+1}\|_1 \\ &\leq \sum_{v_j \in N(v_i) \cap S} p_{ij} y_{j,t} - (1 - \prod_{v_j \in N(v_i) \cap S} (1 - p_{ij} y_{j,t})) \\ &\leq \sum_{v_j \in N(v_i) \cap S} \sum_{v_k \in N(v_i) \cap S} p_{ij} y_{j,t} p_{ik} y_{k,t} \\ &\leq \sum_{v_j \in N(v_i) \cap S} \sum_{v_k \in N(v_i) \cap S} p_{ij} p_{ik} \leq \sum_{v_j \in S} p_{ij} \sum_{v_k \in S} p_{ik} \leq \frac{|S|^2}{|V|^2}. \end{aligned}$$

Since $|V| \gg |S|$, $\|\hat{y}_{t+1} - y_{t+1}\|_1 \rightarrow 0$. The theorem follows. $\square$

A.4 Proof of Proposition 5.1

Proposition 5.1. *Let S denote a set of nodes. Given a node $v \notin S$, a set-to-value function $f: \mathbb{R}^* \rightarrow \mathbb{R}$ is monotonic iff $f(S \cup \{v\}) \geq f(S)$. With the sum pooling over any non-negative parametric function $g(\cdot)$, i.e., $\sum_{u \in S} g(u)$, the function f is monotonic.*

PROOF. By definition, we have

$$f(S \cup \{v\}) - f(S) = \sum_{u \in S} g(u) + g(v) - \sum_{u \in S} g(u) = g(v).$$

Since $g(v)$ is non-negative, the proposition follows. $\square$

Corollary A.1. *The non-negative linear combination of a set of monotonic functions is also monotonic.*

PROOF. According to the definition,

$$f_i(S \cup \{v\}) \geq f_i(S), \forall f_i \in F,$$

where F denotes a set of submodular functions. By multiplying a non-negative scalar w_i for each inequality,

$$w_i(f_i(S \cup \{v\})) \geq w_i(f_i(S)), \forall f_i \in F \wedge w_i \geq 0.$$

Summing up the the inequalities, the lemma follows. $\square$

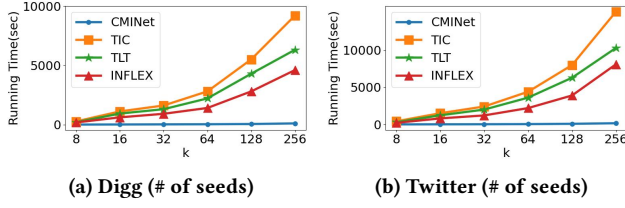


Figure 2: Running Time.

A.5 Proof of Proposition 5.2

Proposition 5.2. Let S^1 and S^2 denote two sets of nodes, where $S^1 \subseteq S^2$. Given a node $v \notin S^2$, a set-to-value function $f: \mathbb{R}^* \rightarrow \mathbb{R}$ is submodular iff $f(S^1 \cup \{v\}) - f(S^1) \geq f(S^2 \cup \{v\}) - f(S^2)$. With the sigmoid activation over any non-negative and monotonic function h , the function f is submodular.

PROOF. Suppose that the function h is first-order differentiable,

$$\partial f = \text{sigmoid}(h)(1 - \text{sigmoid}(h))\partial h.$$

Since $h \geq 0$, $\text{sigmoid}(h)(1 - \text{sigmoid}(h))$ is strictly decreasing. Besides, $\partial h \geq 0$ because g is monotonic, concluding the proof. $\square$

Corollary A.2. The non-negative linear combination of a set of submodular functions is also submodular.

PROOF. According to the definition,

$$f_i(S_1 \cup \{v\}) - f_i(S_1) \geq f_i(S_2 \cup \{v\}) - f_i(S_2), \forall f_i \in F,$$

where F denotes a set of submodular functions. By multiplying a non-negative scalar w_i for each inequality,

$$w_i(f_i(S_1 \cup \{v\}) - f_i(S_1)) \geq w_i(f_i(S_2 \cup \{v\}) - f_i(S_2)), \forall f_i \in F \wedge w_i \geq 0.$$

Summing up the the inequalities, the lemma follows. $\square$

A.6 Proof of Theorem 5.2

Theorem 5.2. CMINet allows $(1 - 1/e)$ -approximate solutions for influence maximization.

PROOF. Since the probability y_i^m supports the monotonicity and submodularity, the influence spread function $\sigma^m(S, c) = \sum_{v_i \in V} y_i^m$ of m^{th} -channel is also monotone and submodular according to Corollaries A.1 and A.2. Similarly, the objective

$$O(S) = \sum_{m \in [1, M]} w^m \sigma^m(S, c)$$

is also monotonic and submodular since $w^m \geq 0, \forall m \in [1, M]$. Following [13], there is a greedy algorithm, s.t. $\frac{O(S)}{O(S^*)} = 1 - 1/e$, where S is the output of the algorithm, and S^* is the optimum. $\square$

B Running Time

Fig. 2 compares the running time of different approaches for Digg and Twitter datasets by varying k , where k is the number of seed nodes on diffusion prediction. CMINet outperforms all baselines significantly, by reducing about two orders of magnitude of running time compared with TIC (91 $\times$), TLT (82 $\times$), and INFLEX (59 $\times$), which are very inefficient due to the Monte-Carlo simulation [5]. In contrast, CMINet adopts DiffGNN to model the high-order influence diffusion process and employs CMIP to exploit the correlation

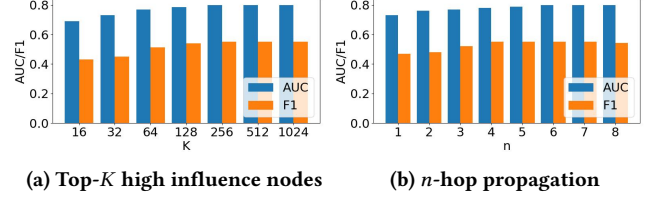


Figure 3: Sensitivity Test.

between multi-channel seed nodes, and considers the multimedia content from historical events.

C Sensitivity Tests

We use the Twitter dataset to perform sensitivity tests to evaluate the impact of various parameter settings on CMINet. Fig. 3(a) reports the performance of CMINet over different numbers of top- K high-influence nodes in IOT. As shown, the performance of CMINet improves as K increases but becomes saturated when K is greater than 256, because we only require 128 nodes in embedding dimension to form a complete metric space for transportation. Note that we do not evaluate the model with K smaller than 16 because there are no sufficient nodes as the anchors for projection. Finally, Fig. 3(b) evaluates the performance of CMINet over different numbers of propagation hops in CMIP. Overall, the performance increases with the increment on n , because the high-order neighborhood nodes also implicitly influence the target nodes. The improvement becomes saturated when n is greater than 4 because the target nodes are less influenced by nodes far away. By digging into details, we observe that most of them have similarity smaller than 0 and thus are ignored by the *Relu* activation.

D Pseudocode

Algorithm 1 CMINet

Require: The multi-channel network $\mathcal{G} = \{G^m\}_{m \in [1, M]}$, multifaceted seed nodes $\mathcal{S} = \{S^m\}_{m \in [1, M]}$, and the context $\mathcal{C}$

Ensure: The newly influenced users $\mathcal{Y} = \{Y^m\}_{m \in [1, M]}$ of each channel.

```

1: for  $m \leftarrow 1$  to  $M$  do
2:    $H^m = \text{DiffGNN}^m(G^m)$ 
3:    $X^1 = H^1$ 
4: for  $m \leftarrow 2$  to  $M$  do
5:    $X^m = \text{IOT}(H^m, H^1)$ 
6: for  $m \leftarrow 1$  to  $M$  do
7:    $c^m = \text{SMFE}(\mathcal{C}, X^m, S^m)$ 
8: for  $S^m \in \mathcal{S}$  do
9:   for  $j \leftarrow 1$  to  $n$  do
10:    for  $v_i \leftarrow N_j(S^m)$  do
11:       $y_{inter,i}^m = \sum_{v_j \in S^m} \text{Relu}(x_i^{mT} \text{diag}(c) x_j^m)$ 
12:       $y_{intra,i}^m = \sum_{k \in [1, M] \wedge y_i \in Y^k} \text{Relu}(x_i^{mT} \text{diag}(c) x_i^k)$ 
13:       $\hat{y}_i^m = 2 * \text{sigmoid}(y_{inter,i}^m + y_{intra,i}^m) - 1$ 
14:      if  $\hat{y}_i^m > 0.5$  then
15:         $S^m.append(v_i)$ 
16:    $Y^m = S^m$ 
17: return  $\mathcal{Y}$ 

```