HRL-GAT: A Hybrid Reinforcement Learning Framework with Graph Attention Networks for Influence Maximization

Guoyu Zhang[†], Huan Li[*,†], Xinyue Mo[*]

School of Cyberspace Security/School of Cryptology, Hainan University, Renmin Avenue 58, Haikou, 570228, Hainan, China.

*Corresponding author(s). E-mail(s): lihuan@hainanu.edu.cn; moxinyue@hainanu.edu.cn;

[†]These authors contributed equally to this work.

Abstract

Influence maximization seeks a seed set that yields the largest expected diffusion, but practical deployment on large networks is hindered by the combinatorial search space and the expensive evaluation of diffusion outcomes. Classical approximation methods are often bottlenecked by repeated influence estimation, whereas learning-based approaches can become unreliable when the action space spans the entire node set and the policy must balance quality with diversity. This work presents HRL-GAT, a Hybrid Learning Framework for influence maximization under the Weighted Independent Cascade model. The central idea is to couple learning with a structured reduction of the decision space so that policy optimization focuses on a compact set of high-impact candidates while still accounting for redundancy among selected seeds. A diffusion-aligned representation is learned to encode network structure, and a lightweight screening mechanism constructs a budget-adaptive candidate pool that retains high-quality seeds while controlling computational cost. Seed selection is then optimized as a finite-horizon decision process using stable policy updates and marginal diffusion gain as the training signal. Experiments on twelve real-world networks show that HRL-GAT consistently achieves higher expected influence spread than seven representative baselines across seed budgets, while maintaining stable training behavior and practical scalability.

Keywords: Influence maximization; graph attention networks; reinforcement learning; proximal policy optimization; social networks

1 Introduction

Information diffusion is a fundamental process in modern complex networks, where large populations of interacting entities collectively shape the spread of information, opinions, and behaviors. Such networks underpin a wide range of real-world applications, including viral marketing [1], public opinion dynamics [2], and epidemic mitigation. A central question in this context is how to identify a small set of influential individuals whose activation can maximize the expected diffusion, i.e., the *Influence Maximization* (IM) problem. This problem has attracted sustained attention and has been systematically reviewed in several recent surveys [3, 4]. However, IM remains computationally challenging: under standard settings the influence function is submodular and the optimization is NP-hard [5]. As a classic function, CELF accelerates the greedy framework via lazy evaluations to reduce redundant marginal-gain computations [6]. However, it still requires many repeated influence evaluations during iterative selection, which can be time-consuming on large-scale networks.

In recent years, learning-based approaches have emerged as a major direction for influence maximization, aiming to improve practicality by leveraging deep models to approximate diffusion-related signals and guide seed selection. Representative studies employ graph representation learning [7–9] or GNN-style encoders [10, 11] to score nodes or estimate influence more efficiently, so that seed selection can be performed with amortized inference rather than expensive repeated evaluations. However, these deep learning methods often rely on large amounts of training supervision generated from simulations or costly offline computations, and their performance may be sensitive to diffusion settings and distribution shifts across networks, requiring retraining or careful adaptation [12, 13].

In parallel, reinforcement learning (RL) [14–18] has been increasingly adopted to model IM as a sequential decision-making problem, where an agent selects seeds step by step based on learned network representations. While RL-based methods can flexibly incorporate objectives and constraints, they typically suffer from high sample complexity and training instability, and still depend on expensive influence estimation during training [19, 20]. Moreover, compared with classical submodular methods, learning-based approaches usually provide limited theoretical guarantees and may be harder to deploy under strict latency or robustness requirements.

To address the above limitations in scalability, efficiency, and robustness of learning-based influence maximization, a hybrid learning framework termed HRL-GAT is introduced for the Weighted Independent Cascade (WIC) diffusion model [21, 22]. The framework is organized as a two-stage pipeline that couples representation learning with sequential decision-making. First, a Graph Attention Network (GAT) encoder [23] is employed to learn diffusion-aware node representations by modeling fine-grained structural dependencies, thereby providing informative features for policy learning under heterogeneous network structures. Second, a lightweight candidate-screening mechanism, Expected Cascade Multiple Reward (ECMR), is designed to construct a compact yet high-quality candidate seed pool, which substantially reduces the action space and alleviates the computational burden induced by large-scale graphs. On top of the ECMR-filtered candidate set, a Proximal Policy Optimization

(PPO) agent [24] is trained to select seed nodes sequentially, enabling stable policy updates and improving sample efficiency while directly optimizing the diffusion objective.

The main contributions are summarized as follows:

- **Hybrid two-stage framework for scalable IM.** HRL-GAT integrates GAT-based representation learning with PPO-based sequential seed selection under WIC, improving practical scalability for large networks.
- **Diffusion-aware node embeddings.** The GAT encoder captures fine-grained structural dependencies and produces informative node embeddings that enhance policy learning and robustness across diverse network topologies.
- **ECMR candidate screening.** ECMR constructs a compact high-quality candidate seed set to shrink the action space and reduce runtime, while preserving influence quality.
- **Stable and sample-efficient sequential selection.** A PPO agent is trained on the ECMR-reduced action space to enable stable optimization and efficient sequential seed selection aligned with the diffusion objective.

The remainder of this paper is organized as follows. Section 2 reviews related works on influence maximization. Section 3 introduces the WIC diffusion model and the formulation of the problem. Section 4 presents the proposed framework in detail. Section 5 reports the experimental results. Finally, Section 6 concludes the paper and outlines directions for future work.

2 Related Works

This section reviews representative approaches to the IM problem, covering classical optimization methods and recent learning-based frameworks.

2.1 Classical Methods

Early studies formulate IM as a combinatorial optimization problem and exploit the monotonicity and submodularity of the influence spread function under stochastic diffusion assumptions to obtain approximation guarantees. The greedy algorithm is a canonical approach that iteratively selects the node with the largest marginal gain [5]. To improve practical efficiency, CELF and CELF++ incorporate lazy evaluation and priority-queue updates to reduce redundant marginal-gain recomputation during greedy selection [25].

RIS constitutes another major line for scalable IM. RIS-based methods sample Reverse Reachable (RR) sets and transform IM into a coverage-style selection problem, where the seed set is chosen to cover as many RR sets as possible. Representative algorithms such as TIM/TIM+ and IMM provide provable approximation quality with near-linear running time by carefully controlling the number of RR samples [26]. Tang et al. further establish near-linear-time guarantees via refined sampling analysis for RIS-style methods [27]. Subsequent work improves sampling efficiency and stopping rules, including adaptive control of sample size (SSA/D-SSA) [28] and probability-/structure-aware refinements for large graphs [29].

Besides approximation algorithms, heuristic and meta-heuristic[30–33] methods are also widely studied. Heuristic approaches rank nodes using structural measures such as degree, betweenness, and closeness centralities, and have been extended with richer structural signals such as k-core decomposition and community-aware ranking [34]. Meta-heuristics, including GA, PSO, ACO, and SA, treat IM as a global search over candidate seed sets and iteratively refine solutions through evolutionary or swarm-based updates [35]. Related variants further incorporate community structure or hybrid initialization to improve solution search in large networks [36, 37].

2.2 Machine Learning Methods

Learning-based approaches aim to improve practicality by leveraging representation learning and sequential decision-making. Deep learning methods commonly employ graph neural networks and related architectures to encode network structure and produce node embeddings for influence estimation or seed selection [38]. Cascade-driven models such as DeepInf learn influence patterns from observed diffusion traces and have inspired subsequent variants for more expressive influence modeling [39, 40]. More recent end-to-end frameworks learn seed-set construction using deep graph representations and are designed to generalize seed selection across instances [7, 41]. In addition, embedding-based IM pipelines combine learned node representations with downstream combinatorial selection to support repeated IM queries more efficiently [42, 43].

RL formulates IM as a sequential selection process, where an agent chooses seed nodes step by step to maximize expected diffusion rewards. Early studies adopt DQN-style learning for node selection [14], and subsequent work explores actor–critic and policy-gradient frameworks for policy learning on graphs [16]. Recent methods integrate graph encoders with RL to better capture network structure and improve policy learning, and have been applied to large-scale graphs and related diffusion settings [44–46]. Extensions to competitive or multi-agent settings have also been studied, where multiple policies interact over shared diffusion processes [47, 48].

3 Preliminaries

3.1 WIC Diffusion Model

The WIC model is adopted as the underlying diffusion process for influence maximization. It describes how activations initiated from a seed set propagate through a social network with heterogeneous edge weights, and the expected number of activated nodes under WIC defines the influence spread of a seed set. We adopt WIC as the diffusion process. Given a graph $G = (V, E)$, where V is the set of nodes and E is the set of edges, each edge $(u, v) \in E$ is associated with a propagation probability defined as

$$p_{uv} = \frac{1}{d_v}, \tag{1}$$

where d_v is the degree of node v. The diffusion unfolds in discrete steps. Let $S \subseteq V$ be the initial seed set. At step t, every newly activated node u attempts to activate each inactive neighbor v with probability p_{uv}. Each attempt is independent. Once a node

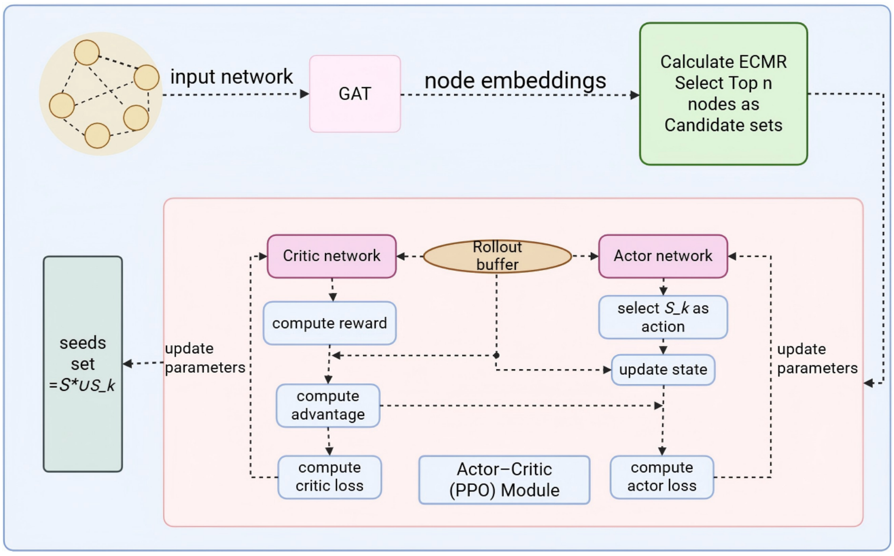


Fig. 1 Overall framework of the proposed GAT–PPO-based influence maximization model.

becomes active, it remains active permanently. The process stops when no further activations are possible.

3.2 Problem Formulation

The IM problem is a formal optimization task defined on a network. The IM problem can be formally defined as follows.

Given a graph $G = (V, E)$, a diffusion model $\mathcal{M}$, and a budget k, the goal is to select a seed set S of size k that maximizes the expected influence:

$$S^* = \arg\max_{S \subseteq V, |S|=k} \sigma(S), \tag{2}$$

where $\sigma(S)$ is the expected number of activated nodes under model $\mathcal{M}$.

It has been shown that the IM problem is NP-hard, and $\sigma(S)$ is monotone and submodular under Independent Cascade (IC) and WIC [5]. However, the exact evaluation of $\sigma(S)$ requires averaging over exponentially many diffusion realizations. In practice, it is estimated by Monte Carlo (MC) simulations.

4 Our Method

4.1 Method Overview

The overall procedure of HRL-GAT is summarized in Algorithm 1.

As illustrated in Fig. 1, the proposed HRL-GAT framework consists of three main components: a GAT-based encoder, the ECMR candidate constructor, and a PPO agent for sequential seed selection.

Algorithm 1 Overview of the HRL-GAT Framework

Input: graph $G = (V, E)$; budget k; candidate multiplier c; diffusion model $\mathcal{M}$
Output: optimal seed set S^*
Step 1: Node Embedding
Generate node embeddings $\mathbf{Z}$ using pretrained GAT (Algorithm 2).
Step 2: Candidate Construction
Compute ECMR scores for all nodes and select top-$c{\cdot}k$ candidates (Algorithm 3).
Step 3: Reinforcement Learning
Train PPO agent to select k seeds sequentially (Algorithm 4).
Step 4: Output
Output the final seed set selected by the PPO agent (same as the algorithm output)).
return S^*.

1. **Node embedding with GAT.** A multi-layer GAT encoder maps original node features and graph structure to expressive node embeddings. A contrastive pre-training strategy enhances the structural discriminability and diffusion awareness of the learned representations.
2. **Candidate seed set construction via ECMR.** An ECMR heuristic integrates one-hop and two-hop propagation effects under WIC, degree centrality, and clustering coefficient. The ECMR score ranks all nodes, and the top $c{\cdot}k$ nodes are retained as a compact candidate seed set, which significantly reduces the action space.
3. **Sequential seed selection with PPO.** A PPO agent is trained to sequentially select k seeds from the ECMR-based candidate set. At each step, the agent observes a state representation that jointly encodes node embeddings, the current seed set, and lightweight structural statistics, from which it samples an action corresponding to the next seed node. The actor–critic architecture with clipped policy updates stabilizes training and improves sample efficiency.

This design reduces the high-dimensional action space, improves the stability of RL training, and explicitly optimizes the diffusion objective. ECMR-based candidate filtering injects structural priors into the RL process, the GAT encoder provides diffusion-aware node embeddings, and the PPO agent learns a robust seed-selection policy driven by MC estimates of influence spread.

4.2 Node Embedding with GAT

The pretraining procedure of the GAT encoder is summarized in Algorithm 2. Let $G = (V, E)$ denote a graph with $|V| = n$ and $|E| = m$. Each node $i \in V$ is associated with an initial feature vector $\mathbf{x}_i \in \mathbb{R}^{d_0}$; when raw attributes are unavailable, structural features are concatenated with learnable embeddings. Let $\mathcal{N}(i)$ represent the (in-)neighborhood of i with self-loops included, and stack node features as $\mathbf{X} \in \mathbb{R}^{n \times d_0}$.

A GAT layer maps $\mathbf{H}^{(\ell)} \in \mathbb{R}^{n \times d_\ell}$ to $\mathbf{H}^{(\ell+1)} \in \mathbb{R}^{n \times d_{\ell+1}}$ via masked self-attention:

$$\mathbf{q}_i^{(\ell)} = \mathbf{W}_q^{(\ell)} \mathbf{h}_i^{(\ell)}, \quad \mathbf{k}_j^{(\ell)} = \mathbf{W}_k^{(\ell)} \mathbf{h}_j^{(\ell)}, \quad \mathbf{v}_j^{(\ell)} = \mathbf{W}_v^{(\ell)} \mathbf{h}_j^{(\ell)}, \tag{3}$$

Algorithm 2 Node Embedding with GAT

Require: graph $G = (V, E)$, input features $\mathbf{x}$, number of layers L, number of heads K
Ensure: node embeddings $\mathbf{z}$

```
Initialize H^(0) ← x
for ℓ = 1 to L do
    for each node i ∈ V do
        Linear projections according to Eq. (3)
        Compute attention logits according to Eq. (4)
        Compute masked attention weights according to Eq. (5)
        Aggregate neighbors and update representations according to Eq. (6)
        Apply residual connection and normalization according to Eq. (9), then apply dropout
    end for
end for
Optimize the pretraining objective in Eq. (12) using Eq. (10) and Eq. (11)
return z ← H^(L)
```

$$e_{ij}^{(\ell)} = \mathrm{LeakyReLU}\Big(\mathbf{a}^{(\ell)\top}\big[\mathbf{q}_i^{(\ell)} \,\|\, \mathbf{k}_j^{(\ell)}\big]\Big), \quad j \in \mathcal{N}(i), \tag{4}$$

$$\alpha_{ij}^{(\ell)} = \frac{\exp\Big(e_{ij}^{(\ell)}\Big)}{\sum_{t \in \mathcal{N}(i)} \exp\Big(e_{it}^{(\ell)}\Big)}, \tag{5}$$

$$\tilde{\mathbf{h}}_i^{(\ell+1)} = \sum_{j \in \mathcal{N}(i)} \alpha_{ij}^{(\ell)} \mathbf{v}_j^{(\ell)}, \qquad \mathbf{h}_i^{(\ell+1)} = \phi\Big(\mathrm{BN}\Big(\tilde{\mathbf{h}}_i^{(\ell+1)}\Big)\Big). \tag{6}$$

Where, $\mathbf{W}_q^{(\ell)}, \mathbf{W}_k^{(\ell)}, \mathbf{W}_v^{(\ell)}$ are learnable projection matrices, $\mathbf{a}^{(\ell)}$ is the attention vector, $\|$ denotes concatenation, and $\phi(\cdot)$ is an activation function (set to ELU). The softmax in Eq. (5) is masked to $\mathcal{N}(i)$, and self-loops are included to preserve node-specific information.

To enhance expressiveness while maintaining stable optimization, K attention heads are adopted. For each head $h \in \{1, \ldots, K\}$, the head-specific representation $\tilde{\mathbf{h}}_{i,h}^{(\ell+1)}$ follows Eq. (6), and the multi-head output is formed by concatenation in intermediate layers and averaging in the final layer:

$$\mathbf{u}_i^{(\ell+1)} = \begin{cases} \big\|_{h=1}^{K} \tilde{\mathbf{h}}_{i,h}^{(\ell+1)}, & \text{if } \ell < L-1, \\ \frac{1}{K} \sum_{h=1}^{K} \tilde{\mathbf{h}}_{i,h}^{(\ell+1)}, & \text{if } \ell = L-1. \end{cases} \tag{7}$$

Further stability across layers is promoted through residual connections with dimension matching:

$$\mathbf{r}_i^{(\ell)} = \begin{cases} \mathbf{R}^{(\ell)} \mathbf{h}_i^{(\ell)}, & \text{if } \dim(\mathbf{u}_i^{(\ell+1)}) \neq \dim(\mathbf{h}_i^{(\ell)}), \\ \mathbf{h}_i^{(\ell)}, & \text{otherwise}, \end{cases} \tag{8}$$

yielding the layer output

$$\mathbf{h}_i^{(\ell+1)} = \phi\Big(\mathrm{BN}\Big(\mathbf{u}_i^{(\ell+1)} + \mathbf{r}_i^{(\ell)}\Big)\Big)\,. \tag{9}$$

Dropout is applied to both attention coefficients and node features to mitigate overfitting.

Embedding pretraining aligns node representations with multi-hop structural cues while avoiding expensive label construction, and is implemented through a two-view contrastive objective. Let $\mathcal{T}$ denote a distribution over graph augmentations, including edge dropout, feature masking, and subgraph sampling. Sampling $t_1, t_2 \sim \mathcal{T}$ yields two views $G^{(1)} = t_1(G)$ and $G^{(2)} = t_2(G)$, which are encoded by a shared GAT to produce $\mathbf{Z}^{(1)}, \mathbf{Z}^{(2)} \in \mathbb{R}^{n\times d}$ with rows $\mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)}$. With cosine similarity $\mathrm{sim}(\mathbf{u},\mathbf{v}) = \frac{\mathbf{u}^\top \mathbf{v}}{\|\mathbf{u}\|\,\|\mathbf{v}\|}$ and temperature $\tau > 0$, the InfoNCE loss is

$$\mathcal{L}_{\mathrm{con}} = -\frac{1}{n}\sum_{i=1}^{n} \log \frac{\exp\Big(\mathrm{sim}\big(\mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)}\big)/\tau\Big)}{\sum_{j=1}^{n} \exp\Big(\mathrm{sim}\big(\mathbf{z}_i^{(1)}, \mathbf{z}_j^{(2)}\big)/\tau\Big)}. \tag{10}$$

To encourage local smoothness consistent with diffusion along edges, the objective is augmented with a Laplacian-style regularizer:

$$\mathcal{L}_{\mathrm{smooth}} \;=\; \frac{1}{|E|}\sum_{(i,j)\in E} w_{ij}\,\big\|\mathbf{z}_i - \mathbf{z}_j\big\|_2^2, \tag{11}$$

where w_{ij} are proportional to the WIC edge probabilities p_{ij}. The overall objective becomes

$$\min_{\Theta_{\mathrm{GAT}}} \; \mathcal{L}_{\mathrm{pre}} \;=\; \mathcal{L}_{\mathrm{con}} + \lambda_{\mathrm{s}}\,\mathcal{L}_{\mathrm{smooth}}, \tag{12}$$

with $\lambda_{\mathrm{s}} \geq 0$. After pretraining, the encoder may be frozen for stability or fine-tuned end-to-end jointly with RL.

The resulting node embeddings are $\mathbf{Z} = \mathbf{H}^{(L)} = [\mathbf{z}_1; \ldots; \mathbf{z}_n] \in \mathbb{R}^{n\times d}$. The PPO agent operates on per-node action features that capture both node quality and diversity relative to the selected seeds S_t. Accordingly, for node i at step t, the action feature vector is defined as

$$\mathbf{f}_i^{(t)} \;=\; \big[\,\mathbf{z}_i \,\|\, \bar{\mathbf{z}}_{S_t} \,\|\, \delta_i^{(t)} \,\|\, \psi_i\,\big], \qquad \bar{\mathbf{z}}_{S_t} = \frac{1}{|S_t|}\sum_{v\in S_t} \mathbf{z}_v, \qquad \delta_i^{(t)} = 1 - \max_{v\in S_t} \mathrm{sim}(\mathbf{z}_i, \mathbf{z}_v), \tag{13}$$

where ψ_i stacks lightweight structural scalars such as degree, clustering coefficient, and the ECMR score. The diversity term $\delta_i^{(t)}$ down-weights candidates that are overly similar to the current seeds, thereby mitigating redundancy.

Algorithm 3 Candidate Seed Set Construction via ECMR

Input: graph $G = (V, E)$; degree d_v; clustering C_v; budget k; multiplier c
Output: candidate set $\mathcal{C}$

for each node $v \in V$ **do**:
  One-hop contribution $I_1(v)$ as in Eq. (14)
  Two-hop contribution $I_2(v)$ as in Eq. (15)
  Compute ECMR(v) as in Eq. (16)
end for
Sort nodes by ECMR(v) in descending order
Select top-$c \cdot k$ candidates as in Eq. (17)
return $\mathcal{C}$

4.3 Candidate Seed Set Construction

The ECMR-based candidate construction procedure is summarized in Algorithm 3. Allowing the RL agent to select seeds directly from the full node set V is impractical on large graphs, as the action space $|V|$ commonly reaches 10^5–10^7. To reduce computational burden without sacrificing solution quality, a heuristic filtering stage is introduced to restrict decisions to a compact candidate seed set. This stage is driven by ECMR, which evaluates the potential influence of each node using local structural signals and probabilistic diffusion cues under WIC.

For a node $v \in V$, the ECMR score integrates (i) one-hop and two-hop propagation effects under WIC, (ii) degree centrality, and (iii) clustering coefficient. Let $\mathcal{N}(v)$ denote the neighbors of v and d_v its degree. The one-hop expected activation contribution is defined as

$$I_1(v) \;=\; \sum_{u \in \mathcal{N}(v)} p_{vu}, \qquad p_{vu} = \frac{1}{d_u}. \tag{14}$$

For each two-hop path $v \to u \to w$, where $u \in \mathcal{N}(v)$ and $w \in \mathcal{N}(u) \setminus \{v\}$, the activation probability is attenuated by a discount factor $\eta \in (0, 1)$, yielding

$$I_2(v) \;=\; \sum_{u \in \mathcal{N}(v)} \sum_{\substack{w \in \mathcal{N}(u) \\ w \neq v}} \eta \cdot p_{vu} \cdot p_{uw}. \tag{15}$$

In the experiments, η is fixed to 0.5, reflecting the intuition that two-hop influence is weaker than one-hop influence.

Let $C_v \in [0, 1]$ denote the clustering coefficient of node v, and let $d_{\max} = \max_{x \in V} d_x$ be the maximum degree in the graph. The ECMR score is then defined as

$$\mathrm{ECMR}(v) \;=\; \Big(1 + I_1(v) + I_2(v)\Big) \cdot \left(\frac{d_v}{d_{\max}} + \big(1 - C_v\big)\right). \tag{16}$$

Algorithm 4 Sequential Node Selection via PPO

Input: candidate set $\mathcal{C}$, graph G, budget k, diffusion model $\mathcal{M}$
Output: optimized policy π_{θ^*}, selected seed set S^*

Initialize actor π_θ and critic V_ϕ
Initialize $S_0 \leftarrow \emptyset$; initialize state s_0
for each episode do:
  for $t = 1$ to k **do:**
    Sample $a_t \sim \pi_\theta(\cdot \mid s_t)\ \mathcal{C}$
    Reward r_t as marginal gain in Eq. (18)
    Update seed set $S_{t+1} = S_t \cup \{a_t\}$; observe s_{t+1}
  end for
  Compute advantages via GAE as in Eq. (21)
  Policy ratio $r_t(\theta)$ as in Eq. (19)
  PPO clipped objective as in Eq. (20)
  Critic loss as in Eq. (22)
  Total loss with entropy bonus as Eq. (23); update θ, ϕ
  Output S^* under final policy π_{θ^*}
return S^*

After computing ECMR$(\cdot)$ for all nodes, the nodes are sorted in descending order of score. Given a budget k, the candidate seed set is chosen as the top-$c \cdot k$ nodes:

$$\mathcal{C} \;=\; \{\, v \in V \;\mid\; \mathrm{rank}(v) \le c \cdot k \,\}, \tag{17}$$

where $c \geq 1$ is a multiplier controlling the trade-off between candidate set size and exploration flexibility.

This ECMR-based filtering serves two purposes: it reduces the action space from $|V|$ to $O(c \cdot k)$, thereby accelerating both training and inference, and it injects structural priors into policy learning by steering the PPO agent toward high-potential regions of the graph. Empirically, the top-$c \cdot k$ ranking retains the vast majority of high-quality seeds while substantially lowering computational overhead.

4.4 Node Selection via PPO

The PPO-based policy update for HRL-GAT is summarized in Algorithm 4. IM is formulated as a sequential decision-making process. At step t, the agent observes state s_t, selects an action a_t corresponding to a node $v_t \in \mathcal{C}$ from the candidate set $\mathcal{C}$, and receives a reward defined as the marginal influence gain:

$$r_t \;=\; \sigma(S_t \cup \{a_t\}) - \sigma(S_t), \tag{18}$$

where $\sigma(S)$ denotes the expected cascade size of a seed set S under WIC. The interaction terminates once k seeds have been chosen, i.e., $|S_T| = k$.

The policy $\pi_\theta(a|s)$ is parameterized by an actor network, while the state-value function $V_\phi(s)$ is modeled by a critic network. The actor produces a distribution over candidate nodes and the critic estimates the expected cumulative reward, yielding an actor–critic architecture that reduces the variance of policy-gradient updates.

To improve training stability, PPO is adopted by constraining policy updates through a clipped surrogate objective. Specifically, PPO updates the policy by reusing trajectories sampled from the previous policy, while ensuring that the updated policy does not change too drastically in one step. To quantify the change of action probabilities induced by the update, PPO introduces the following probability ratio between the new policy π_θ and the behavior policy $\pi_{\theta_{\text{old}}}$: Let θ_{old} denote the policy parameters before an update, and define the probability ratio:

$$r_t(\theta) \;=\; \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}. \tag{19}$$

Using $r_t(\theta)$ as an importance-sampling factor, PPO constructs a surrogate objective weighted by the advantage estimate $\hat{A}_t$, and further clips the ratio to prevent overly large policy updates:

$$L^{\text{PPO}}(\theta) \;=\; \mathbb{E}_t\left[\min\Big(r_t(\theta)\,\hat{A}_t,\; \text{clip}\big(r_t(\theta),\, 1-\epsilon,\, 1+\epsilon\big)\,\hat{A}_t\Big)\right], \tag{20}$$

where ϵ is a small constant and $\hat{A}_t$ is the advantage estimate.

Advantage estimation is computed via Generalized Advantage Estimation (GAE):

$$\hat{A}_t = \sum_{l=0}^{\infty}(\gamma\lambda)^l\,\delta_{t+l}, \qquad \delta_t = r_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t), \tag{21}$$

with discount factor $\gamma \in (0,1]$ and bias–variance parameter $\lambda \in [0,1]$. The critic is trained by minimizing the squared error:

$$L^{\text{critic}}(\phi) = \mathbb{E}_t\Big[\big(V_\phi(s_t) - \hat{R}_t\big)^2\Big], \tag{22}$$

where $\hat{R}_t = \hat{A}_t + V_\phi(s_t)$ denotes the target return.

The overall objective combines the policy loss, the value loss, and an entropy regularizer for exploration:

$$L(\theta,\phi) = -L^{\text{PPO}}(\theta) + \beta\, L^{\text{critic}}(\phi) - \eta\,\mathbb{E}_t\big[\mathcal{H}(\pi_\theta(\cdot|s_t))\big], \tag{23}$$

where β and η balance the value and entropy terms, respectively. This PPO-based formulation stabilizes policy optimization through clipping, improves temporal credit assignment via GAE, and maintains sufficient exploration through entropy regularization, enabling effective optimization of the diffusion objective on large action spaces.

4.5 Time Complexity Analysis

The time complexity of HRL-GAT is analyzed as follows. Let $n = |V|$ and $m = |E|$ denote the numbers of nodes and edges, respectively, k be the seed budget, c be the candidate multiplier in ECMR, L be the number of GAT layers, H be the number of attention heads, and d be the embedding dimension.

For the GAT encoder, computing attention coefficients and aggregating neighbor features in one layer costs $O(mHd)$, so a forward pass through L layers has complexity $O(LmHd)$. If the encoder is pretrained for E epochs, the total pretraining cost is $O(ELmHd)$; this cost is incurred offline and can be reused across subsequent runs on the same graph.

For ECMR, computing the one-hop term $I_1(v)$ takes $O(d_v)$ per node, while the two-hop term $I_2(v)$ iterates over neighbors-of-neighbors and has worst-case complexity $O(\sum_v d_v^2)$, which can be upper-bounded by $O(md_{\max})$ where $d_{\max} = \max_v d_v$. After obtaining ECMR$(\cdot)$ for all nodes, sorting to select the top-ck candidates requires $O(n \log n)$. Hence, the ECMR-based candidate construction costs $O(md_{\max} + n \log n)$.

During PPO training, each episode selects k seeds sequentially from the candidate pool $\mathcal{C}$ with $|\mathcal{C}| = ck$. At each decision step, evaluating the actor and critic over candidates yields a worst-case cost of $O(|\mathcal{C}|d)$, resulting in $O(k|\mathcal{C}|d) = O(ck^2 d)$ per episode for policy evaluation. Reward computation typically dominates: when the marginal gain in Eq. (18) is estimated by MC simulations under WIC, each cascade simulation has expected cost $O(m)$, and using R simulations yields $O(Rm)$ per reward. Over k steps, the reward computation is $O(kRm)$ per episode. If training runs for T episodes, the overall training complexity is $[O\big(T(ck^2 d + kRm)\big)$, where kRm is usually the leading term in practice.

After training, inference requires k sequential selections from $\mathcal{C}$ and thus has worst-case complexity $O(ck^2 d)$. If the final influence spread is reported using R MC cascades, evaluation adds an $O(Rm)$ term. Overall, the computationally intensive GAT pre-training and PPO optimization are performed offline, while online usage on a fixed graph is mainly determined by candidate-based policy evaluation and MC influence evaluation.

5 Experiments

5.1 Influence Evaluation

The quality of a given seed set S produced by HRL-GAT or any baseline is evaluated by estimating its expected influence spread under WIC using MC simulations. For each seed set S, $R = 5000$ independent runs of the WIC diffusion process (as defined in Section 3.1) are performed, and the average final cascade size over these runs is reported as the estimated influence spread:

$$\hat{\sigma}(S) = \frac{1}{R} \sum_{r=1}^{R} \big|\text{Activated}^{(r)}(S)\big|, \tag{24}$$

Algorithm 5 MC Influence Estimation under WIC

Input: graph $G = (V, E)$; seed set $S \subseteq V$; edge probabilities $\{p_{uv}\}$; number of simulations R (e.g., $R = 5000$)
Output: estimated influence spread $\hat{\sigma}(S)$

```
total_spread ← 0
for r = 1 to R do:
    activated ← S
    frontier ← S
    while frontier ≠ ∅ do:
        new_frontier ← ∅
        for each u ∈ frontier do:
            for each neighbor v of u in G do:
                if v ∉ activated then:
                    x ← rand(0, 1)
                    if x < p_uv then:
                        activated ← activated ∪{v}
                        new_frontier ← new_frontier ∪{v}
        end for
    end while
    total_spread ← total_spread +|activated|
end for
σ̂(S) ← total_spread /R
return σ̂(S)
```

where $\text{Activated}^{(r)}(S)$ denotes the set of nodes that are active at the end of the r-th simulation run starting from seed set S. Propagation probability is precomputed as $p_{uv} = 1/d_v$ for each directed edge $(u, v) \in E$, where d_v is the (in-)degree of node v. During each simulation, newly activated nodes attempt to activate their inactive neighbors once, with successful activations determined by independent Bernoulli trials with success probability p_{uv}. The choice $R = 5000$ provides a good trade-off between estimation variance and computational cost, and yields stable comparisons across different methods.

For all datasets described below, the influence spread of each method is evaluated under WIC using the MC procedure in Algorithm 5.

5.2 Datasets

Experiments are conducted on twelve widely used real-world networks to evaluate the effectiveness and robustness of HRL-GAT. The datasets span multiple domains: co-authorship networks include Astroph, CondMat, GrQc, and DBLP [49]; the communication network is Email [50]; social/trust/collaboration networks include Facebook [51], Hamster, Jazz, and PGP [50]; the infrastructure network is Power-Grid [50]; the user–item interaction network is CiaoDVD [52]; and the contact network is Sex [53]. Table 1 summarizes the basic statistics of these datasets.

Table 1 Statistics of the benchmark datasets.

Dataset	#Nodes $\|V\|$	#Edges $\|E\|$	$d_{\max}$	*degree_avg*
Astroph	18,771	198,050	504	21.10
CiaoDVD	4,658	33,116	362	14.22
CondMat	23,133	93,439	279	8.08
DBLP	317,080	1,049,866	343	6.62
Email	1,133	5,451	71	9.62
Facebook	22,470	170,823	709	15.20
GrQc	5,241	14,484	81	5.53
Hamster	2,426	16,631	273	13.71
Jazz	198	2,741	100	27.69
PGP	10,680	24,316	205	4.55
PowerGrid	4,941	6,594	19	2.67
Sex	10,106	39,016	311	7.72

For each dataset, the largest connected component is used, and node features are normalized; the GAT-based initialization follows Section 4.2.

5.3 Baselines

To comprehensively evaluate the proposed HRL-GAT model, it is compared with representative baselines covering classical heuristics, community-aware methods, and deep RL-based approaches for IM:

DC [54] selects seeds with the largest degrees and serves as a simple yet strong topological heuristic baseline.
k-Core [55] decomposition method ranks nodes by their core indices; nodes in higher shells (inner cores) are regarded as more influential.
CSP [21] is a community-aware heuristic that detects structural modules and selects seeds from influential modules using a composite structural score.
S2V-DQN [56] is a generic deep RL framework for combinatorial optimization on graphs, which uses the structure2vec network to embed graph states and a DQN to learn a greedy seed-selection policy.
ToupleGDD [45] couples graph neural networks with a Double DQN architecture, leveraging cascading-aware embeddings to learn a diffusion-aware seed-selection strategy.
BiGDN [57] employs a bidirectional GNN with multi-head attention inside a deep Q-network, and uses pre-trained influence-prediction embeddings to enhance RL-based IM.
ENRENEW [58] is a hybrid neural method that combines enhanced feature extraction with RL, providing a strong learning-based baseline for IM.

5.4 Hyperparameter Analysis

This subsection evaluates the sensitivity of the proposed HRL-GAT framework to key hyperparameters in the reinforcement learning component. To isolate the impact of PPO, the GAT encoder and ECMR-based candidate construction are fixed, and only

Table 2 Expected influence spread of HRL-GAT for different candidate multipliers c under varying seed budgets k.

k	c=2	c=3	c=4	c=5	c=6	c=7	c=8
5	128.30	125.86	128.66	128.09	127.55	**131.10**	129.70
10	196.98	198.78	198.17	193.95	**200.41**	199.39	198.09
15	247.13	247.44	246.83	245.14	247.05	249.24	**249.90**
20	285.91	288.59	285.92	289.62	287.01	289.38	**292.25**
25	318.26	316.80	322.48	322.84	321.08	323.57	**323.90**
30	345.56	347.70	347.94	346.10	346.32	**349.70**	345.56
35	370.39	370.85	375.69	370.39	375.88	**377.13**	373.31
40	394.09	394.74	399.30	396.47	397.12	**399.51**	388.94
45	414.10	413.49	418.99	416.42	416.17	417.32	**419.59**
50	429.60	433.71	433.11	436.44	432.88	**439.50**	436.41

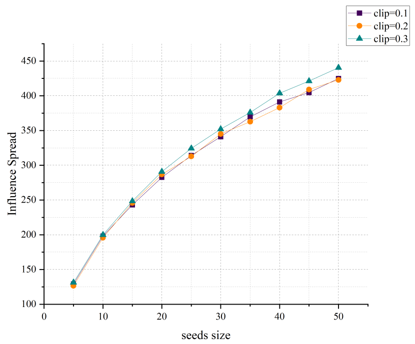
PPO-related hyperparameters are varied. All experiments are conducted on the Email network and averaged over multiple random seeds. The analysis proceeds in two steps: first, the tested ranges are determined based on established practice and prior empirical guidance; second, the experimental outcomes are summarized and interpreted.

5.4.1 Hyperparameter Ranges

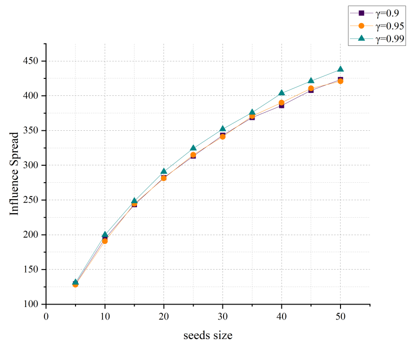
The candidate set size controls the action space available to the agent. Following the common candidate-restriction strategy in learning-based IM and the design in CoreQ, where the candidate list length is parameterized as $m \cdot K$ and a moderate expansion is recommended as a quality–cost compromise [59], the candidate multiplier is set to cover both compact and moderately expanded regimes, namely $c \in \{2, 3, 4, 5, 6, 7, 8\}$. For PPO, the clip parameter is selected from the standard range used in the original PPO formulation and widely adopted in practice, $\epsilon \in \{0.1, 0.2, 0.3\}$ [60]. The discount factor is chosen from typical deep RL values that reflect increasing emphasis on long-horizon credit assignment, $\gamma \in \{0.9, 0.95, 0.99\}$, consistent with common settings and recent discussions on discounting in RL [61, 62]. For GAE, prior results indicate that performance tends to improve as λ increases and becomes relatively stable near $\lambda \approx 1$, with $\lambda \approx 0.95$ often serving as a robust choice [63]; accordingly, the tested range is restricted to the practically relevant high-λ regime $\lambda \in \{0.9, 0.95, 1.0\}$. Finally, the learning rate is tuned around the robust PPO default recommended in implementation studies, taking $\mathrm{lr} \in \{1, 3, 5, 7\} \times 10^{-4}$ centered at 3×10^{-4} [64].

5.4.2 Hyperparameter Experimental Results

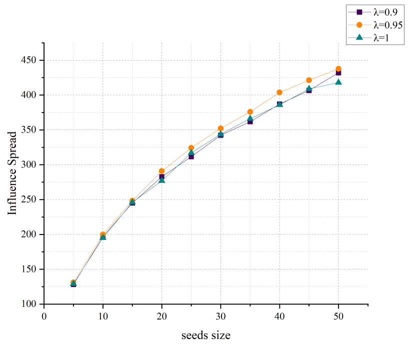
Table 2 reports the influence spread under different candidate multipliers c across seed budgets k. Overall, the spread improves as the candidate pool expands from very compact settings, while gains diminish once c reaches a moderately expanded regime. In particular, the best or near-best performance is consistently achieved when the candidate set size is around $6k$–$8k$, and $c = 7$ provides a stable trade-off across budgets. Consequently, $c = 7$ is used as the default multiplier in the main experiments.



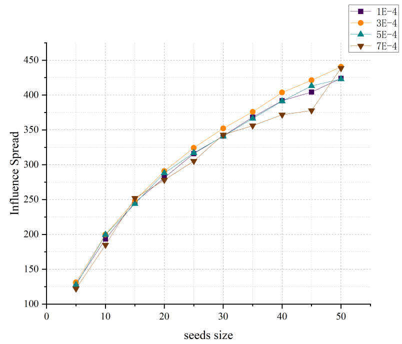
(a) Clipping parameter ϵ.



(b) Discount factor γ.



(c) GAE parameter λ.



(d) Learning rate.

Fig. 2 Sensitivity of PPO-related hyperparameters on the Email dataset under varying seed budgets k.

Fig. 2 summarizes the remaining PPO-related hyperparameters. For the clip parameter, larger ϵ values yield consistently better influence spread in the tested range, and $\epsilon = 0.3$ performs best without observable instability, leading to its adoption as default. For the discount factor, increasing γ improves performance across budgets, and $\gamma = 0.99$ consistently dominates, indicating that emphasizing long-term returns benefits sequential seed selection. For GAE, the three settings produce very close curves, confirming robustness in the high-λ regime, with $\lambda = 0.95$ slightly more favorable and stable overall. For the learning rate, 3×10^{-4} achieves the best or near-best performance across most budgets, whereas smaller values converge more slowly and larger values introduce additional variability; thus, $\mathrm{lr} = 3 \times 10^{-4}$ is selected.

Overall, the results demonstrate that HRL-GAT remains stable within standard PPO tuning ranges. The final configuration adopted in the main experiments is $c = 7$, $\epsilon = 0.3$, $\gamma = 0.99$, $\lambda = 0.95$, and $\mathrm{lr} = 3 \times 10^{-4}$.

5.5 Influence Spread Comparison

Fig. 3 reports the expected influence spread of all methods on the twelve real-world networks under varying seed budgets. Overall, HRL-GAT achieves the highest or

Table 3 Comparison of influence spread between HRL-GAT and its ablation variants on the Email dataset. Best results are highlighted in bold.

Budget (k)	HRL-GAT (Full)	w/o Pretrain	w/o ECMR	w/o RL	w/o Hybrid
5	**132.74**	131.18	124.07	127.26	127.09
10	**199.90**	198.17	193.48	191.49	197.58
15	**248.36**	240.98	239.81	244.69	246.55
20	**290.84**	284.17	275.92	286.86	283.22
25	**324.30**	314.17	306.17	316.28	304.29
30	**352.08**	343.24	339.04	351.17	333.13
35	**375.88**	362.55	366.86	370.28	364.04
40	**403.75**	389.45	395.76	366.49	391.38
45	**421.24**	401.40	406.17	386.77	415.48
50	**440.57**	430.62	432.16	418.10	435.26

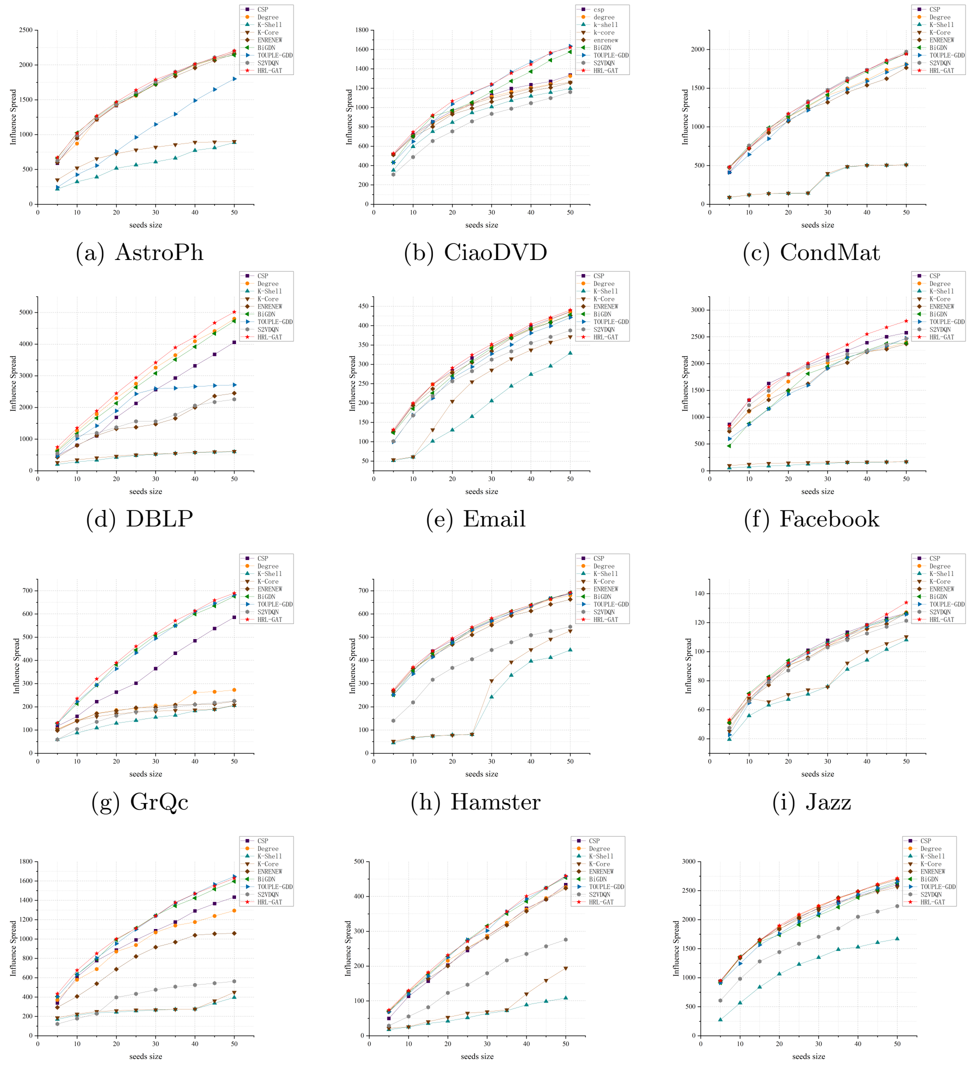
near-highest spread on most datasets. On large and structurally heterogeneous graphs such as AstroPh, DBLP, and Facebook, classical heuristics such as Degree and k-core exhibit a clear performance drop, suggesting that purely local connectivity and static rankings are insufficient to capture diffusion potential. In contrast, HRL-GAT maintains consistently larger cascades, indicating that diffusion-aware embeddings together with sequential selection better exploit complex structures and mitigate redundant seeds.

On smaller or more homogeneous networks such as Jazz, Hamster, and PGP, differences are less pronounced for very small budgets and some heuristics remain competitive. As the budget increases, however, HRL-GAT typically gains more influence than both heuristic baselines and learning-based competitors including S2V-DQN, ToupleGDD, BiGDN, and ENRENEW, reflecting the increasing importance of diversity-aware selection and candidate filtering when more seeds are available. These results support the effectiveness of combining GAT embeddings, ECMR-based candidate construction, and PPO-driven policies for influence maximization.

Fig. 4 compares the running time of different methods under varying seed budgets k. Overall, heuristic baselines are the most efficient. HRL-GAT shows a moderate inference cost that increases with k due to sequential seed selection. Compared with TOUPLE-GDD, HRL-GAT is consistently much faster across all budgets. Compared with S2VDQN, HRL-GAT is faster for small-to-medium budgets, while it becomes slightly slower when k is large. Overall, HRL-GAT achieves a favorable balance between efficiency and influence spread.

5.6 Ablation Study

The ablation study is conducted on the Email dataset to quantify the contribution of each component in HRL-GAT. The full model is compared with four variants: w/o Pretrain removes contrastive pre-training for the GAT encoder, w/o ECMR replaces ECMR with Degree Centrality for candidate construction, w/o RL removes PPO and



(a) AstroPh (b) CiaoDVD (c) CondMat

(d) DBLP (e) Email (f) Facebook

(g) GrQc (h) Hamster (i) Jazz

(j) PGP (k) PowerGrid (l) Sex

Fig. 3 Influence spread of HRL-GAT and baselines on twelve real-world networks under varying seed budgets k.

selects seeds purely by the heuristic score with $\omega = 1.0$, and w/o Hybrid removes heuristic guidance during selection with $\omega = 0.0$. Table 3 reports the expected influence spread under seed budgets $k \in \{5, 10, \ldots, 50\}$.

As shown in Table 3, the full HRL-GAT consistently achieves the best spread for all budgets, demonstrating that the proposed hybrid design is beneficial throughout the entire selection horizon. Replacing ECMR with Degree Centrality leads to the most pronounced degradation, especially at small budgets, where the spread drops

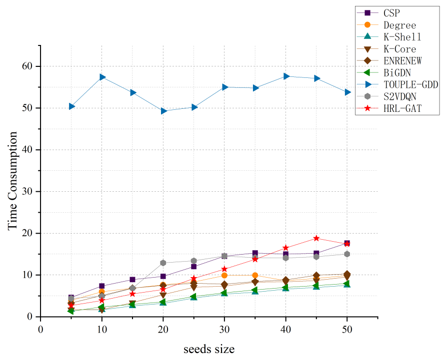


Fig. 4 Running time comparison of all methods on the Email network under different seed budgets k.

from 132.74 to 124.07 at $k = 5$, indicating that diffusion- and structure-aware candidate filtering provides a substantially higher-quality action space. Removing PPO and relying on the heuristic alone becomes increasingly detrimental as k grows, with the spread decreasing to 418.10 at $k = 50$ compared with 440.57 for the full model, suggesting that sequential policy learning is important for reducing overlap and handling diminishing returns in larger seed sets. Disabling pre-training causes a consistent but milder reduction across budgets, for example from 324.30 to 314.17 at $k = 25$, which confirms that diffusion-aligned embeddings improve the quality of the learned policy. Finally, removing the hybrid mechanism also reduces performance, such as 415.48 versus 421.24 at $k = 45$, supporting the role of heuristic guidance in stabilizing exploration and improving decision quality.

Overall, the ablation results confirm that diffusion-aware GAT pre-training, ECMR-based candidate construction, PPO-based sequential optimization, and the hybrid evaluation strategy are all necessary to achieve robust influence maximization performance.

6 Conclusion

This paper addresses the scalability and instability challenges inherent in RL-based IM by proposing HRL-GAT, a novel hybrid framework. The approach bridges structural heuristics and RL through three synergistic components: (i) a contrastively pre-trained GAT that captures fine-grained, diffusion-aware node representations; (ii) a structural prior-guided candidate generation mechanism based on ECMR, which drastically reduces the valid action space; and (iii) a PPO-based actor–critic agent that learns robust sequential seed selection policies.

Extensive empirical evaluations on twelve real-world datasets demonstrate that HRL-GAT consistently outperforms state-of-the-art heuristics, meta-heuristics, and RL baselines in terms of expected influence spread. Furthermore, the ablation studies confirm the critical role of each component, highlighting that integrating ECMR with learning-based optimization significantly enhances both sample efficiency and solution quality. These results suggest that hybridizing domain-specific heuristics with modern

RL is a promising direction for solving large-scale combinatorial optimization problems on graphs.

Future work will focus on two main directions: extending HRL-GAT to dynamic networks where topological structures and diffusion probabilities evolve over time, and adapting the framework to more complex propagation scenarios, such as competitive IM or complementary product adoption.

Declarations

Funding

This work is supported by Hainan Provincial Natural Science Foundation of China (Grant number: 623RC455, 623RC457, 425QN244), Scientific Research Fund of Hainan University (Grant number: KYQD (ZR)-22096, KYQD(ZR)-22097), Lanzhou University-Hainan University Technical Service Project (HD-KYH-2024424).

Author contributions

Funding acquisition was carried out by H.L. and X.M. Methodology was contributed by G.Z. and H.L. Software-related work was done by G.Z. The original draft was written by G.Z., H.L., and X.M. The review and editing of the writing were completed by H.L. and X.M. All authors have read and agreed to the published version of the manuscript.

Conflict of interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Bello-Orgaz, G., Jung, J.J., Camacho, D.: Social big data: Recent achievements and new challenges. Information Fusion **28**, 45–59 (2016) https://doi.org/10.1016/j.inffus.2015.08.005

[2] Eom, Y.-H., Jo, H.-H.: Tail-scope: Using friends to estimate heavy tails of degree distributions in large-scale complex networks. Scientific Reports **5**(1) (2015) https://doi.org/10.1038/srep09752

[3] Jaouadi, M., Ben Romdhane, L.: A survey on influence maximization models. Expert Systems with Applications **248**, 123429 (2024) https://doi.org/10.1016/j.eswa.2024.123429

[4] Solanki, S., Kumar, M., Kumar, R.: A survey on information diffusion and competitive influence maximization in social networks. Social Network Analysis and Mining **15**(1), 41 (2025) https://doi.org/10.1007/s13278-025-01459-2

[5] Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence through a social network. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '03, pp. 137–146. Association for Computing Machinery, New York, NY, USA (2003). https://doi.org/10.1145/956750.956769 . https://doi.org/10.1145/956750.956769

[6] Ma, L., Shao, Z., Li, X., Lin, Q., Li, J., Leung, V.C.M., Nandi, A.K.: Influence maximization in complex networks by using evolutionary deep reinforcement learning. IEEE Transactions on Emerging Topics in Computational Intelligence **7**(4), 995–1009 (2023) https://doi.org/10.1109/TETCI.2021.3136643

[7] Ling, C., Jiang, J., Wang, J., Thai, M., Xue, L., Song, J., Qiu, M., Zhao, L.: Deep Graph Representation Learning and Optimization for Influence Maximization (2023). https://arxiv.org/abs/2305.02200

[8] Song, N., Sheng, W., Sun, Y., Lin, T., Wang, Z., Xu, Z., Yang, F., Zhang, Y., Li, D.: Online dynamic influence maximization based on deep reinforcement learning. Neurocomputing **618**, 129117 (2025) https://doi.org/10.1016/j.neucom.2024.129117

[9] Panagopoulos, G., Tziortziotis, N., Vazirgiannis, M., Pang, J., Malliaros, F.D.: Learning graph representations for influence maximization. Social Network Analysis and Mining **14**(1), 203 (2024) https://doi.org/10.1007/s13278-024-01311-z

[10] Kumar, S., Mallik, A., Khetarpal, A., Panda, B.S.: Influence maximization in social networks using graph embedding and graph neural network. Information Sciences **607**, 1617–1636 (2022) https://doi.org/10.1016/j.ins.2022.06.075

[11] Liu, W., Wang, S., Ding, J.: Influence Maximization Based on Adaptive Graph Convolution Neural Network in Social Networks. Electronics **13**(16), 3110 (2024) https://doi.org/10.3390/electronics13163110

[12] Lin, R., Yao, R., Wang, Y., Lin, J., Wu, Z., Tang, Y.: Influence Maximization in Multi-layer Social Networks Based on Differentiated Graph Embeddings. arXiv (2025). https://doi.org/10.48550/ARXIV.2508.10289

[13] Panagopoulos, G., Tziortziotis, N., Vazirgiannis, M., Malliaros, F.D.: Maximizing Influence with Graph Neural Networks. arXiv (2021). https://doi.org/10.48550/ARXIV.2108.04623

[14] Li, H., Xu, M., Bhowmick, S.S., Rayhan, J.S., Sun, C., Cui, J.: Piano: Influence maximization meets deep reinforcement learning. IEEE Transactions on Computational Social Systems **10**(3), 1288–1300 (2023) https://doi.org/10.1109/TCSS.2022.3164667

[15] Chen, T., Yan, S., Guo, J., Wu, W.: ToupleGDD: A Fine-Designed Solution of Influence Maximization by Deep Reinforcement Learning. IEEE Transactions on

Computational Social Systems **11**(2), 2210–2221 (2024) https://doi.org/10.1109/TCSS.2023.3272331 [cs]

[16] Yang, S., Du, Q., Zhu, G., Cao, J., Chen, L., Qin, W., Wang, Y.: Balanced influence maximization in social networks based on deep reinforcement learning. Neural Networks **169**, 334–351 (2024) https://doi.org/10.1016/j.neunet.2023.10.030

[17] Wang, J., Cao, Z., Xie, C., Li, Y., Liu, J., Gao, Z.: DGN: Influence maximization based on deep reinforcement learning. The Journal of Supercomputing **81**(1), 130 (2025) https://doi.org/10.1007/s11227-024-06621-9

[18] Halal, T., Cautis, B., Groz, B., Gao, R.: Topic-aware influence maximization with deep reinforcement learning and graph attention networks. Data Mining and Knowledge Discovery **39**(6), 71 (2025) https://doi.org/10.1007/s10618-025-01133-3

[19] Chen, H., Wilder, B., Qiu, W., An, B., Rice, E., Tambe, M.: Complex contagion influence maximization: a reinforcement learning approach. In: Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence. IJCAI '23 (2023). https://doi.org/10.24963/ijcai.2023/614 . https://doi.org/10.24963/ijcai.2023/614

[20] Li, Y., Gao, H., Gao, Y., Guo, J., Wu, W.: A Survey on Influence Maximization: From an ML-Based Combinatorial Optimization. ACM Transactions on Knowledge Discovery from Data **17**(9), 1–50 (2023) https://doi.org/10.1145/3604559

[21] Beni, H.A., Bouyer, A., Azimi, S., Rouhi, A., Arasteh, B.: A fast module identification and filtering approach for influence maximization problem in social networks. Information Sciences **640**, 119105 (2023) https://doi.org/10.1016/j.ins.2023.119105

[22] Bouyer, A., Beni, H.A., Arasteh, B., Aghaee, Z., Ghanbarzadeh, R.: FIP: A fast overlapping community-based influence maximization algorithm using probability coefficient of global diffusion in social networks. Expert Systems with Applications **213**, 118869 (2023) https://doi.org/10.1016/j.eswa.2022.118869

[23] Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. ArXiv **abs/1710.10903** (2017)

[24] Gu, Y., Cheng, Y., Chen, C.L.P., Wang, X.: Proximal policy optimization with policy feedback. IEEE Transactions on Systems, Man, and Cybernetics: Systems **52**(7), 4600–4610 (2022) https://doi.org/10.1109/TSMC.2021.3098451

[25] Goyal, A., Lu, W., Lakshmanan, L.V.S.: CELF++: optimizing the greedy algorithm for influence maximization in social networks. In: Proceedings of the 20th

International Conference Companion on World Wide Web, pp. 47–48. ACM, Hyderabad India (2011). https://doi.org/10.1145/1963192.1963217

[26] Tang, Y., Xiao, X., Shi, Y.: Influence maximization: near-optimal time complexity meets practical efficiency. In: Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data. SIGMOD '14, pp. 75–86. Association for Computing Machinery, New York, NY, USA (2014). https://doi.org/10.1145/2588555.2593670 . https://doi.org/10.1145/2588555.2593670

[27] Tang, Y., Shi, Y., Xiao, X.: Influence Maximization in Near-Linear Time: A Martingale Approach. In: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, pp. 1539–1554. ACM, Melbourne Victoria Australia (2015). https://doi.org/10.1145/2723372.2723734

[28] Huang, K., Wang, S., Bevilacqua, G., Xiao, X., Lakshmanan, L.V.S.: Revisiting the stop-and-stare algorithms for influence maximization. Proc. VLDB Endow. **10**(9), 913–924 (2017) https://doi.org/10.14778/3099622.3099623

[29] Gong, Y., Liu, S., Bai, Y.: A probability-driven structure-aware algorithm for influence maximization under independent cascade model. Physica A: Statistical Mechanics and its Applications **583**, 126318 (2021) https://doi.org/10.1016/j.physa.2021.126318

[30] Li, H., Zhang, R., Liu, X.: An efficient discrete differential evolution algorithm based on community structure for influence maximization. Applied Intelligence **52**(11), 12497–12515 (2022) https://doi.org/10.1007/s10489-021-03021-x

[31] Li, H., Zhang, R., Zhao, Z., Liu, X., Yuan, Y.: Identification of top-k influential nodes based on discrete crow search algorithm optimization for influence maximization. Applied Intelligence **51**(11), 7749–7765 (2021) https://doi.org/10.1007/s10489-021-02283-9

[32] Li, H., Zhang, R., Zhao, Z., Yuan, Y.: An Efficient Influence Maximization Algorithm Based on Clique in Social Networks. IEEE Access **7**, 141083–141093 (2019) https://doi.org/10.1109/ACCESS.2019.2943412

[33] Li, H., Zhang, R., Zhao, Z., Liu, X.: LPA-MNI: An Improved Label Propagation Algorithm Based on Modularity and Node Importance for Community Detection. Entropy **23**(5), 497 (2021) https://doi.org/10.3390/e23050497

[34] Kumar, S., Gupta, A., Khatri, I.: CSR: A community based spreaders ranking algorithm for influence maximization in social networks. World Wide Web **25**(6), 2303–2322 (2022) https://doi.org/10.1007/s11280-021-00996-y

[35] Bucur, D., Iacca, G.: Influence Maximization in Social Networks with Genetic Algorithms. In: Squillero, G., Burelli, P. (eds.) Applications of Evolutionary Computation, pp. 379–392. Springer, Cham (2016)

[36] Tang, J., Zhang, R., Yao, Y., Zhao, Z., Wang, P., Li, H., Yuan, J.: Maximizing the spread of influence via the collective intelligence of discrete bat algorithm. Knowledge-Based Systems **160**, 88–103 (2018) https://doi.org/10.1016/j.knosys.2018.06.013

[37] Tang, J., Zhang, R., Yao, Y., Zhao, Z., Chai, B., Li, H.: An adaptive discrete particle swarm optimization for influence maximization based on network community structure. International Journal of Modern Physics C **30**(06), 1950050 (2019) https://doi.org/10.1142/S0129183119500505

[38] Tang, J., Song, S., Du, Q., Yao, Y., Qu, J.: Graph convolutional networks with the self-attention mechanism for adaptive influence maximization in social networks. Complex & Intelligent Systems **10**(6), 8383–8401 (2024) https://doi.org/10.1007/s40747-024-01604-y

[39] Qiu, J., Tang, J., Ma, H., Dong, Y., Wang, K., Tang, J.: DeepInf: Social Influence Prediction with Deep Learning. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2110–2119 (2018). https://doi.org/10.1145/3219819.3220077

[40] Liu, Y., Cao, J., Wu, J., Pi, D.: Modeling the social influence of COVID-19 via personalized propagation with deep learning. World Wide Web **26**(4), 2075–2097 (2023) https://doi.org/10.1007/s11280-022-01129-9

[41] Chowdhury, T., Ling, C., Jiang, J., Wang, J., Thai, M.T., Zhao, L.: Deep graph representation learning for influence maximization with accelerated inference. Neural Networks **180**, 106649 (2024) https://doi.org/10.1016/j.neunet.2024.106649

[42] Wang, Z., Chen, X., Li, X., Du, Y., Lan, X.: Influence maximization based on network representation learning in social network. Intelligent Data Analysis **26**(5), 1321–1340 (2022) https://doi.org/10.3233/IDA-216149

[43] Sonia, Sharma, K., Bajaj, M.: DeepWalk Based Influence Maximization (DWIM): Influence Maximization Using Deep Learning. Intelligent Automation & Soft Computing **35**(1), 1087–1101 (2023) https://doi.org/10.32604/iasc.2023.026134

[44] Liu, C., Fan, C., Zhang, Z.: Finding Influencers in Complex Networks: An Effective Deep Reinforcement Learning Approach. The Computer Journal **67**(2), 463–473 (2024) https://doi.org/10.1093/comjnl/bxac187

[45] Chen, T., Yan, S., Guo, J., Wu, W.: ToupleGDD: A Fine-Designed Solution of Influence Maximization by Deep Reinforcement Learning. IEEE Transactions on Computational Social Systems **11**(2), 2210–2221 (2024) https://doi.org/10.1109/TCSS.2023.3272331 . arXiv:2210.07500 [cs]

[46] Sun, Y., Wu, J., Song, N., Lin, T., Li, L., Li, D.: Deep reinforcement learning-based influence maximization for heterogeneous hypergraphs. Physica A: Statistical Mechanics and its Applications **660**, 130361 (2025) https://doi.org/10.1016/j.physa.2025.130361

[47] Ali, K., Wang, C.-Y., Yeh, M.-Y., Chen, Y.-S.: Addressing Competitive Influence Maximization on Unknown Social Network with Deep Reinforcement Learning. In: 2020 IEEE/ACM International Conference on Advances in Social Networks Analysis And Mining (ASONAM), pp. 196–203. IEEE, The Hague, Netherlands (2020). https://doi.org/10.1109/ASONAM49781.2020.9381471

[48] Liu, Y., Sze, W., Gao, X., Chen, G.: Multiple Agents Reinforcement Learning Based Influence Maximization in Social Network Services. In: Hacid, H., Kao, O., Mecella, M., Moha, N., Paik, H.-y. (eds.) Service-Oriented Computing vol. 13121, pp. 431–445. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-91431-8_27

[49] Leskovec, J., Krevl, A.: SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data (2014)

[50] Rossi, R.A., Ahmed, N.K.: The network data repository with interactive graph analytics and visualization. In: AAAI (2015). https://networkrepository.com

[51] Rozemberczki, B., Allen, C., Sarkar, R.: Multi-scale Attributed Node Embedding (2019)

[52] Guo, G., Zhang, J., Thalmann, D., Yorke-Smith, N.: Etaf: An extended trust antecedents framework for trust prediction. In: Proceedings of the 2014 International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 540–547 (2014)

[53] Rocha, L.E.C., Liljeros, F., Holme, P.: Simulated Epidemics in an Empirical Spatiotemporal Network of 50,185 Sexual Contacts. PLOS Computational Biology **7**(3), 1–9 (2011) https://doi.org/10.1371/journal.pcbi.1001109 . Publisher: Public Library of Science

[54] Freeman, L.C.: Centrality in social networks conceptual clarification. Social Networks **1**(3), 215–239 (1978) https://doi.org/10.1016/0378-8733(78)90021-7

[55] Dorogovtsev, S.N., Goltsev, A.V., Mendes, J.F.F.: k -Core Organization of Complex Networks. Physical Review Letters **96**(4), 040601 (2006) https://doi.org/10.1103/PhysRevLett.96.040601

[56] Dai, H., Khalil, E.B., Zhang, Y., Dilkina, B., Song, L.: Learning combinatorial optimization algorithms over graphs. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. NIPS'17, pp. 6351–6361. Curran Associates Inc., Red Hook, NY, USA (2017)

[57] Zhu, W., Zhang, K., Zhong, J., Hou, C., Ji, J.: BiGDN: An end-to-end influence maximization framework based on deep reinforcement learning and graph neural networks. Expert Systems with Applications **270**, 126384 (2025) https://doi.org/10.1016/j.eswa.2025.126384

[58] Zareie, A., Sheikhahmadi, A., Jalili, M., Fasaei, M.S.K.: Finding influential nodes in social networks based on neighborhood correlation coefficient. Knowledge-Based Systems **194**, 105580 (2020) https://doi.org/10.1016/j.knosys.2020.105580

[59] Ahmad, W., Wang, B.: A learning-based influence maximization framework for complex networks via K-core hierarchies and reinforcement learning. Expert Systems with Applications **259**, 125393 (2025) https://doi.org/10.1016/j.eswa.2024.125393

[60] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal Policy Optimization Algorithms. arXiv (2017). https://doi.org/10.48550/arXiv.1707.06347

[61] Kim, M., Kim, J.-S., Choi, M., Park, J.-H.: Adaptive discount factor for deep reinforcement learning in continuing tasks with uncertainty. Sensors (Basel, Switzerland) **22** (2022)

[62] Tang, Y., Rowland, M., Munos, R., Valko, M.: Taylor Expansion of Discount Factors (2021). https://arxiv.org/abs/2106.06170

[63] Schulman, J., Moritz, P., Levine, S., Jordan, M., Abbeel, P.: High-Dimensional Continuous Control Using Generalized Advantage Estimation (2018). https://arxiv.org/abs/1506.02438

[64] Huang, S., Dossa, R.F.J., Raffin, A., Kanervisto, A., Wang, W.: The 37 implementation details of proximal policy optimization. In: The ICLR Blog Track 2023 (2022). https://elib.dlr.de/191986/