2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)

Addressing Competitive Influence Maximization on Unknown Social Network with Deep Reinforcement Learning

Khurshed Ali*‡, Chih-Yu Wang†, Mi-Yen Yeh*, and Yi-Shin Chen‡
*TIGP-SNHCC, Institute of Information Science, Academia Sinica, Taipei, Taiwan
†Research Center for Information Technology Innovation, Academia Sinica, Taipei, Taiwan
‡Institute of Systems and Applications, National Tsing Hua University, Hsinchu, Taiwan
Email: *(khurshed, miyen)@iis.sinica.edu.tw, †cywang@citi.sinica.edu.tw, ‡yishin@gmail.com

***Abstract*—Recent studies have considered the reinforcement and deep reinforcement learning models to address the competitive influence maximization (CIM) problem. However, these models assume complete network topology information is available to address the CIM problem. This assumption is unrealistic as it is difficult to obtain complete social network data and requires exhaustive efforts to obtain it. In this work, we propose a deep reinforcement learning-based (DRL) model to tackle the competitive influence maximization on unknown social networks. Our proposed model has a two-fold objective: the first is to identify the time when to explore the network to collect network information. The second is to determine key influential users from the explored network, using optimal seed-selection strategy considering the competition in the social network. Moreover, we integrate the transfer learning in DRL to improve the training efficiency of DRL models. Experimental results show that our proposed DRL and transfer learning-based DRL models achieve significantly better performance than heuristic-based methods.**

***Index Terms*—Influence Maximization, Competitive Influence Maximization, Deep Reinforcement Learning, Social Network Analysis, Transfer Learning**

I. Introduction

Companies employ social media, such as Facebook, Twitter, and YouTube, to promote their products among the masses. These companies select the key users in a social network who can spread the company product's information and expect many users to buy the product. Such a selection of key influential users with an expectation of maximum reward is known as the *influence maximization (IM)* problem. The IM problem has been widely studied in the research domain [1]–[6].

A more realistic and practical scenario is when multiple companies, such as Samsung and HTC, are promoting similar products (smartphones) in a social network to maximize their respective profit than the competitor. This problem has been termed as a *competitive influence maximization (CIM)*. Bharathi *et. al* [7] is the pioneer to address the CIM problem using game-theory. Some studies addressed the CIM problem by extending traditional IM-based approaches [8]–[10]. Recently, reinforcement and deep-reinforcement learning-based approaches are proposed to tackle the CIM problem [11]–[14]. However, these studies assume that the complete structure of the network is known, and are only concerned about selecting key users to maximize the profit. This assumption is impractical in a scenario where companies are unaware of the underlying social network, and complete network topology is not given. Under such unknown and partially visible networks, companies need to first explore the network before selecting key users for maximum information spread.

Consider a motivational example shown in Figure 1 where two parties are promoting their products for two rounds. Assume that parties can select a single user to propagate information or explore the network through multiple users at each round but can not do both simultaneously. Figure 1 (a) represents a given social network where users v_1, v_2, v_6, and v_8 links are known while other users connections are hidden. Let Parties A and B select v_6 and v_2 considering the user's number of neighbors for product promotion in the first round. Party A can influence v_4, v_7, and v_{13} users, while party B can influence v_5 only since the first party already activates v_4. In the second round, party A and party B choose users v_1 and v_8, respectively, as seed nodes, resulting in party A's information propagation to v_3 and party B to none since users v_7 and v_8 are activated. In total, Party A activates six users, and party B influences three users, including seeds after two rounds, as shown in Figure 1 (b). However, if parties decide to explore the network in the first round, then both parties get more network visibility to select better influential users. For instance, party A explores the network through users v_4, v_9 while party B probes through users v_{11} and v_{12}. In the second round, party A decides to select user v_4 as a seed node with more neighbors while party B chooses user v_1 as a seed node. In total, party A activates nine users while B activates four users (as shown in Figure 1(c)). Thus, it is critical to find an optimal strategy that assists when to probe the network and select key influential users from the observed network to promote the product in the social network.

In light of the above issues, we address the following

*Chih-Yu Wang is the corresponding author. This work was supported by the Ministry of Science and Technology under Grant MOST 105-2221-E-001-003-MY3, 108-2628-E-001-003-MY3, 107-2221-E-001-009-MY3, 106-3114-E-002-008, and the Academia Sinica under Thematic Research Grant.

IEEE/ACM ASONAM 2020, December 7-10, 2020
978-1-7281-1056-1/20/$31.00 © 2020 IEEE

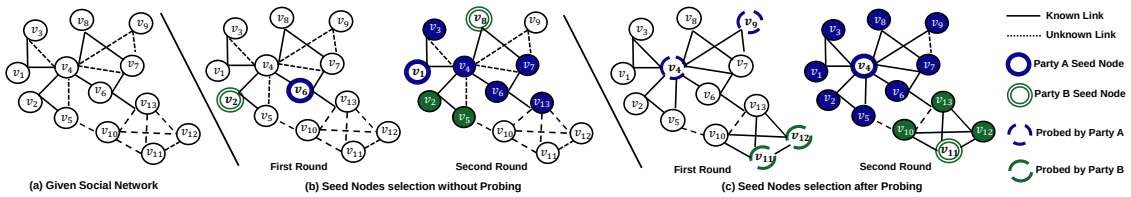


Fig. 1: Motivational Example

challenging questions. First, when to explore the network? Second, how to select the key users after observing the partial network given that other parties are competing in the same network? Consequently, we propose a competitive influence maximization on an unknown social network (CIM-UN) problem. This problem considers multiple parties need to devise the strategy to maximize their profit by probing and investing the budget on the social network. Inspired by the impressive performance of *deep reinforcement learning* (DRL) models [15]–[17], we propose a DRL-based framework to solve the CIM problem on an unknown network by finding an optimal policy that determines when to explore the network and how to select seed nodes. Further, the proposed framework considers competition, budget, and probing limit constraints when finding the optimal strategy to maximize the reward. To improve the DRL model's training efficiency, we integrate transfer learning in DRL to tackle the CIM problem on unknown social networks.

In short, our major contributions are as follows.

- To the best of our knowledge, we are the first to address a competitive influence maximization on an unknown social network using a competitive linear threshold model.
- We propose a DRL-based framework to find a trade-off between probing the network and investing the budget to maximize the reward. The DRL-based agent needs to find an optimal strategy that consists of when to explore the unknown network to get more network visibility and select influential seed nodes from the explored network.
- To boost the training efficiency, we integrate the transfer learning in DRL to quickly learn the policy on unknown target networks.

II. Related Work

Influence Maximization (IM) problem has got ample research attention for more than a decade. Domingo and Richardson *et al.* [1], [2], in their seminal work, formulated it as an algorithm problem and utilized Markov Random field to tackle it. Kempe *et al.* [3] proved the IM problem's NP-hardness and proposed a greedy-based algorithm that achieved an approximation ratio of $(1 - 1/e)$. However, the greed-based approach was computationally expensive and lacks scalability on large networks. Consequently, subsequent studies addressed the scalability, efficiency, and quality of the IM problem [4], [18]. Monte-Carlo [19]–[21] and heuristic-based methods [18], [22] are also proposed to address the scalability issue of IM problem.

Competitive Influence Maximization (CIM) is a more practical scenario of the IM problem where multiple companies are campaigning on a social network considering their budget constraints and maximizing their respective profit. Bharti *et al.* [7] is the pioneer to formulate the CIM problem and addressed it using game theory. Further, Bharti *et al.* [7] proved that at least $(1 - 1/e)$ optimal influence spread can be obtained when the other party' seed nodes are accurately predicted. Many research studies are proposed to address the CIM problem when the competitor's strategy is known [8], [9], [23]–[25]. Most of these studies extended the traditional IM-based approaches to address the CIM problem.

In recent studies, learning-based approaches are proposed to tackle the CIM problem [11]–[14]. Lin *et al.* [11] proposed a reinforcement learning-based (RL) method to tackle the CIM problem by considering seeds selection in multiple rounds. Seeds selection in the first round, as traditional approaches, and single strategy is not an optimal choice across multiple social networks, as discussed in [11]. Further, Ali *et al.* [13] proposed an RL-based model to tackle the time-constrained CIM problem. The main objective of their work [13] is to find an optimal time to invest budget and the strategy to select seeds for maximizing the reward. Ali *et al.* [12] employed transfer learning (TL) in RL to boost the RL-based model's training efficiency for CIM problem. Recently, Chung *et al.* [14] proposed a DRL-based approach to address the CIM problem. Chung *et al.* [14] assumed the complete network information is provided and employed a community-based quota policy to invest budget. However, the complete network information is difficult to obtain and need to be collected through extensive work such as surveys and so. In this work, we propose a DRL-based method to explore the unknown network and select influential users from an explored network to maximize the reward.

III. Problem Formulation

We model a social network, $G = (V, E)$, as a weighted and directed graph for competitive influence maximization. In the beginning, only the set of nodes $V = \{v_1, v_2, ...v_n\}$ is known, and E, set of edges, is unknown. Here, the total number of nodes is represented as $|V|$, and an edge represents a relationship between two users in the social network. Further, there are set of $\mathcal{P} = \{p_1, p_2, ...p_k\}$ parties promoting their products, and each user v can buy at most one product from the parties. Besides, there are T rounds where each party p can probe m_p nodes or select seed nodes k_p at each round. Each party has a maximum of K_p budget

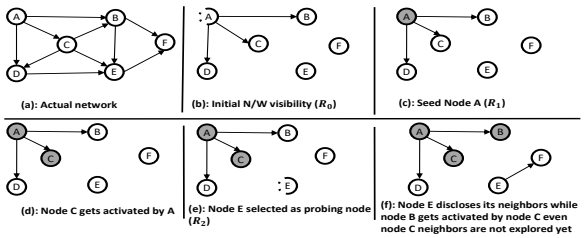


Fig. 2: Influence Propagation and Probing Example

to invest. Once parties have selected seed nodes or probed the network at each round, we propagate each party's ideas at the same time using the competitive diffusion process. A node u gets activated by party p if it accepts the idea of party p or remains inactive if not activated by any party. We have used a competitive linear threshold model [10], [11] as the competitive diffusion process in our experiments.

Definition 3.1: COMPETITIVE LINEAR THRESHOLD (CLT) [10], [11]: Given a graph $G = (V, E)$ and set of party $\mathcal{P}$, each node v picks an activation threshold ρ_v. At round t, the node v is activated by party $p \in \mathcal{P}$ if the total weight of its active in-neighbors exceeds the activation threshold, i.e., $\sum_{u \in \mathcal{O}_t^p} w_{u,v} > \rho_v$, where $\mathcal{O}_t^p \subseteq \mathcal{V}$ is the set of activated nodes by party p before t^{th} round and $w_{u,v}$ is the edge weight from node u to v.

We propagate each party's influence at the same time once the seed nodes are selected, or the network is probed. The rule of conflict is addressed by majority rule. The majority rule is that the node v gets activated by party p whose total influence is highest on node v, that is , $\sum_{u \in \mathcal{O}_t^p} w_{u,v} > \sum_{u \in \mathcal{O}_t^j} w_{u,v}$ than any other party j. If two parties have the same highest influence to activate the node v, then the first party gets a preference to activate node v. Moreover, we illustrate the working of information diffusion and probing in an unknown social network in Figure 2.

Figure 2 (a) shows the real network topology information. However, we are provided an unknown or partially visible network such as Figure 2 (b) before training. The party determines to invest budget and selects a node with a maximum degree from an explored network, that is, node A in the first round (R_1). We propagate the influence of node A and assume that node C gets activated. In the next round, R_2, the party determines to explore the network rather than selecting a seed node(s). Node E is selected as probing node and discloses its neighbors as in Figure 2 (f). After probing, we propagate the influence of node C, activated in the previous round, to its neighbors, even its neighbors are not disclosed yet. In short, parties can select the seed nodes from the explored network while influence is propagated considering the real network topology (as Figure 2 (a)). Intuition for such information diffusion is like the company campaigning its product information on the Facebook network, even without knowing the users' actual friendship connections.

Definition 3.2: COMPETITIVE INFLUENCE MAXIMIZATION ON UNKNOWN SOCIAL NETWORK (CIM-UN): The CIM-UN problem consist of T rounds, where each party p determines to select k_p seed nodes or probe m_p nodes from the unknown network G to explore the network at each round. The influence is propagated using the CLT diffusion model at each round and continues till maximum T rounds, or no more nodes are left to get activated.

The objective of each party is to find an optimal policy against the competitor's strategy to maximize the reward. The optimal policy is to find a trade-off between exploring the network and investing the budget to maximize the reward taking budget constraints and competition into consideration.

IV. METHODOLOGY

A. Background

Reinforcement learning (RL) is a machine learning technique where an agent learns to find an optimal policy to solve the task (or maximize the accumulative reward) by keeping interacting with the environment [26]. RL models value function (V) and action function (Q) to estimate how good the policy (π) is in maximizing the reward in the long run. Though RL has achieved impressive results in various applications, it is still intractable to tackle large-scale problems due to high-dimensional state-action pair space growth. The recent success and fast computational resources allow us to tackle the high dimensional and complex problems using deep learning. The deep learning can provide the approximated Q-values with the help of neural networks rather than learning action values at every state through Q-learning. Mnih *et. al* [15], [16] is a pioneer who proposed a Deep Q-Network (DQN) consisting of neural network and multiple hidden layers. Deep Q-network outputs the vector of action values, that is, $Q(s, \cdot; \theta)$, in m size when provided an n-dimensional input state. To overcome the instability of the Q estimation, Mnih *et. al* [15] proposed a target Q network similar to online network except the parameters θ^- are frozen for a certain iterations and later updated through the online network, i.e., $\theta^- = \theta$. Experience replay is another key component of the Deep Q-network framework for stability [27].

B. Proposed Framework

Although RL-based models have achieved significant success in tackling the competitive influence maximization [11]–[13] on social networks, these models lack scalability to tackle the state space growth of social networks. Chung *et al.* [14] proposed a DRL-based approach to address the CIM problem by integrating the community structure of the social network with it. Their proposed model assumes the complete network topology is visible. However, such an assumption is impractical in real settings where it is hard to obtain complete network data. In this work, we propose a Deep Reinforcement Learning-based framework to address the CIM on unknown social networks.

Figure 3 illustrates the flow of our proposed deep q-learning framework. The agent is provided a partially explored state of the network and determines a set of actions through deep Q network to achieve the maximum reward, that is, a number of nodes activated. Different from [14], an agent needs to

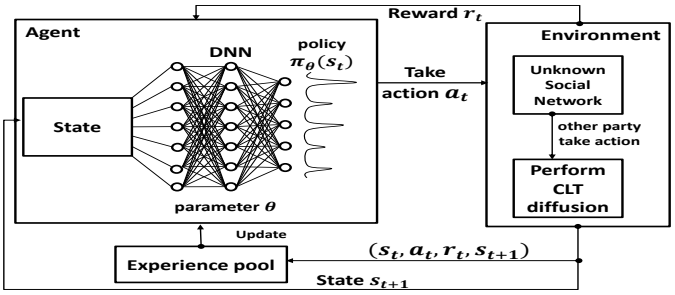


Fig. 3: Deep Q-learning framework

determine the policy when to probe (explore) the network and when to invest the budget given the partially observed network state. At the first time-stamp, we only know the number of nodes in the network and initial network visibility, so the state is computed considering this information only. With the elapsed training, the network is gradually explored if the agent chooses a probing strategy, and the state would be computed based on the up-to-date current network visibility status. We explore the network using the popular 'Jump-Crawl' method [28] when the agent determines the best action as probing given the current state of the network. We propagate the influence of multiple parties using the CLT diffusion model after parties determined their action, i.e., selected key seed nodes k_p, among network and then stored the state transition in the experience pool.

The proposed DRL agent's objective is to learn an optimal policy, π, that consists of probing and seed selection strategy, which maximizes its expected accumulated reward.

Environment. We represent the competitive influence diffusion as an environment. It disseminates the active nodes' influence on other in-active nodes using a competitive linear threshold model, as discussed in section III.

Reward. The DRL agent receives the reward as the delayed reward, that is, the number of nodes activated till the last round. Specifically, r_t^p as, 0 if $t < T$ and $|\mathcal{V}^p|$ if $t = T$, where $|\mathcal{V}^p|$ denotes the number of nodes influenced by party p till the last round.

Action. We extend the meta-learning approach proposed in [11], [14] for action space to include the probing strategy to explore the network. Specifically, we include a 'Jump-Crawl' [28] method in existing meta-based action space for exploring the network. The agent can probe m nodes to explore the network or select any existing IM-based strategy to invest the budget k, that is, select seed nodes. Further, we select seed nodes considering the explored social network contrary to a complete topology consideration as in [11], [14]

- Jump-Crawl: The idea of this approach is to either jump to a uniformly random node for exploring the network or crawl along an edge from the set of an already visited node to one of its neighbors. Once a node is visited, it reveals all of its neighbors [28], [29].
- IM-Based Strategies for seed selection: We select seed nodes from the explored network using *MaxDegree, MaxWeight, Blocking, SubGreedy, or Voting* strategy. Readers are requested to refer [11], [13] for complete IM-based strategy discussion.

State. The main backbone for our transfer learning in DRL is to design state features in such a way that any unknown social network with different topological structures can have similar state representations. When networks of varying sizes are represented in the same representation, then it would be easy to transform the learned policy from one network to another without redefining the neural network structure for each network. We revised the state features proposed in [11] to accommodate the unknown social network.

1) Number of unexplored nodes
2) Number of in-active nodes
3) Maximum out-degree of explored but in-active nodes
4) Maximum out-edge weight summation of in-active explored nodes
5) Summation of the out-edge weight of in-active explored nodes, which are neighbors of second party active nodes.

We normalize each state feature with its original value and transform them into numerical representation such as [3, 2, 1, 0] to avoid the network scale gap variation from network to network. We discuss the state normalization process in the following example with only two features, and the remaining features normalization is carried out in the same way.

Example: Let us refer to the same unknown social network given in Figure 2 (b). There are six nodes in the network, and only node A is explored (probed) at this time. So, we normalize the first feature as the number of unexplored nodes divided by the total number of nodes, that is, $5/6 = 0.83$. Now, let us consider the third feature for normalization, that is, the maximum out-degree of in-active explored nodes. Since we do not know the maximum degree as edges are unknown except N, the total number of nodes. We assume that at-most a node can have $(N-1)$ neighbors in a social network. So, the third state feature is normalized as $3/5 = 0.6$, that is, an out-degree of node A divided by a maximum degree in a social network. Further, We represent the decimal digits of normalized features in numerical form to avoid the decimal digits in state representation. For instance; the decimal value greater than 0.6 is represented as 3, decimal value > 0.2 as 2, decimal value > 0 as 1, and 0 otherwise.

Such normalized state features representation, as a one-dimensional vector representation, will assist the neural network to have the same hyper-parameters of the network state and transform the learned hyper-parameters from one unknown social network to another easily. Besides, DRL will leverage the past learning to learn on a new unknown target network quickly.

Deep Q network. We create a Deep Q network similar to the neural network structures proposed in [14], [16] in our proposed framework to estimate the Q-value. Contrary to [14], we compute the state features to represent the environment state based on an unknown social network and its explored part, as discussed earlier.

C. Transfer Learning in DRL for CIM

Ali *et. al* [12] proposed a transfer learning (TL) in reinforcement learning to tackle the CIM problem. Recently, some studies have considered integrating transfer learning in DRL [30], [31]. Du *et. al* [31] discusses the integration of TL in DRL by utilizing deep learning models transfer techniques. That is, either apply the pre-trained models directly without further training on the new task or transform the learned weights from pre-trained models for new tasks and fine-tune it before evaluation [32]. Likewise, we propose a similar approach to integrate transfer learning in DRL to tackle the CIM problem on an unknown social network. First, we train the model on a smaller unknown social network, termed as Unknown Source Social Network, for hundreds of thousands of times. It takes less training time to learn the policy as an unknown network is quite small. Once we get the pre-trained model, we then copy the weights, biases, and Q-values from it for further fine-tuning or evaluation on a larger unknown target network, termed Unknown Target Social Network. This way, it avoids the random initialization of weights and biases on an unknown target network and learns faster by utilizing pre-trained weights. Moreover, our proposed DRL and transfer learning in DRL methods are discussed in Algorithms 1 and 2, respectively.

Algorithm 1 DRL for competitive influence maximization on unknown social network (DRL-UN)

Initialize Q action-value function with random weights θ
Initialize target $\hat{Q}$ action-value function with weights $\theta^{-} = \theta$
Initialize experience pool $\mathcal{M}$ to capacity N
Initialize ϵ-decay as 1, anneal to 0.1 with training, learning rate γ = 0.00025
for training episode $s = 1, S$ **do**
    $s_t \leftarrow s_0$, $m_t \leftarrow m_0$
    for $t = 1, T$ **do**
        Select a random action with probability ϵ, otherwise choose action $a_t = argmax_a Q(\phi(s_t), a; \theta)$
        Simulate the competitor' strategy
        Propagate diffusion using CLT model and observe next state s_{t+1}, and reward r_t
        Store transition (s_t, a_t, r_t, s_{t+1}) in $\mathcal{M}$
        $s_t \leftarrow s_{t+1}$
        Sample random minibatch of transitions from $\mathcal{M}$
        Update the Q action-value function
        Reset $\theta^{-} = \theta$ after every C steps
    end for
end for

V. Experiments

A. Experimental Setup

We evaluate the following hypothesis for our proposed DRL and TL-based DRL models in our experiments:

- H1. Can DRL-based models achieve acceptable performance on unknown networks than the heuristic-based methods?

Algorithm 2 Transfer Learning in DRL (DRL-UN(TL))

Initialize Q action-value function with pre-saved weights θ
Initialize target $\hat{Q}$ action-value function with weights $\theta^{-} = \theta$
Initialize experience pool $\mathcal{M}$ to capacity N
Initialize ϵ-decay as 0.2 and learning rate γ = 0.01
for Training episode $s = 1, S$ **do**
    Compute remaining training process similar to DRL-UN algorithm
end for

- H2. If the DRL-based model is trained on small unknown source network, can it achieve attainable performance on unknown target networks with little further training?

We compare the DRL-based models (either trained from scratch or transferred models) performance of unknown social networks with the DRL model having complete network visibility along with Heuristic-based methods. Following is the list of implemented models along with comparison methods.

- **DRL-UN**: DRL Algorithm trained on a target network from scratch for $100,000$ training episodes with initial network visibility as 0.1% of total nodes. Each episode consists of T rounds for action selection (seed selection or probing) and influence propagation. Initial network visibility means 0.1% of the nodes, and their relationships are visible at the start of each training episode. This 0.1% visibility is uniformly drawn from pre-generated 5000 different network visibility random lists for training.
- **DRL-UN(CEL)**: DRL-UN Algorithm trained on the Celegan source network for $200,000$ training episodes. Since Celegan is a small source network, so we choose initial network visibility as 1% for training. Further, we evaluate this model on unknown target networks as-is without fine-tuning.
- **DRL-UN(TL)**: We fine-tune the DRL-UN(CEL) model on unknown target networks for further $30,000$ training episodes with initial network visibility as 0.1%.
- **DRL-OPT**: Deep Q-learning agent trained on a target network for $20,000$ training episodes with complete network visibility.
- **FD: Fixed Degree**: A heuristic strategy that selects seed nodes having a maximum degree at each round without probing the network.
- **AFD: Alternate Fixed Degree**: A heuristic method that probes and selects seed nodes in alternate rounds using *Jump-Crawl* and *MaxDegree* strategies respectively.
- **FR: Fixed Random**: A heuristic strategy that selects seed nodes randomly at each round.
- **AFR: Alternate Fixed Random**: A heuristic strategy that probes and selects seed nodes in alternate rounds using *Jump-Crawl* and *Random* strategies respectively.

We conducted experiments on four different real-world social networks. We selected the Celegan network [1] as an unknown source network while other three networks, as

[1] https://graph-tool.skewed.de/static/doc/collection.html

unknown target networks, are downloaded from Stanford Network Collection [33] website. More information about the statistics of the network is discussed in Table I. In our experiments, we use the CLT diffusion model as discussed in Section III. Further, we used weighted-cased model [19], [24] and fixed edge-weights as 0.4 in evaluation.

TABLE I: Datasets

Name	#Nodes	#Edges	Description
Unknown Source Network			
Celegan (CEL)	297	2,359	A directed neural network
Unknown Target Networks			
Facebook (FB)	4,039	88,234	Social circles from Facebook
Ca-GrQc	5,242	14,496	Collaboration network of Arxiv General Relativity
P2P-Gnutella (P2P)	6,301	20,777	A snapshot of P2P network

Data for training and evaluation: Since we do not have real-time or human-annotated data to train and evaluate our proposed models, so, we generated 5000 initial network visibility random lists of source and target networks for training. Initial network visibility of target networks is set as 0.1% and 1% of the Celegan source network for training. This initial visibility random lists make DRL-based models' training stable by not beginning training from too much random initial visibility. Further, we generated 2000 different initial visibility random lists of unknown target networks for evaluation. We evaluated each method by running against these 2000 different random lists and present an average competition reward of the first party. Each method is evaluated by competing against a competitor's **AFD** strategy. We set a number of rounds, $T = 10$, for action selection, that is, probing or seed-selection, and influence propagation. Besides, Each method can invest budget $k = 1$ in each round when a seed-selection strategy is chosen. We evaluate each method's performance against different initial network visibility settings $n_v = \{0.1, 0.4, 0.7, 1, 3\}$.

B. Experimental Results

1) When the transfer learning-based DRL model is transferred in the same settings: In the first experiment, we set the number of nodes to probe by each party m_p as 10 if the probing strategy is selected as an action. We train the DRL-UN and DRL-UN(CEL) using weighted-cased edge-weights while the number of nodes to probe is fixed as 10 during training and evaluation.

Table II presents the model's performance on three unknown target networks using a weighted cascade model. Column heading under each network represents the initial visibility of the respective target networks. Cell values represent the average number of nodes activated by the first party. DRL-based models, that is, DRL-UN, DRL-UN(CEL), and DRL-UN(TL), performed better than the heuristic methods in all networks in most network visibility settings except on FB network when network visibility is low such as 0.1, and 0.7. We found that DRL-based models took *probing* strategy on FB network only once when the visibility was low and achieved less performance than the *AFD* strategy, which explored and invested budget in alternate rounds. However, the performance of *AFD* strategy is inconsistent on other networks and achieved less reward than the *FD* strategy. Besides, DRL-UN(TL) model achieved significantly better reward than the DRL-UN(CEL) model on all the networks when network visibility was higher. Further, the last row represents the DRL-OPT model's reward with complete network visibility. Intuitively, the DRL model can perform better when having complete network topology information. Besides, the naive random-based strategies, that is, FR and AFR, are not as competitive as the other two heuristic strategies compared with DRL-based models. So, we omitted the results of these two naive random-based strategies results in our following experimental results.

2) When the transfer learning-based DRL model is transferred in different settings: In this experiment, we train the DRL-UN(CEL) source model using weighted-cased edge-weights and number of nodes to probe as 10. Further, we transfer this pre-trained model and fine-tune it on target networks with the number of nodes to probe as 20 if the probing strategy is selected. Besides, the DRL-UN model is trained by selecting a number of nodes as 20.

Table III presents the DRL and heuristic models results when the number of nodes to probe is set at 20. We can observe from results in Table III that the number of nodes activated by the first party is higher than the nodes activated with probing set as 10 (as in results II) when using weighted cascade edge weights in both settings. This shows that if we increase the number of nodes to probe during training and evaluation, then the model can explore more network and have a higher chance of selecting influential users that can activate more users. DRL-UN, DRL-UN(CEL), and DRL-UN(TL) models perform better than the heuristic methods in all networks except in the Facebook network, where AFD strategy performed better in most network visibility settings. DRL-based models took *probing* strategy few times (maybe once) on FB network when network visibility was low and selected seeds from this explored part, which resulted in less reward. Similar to results shown in Table II, the performance of heuristic strategies, *FD and AFD*, is inconsistent on all the network. Further, DRL-UN(CEL) model, without further retraining to accommodate different settings, could not achieve better reward than other two DRL-based models.

3) Adaptability of DRL models with different structure: Next, we discuss the DRL model's adaptability to a different network structure, such as changing edge-weights from weighted cascade to 0.4 for all the edges. Our objective is to analyze the effect of different edge-weights than the DRL models were trained. Table IV presents the DRL-based, and naive heuristic-based models results evaluated using fixed edge-weights, that is, 0.4, and the number of nodes to probe is set as 10. More nodes are activated when we fixed the

TABLE II: When models are evaluated using weighted cascade edge-weights and nodes to probe $= 10$

	FB					P2P					Ca-GrQc				
	0.1	0.4	0.7	1	3	0.1	0.4	0.7	1	3	0.1	0.4	0.7	1	3
Model															
Heuric-Based															
FD	8	30	40	51	115	15	39	46	48	51	8	23	27	26	29
AFD	43	45	49	65	124	24	24	24	26	29	14	14	14	14	16
FR	9	15	13	19	15	14	22	23	23	23	8	18	18	17	16
AFR	8	8	9	9	7	11	11	11	11	11	8	8	8	8	8
DRL-Based															
DRL-UN	**30**	**37**	**53**	**61**	**135**	**31**	**47**	**57**	**62**	**83**	**21**	**29**	**33**	**36**	**47**
DRL-UN(CEL)	23	36	53	62	156	30	46	55	60	77	20	24	25	26	31
DRL-UN(TL)	**23**	**33**	**51**	**65**	**155**	**30**	**50**	**64**	**71**	**98**	**21**	**27**	**30**	**33**	**42**
DRL-OPT			2432					200					72		

TABLE III: When models are evaluated using weighted cascade edge-weights and nodes to probe $= 20$

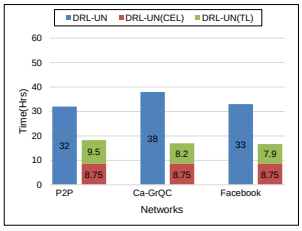
	FB					P2P					Ca-GrQc				
	0.1	0.4	0.7	1	3	0.1	0.4	0.7	1	3	0.1	0.4	0.7	1	3
Model															
Heuric-Based															
FD	8	30	46	51	126	15	39	45	48	51	8	23	25	26	29
AFD	54	59	78	80	145	26	25	26	27	31	12	11	11	11	11
DRL-Based															
DRL-UN	**28**	**36**	**55**	**63**	**137**	**44**	**65**	**81**	**91**	**129**	**27**	**32**	**35**	**38**	**48**
DRL-UN(CEL)	36	42	55	70	161	38	49	56	61	78	22	24	26	27	31
DRL-UN(TL)	**31**	**43**	**60**	**73**	**162**	**44**	**63**	**72**	**88**	**119**	**27**	**32**	**36**	**38**	**48**
DRL-OPT			2432					200					72		

edge-weights as 0.4 than with the weighted-cascade results shown in Table II. DRL-UN/DRL-UN(TL) model achieved better results on all the networks than the naive heuristic-based methods in most network visibility settings except on FB/Ca-GrQc networks, respectively when network visibility is higher than 0.4. It should be noted that the DRL-UN and DRL-UN(TL) models are trained using the *weighted-cascade* model with initial network visibility set as 0.1. The result of both the models against 0.1 initial network visibility is quite better than naive heuristic-based methods on all three networks. Nevertheless, the performance gap between DRL-UN/DRL-UN(TL) and *FD* heuristic strategy is not huge on some network visibility settings. It can be a result of policy learned by the DRL-UN/DRL-UN(TL) model against 0.1 network visibility on the FB/Ca-GrQc network, which is not as competitive as *FD* heuristic strategy when the network visibility gets higher.

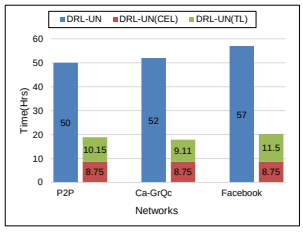
In general, DRL-based models performed better than naive heuristic-based strategies in most of network visibility settings on all three networks and adopted well with different edge-weight settings. Besides, the performance of the transfer-learning based model, that is, DRL-UN(TL), is better or similar as compared to heuristic-based as well as the DRL-UN methods. It validates both hypotheses.

C. Training Time Efficiency

Figure 4 shows the training time taken by each DRL-based model with number of nodes set as 10 and 20. DRL-UN(CEL) is trained only once on the Celegan source network and transferred to other target networks. So, the time of DRL-UN(CEL) is the same, that is, 8.75 hours on all networks. DRL-UN(TL) took around $8-10$ hours on all networks when the number of nodes to probe is set at 10. However, DRL-UN training time is more than 30 hours on all networks in the same setting. The training time of DRL-UN is more than 50 hours in all networks when we set the number of nodes to probe as 20. While the DRL-UN(TL) training time is around $9-11$ hours when trained with the number of nodes as 20 on all networks. It can be seen from Figure 4 that we can save significant training time when integrating transfer learning in DRL on even medium-sized networks.



(a) Nodes to probe = 10



(b) Nodes to probe = 20

Fig. 4: Training time of DRL-UN and DRL-UN(TL)

VI. Conclusion

In this work, we formulate a competitive influence maximization for unknown social networks. We proposed a DRL-based approach to tackle the CIM problem on unknown social networks. To boost the training time efficiency, we integrated the transfer learning in the DRL approach. Experimental results show that DRL-based approaches achieved significantly

TABLE IV: When models are evaluated using fixed edge-weights as 0.4 and nodes to probe $= 10$

	FB					P2P					Ca-GrQc				
	0.1	0.4	0.7	1	3	0.1	0.4	0.7	1	3	0.1	0.4	0.7	1	3
Model															
Heuric-Based															
FD	484	990	1251	1340	1639	64	146	162	166	171	1240	1846	1982	2104	2411
AFD	896	935	975	979	1052	73	73	75	77	84	1250	1284	1299	1342	1376
DRL-Based															
DRL-UN	**1052**	**1129**	**1234**	**1265**	**1468**	**107**	**178**	**205**	**216**	**279**	**1428**	**1973**	**2105**	**2211**	**2508**
DRL-UN(CEL)	767	1106	1317	1419	1777	119	193	224	233	264	1303	1729	1881	1920	2303
DRL-UN(TL)	**802**	**1106**	**1319**	**1431**	**1773**	**119**	**194**	**224**	**234**	**264**	**1303**	**1731**	**1881**	**1920**	**2303**
DRL-OPT			2632					452					1510		

better performance than naive heuristic-based approaches. Besides, the transfer learning-based DRL approach also achieved better results than the heuristic-based methods. Nevertheless, DRL-UN(CEL) model took quite less time to train on the Celegan source network. It can be applied as-is or with fine-tuning for a small number of training episodes on unknown target networks when time and computational resources are a concern. Experimental results show that the transfer learning-based DRL approach saved significant training time without much performance loss.

REFERENCES

[1] P. Domingos and M. Richardson, "Mining the network value of customers," in *Proc. of ACM SIGKDD*, 2001, pp. 57–66.

[2] M. Richardson and P. Domingos, "Mining knowledge-sharing sites for viral marketing," in *Proc. of ACM SIGKDD*, 2002, pp. 61–70.

[3] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proc. of ACM SIGKDD*, 2003, pp. 137–146.

[4] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, "Cost-effective outbreak detection in networks," in *Proc. of ACM SIGKDD*, 2007, pp. 420–429.

[5] Y. Tang, Y. Shi, and X. Xiao, "Influence maximization in near-linear time: A martingale approach," in *Proc. of ACM SIGMOD*, 2015, pp. 1539–1554.

[6] J. Shang, S. Zhou, X. Li, L. Liu, and H. Wu, "Cofim: A community-based framework for influence maximization on large-scale networks," *Knowledge-Based Systems*, vol. 117, pp. 88–100, 2017.

[7] S. Bharathi, D. Kempe, and M. Salek, "Competitive influence maximization in social networks," in *International Workshop on Web and Internet Economics*. Springer, 2007, pp. 306–311.

[8] C. Budak, D. Agrawal, and A. El Abbadi, "Limiting the spread of misinformation in social networks," in *Proc. of ACM WWW*, 2011, pp. 665–674.

[9] A. Borodin, Y. Filmus, and J. Oren, "Threshold models for competitive influence in social networks," in *International Workshop on Internet and Network Economics*. Springer, 2010, pp. 539–550.

[10] X. He, G. Song, W. Chen, and Q. Jiang, "Influence blocking maximization in social networks under the competitive linear threshold model," in *Proc. of SIAM ICDM*, 2012, pp. 463–474.

[11] S.-C. Lin, S.-D. Lin, and M.-S. Chen, "A learning-based framework to handle multi-round multi-party influence maximization on social networks," in *Proc. of ACM SIGKDD*, 2015, pp. 695–704.

[12] K. Ali, C.-Y. Wang, and Y.-S. Chen, "Boosting reinforcement learning in competitive influence maximization with transfer learning," in *Proc. of IEEE/WIC/ACM WI*, 2018, pp. 395–400.

[13] ——, "A novel nested q-learning method to tackle time-constrained competitive influence maximization," *IEEE Access*, vol. 7, pp. 6337–6352, 2019.

[14] T.-Y. Chung, K. Ali, and C.-Y. Wang, "Deep reinforcement learning-based approach to tackle competitive influence maximization," in *Proc. of MLG workshop*, 2019.

[15] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[16] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, 2015.

[17] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton *et al.*, "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, p. 354, 2017.

[18] C. Wang, W. Chen, and Y. Wang, "Scalable influence maximization for independent cascade model in large-scale social networks," *Data Mining and Knowledge Discovery*, vol. 25, no. 3, pp. 545–576, 2012.

[19] W. Chen, Y. Wang, and S. Yang, "Efficient influence maximization in social networks," in *Proc. of ACM SIGKDD*, 2009, pp. 199–208.

[20] A. Goyal, W. Lu, and L. V. Lakshmanan, "Celf++: optimizing the greedy algorithm for influence maximization in social networks," in *Proc. of ACM WWW*, 2011, pp. 47–48.

[21] S. Cheng, H. Shen, J. Huang, G. Zhang, and X. Cheng, "Staticgreedy: solving the scalability-accuracy dilemma in influence maximization," in *Proc. of ACM CIKM*, 2013, pp. 509–518.

[22] W. Chen, Y. Yuan, and L. Zhang, "Scalable influence maximization in social networks under the linear threshold model," in *Proc. of IEEE ICDM*, 2010, pp. 88–97.

[23] J. Kostka, Y. A. Oswald, and R. Wattenhofer, "Word of mouth: Rumor dissemination in social networks," in *International Colloquium on Structural Information and Communication Complexity*. Springer, 2008, pp. 185–196.

[24] W. Chen, A. Collins, R. Cummings, T. Ke, Z. Liu, D. Rincon, X. Sun, Y. Wang, W. Wei, and Y. Yuan, "Influence maximization in social networks when negative opinions may emerge and propagate." in *Proc. of SIAM ICDM*, 2011, pp. 379–390.

[25] S. Goyal, H. Heidari, and M. Kearns, "Competitive contagion in networks," *Games and Economic Behavior*, 2014.

[26] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1, no. 1.

[27] L.-J. Lin, "Self-improving reactive agents based on reinforcement learning, planning and teaching," *Machine learning*, vol. 8, no. 3-4, pp. 293–321, 1992.

[28] M. Brautbar and M. J. Kearns, "Local algorithms for finding interesting individuals in large networks," 2010.

[29] B. Wilder, N. Immorlica, E. Rice, and M. Tambe, "Maximizing influence in an unknown social network," in *Proc. of AAAI*, 2018.

[30] A. Barreto, D. Borsa, J. Quan, T. Schaul, D. Silver, M. Hessel, D. Mankowitz, A. Zidek, and R. Munos, "Transfer in deep reinforcement learning using successor features and generalised policy improvement," in *Proc. of PMLR ICML*, 2018, pp. 501–510.

[31] Y. Du, "Improving deep reinforcement learning via transfer," in *Proc. of AAMAS*, 2019, pp. 2405–2407.

[32] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proc. of NIPS*, 2014, pp. 3320–3328.

[33] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection," http://snap.stanford.edu/data, Jun. 2014.