Adversarial Enhancement for Community Detection in Complex Networks

Jiajun Zhou, Zhi Chen, Min Du, Lihong Chen, Shanqing Yu,
Feifei Li, Guanrong Chen, *Fellow, IEEE*, and Qi Xuan, *Member, IEEE*

Abstract—Community detection plays a significant role in network analysis. However, it also faces numerous challenges like adversarial attacks. How to further improve the performance and robustness of community detection for real-world networks has raised great concerns. In this paper, we propose a concept of *adversarial enhancement for community detection*, and present two adversarial enhancement algorithms: one is named *adversarial enhancement via genetic algorithm* (AE-GA), in which the modularity and the number of clusters are used to design a fitness function to solve the resolution limit problem; and the other is called *adversarial enhancement via vertex similarity* (AE-VS), integrating multiple information of community structures captured by diverse vertex similarities, which scales well on large-scale networks. The two algorithms are tested along with six existing community detection algorithms on four real-world networks. Comprehensive experimental results show that, by comparing with two traditional enhancement strategies, our methods help six community detection algorithms achieve more significant performance improvement. Moreover, experiments on the corresponding adversarial networks indicate that our methods can rebuild the network structure destroyed by adversarial attacks to certain extent, achieving stronger defense against community detection deception.

Index Terms—Community detection; Community deception; Adversarial enhancement; Adversarial attack; Genetic algorithm.

1 Introduction

Recently, as an interdiscipline, network science has been widely applied to model complex systems in different fields like sociology, biology, transportation and computer science [1]–[3]. Real-world networks share various common properties such as power-law degree distribution [4], [5], small-world features [6], and community structures [7]. In particular, community structure is very important in network analysis. In networks, vertices are organized into groups, called *communities*, *clusters* or *modules*, with dense connections within groups and sparse connections between them. For instance, in co-author networks, communities are formed by scientists with similar research interests in close fields; in social networks like Facebook, they can represent people focusing on similar topics. Many recent researches suggest that network properties at the community level are quite different from those at the global level, and thus ignoring community structure may miss many interesting features [8]. As a matter of fact, identifying communities in networks has played a significant role in exploiting essential network structures.

Since Girvan and Newman [7] first proposed a community detection method based on edge betweenness, a large number of techniques have been developed to detect community structures in networks. Traditional approaches inlcude spectral clustering [9], [10], hierarchical clustering [7], statistical inference [11]–[14] and modularity optimization. Other approaches involve random walk dynamics [15], [16], cluster synchronization [17], [18], and so on.

However, since these approaches rely on the topological structure of the underlying network, their capability to discover the true community structure faces numerous challenges. The first challenge is about the integrity and accuracy of the network. Real networks are often incomplete and suffer from missing edges, since not all real-world relationships are reflected in a single network. For instance, users in social networks like Twitter seldom follow all their friends in activities. Moreover, missing edges also occur when crawling datasets from online networks with privacy restrictions. On the other hand, the accuracy of a network is very likely to be questioned when the information encoded in the network topology is perturbed by noise, especially when the network suffers from adversarial attacks, which leads to the degradation of the performance of many network analysis methods. In particular, adversarial attacks against community detection aim to hide target communities or sensitive edges [19], and finally generate specific adversarial networks, which can strongly impact the performance of community detection algorithms. Existing community detection methods rarely consider missing edges and noise in networks, increasing the risk to obtain wrong community structures.

Another challenge is the lack of a consensus on the formal definition of a network community structure [20]. Currently, there are no universal standards for the definition of community, and a large number of community detection al-

- *J. Zhou, L. Chen, S. Yu, and Q. Xuan are with the Institute of Cyberspace Security, College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China, and also with the Big Search in Cyberspace Research Center, Zhejiang Lab, Hangzhou 311121, China. E-mail: jjzhou012@163.com, {yushanqing, 2111803032, xuanqi}@zjut.edu.cn).*
- *Z. Chen and M. Du are with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720, USA. E-mail: {zhichen98, min.du}@berkeley.edu.*
- *F. Li is with the Department of Computer Science, University of Utah, Salt Lake City, UT 8411, USA. E-mail: lifeifei@cs.utah.edu.*
- *G. Chen is with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong SAR, China. E-mail: eegchen@cityu.edu.hk.*
- *J. Zhou and Z. Chen make equal contribution.*
- *Corresponding author: Qi Xuan.*

gorithms based on different technologies and ideas have been proposed, which led to a quality discrepancy among different results. Moreover, modularity optimization has a resolution limit [21]. Clusters consisting of a number of vertices smaller than a threshold would not be detected because these clusters tend to merge into larger ones by modularity optimization. Large, but local and sparse, communities probably tend to be divided into smaller ones during community partition.

It is believed that such challenges are mostly from unstable network structures. Networks with sparse community structures are vulnerable to adversarial attacks which can destroy network structures, leading to community detection deception. Generally, communities with weak structures will be absorbed from the outside or disintegrated from the inside of the network. Optimizing the network structure and improving the stability of the network may be an effective way to deal with these challenges. A heuristic idea comes from the fact that community structures show a high connection density of intra-communities and a sparse one of inter-communities. Agglomerating the intra-communities by adding edges between internal vertices and divisiving the inter-communities by removing edges between communities, therefore, can strengthen the community structure in a network. Another idea for enhancing community detection in original, incomplete or adversarial networks is to enhance network structure with edge prediction, of which the task is to complement missing edges or predict future edges between pairwise vertices based on the current network structure. The vertex similarity indices can be used to guide network structure optimization, according to the following two assumptions: 1) vertices in the same community are aggregated based on their high similarity; 2) a larger similarity of pairwise vertices leads to a higher likehood of edges between them [22].

In this paper, we propose a concept of adversarial enhancement, and develop two adversarial enhancement algorithms to optimize network structures for community detection. The main contributions of our work are summarized as follows:

- First, we propose the technique of adversarial enhancement to improve the performance of existing community detection algorithms. To the best of our knowledge, our work is the first for enhancing community detection in both real-world networks and adversarial networks.
- Second, we develop two adversarial enhancement algorithms, namely *adversarial enhancement via genetic algorithm* (AE-GA) based on modularity and *adversarial enhancement via vertex similarity* (AE-VS), and compare them with traditional enhancement algorithms on four real-world networks. Experimental results demonstrate the superiority of our methods in helping six community detection algorithms to achieve significant improvement of performances.
- Third, we test the four enhancement algorithms on adversarial networks, the results show that both AE-GA and AE-VS can rebuild the network structure destroyed by adversarial attacks and achieve stronger defense ability against community detection deception.
- Finally, since our methods are designed to resolve the resolution limit in modularity optimization, they can help various community detection algorithms to achieve consensus, i.e., getting consistent partition.

The rest of the paper is organized as follows. First, in Sec. 2, we review the related works on adversarial attacks for networks and traditional enhancement for community detection. Then, in Sec. 3, we describe our approaches in detail. Thereafter, we present extensive experiments in Sec. 4, with a series of discussions. Finally, we conclude the paper and outline future work in Sec. 5.

2 Related work

2.1 Adversarial Attacks on Networks

Related studies about adversarial attacks on networks or graph data are just at the beginning but raise more and more concerns. For instance, in a social network, an adversary can easily disguise himself by adding a very small number of friendship connections with strangers and deleting connections with friends, which may have severe consequences. Before the concept of *adversarial attack* was introduced, several studies have focused on destroying the network structure. Holme et al. [23] investigated the vulnerability of various networks subject to attacks on vertices or edges, and found that the network structure changes and several metrics degrade as important vertices or edges are removed. Bellingeri et al. [24] found that sequential deletion of vertices in decreasing order of betweenness centrality was the most efficient attack strategy when using the size of the largest connected component (LCC) as network performance index. Karrer et al. [25] proposed a method for perturbing networks and a metric of robustness, and then use them to assess the significance of community structure in various networks.

Adversarial attacks against graph algorithms and models that applied to link prediction, node classification, or community detection are widely studied in recent years. For community detection, Waniek et al. [26] proposed a simple heuristic method deployed by intra-community edge deletion and inter-community edge addition, and introduced a measure of concealment to express how well a community is hidden. Fionda et al. [19] introduced and formalized the community deception problem, and proposed an community deception algorithm based on safeness, which achieves a success in hidding a target community. Chen et al. [27] formulated the community deception problem and developed an effective strategy, namely genetic algorithm (GA)-based Q-Attack, to achieve deception by negligibly rewiring networks.

For other tasks, Yu et al. [28] proposed both heuristic and evolutionary approaches to hide sensitive links from being predicted, achieving privacy protection. Dai et al. [29] proposed a reinforcement learning-based attack approach, and showed that graph neural network (GNN) models are vulnerable to such attacks, in both graph-level and node-level classification tasks. Moreover, Bojchevski et al. [30] proposed an adversarial attack on network embedding based on random walks, suggesting that effective adversarial perturbations can destroy the network structure and lower the quality of the embeddings. Zugner et al. [31] proposed an adversarial attack on node classification and modeled

misclassification by generating imperceptible perturbations that can confuse the node features and graph structure.

2.2 Traditional Enhancement of Community Detection

Because of the deficiency of many existing community detection methods, how to improve their performance in complicated real applications has become an important issue.

Most strategies suggest first preprocessing networks and then feeding them into community detection algorithms so as to improve their performances. For instance, it was found that the resolution limits of modularity optimization can be alleviated by weighting network edges in different ways, which make it more suitable for community detection. Meo et al. [32] introduced a measure of κ-path edge centrality and proposed a weighting algorithm called WERW-κPath to effectively compute the centrality as edge weight, which is better for community detection. Sun [33] weighted networks via a series of edge centrality indices and detected communities in the weighted network using a function that considers both links and link weights. Moreover, Lai et al. [34] considered random walk for simulation on dynamic processes, and applied it to enhance modularity-based methods, based on the intuition that pairwise vertices in the same community have similar dynamic patterns. Interestingly, Li et al. [35] proposed an edge enhancement approach for motif-aware community detection, called EdMot, which not only can leverage higher-order connections of the network, but also can resolve the hypergraph fragmentation issue. Their method transfers the network into motif-based hypergraph and partitions it into modules, and then a new edge set is constructed to enhance the connectivity structure of the original network by fully connecting all modules. Lancichinetti et al. [36] proposed consensus clustering algorithm, which combines the information of different outputs to obtain a more representative partition, to analyze the time evolution of clusters in dynamics networks. Dahlin et al. [37] proposed the ensemble cluster that combines the ensemble method with clustering, and improve community detection by aggregating multiple runs of algorithms.

On the other hand, model-based methods tend to integrate the enhancement into the whole community detection procedure. For example, He et al. [38] provided a framework to enhance the ability of NMF models to detect communities, which use the NMF method to train a stochastic model constrained by vertex similarity.

3 The Proposed Methods

We first formulate the adversarial enhancement problem, and then present our methods. The notations in this paper are listed in TABLE 1.

TABLE 1
Notations used in this paper.

Symbol	Description
$\mathcal{G}$	The target graph (network)
$\mathcal{V}, \mathcal{E}, \mathcal{M}$	Sets of vertices, edges, communities in graph $\mathcal{G}$
v, e	Vertex, edge in graph $\mathcal{G}$
$\mathcal{S}$	The studied community detection method
$\mathcal{M}_{\mathcal{G}}^{\mathcal{S}}$	Set of communities found by $\mathcal{S}$ in graph $\mathcal{G}$
n, m, M	Numbers of vertices, edges, communities in graph $\mathcal{G}$
$\mathcal{M}_{real}$	The ground-truth community partition in graph $\mathcal{G}$
$\mathcal{E}_{mod}$	The adversarial enhancement scheme
$\mathcal{E}_{add}, \mathcal{E}_{del}$	The schemes of edge addition and edge deletion
$\mathcal{G}^*$	The rewired graph (network)
$\mathcal{E}^*, \mathcal{M}^*$	Sets of edges, communities in graph $\mathcal{G}^*$
β_a, β_d	Sample rates of edge addition and deletion
$\mathcal{Q}$	Modularity
$\mathcal{P}$	Population
$\mathcal{F}$	Set of fitness in population $\mathcal{P}$
$\mathcal{P}_s$	Size of population
$\mathcal{P}_c$	Crossover rate
$\mathcal{P}_m$	Mutation rate
$\mathcal{P}_e$	Elitist pres rate
$\mathcal{T}_{ga}$	Number of iterations
$\mathcal{G}_{co}, \mathcal{A}_{co}$	Co-occurrence graph and its consensus matrix
$\mathcal{T}$	Threshold of prune in $\mathcal{G}_{co}$
$\mathcal{G}_{co}^{\mathcal{T}}$	Co-occurrence graph pruned with threshold $\mathcal{T}$
$\mathcal{M}^{\mathcal{T}}$	Community partition in pruned graph $\mathcal{G}_{co}^{\mathcal{T}}$
c	Consensus of a cluster
C	Consensus score of cluster partition
z	Number of partitions in AE-VS-E
$\mathcal{A}_{\mathsf{cn}}, \ldots, \mathcal{A}_{\mathsf{rwr}}$	Similarity score matrices for graph $\mathcal{G}$
$\mathcal{K}$	Number of selected connected components (EdMot).
κ	Length of walk path (WERW-κPath)
ρ	Iterations of walk (WERW-κPath)
β	Attack cost in adversarial attack algorithms

3.1 Problem Formulation

A assume that an undirected and unweighted network is represented by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, which consists of a vertex set $\mathcal{V} = \{v_i \mid i = 1, \ldots, n\}$ and an edge set $\mathcal{E} = \{e_i \mid i = 1, \ldots, m\}$. The task of community detection in a network is to find a vertex partition $\mathcal{M} = \{\mathcal{M}_i \mid i = 1, \ldots, M\}$, with $\bigcup \mathcal{M}_i = \mathcal{V}$ and $\mathcal{M}_i \bigcap \mathcal{M}_j = \emptyset$ for $i \neq j$, where set $\mathcal{M}_i$ is called a *community*. The ground-truth community labels of the network is denoted as $\mathcal{M}_{real}$. Note that the community overlapping problem will not be considered in this paper.

In the adversarial enhancement scenario, a network will be rewired, i.e., a series of edges will be modified, which will be denoted in the following set form:

$$\mathcal{E}_{mod} = \{+\mathcal{E}_{add}, -\mathcal{E}_{del}\}, \tag{1}$$

where $\mathcal{E}_{add}$ represents the set of edges added to the network and $\mathcal{E}_{del}$ represents the set of edges removed from the network, respectively, denoted as:

$$\begin{aligned} \mathcal{E}_{add} &= \{\tilde{e}_j \mid j = 1, \ldots, \beta_a m;\ \forall \tilde{e}_j \notin \mathcal{E}\}, \\ \mathcal{E}_{del} &= \{\tilde{e}_j \mid j = 1, \ldots, \beta_d m;\ \forall \tilde{e}_j \in \mathcal{E}\}. \end{aligned} \tag{2}$$

Notably, in Eq. (2), for each enhancement scheme, two sample rates β_a and β_a are set to control the quantity of edge addition and deletion, respectively. Then, based on the modification scenario $\mathcal{E}_{mod}$, the connectivity structure of the original network is enhanced to generate a rewired network:

$$\mathcal{G}^* = (\mathcal{V}, \mathcal{E}^*) \quad \text{with } \mathcal{E}^* = \mathcal{E} + \mathcal{E}_{mod}. \tag{3}$$

In this way, we can find the solution E_{mod} to optimize the network structure by a rewiring process. For the rewired

networks obtained by adversarial enhancement algorithms, the community detection methods perform significantly better and the new partition result $\mathcal{M}^*$ is closer to the ground-truth communities, i.e., there is a significant improvement in evaluation metrics after assigning $\mathcal{M}^*$ to the original network.

3.2 Modularity-Based Adversarial Enhancement

Adversarial enhancement can be considered as an optimization problem. Based on modularity, we propose the first adversarial enhancement algorithm, namely *adversarial enhancement via genetic algorithm* (AE-GA) based on modularity, which aims to determine optimal edge modification to rewire the connections among communities.

3.2.1 Network Rewiring

Previous works [19], [26] have shown that intra-community edge deletion and inter-community edge addition can effectively deploy community deception attacks. Therefore, by contrast, edge modification schemes in AE-GA are designed with two basic operations, i.e., intra-community edge addition and inter-community edge deletion, which can stabilize the community structure.

Specifically, for an original network $\mathcal{G}$, the neighbor set of a target vertex v_i is denoted as $\mathcal{V}_i^- = \{v_j \mid v_j \neq v_i; (v_i, v_j) \in \mathcal{E}\}$, while its nonneighbor set is denoted as $\mathcal{V}_i^+ = \mathcal{V} - \mathcal{V}_i^- - \{v_i\}$. Then, a prior community detection algorithm $\mathcal{S}$ is applied to partition $\mathcal{G}$ to obtain a community partition $\mathcal{M}_\mathcal{G}^\mathcal{S} = \{\mathcal{M}_i \mid i = 1, \ldots, k\}$. The candidate set of intra-community edge addition for v_i is denoted as

$$\mathcal{E}_i^{add} = \{(v_i, v_j) \mid v_j \in \mathcal{M}_l \cap \mathcal{V}_i^+\}, \tag{4}$$

where $\mathcal{M}_l$ is the community that v_i belongs to. Similarly, the candidate set of inter-community edge deletion for v_i is denoted as

$$\mathcal{E}_i^{del} = \{(v_i, v_j) \mid v_j \in \overline{\mathcal{M}_l} \cap \mathcal{V}_i^-\}, \tag{5}$$

where $\overline{\mathcal{M}_l} = \mathcal{V} - \mathcal{M}_l$.

However, as mentioned above, modularity optimization suffers from resolution limit that small clusters tend to merge into larger ones when they have a size smaller than a specified threshold. We thus consider some adjustments during network rewiring to deal with this problem. We make a comparison between the number of communities in the estimated partition and that in the ground truth:

- If $M_S > M_{real}$, extra inter-community addition is available during adversarial enhancement;
- If $M_S < M_{real}$, extra intra-community deletion is available during adversarial enhancement;
- If $M_S = M_{real}$, both the inter-community addition and intra-community deletion are inoperative.

Here, M_{real} is the number of the ground-truth communities in $\mathcal{G}$ and M_S is the number of communities found by the specific community detection method $\mathcal{S}$.

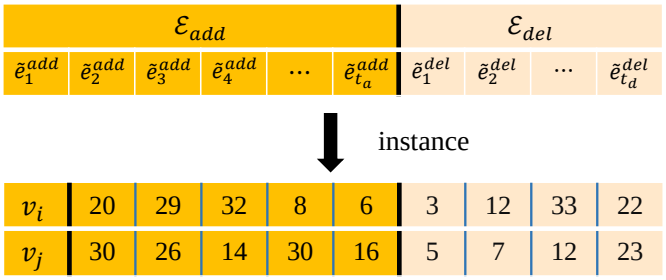

Fig. 1. The diagram of chromosome in AE-GA. It consists of two parts including edge addition segment $\mathcal{E}_{add}$ and edge deletion segment $\mathcal{E}_{del}$. The instance of chromosome is initialized in the experiment for Karate dataset, with an edge addition segment of length 5 and an edge deletion segment of length 4.

3.2.2 Evolutionary Optimization

We use the genetic algorithm (GA) due to its good performance in solving combination optimization problems. Specifically, we design the encoding scheme of chromosome and the function of fitness as follows.

- **Chromosome**
  A chromosome here represents an adversarial enhancement scheme $\mathcal{E}_{mod}$, consisting of two parts: $\mathcal{E}_{add}$ and $\mathcal{E}_{del}$, where a gene denotes an edge modification operation, including edge addition or deletion. The diagram of chromosome is shown in Fig. 1.
- **Fitness**
  Modularity is commonly used to measure the quality of community partition for a network with unknown community structure. The basic idea is to compare the network with the corresponding null model. Here, the fitness is defined as

$$f = \frac{|\mathcal{Q}|}{e^{|M_S - M_{real}|}}, \tag{6}$$

  where $\mathcal{Q}$ is the modularity of the partition for the target network. Individuals with larger modularity and more accurate partition will generally have larger fitness.

The procedure of AE-GA is shown in Algorithm 1. As mentioned above, AE-GA requires the prior knowledge of the community structure, which guides the edge update. We feed the target network $\mathcal{G}$ into community detection algorithm $\mathcal{S}$ to obtain a general community partition $\mathcal{M}_\mathcal{G}^\mathcal{S}$ and then construct the candidate edge sets (line 1).

In this scheme, during **Initialization**, a parental generation $\mathcal{P} = \left\{\mathcal{E}_i^{mod} \mid i = 1, \ldots, \mathcal{P}_s\right\}$ is randomly generated with a population size $\mathcal{P}_s$ and each individual $\mathcal{E}_i^{mod}$ in the population has an unfixed size, i.e., the quantity of modified edges is not fixed for each initial enhancement scheme (line 2). During **selection**, the operation is conducted on *roulette*, which means that the probability for an individual to be selected is proportional to its fitness (line 6). **Crossover** is the process of combining the parental generation to obatin new schemes and we apply *multi-point crossover* to swap gene segments between two parental chromosomes (line 7). **Mutation** prevents the algorithm from falling into local optimization. We traverse each gene in the chromosome

Algorithm 1: AE-GA

Input: Target network $\mathcal{G}$, community detection algorithm $\mathcal{S}$, parameter for GA($\mathcal{P}_s$, $\mathcal{P}_c$, $\mathcal{P}_m$, $\mathcal{P}_e$, $\mathcal{T}_{ga}$), sample rate β_a, β_d .
Output: New community partition $\mathcal{M}^*$

1 $\mathcal{M}_{\mathcal{G}}^{\mathcal{S}} \leftarrow$ communityDetection$(\mathcal{S}, \mathcal{G})$;
2 $\mathcal{P}, \mathcal{F} \leftarrow$ popInitialize$(\mathcal{G}, \mathcal{M}_{\mathcal{G}}^{\mathcal{S}}, \mathcal{P}_s, \beta_a, \beta_b)$;
3 Initialize current generation $i = 0$;
4 **while** $i < \mathcal{T}_{ga}$ **do**
5 $\quad \mathcal{P}_{elitist} \leftarrow$ retain$(\mathcal{F}, \mathcal{P}, \mathcal{P}_e)$;
6 $\quad \mathcal{P}_{select} \leftarrow$ selection$(\mathcal{F}, \mathcal{P})$;
7 $\quad \mathcal{P}_{crossover} \leftarrow$ crossover$(\mathcal{P}_{select}, \mathcal{P}_c)$;
8 $\quad \mathcal{P}_{mutate} \leftarrow$ mutation$(\mathcal{P}_{crossover}, \mathcal{P}_m, \mathcal{M}_{\mathcal{G}}^{\mathcal{S}})$;
9 $\quad$ Calculate the fitness of individuals: $\mathcal{F} \leftarrow$ getFitness$(\mathcal{G}, \mathcal{S}, \mathcal{P}_{mutate})$;
10 $\quad \mathcal{P} \leftarrow$ getNextGeneration$(\mathcal{P}_{mutate}, \mathcal{P}_{elitist})$
11 Get the individual with highest fitness from the last population: $\mathcal{E}_{mod} \leftarrow$ getBestIndividual$(\mathcal{F}, \mathcal{P})$;
12 Rewire the original network to obtain $\mathcal{G}^*$ via Eq. 3 ;
13 Feed $\mathcal{G}^*$ into $\mathcal{S}$ to obtain new community partition: $\mathcal{M}^* \leftarrow$ communityDetection$(\mathcal{S}, \mathcal{G}^*)$. ;
14 **end** ;
15 **return** $\mathcal{M}^*$;

and conduct the mutation operation with a mutation rate $\mathcal{P}_m$ (line 8). In so doing, we randomly replace the edge modification operation $\tilde{e}^{add}$ or $\tilde{e}^{del}$ with another one. Finally, **elitist preservation** is applied to retain excellent individuals, which refers to enhancement schemes with higher fitness. In particular, we retain excellent individuals by replacing the worst 20% of the offspring with the best 20% of the parents (line 5). Evolution is a process of iteration and we set the number of iterations $\mathcal{T}_{ga}$ as the evolutionary generation. The evolutionary optimization stops when it is convergent or this condition is satisfied.

3.3 Similarity-Based Adversarial Enhancement

Empirically, vertices in the same community can be aggregated due to their high similarity. Therefore, we adopt the similarity indices to aggregate those vertices of high similarity, i.e., considering the similarity indices as the guidance of edge modification. Based on this, we propose another adversarial enhancement algorithm, namely *adversarial enhancement via vertex similarity* (AE-VS), which rewires a network via multiple similarity indices and aggregates corresponding community partitions to generate a more accurate community structure. The framework of AE-VS is shown in Fig. 2, and the procedure of AE-VS is shown in Algorithm 2.

3.3.1 Network Rewiring

Vertex similarity can be defined as the number of common features that a pair of vertices share [39]. Zhou et al. [40] compared ten local similarity indices on six real networks and found that *Resource Allocation (RA)* index has the best overall performance. Other similarity indices based on paths or random walks are summarized and compared in [41]. Here, we adopt a variety of similarity indices, as shown in TABLE 2, to guide enhancement strategy. It is worth noting that many other similarity indices can also be applied into this scheme.

During network rewiring, we calculate the similarity matrix for each index, which consists of similarity scores of arbitrary pairwise vertices (line 1). Enhancement schemes are operated by sampling nonexistent edges from the candidate

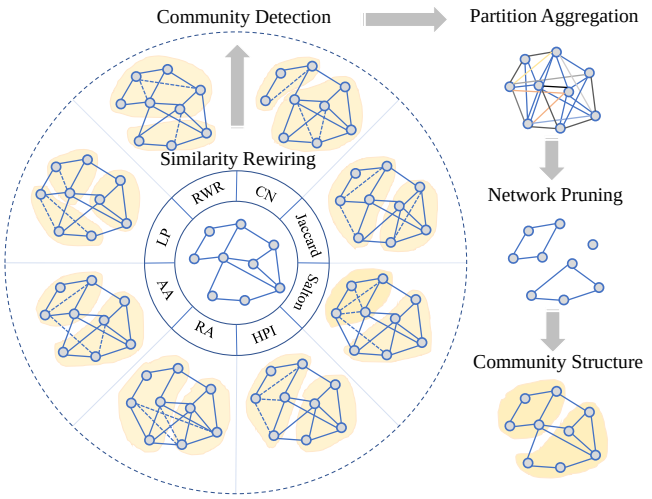

Fig. 2. The framework of AE-VS.

Algorithm 2: AE-VS

Input: Target network $\mathcal{G}$, community detection algorithm $\mathcal{S}$, sample rate β_a.
Output: New community structure M^*

1 Compute similarity matrices for indices in TABLE 2: $\{\mathcal{A}_{\mathsf{cn}}, \ldots, \mathcal{A}_{\mathsf{rwr}}\} \leftarrow$ computeSimilarity$(\mathcal{G})$;
2 Obtain enhancement schemes via sampling: $\{\mathcal{E}_{mod}^1, \ldots, \mathcal{E}_{mod}^z\} \leftarrow$ sample$(\beta_a, \{\mathcal{A}_{\mathsf{cn}}, \ldots, \mathcal{A}_{\mathsf{rwr}}\})$;
3 Update graph via network rewiring : $\{\mathcal{G}_1^*, \ldots, \mathcal{G}_z^*\} \leftarrow$ rewire$(\mathcal{G}, \{\mathcal{E}_{mod}^1, \ldots, \mathcal{E}_{mod}^z\})$;
4 Obtain multiple partitions via community detection: $\{\mathcal{M}_1^*, \ldots, \mathcal{M}_z^*\} \leftarrow$ communityDetection$(\mathcal{S}, \{\mathcal{G}_1^*, \ldots, \mathcal{G}_z^*\})$;
5 Get a co-occurrence network from multiple partitions: $\mathcal{G}_{co}, \mathcal{A}_{co} \leftarrow$ getCoNetwork$(\{\mathcal{M}_1^*, \ldots, \mathcal{M}_z^*\})$;
6 Find an optimal pruning threshold and core communities: $\mathcal{T}, \mathcal{M}_{core}^{\mathcal{T}} \leftarrow$ getOptimalThreshold$(\mathcal{G}_{co}, z)$;
7 Get the final partition by assigning isolated vertices to core communities: $\mathcal{M}^* \leftarrow$ getFinalPartition$(\mathcal{M}_{core}^{\mathcal{T}}, \{\mathcal{A}_{\mathsf{cn}}, \ldots, \mathcal{A}_{\mathsf{rwr}}\})$;
8 **end** ;
9 **return** M^*;

TABLE 2
Similarity indices used in AE-VS.

Category	Similarity index
Local	CN, Jaccard [42], Salton [43], HPI [44], AA [45], RA [40]
Path	LP [46]
Random walk	RWR [47]

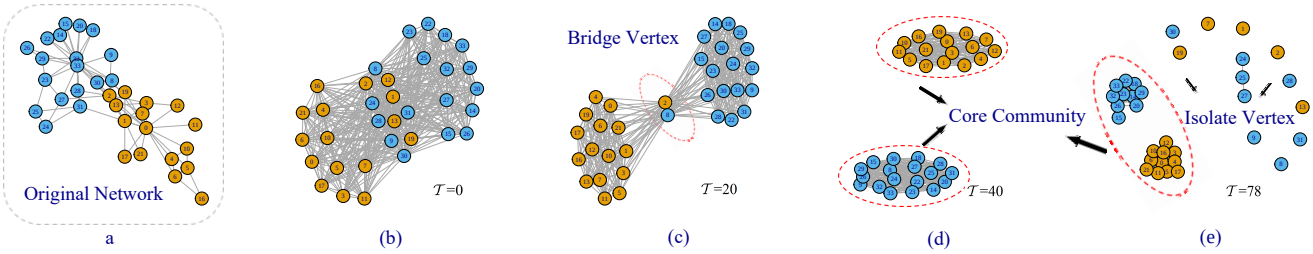

Fig. 3. Visualization of network pruning in the co-occurrence network of Karate network. The last four represent the co-occurrence network pruned with various values of threshold $\mathcal{T}$. Note that vertices with the same color share the same ground truth community label.

set $\{(e_i, e_j) | \ i \neq j; (e_i, e_j) \notin \mathcal{E})\}$ with a sample rate β_a, and the probability for an edge to be selected is proportional to its similarity score (line 2). Note that we only consider edge addition in AE-VS. We update the target network $\mathcal{G}$ using the enhancement schemes and feed these rewired networks into the community detection algorithm to obtain a series of community partitions (lines 3, 4).

3.3.2 Partition Ensemble

Due to the diversity of similarity indices and enhancement schemes, these partitions are likely to be non-unique and not necessarily better than the original partition $\mathcal{M}$. Ensemble learning, which achieves better classification and prediction performance by integrating multiple weak models, has been used for clustering tasks. Previous studied consensus and ensemble clustering [36], [37] , showing that these techniques can be combined with existing clustering methods and enhance the stability and accuracy of community partitions.

We aggregate multiple partitions using a consensus matrix $\mathcal{A}_{co} = (a_{ij})_{n \times n}$, in which element a_{ij} indicates the frequency of two vertices i and j assigned to the same community. A weighted co-occurrence graph $\mathcal{G}_{co}$ can be generated from the consensus matrix $\mathcal{A}_{co}$ (line 5). Once pairwise vertices appear in the same community in some partitions, $\mathcal{G}_{co}$ links them and assigns weights that correspond to the frequency of co-occurrence. A larger/smaller weight means a higher/lower likelihood that the pairwise vertices belong to the same community. In other words, edges with larger weights tend to be intra-community edges while those with lower weights can be considered as inter-community edges. Since the co-occurrence network $\mathcal{G}_{co}$ is constructed from all partitions, we have no access to deploy intra-community edge addition without new partitions, but we can prune $\mathcal{G}_{co}$ to remove inter-community edges by setting a threshold $\mathcal{T}$. During pruning, all edges with weights less than $\mathcal{T}$ are considered as inter-community edges and will be removed from $\mathcal{G}_{co}$.

A visualization of network pruning in the co-occurrence network of Karate network is shown in Fig. 3. In this paper, we use eight similarity indices, and for each index, ten samples are performed, to generate a total of eighty partitions, which determine the domain of threshold $\mathcal{T} \in \{1, 2, \ldots, 80\}$. The original Karate network is shown in Fig. 3 (a). There are two communities in Karate network, with the vertices of the same color sharing the same ground-truth community label. Fig. 3 (b) shows the co-occurrence network, which aggregates the information of 80 partitions and has dense connections. The last three subgraphs show the different pruned co-occurrence networks with various thresholds. When $\mathcal{T} = 20$, the pruned co-occurrence network still has only one connected component but two bridge vertices emerge, as shown in Fig. 3 (c). With the increase of the threshold, $\mathcal{G}_{co}$ is divided into two connected components, matching exactly with the two clusters in the original network, as shown in Fig. 3 (d). When the threshold approaches the upper limit, generally, we'll get several small connected components that contain few vertices, or even isolated vertices, as shown in Fig. 3 (e). This phenomenon indicates that the selection of threshold actually influences the result of partition.

In order to address the resolution limit problem, we optimize the threshold via a traversal procedure (line 6). The domain of threshold is $\{1, 2, \ldots z\}$, i.e., $\mathcal{T}$ will always be smaller than the number of partitions z. We prune $\mathcal{G}_{co}$ with a threshold $\mathcal{T}$ to yield a pruned network $\mathcal{G}_{co}^{\mathcal{T}}$, and evaluate the cluster partition of $\mathcal{G}_{co}^{\mathcal{T}}$ using *cluster consensus* metric, which can quantify the stability of clusters [48]. For a pruned co-occurrence network $\mathcal{G}_{co}^{\mathcal{T}}$ with cluster partition $\mathcal{M}^{\mathcal{T}} = \{\mathcal{M}_k \mid k = 1, \ldots, M^{\mathcal{T}}\}$, the consensus of cluster $\mathcal{M}_k$ is defined as

$$c(\mathcal{M}_k) = \frac{1}{M_k \left(M_k - 1\right)/2} \sum_{\substack{i,j \in \mathcal{M}_k \\ i<j}} \mathcal{A}_{co}(i, j), \tag{7}$$

where M_k is the size of $\mathcal{M}_k$. The optimal threshold corresponds to the maximum partition score, which can be computed via a weighted sum of cluster consensus, as follows:

$$C(\mathcal{M}^{\mathcal{T}}) = \sum_{k=1}^{M^{\mathcal{T}}} \frac{M_k}{n} c(\mathcal{M}_k), \tag{8}$$

$$\mathcal{T} = \arg\max_{\mathcal{T}} C(\mathcal{M}^{\mathcal{T}}). \tag{9}$$

After pruning, the co-occurrence network is split into several connected components, and those with large sizes will be treated as core communities $\mathcal{M}_{core}^{\mathcal{T}}$. Generally, we'll get several small connected components that contain few vertices, or even isolated vertices, when pruning network with a relatively large threshold. In order to get a final partition, these vertices in small connected components will be treated as isolated ones and assigned to the core community with

which it has the maximum average similarity score (line 7). The ID of a core community can be obtained by computing

$$\arg\max_{k} \frac{1}{8M^k_{core}} \sum_{j \in \mathcal{M}^k_{core}} \sum \{\mathcal{A}_{\mathsf{cn}}(i,j), \ldots, \mathcal{A}_{\mathsf{rwr}}(i,j)\}, \tag{10}$$

where M^k_{core} is the size of $\mathcal{M}^k_{core}$ which is a core community in $\mathcal{M}^{\mathcal{T}}_{core}$. Note that it is divided by 8, since here we use eight similarity indices in AE-VS.

TABLE 3
Real-world networks. M_{real} is the number of ground-truth communities in $\mathcal{G}$ and $M^{\mathcal{S}}_{\mathcal{G}}$ is the number of communities found by the specific community detection method $\mathcal{S}$.

Network	V	E	M_{real}	$M^{\mathcal{S}}_{\mathcal{G}}$					
				INF	FG	WT	LOU	LP	N2V_KM
Karate	34	78	2	3	3	4	4	2	2
Polbooks	105	441	3	6	4	5	4	4	3
Football	115	613	12	12	6	10	10	9	12
Polblogs	1490	19090	2	306	277	416	276	272	2

4 Experiments

4.1 Datasets

We use four real-world networks to evaluate the effectiveness of our adversarial enhancement algorithms. For all datasets, the edges are treated as undirected and the ground-truth community labels are provided. TABLE 3 gives an overview of the networks, including the number of communities detected by each test method.

- **Zachary Karate club (Karate)** [49]. The network is about the pattern of relationship among the members of a Karate club at an American university, which splits into two groups after a dispute.
- **Books about US politics (Polbooks)** [50]. It is a network of books about US politics published in 2004 for presidential election. Links between books represent their frequent purchasing by the same buyers.
- **American College football (Football)** [7]. The network is about the schedule of games between American college football teams in regular season Fall 2000.
- **Political blogs (Polblogs)** [51]. The network consists of hyperlinks between weblogs on US politics recorded in 2005.

4.2 Evaluation Metrics

Benefit from the availablilty of the ground-truth community labels, we evaluate the community partitions using supervised metrics like normalized nutual information [52] and adjusted rand index. Note that we design the fitness in AE-GA using modularity $\mathcal{Q}$, thus it is not suitable to use it as the evaluation metric. The two evaluation metrics are briefly introduced in the following.

- **Normalized Mutual Information (NMI)** [52]. NMI is a commonly used criterion to evaluate the similarity of two clustering results. It quantifies how much information the estimated partition contains in the real partition. For two clusters X and Y, the NMI is defined as:

$$I_{\text{norm}}(X,Y) = \frac{2I(X,Y)}{H(X)+H(Y)}, \tag{11}$$

  where $I(X,Y) = H(Y) - H(X|Y)$ is the mutual information of X and Y, $H(Y)$ is the Shannon entropy of Y, and $H(X|Y)$ is the conditional entropy of X given Y.
- **Adjusted Rand Index (ARI)** [53]. ARI is the corrected-for-chance version of the Rand index, which measures the degree of agreement between an estimated partition and a real partition. It is defined as

$$ARI = \frac{RI - E[RI]}{\max(RI) - E[RI]}. \tag{12}$$

Both NMI and ARI require the ground-truth community labels for evaluation purpose and the values are in the range between 0 to 1. Note that ARI can yield negative values if the index is less than the expected index [53]. For both metrics, a larger value indicates a better partition.

4.3 Community Detection Methods

We consider the following six community detection algorithms in our experiments. The first five are available in the Python version of the `igraph` [1] library. The implement of `Node2vec` is available online [2].

- **Infomap (INF)** [54]. Infomap decomposes a network into modules by compressing the description of the information flow, i.e., it detects communities by minimizing the encoding length for a random walk.
- **Fast Greedy (FG)** [55]. This is a down-top hierarchical agglomeration algorithm. It merges individual vertices into communities based on a greedy modularity maximization strategy.
- **WalkTrap (WT)** [15]. It detects communities based on the idea that short random walks tend to stay in the same community.
- **Louvain (LOU)** [56]. This is a multi-level modularity optimization algorithm. It initializes each vertex with a separate community, and moves vertices between communities iteratively in a way that maximizes the vertices' local contributions to the overall modularity score.
- **Label Propagation (LP)** [57]. This method detects communities by initializing each vertex with a unique label and re-assigning each vertex the dominant label in its neighbourhood in each iteration.
- **Node2vec + Kmeans (N2V_KM)** [58]. This is a network embedding method, which learns lower-dimensional representations for vertices by biased random walk and skip-Gram. The K-means algorithm is then used to detect communities by clustering the embedded vectors of vertices in an Euclidean space.

We take the number of clustering K in K-means the same as the ground truth. Then, we obtain the same number of communities as the ground truth after feeding the network into N2V_KM according to TABLE 3.

1. https://igraph.org/python/
2. https://github.com/eliorc/node2vec

4.4 Adversarial Attack Methods

In order to test the defense capability of our enhancement algorithms against adversarial attacks, we also generate a series of adversarial graphs from each dataset by using the following two adversarial attack methods.

- **$\mathcal{Q}$-Attack** [27]. It is an evolutionary attack strategy based on genetic algorithm, in which the modularity is used to design the fitness function. This strategy deploys attack via negligible network rewiring, which doesn't change the degree of vertices, and achieves the state-of-the-art attack effect.
- **$\mathcal{D}_m$-Deception via Modularity** [19]. $\mathcal{D}_m$ is a community deception algorithm based on modularity, which can hide a target community via intra-community edge deletion and inter-community edge addition.

4.5 Baseline Enhancement Methods

We compare our adversarial enhancement algorithms with the following two baseline methods.

- **EdMot** [35]. It is an edge enhancement approach for motif-aware community detection via network rewiring and is proposed to address the hypergraph fragmentation issue.
- **WERW-Kpath** [32]. It is an enhancement approach for community detection via network weighting. It exploits random walks to compute the κ-path edge centrality, which is then used to weight the edges.

4.6 Experiment Setup

For all datasets, edges in graphs are treated as undirected and self-loops will be removed. The parameter settings for the proposed algorithms and baselines are shown in TABLE 4.

In AE-GA, the general parameters of GA are set as empirical values, and the proportion of elitist preservation $\mathcal{P}_e$ is set to 20%. Note that sample rate controls the upper limit of the chromosome size during Initialization, i.e., each chromosome will be initialized with an unfixed size not larger than $(\beta_a + \beta_b)m$. The sample rate of edge addition β_a is fixed to 3.0 for all experiments while the sample rate of edge deletion β_d is unfixed (0 for original networks and 0.2 for adversarial networks). For AE-VS, the selection of sample rate varies across datasets, and we will study in Sec. 4.7.2 the impact of the sample rate on the performance of AE-VS. Specifically, the sample rate β_a is set to 1.5, 2.7, 0.2 and 2.7 for Karate, Polbooks, Football and Polblogs, respectively.

For the community detection algorithm N2V_KM, we use the default setting for parameters in node2vec. Specifically, the walk length is 80, the number of walks per node is 10, the embedding dimension is 128, and both return hyper parameter p and in-out parameter q are equal to 1. The number of ground-truth communities is set as the input to K-means. Moreover, because of the randomness of LP and N2V_KM, we repeat experiments for 20 times and report the average results of community detection.

We generate adversarial networks for the two smaller datasets, Karate and Polbooks, via $\mathcal{Q}$-Attack, and use $\mathcal{D}_m$ to attack the other two larger datasets. Details of the adversarial networks are shown in TABLE 5.

TABLE 4
Parameters setting for the proposed algorithms and baselines.

Method	Parameter	Original	Adversarial
AE-GA	βa	3.0	3.0
	βb	0.0	0.2
	$\mathcal{P}_s$	120	
	$\mathcal{P}_c$	0.8	
	$\mathcal{P}_m$	0.02	
	$\mathcal{P}_e$	0.2	
	$\mathcal{T}_{ga}$	1000	
AE-VS	βa	(1.5, 2.7, 0.2, 2.7)	
EdMot	$\mathcal{K}$	1	
WERW-κPath	κ	$0.1m \sim 0.5m$	
	ρ	400 $\sim$ 2000	

TABLE 5
Details of adversarial networks.

Adversarial network	Attack method	Parameter	
		$\mathcal{S}$	β
Adv(Karate)	$\mathcal{Q}$-Attack	INF	10
Adv(Polbooks)	$\mathcal{Q}$-Attack	LOU	40
Adv(Football)	$\mathcal{D}_m$	WT	100
Adv(Polbolgs)	$\mathcal{D}_m$	LOU	5

4.7 Experiment Results

In order to verify the effectiveness of the adversarial enhancement on both the original networks and the networks under adversarial attacks, we respectively use the above-mentioned four methods, including AE-GA, AE-VS, EdMot and WERW-κPath, to show a crosswise comparison, with results reported in TABLE 6 , TABLE 7 and Fig. 5, where the scores are averaged over 20 runs for each method.

4.7.1 Adversarial Enhancement in Original Networks

TABLE 6 reports the results of adversarial enhancement for six community detection algorithms on four original networks. The results are expressed as the estimated values, where the average relative improvement rate (IMP) of metrices is also provided. For each column, we highlight the best result in bold. Based on the results, we have the following findings.

1) Our methods significantly outperform baselines on all original networks in most cases, i.e., both AE-GA and AE-VS achieve significant enhancement effect on six community detection algorithms, indicating stronger generalization and transferability.
2) By comparison, AE-VS not only has better performance, but also has lower time complexity and better interpretability than AE-GA.
3) For each dataset, those community detection algorithms of different performances on the original networks usually obtain more similar community structures during enhancement. Specifically, the standard deviations of the original values of NMI on the

TABLE 6
The results of four enhancement methods on four original networks. For each community detection method, the best result is highlighted in bold.

Karate	NMI							ARI						
	INF	FG	WT	LOU	LP	N2V_KM	IMP	INF	FG	WT	LOU	LP	N2V_KM	IMP
Original	0.711	0.707	0.724	0.618	0.632	0.820	0.00%	0.702	0.680	0.541	0.462	0.599	0.857	0.00%
EdMot	0.630	0.837	0.631	0.630	0.723	**0.837**	2.10%	0.513	0.882	0.513	0.517	0.693	0.882	4.67%
WERW-κPath	**0.838**	0.740	0.698	0.724	0.748	0.806	8.79%	0.803	0.600	0.504	0.541	0.686	0.849	4.40%
AE-GA	0.831	0.837	0.838	0.838	0.582	0.755	11.78%	**0.873**	0.882	**0.882**	0.882	0.587	0.788	32.99%
AE-VS	**0.838**	**1.000**	**0.839**	**1.000**	**0.801**	0.836	**27.65%**	0.803	**1.000**	0.802	**1.000**	**0.723**	**0.882**	**41.63%**
Polbooks	**NMI**							**ARI**						
	INF	FG	WT	LOU	LP	N2V_KM	IMP	INF	FG	WT	LOU	LP	N2V_KM	IMP
Original	0.503	0.531	0.563	0.516	0.552	0.537	0.00%	0.536	0.638	0.681	0.558	0.639	0.644	0.00%
EdMot	0.508	**0.574**	0.521	0.513	0.559	0.553	0.85%	0.567	**0.675**	0.612	0.549	0.655	0.661	0.81%
WERW-κPath	0.515	0.559	0.589	0.537	0.556	0.541	2.99%	0.531	0.671	**0.699**	0.600	0.617	0.648	1.92%
AE-GA	0.554	0.554	0.554	0.553	0.555	0.555	4.00%	0.652	0.652	0.652	0.652	0.642	0.655	6.41%
AE-VS	**0.610**	0.565	**0.589**	**0.559**	**0.607**	**0.578**	**9.71%**	**0.709**	0.663	0.690	**0.674**	**0.665**	**0.668**	**11.02%**
Football	**NMI**							**ARI**						
	INF	FG	WT	LOU	LP	N2V_KM	IMP	INF	FG	WT	LOU	LP	N2V_KM	IMP
Original	0.924	0.708	0.888	0.891	0.877	0.926	0.00%	0.897	0.474	0.815	0.807	0.756	0.892	0.00%
EdMot	0.924	0.713	0.855	0.834	0.891	0.925	-1.33%	**0.897**	0.442	0.713	0.626	0.774	**0.903**	-6.35%
WERW-κPath	0.924	0.777	0.907	0.859	**0.943**	0.926	2.63%	**0.897**	0.577	0.855	0.722	**0.938**	0.892	6.72%
AE-GA	**0.927**	0.855	**0.927**	0.916	0.873	0.925	4.61%	0.890	0.709	0.889	0.864	0.737	0.890	10.36%
AE-VS	0.926	**0.886**	0.922	**0.924**	0.913	**0.927**	**6.17%**	0.890	**0.794**	**0.892**	**0.881**	0.860	0.890	**16.48%**
Polblogs	**NMI**							**ARI**						
	INF	FG	WT	LOU	LP	N2V_KM	IMP	INF	FG	WT	LOU	LP	N2V_KM	IMP
Original	0.407	0.443	0.401	0.442	0.457	0.341	0.00%	0.441	0.528	0.419	0.521	0.551	0.289	0.00%
EdMot	0.428	0.447	0.405	0.449	0.463	0.361	2.65%	0.459	0.536	0.401	0.539	0.556	0.307	1.95%
WERW-κPath	0.406	0.442	0.401	0.439	0.452	0.331	-0.84%	0.443	0.527	0.418	0.522	0.543	0.282	-0.62%
AE-GA	0.453	0.525	0.504	**0.529**	**0.519**	0.381	16.81%	0.472	0.618	0.599	0.622	0.616	0.357	20.35%
AE-VS	**0.506**	**0.536**	**0.518**	0.527	0.519	**0.452**	**23.26%**	**0.614**	**0.633**	**0.612**	**0.629**	**0.618**	**0.468**	**33.37%**

four datasets are (0.073, 0.022, 0.081, 0.042), while those of the corresponding estimated values are (0.16, 0.021, 0.016, 0.030). These may indicate that our methods can make the network structure more stable and achieve consensus partition. Moreover, AE-VS achieves perfect enhancement on some community detection algorithms applied to small datasets. For instance, when enhancing FG and LOU via AE-VS on Karate dataset, both NMI and ARI are equal to 1, suggesting that FG and LOU algorithms can detect community structures completely correctly after enhancement. This also illustrates the increase of standard deviation for Karate network.

4.7.2 *Impact of Sample Rate in AE-VS*

Due to the outstanding performance of AE-VS, we further investigate the impact of the sample rate β_a on the enhancement results, as visualized in Fig. 4. Such impact behaves differently on different networks. According to TABLE 6, we can get a sorting based on the average NMI and ARI, i.e., `Polblogs < Polbooks < Karate < Football`. Specifically, Polblogs has a low average NMI equal to 0.415, and the performance of AE-VS is improved

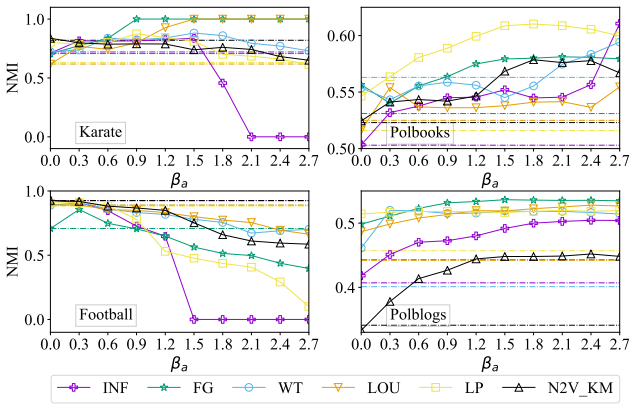

Fig. 4. The impact of sample rate β_a on the performance of AE-VS. The dotted lines of the same color mark the performances obtained by the corresponding community detection algorithm in original networks.

TABLE 7
The results of four enhancement methods on four adversarial networks. For each community detection method, the best result is highlighted in bold.

Karate	NMI							ARI						
	INF	FG	WT	LOU	LP	N2V_M	IMP	INF	FG	WT	LOU	LP	N2V_M	IMP
Original	0.711	0.707	0.724	0.618	0.632	0.820	33.04%	0.702	0.68	0.541	0.462	0.599	0.857	56.20%
Adv	0.487	0.487	0.487	0.487	0.547	0.671	0.00%	0.330	0.330	0.330	0.330	0.445	0.694	0.00%
EdMot	0.487	0.474	0.470	0.487	0.468	0.589	-6.03%	0.330	0.376	0.353	0.330	0.430	0.684	1.79%
WERW-κPath	0.500	0.496	0.480	0.431	0.545	0.580	-4.23%	0.345	0.354	0.313	0.308	0.388	0.572	-7.28%
AE-GA	0.688	0.821	**0.837**	0.666	0.653	0.656	36.48%	**0.715**	0.871	**0.882**	0.655	0.627	0.674	79.91%
AE-VS	**0.727**	**0.837**	0.806	**0.837**	**0.837**	**0.837**	**54.17%**	0.563	**0.882**	0.849	**0.882**	**0.882**	**0.882**	**100.89%**
Polbooks	**NMI**							**ARI**						
	INF	FG	WT	LOU	LP	N2V_M	IMP	INF	FG	WT	LOU	LP	N2V_M	IMP
Original	0.503	0.531	0.563	0.516	0.552	0.537	24.16%	0.536	0.638	0.681	0.558	0.639	0.644	33.19%
Adv	0.418	0.502	0.393	0.343	0.461	0.462	0.00%	0.459	0.598	0.351	0.252	0.534	0.581	0.00%
EdMot	0.501	**0.607**	0.501	0.454	**0.566**	0.515	21.91%	0.559	**0.667**	0.555	0.461	**0.642**	0.625	26.45%
WERW-κPath	0.395	0.392	0.378	0.445	0.440	0.488	-1.59%	0.423	0.467	0.431	0.485	0.515	0.574	4.32%
AE-GA	0.531	0.530	0.531	0.531	0.511	0.528	22.61%	**0.634**	0.632	**0.634**	0.634	0.603	**0.626**	35.60%
AE-VS	**0.541**	0.538	**0.564**	**0.538**	0.521	**0.538**	**25.63%**	0.621	0.634	0.621	**0.664**	0.635	0.618	**36.68%**
Football	**NMI**							**ARI**						
	INF	FG	WT	LOU	LP	N2V_M	IMP	INF	FG	WT	LOU	LP	N2V_M	IMP
Original	0.924	0.708	0.888	0.891	0.877	0.926	7.91%	0.897	0.474	0.815	0.807	0.756	0.892	36.86%
Adv	0.814	0.659	0.793	0.850	0.798	0.918	0.00%	0.498	0.364	0.481	0.696	0.476	0.876	0.00%
EdMot	0.814	0.642	0.844	0.816	0.758	0.921	-0.77%	0.498	0.334	0.634	0.618	0.365	0.886	-1.65%
WERW-κPath	0.814	0.696	0.883	0.843	0.835	0.913	3.15%	0.498	0.431	0.768	0.661	0.685	0.876	15.57%
AE-GA	**0.924**	0.776	**0.924**	**0.907**	0.856	**0.924**	**9.91%**	**0.897**	0.570	**0.897**	0.831	0.531	**0.897**	**36.33%**
AE-VS	0.814	**0.830**	**0.924**	0.904	**0.888**	0.909	9.04%	0.498	**0.646**	**0.897**	**0.843**	**0.777**	0.847	32.94%
Polblogs	**NMI**							**ARI**						
	INF	FG	WT	LOU	LP	N2V_M	IMP	INF	FG	WT	LOU	LP	N2V_M	IMP
Original	0.407	0.443	0.401	0.442	0.457	0.341	21.51%	0.441	0.528	0.419	0.521	0.551	0.289	20.15%
Adv	0.340	0.372	0.336	0.361	0.378	0.263	0.00%	0.376	0.461	0.343	0.445	0.459	0.204	0.00%
EdMot	0.355	0.368	0.345	0.379	0.388	0.287	3.51%	0.421	0.463	0.338	0.480	0.459	0.241	4.98%
WERW-κPath	0.333	0.387	0.347	0.383	0.400	0.249	2.39%	0.378	0.450	0.363	0.445	0.494	0.203	1.97%
AE-GA	0.473	0.527	0.428	**0.532**	**0.530**	0.377	39.85%	0.535	0.620	0.599	0.625	**0.625**	0.345	46.37%
AE-VS	**0.504**	**0.537**	**0.520**	0.529	0.518	**0.448**	**49.07%**	**0.610**	**0.633**	**0.614**	**0.631**	0.617	**0.459**	**55.77%**

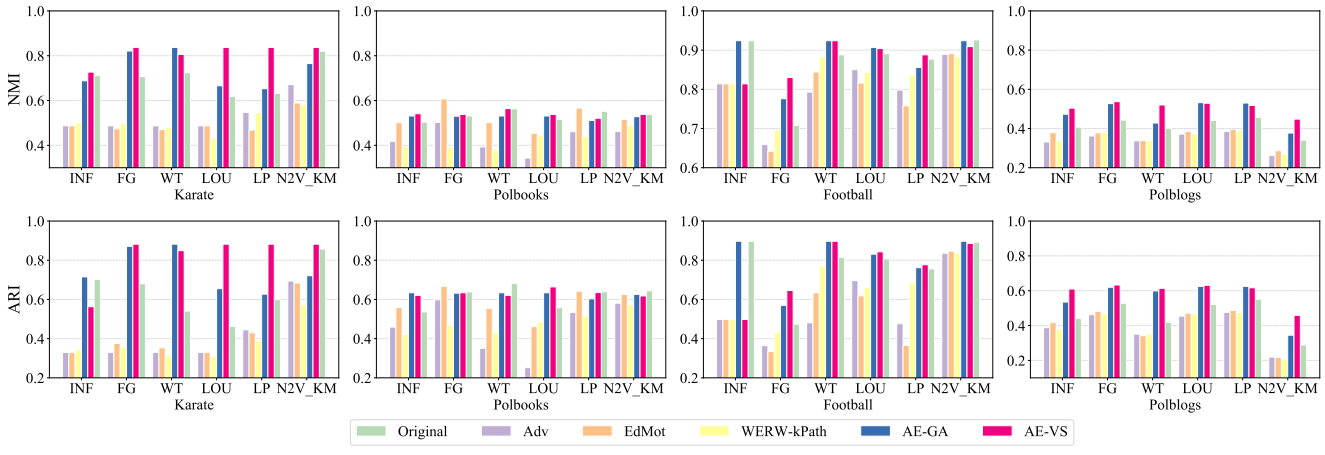

Fig. 5. The community detection performance of various algorithms before and after enhancement for the adversarial networks. Here, it includes the community detection results on the original networks, used as a reference.

with the increase of sample rate; Polbooks has an average NMI equal to 0.534, and its performance curve is messy but basically goes up. For Karate with the average NMI equal to 0.702, five of the six community detection algorithms have relatively stable performances while Infomap suffers from negative enhancement with a relatively large sample rate; Football has an average NMI up to 0.869, the performance of AE-VS drops steadily with the increase of sample rate.

As we can see, in general, the impact of sample rate on the performance of AE-VS is influenced by the network structure. That is, for networks with weak community structures and low average performance metrics, like Polblogs, most community detection algorithms have huge spaces to be enhanced, i.e., the performance of AE-VS is significantly improved with the increase of sample rate. But, for those networks with strong community structures like Football, it is difficult for AE-VS to further enhance the community detection, i.e., adding or deleting more links may even weaken the stable community structure, leading to the degradation of performance.

4.7.3 Adversarial Enhancement in Adversarial Networks

In graph data mining, adversarial attack aims to degrade the performance of algorithms by perturbing the graph structure or attacking the computational process. In social networks, the adversarial attack on community detection or link prediction probably facilitates to hide the real community structure or sensitive links. In order to address the defensive capability of our adversarial enhancement algorithms against such adversarial attacks, we also design experiments on adversarial networks obtained by slightly modifying the original networks via certain adversarial attacks.

The performance of the four enhancement algorithms on a series of adversarial networks is presented in TABLE 7, which is also visualized in Fig. 5 for a more intuitive perspective. Note that here we include the community detection results on the original networks and the adversarial networks as references. We can see that, first, the performance metrics, both NMI and ARI, are significantly smaller in the adversarial networks than those in the original networks, indicating that an adversarial attack has indeed broken the network structure and achieved a community detection deception. Then, during adversarial enhancement, again, our adversarial enhancement algorithms significantly outperform the baselines on all adversarial networks in most cases. In fact, both AE-GA and AE-VS help the six community detection algorithms achieve huge improvements on detection performances, which is even better than the results on the original networks. However, the baselines EdMot and WERW-κ-Path have mediocre performance and may fail with the increase of the attack strength. Such results indicate that our adversarial enhancement algorithms could help partially or even fully recover the network structures destroyed by adversarial attacks, validating their strong defensive capacity.

4.7.4 Visualization of AE-GA on Karate Network

As a case study, we visualize the adversarial enhancement (by AE-GA[3]) for algorithm LOU on Karate network, as shown in Fig. 6.

3. We choose AE-GA rather than AE-VS, since the latter integrates a number of results and thus is relatively difficult to be visualized.

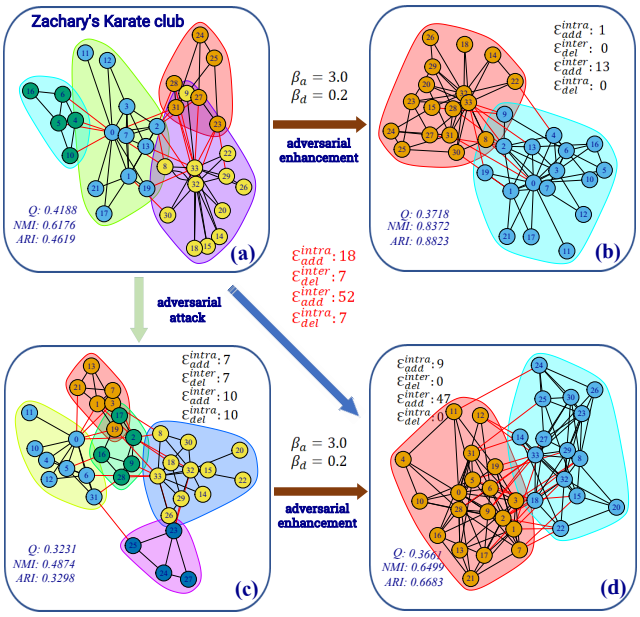

Fig. 6. Adversarial enhancement for LOU on Karate network.

The community structure found by LOU for the original network is shown in Fig. 6 (a), where there are four communities. Since the number of communities found by LOU is more than the ground truth ($M_{real} = 2$), as mentioned in Sec. 3.2.1, extra inter-community edge addition is available when $M_S > M_{real}$ ($S = \texttt{LOU}$). The result of adversarial enhancement on the original network is shown in Fig. 6 (b). This adversarial enhancement scheme consists of one intra-community edge addition and 13 inter-community edge additions, and achieves a significant improvement in community detection, leading to the increase of 35.56% and 91.02% in NMI and ARI, respectively. Specifically, a large number of inter-community edge additions successfully merge small clusters into larger ones, resulting in more accuracy partitions.

Notably, the decrease of modularity here (from 0.4188 to 0.3718) can explain why did not design modularity as fitness function directly. By comparing the information in Figs. 6 (a) and (b), a community partition with larger modularity does not mean closer to the ground-truth community structure. Therefore, in the situations where we know the number of communities in advance, we can combine modularity with the true number of clusters, to obtain more accurate optimization guidance. This has been shown to have excellent performance in enhancing community detection. However, when facing unlabeled networks, the fitness function degrades to modularity, i.e., $f = \mathcal{Q}$, so the enhancement may be weakened to a certain extent.

Now, consider the adversarial network obtain by $\mathcal{Q}$-Attack, as shown in Fig. 6 (c). $\mathcal{Q}$-Attack keeps the number of edges unchanged during community deception and achieves a 22.85% reduction in modularity with an attack cost of 17. As we can see, community structure suffers from structural damage and a new cluster that contains the fringe vertices in the original network is discovered. We then deploy adver-

TABLE 8
The average running time of the four enhancement algorithms. The test is performed on LOU with the same experimental setup.

Method \ Dataset	Karate	Polbooks	Football	Polblogs
EdMot	0.31	1.42	1.73	75.11
WERW-κPath	10.36	121.54	333.49	13203.22
AE-GA	278.40	838.87	2318.16	259200.00
AE-VS	0.51	1.96	2.62	88.36

sarial enhancement with the same setup to this adversarial network and obtain the enhanced network shown in Fig. 6 (d). Compared with the community partition in Fig. 6 (c), LOU achieves a better partition, which even surpasses that in the original network shown in Fig. 6 (a). Such result suggests that our adversarial enhancement can indeed help the existing community detection algorithms defend against adversarial attacks. More interestingly, it seems that such adversarial enhancement not only repairs the broken network structure caused by adversarial attack, but also further optimizes it to obtain a clearer community structure.

4.7.5 Computational Complexity Analysis

In order to compare the efficiency of our two adversarial enhancement algorithms and to see how fast they are, we roughly estimated their time complexity as follows.

- AE-GA runs in time $\mathcal{O}(\mathcal{P}_s \cdot \mathcal{T}_{ga} \cdot |\mathcal{S}|)$, where $|\mathcal{S}|$ is the time complexity of the community detection algorithm to be enhanced. For evolutionary algorithms, a fitness function is used to evaluate the quality of the results during iteration, which directly affects the optimization effeciency and convergence speeds of the algorithms. In AE-GA, the fitness function is computed by modularity $\mathcal{Q}$ and the number of communities $M_{\mathcal{S}}$, obtained from the community structure, which leads to an additional community detection before the calculation of fitness.
- AE-VS runs in time $\mathcal{O}(m \cdot \beta_a + |\mathcal{S}| + |\mathcal{VS}|)$. Note that $\mathcal{O}(|\mathcal{VS}|)$ is the maximum time complexity among selected similarity metrics. The weighted random sampling without replacement has a time complexity of $|E| \cdot \beta_a$. During the process of partition ensemble, which runs in time $\mathcal{O}(n^2)$, the selection of a threshold has a cost of $\mathcal{O}(|\mathcal{E}_{co}|)$, where $\mathcal{E}_{co}$ is the edge set of the co-occurrence network, for which the maximum possible number of edges is $n(n-1)/2$.

Moreover, we evaluate the efficiency of our algorithms by directly comparing the running time with baselines. The average running time (in seconds) of the four algorithms are presented in TABLE 8. As we can see, although the algorithm AE-GA performs well on small networks, it is limited by the optimization mode and does not scale well on large networks. Instead, AE-VS has a comparable time complexity with EdMot but scales well on large networks.

5 Conclusion

In this paper, we proposed adversarial enhancement to improve the performance of existing community detection algorithms. In particular, we put forward two adversarial enhancement algorithms, namely AE-GA and AE-VS, taking both defensive effect and transferability into account. Extensive experimental results demonstrate the superiority of our methods in helping six common community detection algorithms achieve significant performance improvements for both real-world networks and adversarial networks.

Moreover, we designed a fitness function with the number of clusters in AE-GA and the process of threshold selection in AE-VS, which can help the existing community detection algorithms solve the resolution limit in modularity optimization and achieve consensus partitions. Although there is a restriction of time complexity in AE-GA, AE-VS is effective and scales well on large networks.

Finally, our findings inspire more ideas for future works. For instance, the current selected community detection algorithms mainly focus on nonoverlapping community detection, therefore extending adversarial enhancement to overlapping case for community detection, could be interesting topics for further studies.

Acknowledgments

The authors would like to thank all the members in the IVSN Research Group, Zhejiang University of Technology for the valuable discussions about the ideas and technical details presented in this paper.

References

[1] C. Durugbo, W. Hutabarat, A. Tiwari, and J. R. Alcock, "Modelling collaboration using complex networks," *Information Sciences*, vol. 181, no. 15, pp. 3143–3161, 2011.
[2] S. H. Strogatz, "Exploring complex networks," *Nature*, vol. 410, no. 6825, p. 268, 2001.
[3] M. E. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, no. 2, pp. 167–256, 2003.
[4] A.-L. Barabási, "Scale-free networks: a decade and beyond," *Science*, vol. 325, no. 5939, pp. 412–413, 2009.
[5] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.
[6] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, p. 440, 1998.
[7] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.
[8] M. E. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Physical Review E*, vol. 74, no. 3, p. 036104, 2006.
[9] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.
[10] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3-5, pp. 75–174, 2010.
[11] B. Karrer and M. E. Newman, "Stochastic blockmodels and community structure in networks," *Physical Review E*, vol. 83, no. 1, p. 016107, 2011.
[12] T. P. Peixoto, "Hierarchical block structures and high-resolution model selection in large networks," *Physical Review X*, vol. 4, no. 1, p. 011047, 2014.
[13] B. Ball, B. Karrer, and M. E. Newman, "Efficient and principled method for detecting communities in networks," *Physical Review E*, vol. 84, no. 3, p. 036103, 2011.
[14] M. B. Hastings, "Community detection as an inference problem," *Physical Review E*, vol. 74, no. 3, p. 035102.
[15] P. Pons and M. Latapy, "Computing communities in large networks using random walks," in *International Symposium on Computer and Information Sciences*. Springer, 2005, pp. 284–293.
[16] V. Zlatić, A. Gabrielli, and G. Caldarelli, "Topologically biased random walk and community finding in networks," *Physical Review E*, vol. 82, no. 6, p. 066109, 2010.

[17] A. Arenas, A. Díaz-Guilera, and C. J. Pérez-Vicente, "Synchronization reveals topological scales in complex networks," *Physical Review Letters*, vol. 96, no. 11, p. 114102, 2006.

[18] S. Boccaletti, M. Ivanchenko, V. Latora, A. Pluchino, and A. Rapisarda, "Detecting complex network modularity by dynamical clustering," *Physical Review E*, vol. 75, no. 4, p. 045102, 2007.

[19] V. Fionda and G. Pirro, "Community deception or: How to stop fearing community detection algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 4, pp. 660–673, 2017.

[20] M. Ciglan, M. Laclavík, and K. Nørvåg, "On community detection in real-world networks and the importance of degree assortativity," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2013, pp. 1007–1015.

[21] S. Fortunato and M. Barthelemy, "Resolution limit in community detection," *Proceedings of the National Academy of Sciences*, vol. 104, no. 1, pp. 36–41, 2007.

[22] L. Lü and T. Zhou, "Link prediction in complex networks: A survey," *Physica A: Statistical Mechanics and its Applications*, vol. 390, no. 6, pp. 1150–1170, 2011.

[23] P. Holme, B. J. Kim, C. N. Yoon, and S. K. Han, "Attack vulnerability of complex networks," *Physical Review E*, vol. 65, no. 5, p. 056109, 2002.

[24] M. Bellingeri, D. Cassi, and S. Vincenzi, "Efficiency of attack strategies on complex model and real-world networks," *Physica A: Statistical Mechanics and its Applications*, vol. 414, pp. 174–180, 2014.

[25] B. Karrer, E. Levina, and M. E. Newman, "Robustness of community structure in networks," *Physical Review E*, vol. 77, no. 4, p. 046119, 2008.

[26] M. Waniek, T. P. Michalak, M. J. Wooldridge, and T. Rahwan, "Hiding individuals and communities in a social network," *Nature Human Behaviour*, vol. 2, no. 2, p. 139, 2018.

[27] J. Chen, L. Chen, Y. Chen, M. Zhao, S. Yu, Q. Xuan, and X. Yang, "Ga-based q-attack on community detection," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 3, pp. 491–503, 2019.

[28] S. Yu, M. Zhao, C. Fu, H. Huang, X. Shu, Q. Xuan, and G. Chen, "Target defense against link-prediction-based attacks via evolutionary perturbations," *arXiv preprint arXiv:1809.05912*, 2018.

[29] H. Dai, H. Li, T. Tian, X. Huang, L. Wang, J. Zhu, and L. Song, "Adversarial attack on graph structured data," *arXiv preprint arXiv:1806.02371*, 2018.

[30] A. Bojcheski and S. Günnemann, "Adversarial attacks on node embeddings," *arXiv preprint arXiv:1809.01093*, 2018.

[31] D. Zügner, A. Akbarnejad, and S. Günnemann, "Adversarial attacks on neural networks for graph data," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2018, pp. 2847–2856.

[32] P. De Meo, E. Ferrara, G. Fiumara, and A. Provetti, "Enhancing community detection using a network weighting strategy," *Information Sciences*, vol. 222, pp. 648–668, 2013.

[33] P. G. Sun, "Weighting links based on edge centrality for community detection," *Physica A: Statistical Mechanics and Its Applications*, vol. 394, pp. 346–357, 2014.

[34] D. Lai, H. Lu, and C. Nardini, "Enhanced modularity-based community detection by random walk network preprocessing," *Physical Review E*, vol. 81, no. 6, p. 066118, 2010.

[35] P.-Z. Li, L. Huang, C.-D. Wang, and J.-H. Lai, "Edmot: An edge enhancement approach for motif-aware community detection," *arXiv preprint arXiv:1906.04560*, 2019.

[36] A. Lancichinetti and S. Fortunato, "Consensus clustering in complex networks," *Scientific Reports*, vol. 2, p. 336, 2012.

[37] J. Dahlin and P. Svenson, "Ensemble approaches for improving community detection methods," *arXiv preprint arXiv:1309.0242*, 2013.

[38] D. He, H. Wang, D. Jin, and B. Liu, "A model framework for the enhancement of community detection in complex networks," *Physica A: Statistical Mechanics and its Applications*, vol. 461, pp. 602–612, 2016.

[39] D. Lin *et al.*, "An information-theoretic definition of similarity." in *International Conference on Machine Learning*, vol. 98, no. 1998. Citeseer, 1998, pp. 296–304.

[40] T. Zhou, L. Lü, and Y.-C. Zhang, "Predicting missing links via local information," *The European Physical Journal B*, vol. 71, no. 4, pp. 623–630, 2009.

[41] L. Lin-Yuan, "Link prediction on complex networks [j]," *Journal of University of Electronic Science and Technology of China*, vol. 5, 2010.

[42] P. Jaccard, "Étude comparative de la distribution florale dans une portion des alpes et des jura," *Bull Soc Vaudoise Sci Nat*, vol. 37, pp. 547–579, 1901.

[43] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. mcgraw-hill, 1983.

[44] E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai, and A.-L. Barabási, "Hierarchical organization of modularity in metabolic networks," *Science*, vol. 297, no. 5586, pp. 1551–1555, 2002.

[45] L. A. Adamic and E. Adar, "Friends and neighbors on the web," *Social Networks*, vol. 25, no. 3, pp. 211–230, 2003.

[46] L. Lü, C.-H. Jin, and T. Zhou, "Similarity index based on local paths for link prediction of complex networks," *Physical Review E*, vol. 80, no. 4, p. 046122, 2009.

[47] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Computer Networks and ISDN Systems*, vol. 30, no. 1-7, pp. 107–117, 1998.

[48] S. Monti, P. Tamayo, J. Mesirov, and T. Golub, "Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data," *Machine Learning*, vol. 52, no. 1-2, pp. 91–118, 2003.

[49] W. W. Zachary, "An information flow model for conflict and fission in small groups," *Journal of Anthropological Research*, vol. 33, no. 4, pp. 452–473, 1977.

[50] M. E. Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences*, vol. 103, no. 23, pp. 8577–8582, 2006.

[51] A. Lada and G. Natalie, "The political blogosphere and the 2004 us election," in *Proceedings of the 3rd International Workshop on Link Discovery*, vol. 1, 2005, pp. 36–43.

[52] L. Danon, A. Diaz-Guilera, J. Duch, and A. Arenas, "Comparing community structure identification," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2005, no. 09, p. P09008, 2005.

[53] L. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, vol. 2, no. 1, pp. 193–218, 1985.

[54] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proceedings of the National Academy of Sciences*, vol. 105, no. 4, pp. 1118–1123, 2008.

[55] A. Clauset, M. E. Newman, and C. Moore, "Finding community structure in very large networks," *Physical Review E*, vol. 70, no. 6, p. 066111, 2004.

[56] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, p. P10008, 2008.

[57] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Physical Review E*, vol. 76, no. 3, p. 036106, 2007.

[58] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 855–864.