Hide and Seek: Outwitting Community Detection Algorithms

Shravika Mittal, Debarka Sengupta, and Tanmoy Chakraborty, *Member, IEEE*

***Abstract*— Community affiliation of a node plays an important role in determining its contextual position in the network, which may raise privacy concerns when a sensitive node wants to hide its identity in a network. Oftentimes, a target community seeks to protect itself from adversaries so that its constituent members remain hidden inside the network. The current study focuses on hiding such sensitive communities so that the community affiliation of the targeted nodes can be concealed. This leads to the problem of *community deception*, which investigates the avenues of minimally rewiring nodes in a network so that a given target community maximally hides from a community detection algorithm (CDA). We formalize the problem of community deception and introduce Network deception using permanence loss (NEURAL), a novel method that greedily optimizes a *node-centric objective function* to determine the rewiring strategy. Theoretical settings pose a restriction on the number of strategies that can be employed to optimize the objective function, which in turn reduces the overhead of choosing the best strategy from multiple options. We also show that our objective function is *submodular* and *monotone*. When tested on both synthetic and seven real-world networks, NEURAL is able to deceive six widely used CDAs. We benchmark its performance with respect to four state-of-the-art methods on four evaluation metrics. In addition, our qualitative analysis on three other attributed real-world networks reveals that NEURAL, quite strikingly, captures important metainformation about edges that otherwise could not be inferred by observing only their topological structures.**

***Index Terms*— Community detection, community hiding, complex networks, permanence.**

I. Introduction

DETECTING communities from large networks has remained as one of the major research problems in the last two decades. Different heuristics, metrics, and optimization techniques have been proposed to detect communities from multiple types of networks [1]. However, of late, limited efforts have been visible to understand how easily a community detection algorithm (CDA) can be deceived by minimal rewiring of nodes.

In this article, we ask a fundamental question: *how do we hide a target community from being exposed to a CDA, assuming limited rewiring operations are allowed?* In other words, can a node or a community disguise its positioning in the network in order to escape detection [2]? We call this problem hide and seek community (HSC). Answering this question matters since it helps the social network users in hiding their identity from online surveillance[1] [4]. It also helps law-enforcement organizations identify criminal acts deceiving online identity [5]. This may also be useful for counter-terrorism units in order to deploy spies into a terrorist network. The solution to the current problem would help the spies determine who they should start a new friendship with (edge addition) or which existing friendship they should try to break (edge deletion) to conceal their community identity. However, one may argue that the same method can be misused by adversaries. Nonetheless, we believe that our investigation brings issues to light for the plan of novel community detection methods vigorous to deception strategies.

To date, the fundamental question stated above has got very little attention as most of the focus has been concentrated toward building efficient algorithms for community detection. Nagaraja [6] made a pioneering attempt to examine the degree of network information required by an attacker to infer the community membership information. Recently, Waniek *et al.* [2] proposed a heuristic-based solution to evade network centrality analysis. Fionda and Pirrò [5] proposed a novel metric and greedily optimized it to hide the members of a target community from being detected by the CDAs. Liu *et al.* [7] proposed an approach to maximally hide the *entire* community structure (as opposed to a target community) with a minimum rewiring of the network structure.

Here, we pose the HSC problem as a constrained optimization problem. The objective function is designed based on *permanence* [8], a node-centric metric we proposed previously, which has been proved to be highly effective in detecting the entire community structure of a network. Permanence, being a local metric, uses limited information of a node to determine its community membership. We theoretically prove that only two types of edge update operations (intercommunity edge addition and intracommunity edge deletion) are useful for rewiring nodes to optimize our proposed objective function.

Manuscript received October 28, 2020; revised February 2, 2021; accepted February 20, 2021. Date of publication March 12, 2021; date of current version August 2, 2021. This work was supported in part by the Ramanujan Fellowship and in part by the Department of Science and Technology (DST) under Grant ECR/2017/00l691. The work of Tanmoy Chakraborty was supported by the Centre for Artificial Intelligence (CAI), Indraprastha Institute of Information Technology Delhi (IIIT-Delhi). *(Corresponding author: Shravika Mittal.)*

Shravika Mittal and Tanmoy Chakraborty are with the Department of Computer Science and Engineering, Indraprastha Institute of Information Technology Delhi, New Delhi 110020, India (e-mail: shravika16093@iiitd.ac.in; tanmoy@iiitd.ac.in).

Debarka Sengupta is with the Department of Computer Science and Engineering, Indraprastha Institute of Information Technology Delhi, New Delhi 110020, India, and also with the Department of Computational Biology, Indraprastha Institute of Information Technology Delhi, New Delhi 110020, India (e-mail: debarka@iiitd.ac.in).

This article has supplementary downloadable material available at https://doi.org/10.1109/TCSS.2021.3062711, provided by the authors.

Digital Object Identifier 10.1109/TCSS.2021.3062711

[1]Mislove *et al.* [3] showed how by breaking down Facebook user network and attributes of certain users, it is possible to gather private data about other Facebook users.

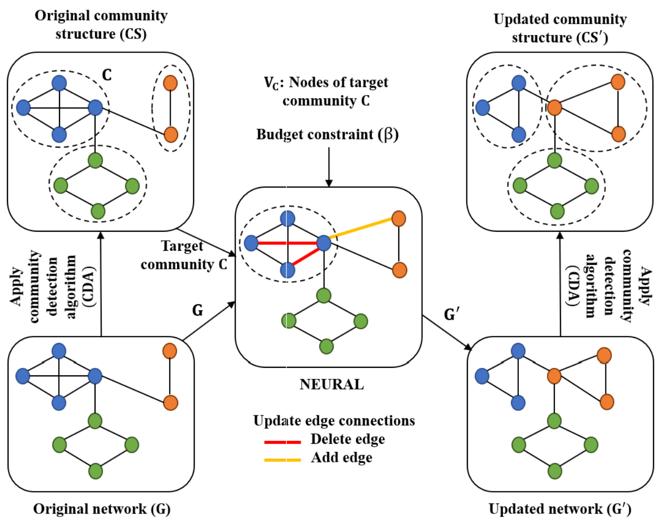
Original community structure (CS)
C
Updated community structure (CS′)
V_C: Nodes of target community C
Budget constraint (β)
Target community C
Apply community detection algorithm (CDA)
G
G′
NEURAL
Update edge connections
Delete edge
Add edge
Original network (G)
Updated network (G′)

Fig. 1. Flow diagram showing the procedure of NEURAL.

We further show that the objective function is *submodular* and *monotone* with respect to the required edge updates. Therefore, we propose **Ne**twork deception **u**sing pe**r**m**a**nence **l**oss (**NEURAL**), a greedy optimization algorithm to optimize the objective function. Given a network G, its community structure CS obtained from a CDA and a target community C whose constituent nodes V_C need to be concealed, NEURAL rewires nodes within the rewiring budget β in such a way that CDA is unable to identify the original community affiliation of V_C (Fig. 1 shows a flow diagram).

Extensive experiments are conducted on both synthetic and seven real-world networks. Six widely used CDAs are considered for deception. We compare NEURAL with four state-of-the-art community deception methods. The performance is measured based on four evaluation metrics (two of them are proposed by us). Our quantitative analysis shows that NEURAL significantly outperforms others across all the data sets and all the evaluation metrics.

We further conduct a detailed qualitative analysis to explain the physical significance of edges selected by the deception methods on three attributed real-world networks—citation network, terrorist network, and breast cancer network. Surprisingly, we observe that NEURAL is able to capture important metainformation of edges that otherwise could not be inferred just by observing the topological structure of networks.

In short, our major contributions are fourfold.

1) *Novel Objective Function:* Our proposed objective function is novel, which considers minimum information of nodes for network rewiring.
2) *Novel Algorithm:* We propose NEURAL, a novel greedy optimization algorithm for community deception.
3) *Quantitative Evaluation:* We perform an extensive evaluation on multiple data sets and show that NEURAL outsmarts existing approaches for hiding the target community within the specific budget.
4) *Qualitative Evaluation:* We further interpret the edges selected for node rewiring by the deception methods and show that NEURAL captures important metainformation of edges in three real-world networks.

Reproducibility: The codes and data sets are available at: https://github.com/mittalshravika/HideAndSeek-NEURAL.

II. Related Work

A. Community Detection

There has been a plethora of research in the detection of communities from a given network. These include traditional clustering-based algorithms such as hierarchical clustering, partitional clustering, and spectral clustering, which group nodes together based on a similarity metric [9]. Another class of CDAs revolves around the optimization of metrics that define the quality of a network partition, such as modularity [10], [11], conductance [12], and cut ratio [13]. Few other methods are based on random walks [14], information theory [15], [16], and spectral algorithms [17], [18]. Algorithms that detect overlapping communities have also been proposed [19], [20]. A detailed study of CDAs can be found in [21] and [22].

B. Community Deception

Another area of interest that has started revolving very recently is *community deception*, i.e., hiding a target community or the entire community structure from getting exposed to CDAs. Nagaraja [6] proposed a counter detection method for hiding a community by adding edges under a certain budget. The endpoints of edges to be added are chosen using vertex-centrality measures (degree centrality, eigenvector centrality, and random initialization). Waniek *et al.* [2] proposed DICE, an algorithm that deletes intracommunity edges (*disconnect internal*) and adds intercommunity edges (*connect external*), inspired by the functioning of modularity. They also devised a metric to quantify the concealment of a target community in the network. Fionda and Pirrò [5] referred to the problem of hiding a community as *community deception*. They devised a greedy optimization algorithm (dubbed SADDEN henceforth) to hide a target community based on *safeness gain*, a new metric that they proposed to quantify how safe a node is under adversarial attack. SADDEN requires the knowledge of the local community rather than knowing the entire community structure of the network to deceive CDAs. Along with this, the authors proposed a metric, called *deception score*, to quantify the effect of the community deception algorithm on the network. They also showed that their method outperforms modularity-based approaches. Recently, Liu *et al.* [7] extended the problem of hiding a target community to hiding the *entire community structure*. They proposed an algorithm for *community structure deception* based on information theory using network entropy minimization.

We consider all the methods mentioned above (Nagaraja, DICE, and SADDEN) as baselines[2] along with a random edge rewiring method, except Liu *et al.* [7] as this method focuses on the deception of the *entire community structure* (instead of a *single target community*); moreover, the metric used in their method (community-based structural entropy) requires entire community information.

C. How NEURAL Is Different From Others?

Table I summarizes how NEURAL is different from the existing methods for community deception. NEURAL uses permanence as a metric to determine how to update a given

[2]To our knowledge, these are the only existing methods that attempted to solve the HSC problem.

TABLE I

COMPARISON OF NEURAL WITH EXISTING METHODS. **E**+, −: EDGE ADDITION AND DELETION. NC: NODE CENTRALITY. **E** → **C**: EDGES CONNECTED WITH THE NODES IN THE TARGET COMMUNITY **C**. QA: QUALITATIVE ANALYSIS

Method	Metric	Strategy	Knowledge	QA
Nagaraja [6]	Modularity	$E+$	NC	No
DICE [2]	Modularity	$E+,-$	$E \rightarrow C$	No
SADDEN [5]	Safeness	$E+,-$	$E \rightarrow C$	No
NEURAL	Permanence	$E+,-$	$E \rightarrow C$	Yes

TABLE II

STATISTICS OF THE REAL-WORLD NETWORKS ($|V|$ AND $|E|$ REPRESENT THE NUMBER OF NODES AND EDGES, RESPECTIVELY; $\langle k \rangle$ (k_{MAX}) REPRESENTS THE AVERAGE (MAXIMUM) DEGREE OF NODES)

Network	$\|V\|$	$\|E\|$	$\langle k \rangle$	k_{max}
Kar	34	78	4.59	17
Dol	62	159	5.13	12
Lesmis	77	154	6.60	36
Polbook	105	441	8.40	25
Adjnoun	112	425	7.60	49
Power	4,941	6,594	2.67	19
Dblp	317,080	1,049,866	4.93	343

network efficiently in order to hide the target community. We perform comprehensive evaluation on both synthetic and real-world networks using four different evaluation metrics. We further perform a qualitative analysis on three attributed networks to understand the significance of the selected edges.

III. PROBLEM FORMULATION

A. Preliminaries

A network $G = (V, E)$ is defined as an undirected graph with V as the set of vertices and E as the set of edges. After applying a CDA on G, we get $CS = (C_1, C_2, \ldots, C_k)$ as the community structure. We only consider communities that are nonoverlapping. For community $C \in CS$, an intracommunity edge $\langle u, v \rangle$ is defined such that $u, v \in C$ and an intercommunity edge $\langle u, v \rangle$ is defined such that $u \in C$ and $v \in C'$ where $C \cap C' = \phi$. $E_{\text{intra}}(C)$ (*resp.* $E_{\text{inter}}(C)$) denotes the set of intra- (*resp.* inter-) community edges corresponding to C.

B. Hide and Seek Community

Our primary goal is to come up with an algorithm that, with minimum edge rewiring, is able to hide a given target community C from a community detection method. In other words, the actual community membership information of nodes inside C should not be revealed by the community detection method. This is done by rearranging the structure of the network using a certain number (β) of edge updates (which we call *budget* for network rewiring). We also assume that each edge update operation will incur a unit cost. One approach would be to search through the entire space for possible edge updates exhaustively and select the ones that are able to hide the target community C the most. However, searching through this huge space of all the possible combinations of edge updates would become computationally expensive in case of large networks. Along with this, such an exhaustive technique would require the knowledge of the entire network and may also depend on the type of CDA that we intend to fool.

To avoid this, we introduce the problem, called HSC, to camouflage a target community C from a community detection method.

*Definition 3.1: (*HSC) For a network $G = (V, E)$, the problem of HSC is to hide a target community C with the help of network edge updates constrained by a parameter β. It can be posed as a constrained optimization problem as follows:

$$\underset{E'(C)}{\operatorname{argmax}} \mathcal{F}(C, E(C), \beta, E'(C)) \tag{1}$$

where, $E(C) = E_{\text{intra}}(C) \cup E_{\text{inter}}(C)$, $E'(C) = (E(C) \cup E_{\text{add}}) \setminus E_{\text{del}}$, and E_{add} (*resp.* E_{del}) indicates the set of edges to be added (*resp.* deleted) to hide C such that $|E_{\text{add}}| + |E_{\text{del}}| \leq \beta$.

IV. METHODOLOGY

We consider *permanence* [8], [23], a node-centric metric[3] to design the objective function $\mathcal{F}$ in (1). We theoretically show that limited edge update operations are required to maximize the *permanence loss* (our objective function). We also show that permanence loss is submodular and monotone with respect to each of the edge update operations. Therefore, we propose NEURAL, a greedy algorithm that makes use of permanence loss in order to hide a target community C. This section first briefly describes permanence, followed by the greedy strategy used in NEURAL.

A. Permanence

Chakraborty *et al.* proposed permanence [8], [23], a vertex-centric metric that quantifies the containment of a node v in a network community C. The formulation of permanence is based on three factors: 1) the internal pull $I(v)$, denoted by the internal connections of a node v within its own community; 2) maximum external pull $E_{\max}(v)$, denoted by the maximum connections of v to its neighboring communities; and 3) internal clustering coefficient of v, $C_{\text{in}}(v)$, denoted by the fraction of actual and possible number of edges among the internal neighbors of v. The above three factors are then suitably combined to obtain the permanence of v as

$$\text{Perm}(v, G) = \frac{I(v)}{E_{\max}(v)} \times \frac{1}{\deg(v)} - (1 - C_{\text{in}}(v)). \tag{2}$$

Fig. 2 shows a toy example to calculate the permanence value of a node.

This metric indicates that a vertex would remain in its own community as long as its internal pull is greater than the external pull or its internal neighbors are densely connected to each other, hence forming a near clique. The permanence for a network G is then defined as $\text{Perm}(G) = (\sum_{v \in V} \text{Perm}(v)/|V|)$. The reasons behind choosing permanence instead of other community scoring metrics, such as (local) modularity [24], [25], conductance, and cut ratio [1], are twofold: 1) permanence is a vertex-centric local metric, which would enable us to update edges incrementally in order to change the network structure without looking into the entire network structure, and 2) permanence has been shown to be superior to other local and global scoring metrics for community detection [22].

[3]Permanence can also be computed for an entire network.

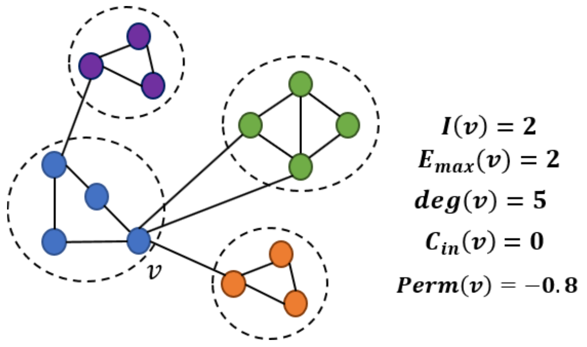

Fig. 2. Toy example demonstrating the calculation of permanence for a node, given the network and the community structure.

B. Proposed Objective Function: Permanence Loss

Our proposed community deception method NEURAL (discussed in Section IV-D) aims to reduce the permanence of the network for a target community C to be hidden from CDAs. We propose to do so because reducing permanence of a vertex would disrupt its containment in the original community, changing the community structure of the network, making it difficult for detection algorithms to identify the original communities. We search for edge updates (addition/deletion of edges) by maximizing the *permanence loss* at every iteration, which is defined as

$$\mathcal{P}_l = \text{Perm}(G) - \text{Perm}(G') \tag{3}$$

where G represents the original network and G' represents the modified network after updating edges (see Fig. 1) with respect to the target community C, elaborated in Sections IV-C and IV-D.

In Section IV-C, we will show that the permanence loss will be affected (positively) only due to the intracommunity edge deletion and intercommunity edge addition. Readers are encouraged to see the Supplementary Material where we show that permanence loss is submodular and monotone with respect to each of the edge updates stated above.

C. Edge Updates

In this section, we describe four possible edge update operations to maximize permanence loss $\mathcal{P}_l$ in NEURAL—intercommunity and intracommunity edge deletion, and intercommunity and intracommunity edge addition.

1) Intercommunity Edge Deletion:

Theorem 4.1: Deleting an intercommunity edge $\langle u, v\rangle$ where $u \in C$ and $v \in C'$ such that $C \cap C' = \phi$ does not result in permanence loss.

Proof: In this proof, we show that deleting an intercommunity edge does not amount to permanence loss. An intercommunity edge deletion just affects the permanence measure for u and v. We will only show the change in permanence for node u (same applies to v). There can be two cases given in the following.

1) $\mathbf{E_{max}(u)}$ Does Not Change After Edge Deletion: In this case, we assume that the maximum external connections for node u remain the same after deleting $\langle u, v\rangle$. Deleting $\langle u, v\rangle$ would not change $C_{\text{in}}(u)$. It would only decrease its degree by 1. Therefore, for permanence loss, we need to see whether $\mathcal{P}_l = \text{Perm}(u, G) - \text{Perm}(u, G') \geq 0$. This reduces to
$$\mathcal{P}_l = \frac{I(u)}{E_{\max}(u)} \times \left[\frac{1}{\deg(u)} - \frac{1}{\deg(u)-1}\right] < 0.$$
Therefore, no permanence loss is possible in this case.

2) $\mathbf{E_{max}(u)}$ Changes After Edge Deletion: In this case, we assume that the deletion of edge $\langle u, v\rangle$ affects the maximum external connections of node u. This is the case where C' is the only community that has the maximum external pull for node u. As a result, along with degree, $E_{\max}(u)$ would also decrease by 1. It would not change $C_{\text{in}}(u)$. Therefore, for permanence loss, we need to see whether $\mathcal{P}_l = \text{Perm}(u, G) - \text{Perm}(u, G') \geq 0$. This reduces to

$$\begin{aligned}\mathcal{P}_l &= I(u)\left[\frac{1}{E_{\max}(u)\times\deg(u)} - \frac{1}{(E_{\max}(u)-1)\times(\deg(u)-1)}\right]\\ &= I(u)\left[\frac{1-E_{\max}(u)-\deg(u)}{E_{\max}(u)\times\deg(u)\times(E_{\max}(u)-1)\times(\deg(u)-1)}\right]\\ &< 0 (E_{\max}(u) \geq 1 \text{ and } \deg(u) \geq 1 \text{ because of edge}\langle u, v\rangle).\end{aligned}$$

Therefore, no permanence loss is possible in the case of deleting an intercommunity edge. ∎

2) Intracommunity Edge Deletion:

Theorem 4.2: Deleting an intracommunity edge $\langle u, v\rangle$ where $u, v \in C$, always results in permanence loss.

Proof: Here, we show that deleting an intracommunity edge always results in permanence loss. We will only show the change in permanence for node u (the same applies to v). Such an edge update would decrease the internal degree and degree of node u by 1. It would not affect $E_{\max}(u)$ (no external connections are being changed). We narrow our search space such that $C_{\text{in}}(u)$ decreases after the deletion of $\langle u, v\rangle$. Therefore, for permanence loss, we need to see whether $\mathcal{P}_l = \text{Perm}(u, G) - \text{Perm}(u, G') \geq 0$. This reduces to

$$\begin{aligned}\mathcal{P}_l &= \frac{1}{E_{\max}(u)}\left[\frac{I(u)}{\deg(u)} - \frac{I(u)-1}{\deg(u)-1}\right]\\ &= \frac{1}{E_{\max}(u)}\left[\frac{\deg(u)-I(u)}{\deg(u)\times(\deg(u)-1)}\right]\\ &\geq 0 \text{ (as } \deg(u) \geq I(u)).\end{aligned} \tag{4}$$

Therefore, deleting an intracommunity edge $\langle u, v\rangle$ would bring in permanence loss in terms of nodes u and v.

The intracommunity edge deletion would also affect the permanence measure for nodes that have both u and v as their neighbors. If so, it would result in a change in their internal clustering coefficient value with all the other factors unchanged. For permanence loss due to such a node w, we need to see whether $\mathcal{P}_l = \text{Perm}(w, G) - \text{Perm}(w, G') \geq 0$. This reduces to

$$\mathcal{P}_l = (1 - C'_{\text{in}}(w)) - (1 - C_{\text{in}}(w)) = C_{\text{in}}(w) - C'_{\text{in}}(w)$$

where $C'_{\text{in}}(w)$ represents the updated internal clustering coefficient of w. In the above equation, $\mathcal{P}_l > 0$ since $C_{\text{in}}(w) > C'_{\text{in}}(w)$. For node w, the number of neighbors is intact, but the edges between its neighbors get reduced by 1 after $\langle u, v\rangle$ is deleted. As a result, the internal clustering coefficient reduces, again resulting in permanence loss. Therefore, deleting $\langle u, v\rangle$ would also bring in permanence loss in terms of their common neighbors. ∎

3) Intercommunity Edge Addition:

Theorem 4.3: Adding an intercommunity edge $\langle u, v\rangle$ where $u \in C$ and $v \in C'$, such that $C \cap C' = \phi$, always results in permanence loss. The loss is more if C' is the community that provides the maximum external pull for node u.

Proof: In this proof, we show that adding an intercommunity edge always causes permanence loss. An intercommunity edge addition just affects permanence for nodes u and v. We will only show the change in permanence for node u (same applies to v). There can be two cases given in the following.

1) $\mathbf{E_{max}(u)}$ Does Not Change After Edge Addition: In this case, we assume that the maximum external connections for node u remain the same after adding $\langle u, v\rangle$. Adding $\langle u, v\rangle$ would have no effect on $C_{\text{in}}(u)$. It would only increase its degree by 1. Therefore, for permanence loss, we need to see whether $\mathcal{P}_l = Perm(u, G) - Perm(u, G') \geq 0$. This reduces to

$$\mathcal{P}_l = \frac{I(u)}{E_{\max}(u)} \times \left[\frac{1}{\deg(u)} - \frac{1}{\deg(u)+1}\right] > 0. \quad (5)$$

Therefore, there is a permanence loss in the case of adding an intercommunity edge such that $E_{\max}(u)$ does not change after edge addition.

2) $\mathbf{E_{max}(u)}$ Changes After Edge Addition: In this case, we assume that the addition of edge $\langle u, v\rangle$ affects the maximum external connections of node u. This is the case where C' is the community that has the maximum external pull for node u. As a result, along with the degree, $E_{\max}(u)$ would also increase by 1. It would not change $C_{\text{in}}(u)$. Therefore, for permanence loss, we need to see whether $\mathcal{P}_l = \text{Perm}(u, G) - \text{Perm}(u, G') \geq 0$. This reduces to

$$\mathcal{P}_l = I(u)\left[\frac{1}{E_{\max}(u)\times\deg(u)} - \frac{1}{(E_{\max}(u)+1)\times(\deg(u)+1)}\right]$$
$$0 = I(u)\left[\frac{1 + E_{\max}(u)+\deg(u)}{E_{\max}(u)\times\deg(u)\times(E_{\max}(u)+1)\times(\deg(u)+1)}\right]$$
$$> 0. \quad (6)$$

Therefore, there is permanence loss in the case of adding an intercommunity edge such that $E_{\max}(u)$ changes after edge addition. ■

Theorem 4.4: The permanence loss is more in case of (6) (i.e., an edge added to the neighboring community from where u experiences the maximum external pull) compared to (5).

Proof: Taking permanence loss in (5) and (6), we get

$$I(u)\left[\frac{1 + E_{\max}(u)+\deg(u)}{E_{\max}(u)\times\deg(u)\times(E_{\max}(u)+1)\times(\deg(u)+1)}\right]$$
$$\geq \frac{I(u)}{E_{\max}(u)} \times \left[\frac{1}{\deg(u)\times(\deg(u)+1)}\right]$$
$$\Rightarrow \deg(u) \geq 0 \text{ which is true.}$$

■

4) Intracommunity Edge Addition:

Theorem 4.5: Adding an intracommunity edge $\langle u, v\rangle$ where $u, v \in C$ does not always ensure a loss in permanence.

Proof: Here, we show that adding an intracommunity edge does not always result in permanence loss. We will only show the change in permanence for node u (the same applies to v).

For this, we consider two parts of permanence separately: 1) ratio of internal–external pull, denoted by $\text{Perm}(G)_1$, and 2) cohesiveness of internal neighbors, denoted by $\text{Perm}(G)_2$.

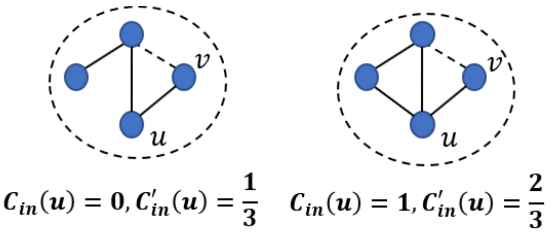
v
u
v
u
$C_{in}(u) = 0, C'_{in}(u) = \frac{1}{3}$
$C_{in}(u) = 1, C'_{in}(u) = \frac{2}{3}$

Fig. 3. Example to demonstrate that permanence loss in terms of cohesiveness of internal neighbors of a node u may not always be positive.

1) *Impact on the Ratio of Internal–External Pull:* In this, we consider the effect of adding an intracommunity edge $\langle u, v\rangle$ on the internal–external pull factor of permanence. This update increases the internal degree and degree for node u by 1. It has no effect on the maximum external connections $E_{\max}(u)$. Therefore, for permanence loss, we need to see whether $\mathcal{P}_{l1} = \text{Perm}(u, G)_1 - \text{Perm}(u, G')_1 \geq 0$. This reduces to

$$\mathcal{P}_{l1} = \frac{1}{E_{\max}(u)}\left[\frac{I(u)}{\deg(u)} - \frac{I(u)+1}{\deg(u)+1}\right]$$
$$= \frac{1}{E_{\max}(u)}\left[\frac{I(u)-\deg(u)}{\deg(u)\times(\deg(u)+1)}\right]$$
$$\leq 0 \text{ (as } I(u) \leq \deg(u)).$$

Therefore, there is no permanence loss with respect to the internal–external pull (first part of (2)).

2) *Impact on Cohesiveness of Internal Neighbors:* In this, we consider the effect of adding an intracommunity edge $\langle u, v\rangle$ on the cohesiveness of internal neighbors. Therefore, for permanence loss, we need to see whether $\mathcal{P}_{l2} = \text{Perm}(u, G)_2 - \text{Perm}(u, G')_2 \geq 0$. This reduces to

$$\mathcal{P}_{l2} = (1 - C'_{\text{in}}(u)) - (1 - C_{\text{in}}(u)) = C_{\text{in}}(u) - C'_{\text{in}}(u)$$

where $C'_{\text{in}}(u)$ represents the updated internal clustering coefficient of u in G and $\mathcal{P}_{l2}$ can be positive or negative depending on how the connections between internal neighbors of u change after introducing its new neighbor v. This is shown using a toy example in Fig. 3. It can be seen that in the first case, $\mathcal{P}_{l2} < 0$, whereas in the second case, $\mathcal{P}_{l2} > 0$.

By combining 1) and 2), we conclude that intracommunity edge addition does not always ensure permanence loss. ■

D. Proposed Algorithm: NEURAL

Since our objective function is submodular and monotone with respect to the possible edge updates that affect permanence loss positively, we propose NEURAL, a greedy algorithm that maximizes permanence loss to rewire nodes within a given budget in order to hide the target community.

NEURAL makes the use of certain edge updates discussed in Section IV-C to rewire the network structure such that the CDAs are not able to detect a target community C. Along with the network, it takes as input β, indicating the budget or the maximum number of edge updates that are allowed. The pseudocode of NEURAL is shown in Algorithm 1 (flow diagram in Fig. 1). At every iteration, it considers an edge update, which contributes toward the maximum loss in permanence for the network, hence greedily updating the original network. For an edge addition, we only consider adding intercommunity edges following Theorems 4.3 and 4.5 as it has been shown that adding an intracommunity edge does not guarantee a loss in permanence in all cases (lines 4–6 of Algorithm 1). In the case of edge deletion, we only consider deleting intracommunity edges following Theorems 4.1 and 4.2 (lines 7–9 of Algorithm 1). Deleting an intercommunity edge does not result in permanence loss in any case; hence, it is not a favorable update. Since NEURAL follows a greedy strategy, for the addition of all the competing intercommunity edges, the one which has the highest permanence loss for the network is considered. The same approach is followed for selecting the best intracommunity edge for deletion. In the end, a choice between the best intercommunity edge to be added and the best intracommunity edge to be deleted is made based on which one contributes more to network permanence loss (lines 10–13 of Algorithm 1).

Note that for computing the best network update at every iteration, we only need node information for a subset of all the nodes present in the network, which reduces the amount of network information being used.

E. Time Complexity of NEURAL

The time complexity of NEURAL is $\mathcal{O}(|V_C| + |E_C|)$, where $|V_C|$ and $|E_C|$ represent the number of nodes and edges (both intracommunity and intercommunity) in the target community C, respectively. This is because, in order to search for edge updates that best contribute toward the permanence loss for hiding C, we only need to go through the nodes and edge connections in the target community, as shown in Section IV-C. Information about the rest of the network is not required. We explore the running time complexity of NEURAL further in the Supplementary Material.

Algorithm 1 NEURAL: Network Deception Using Permanence Loss

```
Input: (i) Network $G$, (ii) target community $C$, (iii) budget $\beta$
Output: Updated Network $G'$
1: $\mathcal{P}_{\mathrm{l,add}} = 0$
2: $\mathcal{P}_{\mathrm{l,del}} = 0$
3: while $\beta > 0$ do
4:     $\mathrm{add_u}, \mathrm{maxComm_u} = \mathrm{getBestNodeForAddition(C)}$ (6)
5:     $\mathrm{add_v} = \mathrm{getBestExternalNodeForAddition(maxComm_u)}$
6:     $\mathcal{P}_{\mathrm{l,add}} = \mathrm{getEdgeAdditionLoss(add_u, add_v)}$
7:     $\mathrm{intraEdge} \leftarrow \mathrm{getConnectingEdges(C)}$
8:     $\mathrm{del_u}, \mathrm{del_v} = \mathrm{getBestEdgeForDeletion(intraEdge)}$ (4)
9:     $\mathcal{P}_{\mathrm{l,del}} = \mathrm{getEdgeDeletionLoss(del_u, del_v, C)}$
10:    if $\mathcal{P}_{\mathrm{l,add}} \geq \mathcal{P}_{\mathrm{l,del}}$ and $\mathcal{P}_{\mathrm{l,add}} > 0$ then
11:        $\mathrm{G} \leftarrow (\mathrm{V}, \mathrm{E} \cup \{\mathrm{add_u}, \mathrm{add_v}\})$
12:    else if $\mathcal{P}_{\mathrm{l,del}} > 0$ then
13:        $\mathrm{G} \leftarrow (\mathrm{V}, \mathrm{E} \backslash \{\mathrm{del_u}, \mathrm{del_v}\})$
14:    end if
15:    $\beta = \beta - 1$
16: end while
17: return G
```

V. Experimental Setup

In this section, we start by briefly describing the data sets, baseline methods, community detection methods we considered for deception, and the evaluation metrics. We then elaborate on the experimental results and the case studies.

A. Synthetic and Real-World Networks

We conduct experiments on two types of networks as follows.

1) *Synthetic Networks:* We use the LFR benchmark [26] and vary the following parameters to generate synthetic networks: N is the number of nodes and μ is the ratio of external connections of a node to degree. The other parameters are set to default as mentioned in the original implementation. Unless otherwise stated, we consider the following setting to generate the default synthetic network: $N = 10\,000$ and $\mu = 0.4$ (as suggested in [8]).
2) *Real-World Networks:* We use seven real-world networks: 1) Zachary's Karate Club (Kar)[4]; 2) Dolphin social network (Dol)[4]; 3) Les Miserables (Lesmis)[4]; 4) Books about U.S. Politics (Polbook)[4]; 5) Word adjacencies (Adjn)[4]; (6) U.S. power grid (power)[4]; and 7) DBLP collaboration network (Dblp).[5] Table II summarizes the statistics of the networks.

Note that we do not require the ground-truth community structure since our primary aim is to deceive a CDA so that after rewiring, the community affiliation of target nodes remains unrevealed.

B. Baseline Methods

We compare NEURAL with four baseline methods, which are given in the following.

1) *Random Algorithm:* It updates the network by randomly selecting the type of edge update (edge addition/deletion), along with the end nodes.
2) *Nagaraja Algorithm [6]:* It updates the network by adding edges between nodes selected on the basis of vertex-centrality measures.
3) *DICE [2]:* It updates the network by randomly adding intercommunity edges or deleting intracommunity edges.
4) *SADDEN [5]:* It updates the network by maximizing the *safeness gain* in every iteration of edge update based on greedy optimization.

C. Community Detection Algorithms

We consider six diverse and widely used CDAs: Louvain (Louv) [27], WalkTrap (Walk) [14], Greedy [28], InfoMap (Info) [15], Label Propagation (Labprop) [29], and Leading Eigenvectors (Eig) [30]. Note that none of these algorithms use permanence as a metric for optimization. Therefore, NEURAL is agnostic to the underlying mechanism of these algorithms.

D. Evaluation Metrics

Here, we briefly describe the metrics used to evaluate the community deception methods. $\uparrow$ (resp. $\downarrow$) indicates higher (*resp.* lower) the value of the metric, better the performance.

1) *Normalized Mutual Information (NMI) $\downarrow$ [31]:* To check how much the deception methods are able to hide a

[4]http://www-personal.umich.edu/∼ mejn/netdata/
[5]http://snap.stanford.edu/data/

particular target community C in the network, we calculate the NMI score between the original community structure of the network, $CS = (C_1, C_2, \ldots, C_k)$, and the new community structure obtained from a CDA on the updated network, $CS' = (C'_1, C'_2, \ldots, C'_{k'})$. The metric ranges from 0 (suggesting no overlap between CS and CS') to 1 (suggesting a complete overlap between CS and CS').

2) *Modified NMI (MNMI)* $\downarrow$*:* For large networks, hiding a target community C may not have a major effect on the *other communities* that are not in immediate contact with C. As a result, to capture how effective a deception method is in hiding C, we may need to measure NMI between the community memberships of nodes in the target communities and their immediate neighbors before and after the edge updates. We call this metric MNMI. Its range is the same as that of NMI.

3) *Community Splits (CommS)* $\uparrow$*:* We propose this metric to define the number of communities in CS' containing the nodes of the target community C in the updated network G'. It ranges from 1 (all nodes in C remain in one community in CS') to $|CS'|$ (all nodes in C get distributed into different communities of CS'). The higher the value of CommS, the wider would be the split of the nodes in C, thereby increasing the deception of the target community

$$\text{CommS} = \sum_{C'_i \in CS'} h(C'_i, C);\ h(C'_i, C) = \begin{cases} 1 & V_C \cap V_{C'_i} \neq \phi \\ 0 & V_C \cap V_{C'_i} = \phi \end{cases}$$

where V_C represents the set of nodes belonging to C and V_{C_i} represents set of nodes belonging to community $C_i \in CS'$.

4) *Community Uniformity (CommU)* $\uparrow$*:* We propose this metric to capture how nodes in the target community C get distributed among communities in the new community structure CS'. It is obtained by calculating the entropy of target community's nodes present among the communities in CS' as follows: $\text{CommU} = \sum_{C'_i \in CS'} -(|V_{C,C'_i}|/|V_C|)\log(|V_{C,C'_i}|/|V_C|)$, where $|V_{C,C'_i}|$ represents the number of nodes in C present in $C'_i \in CS'$ and $|V_C|$ represents the total number of nodes present in C. It ranges from 0 (when all nodes of C remain in one community of CS') to $\log|\mathcal{CS}'|$ (when all nodes of C get distributed into different communities of CS').

VI. Quantitative Evaluation

Here, we present the quantitative analysis of experimental results on both synthetic and real-world networks.

A. Evaluation on Synthetic Networks

We use the default LFR network, set the budget β as the fraction of nodes in the target community C, and vary the fraction from 0.1 to 0.6. The result is averaged over 20 synthetic networks, five randomly selected target communities, and ten runs for each target community. Fig. 4(a)–(c) shows that with an increase of β, NEURAL is able to hide C better (NMI, MNMI scores decrease, and CommS scores increase) showing a parallel between the allowed budget and its effect on community deception.

TABLE III
Comparison on the Default LFR Network, Keeping $\beta = 0.3|V_C|$, Where V_C is the Size of the Target Community

Method	NMI	MNMI	CommS	CommU
Random	0.99	0.97	1.13	0.15
Nagaraja	0.99	0.28	1.21	0.86
DICE	**0.98**	0.90	1.33	0.81
SADDEN	**0.98**	**0.26**	3.64	0.73
NEURAL	**0.98**	**0.26**	**3.80**	**0.94**

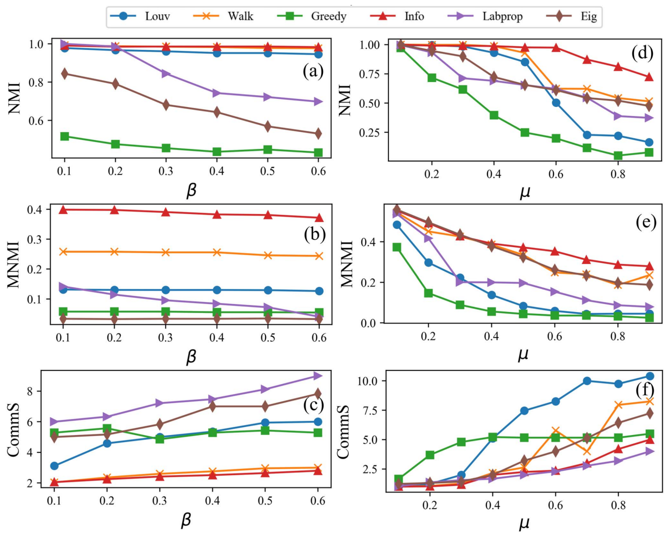

Fig. 4. NMI, MNMI, and CommS on the default synthetic network by varying (a)–(c) β and (d)–(f) μ, keeping $\beta = 0.3|V_c|$, where $|V_C|$ is the number of nodes in the target community.

We further conduct experiments by varying the parameter μ of LFR network from 0.1 to 0.9. Fig. 4(d)–(f) shows that with an increase in μ, the nodes in C are concealed more by NEURAL (NMI, MNMI scores decrease, and CommS scores increase). The above observation matches the expectation that it would be easier to hide a target community, which has more sparse intracommunity connections than the intercommunity connections.

Table III shows that NEURAL delivers comparable (and sometimes better) accuracy on the default synthetic network.

B. Evaluation on Real-World Networks

In case of experiments on real-world networks, we fix β to 30% of the size of the target community C (i.e., $\beta = 0.3|V_C|$). The results reported here are obtained by averaging the performance considering each of the communities as target community at a time and over ten runs for each target community.

For a compact visualization, we rank five competing community deception methods as follows; for each evaluation metric and each CDA, we normalize their scores (using min–max normalization) so that the best performing method gets score 1. Now, if a competing method outperforms others by deceiving all the six CDAs with respect to that evaluation metric, it will secure a composite score of 6.

Fig. 5(a)–(d) shows the composite performance across all the evaluation metrics. Fig. 5(e) shows the composite performance of individual competing methods averaged over all the

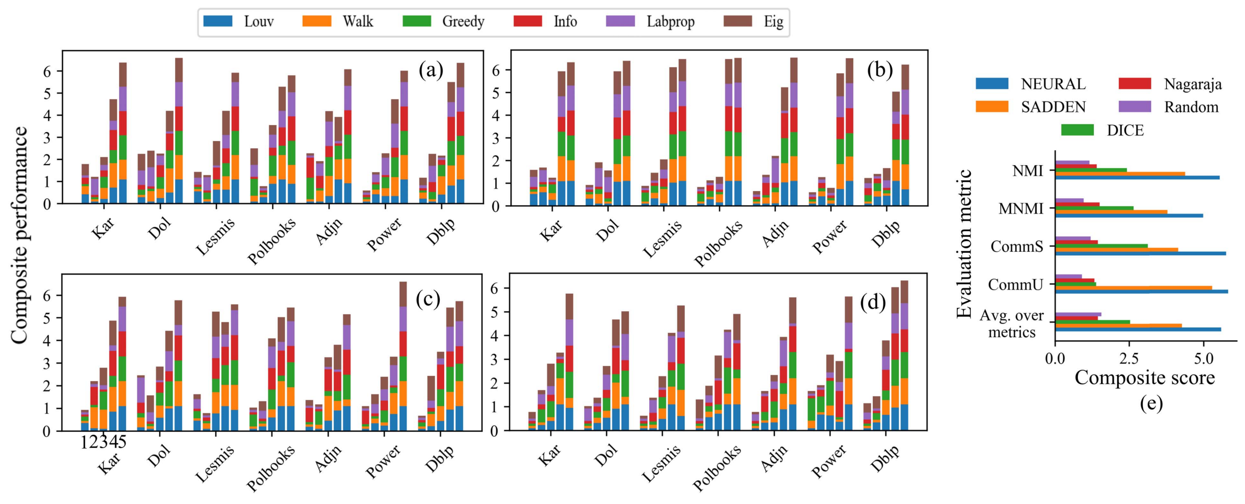

Fig. 5. (Color online) Composite performance of the five competing community deception methods based on (a) NMI, (b) MNMI, (c) CommS, and (d) CommU. Bars in each group (under each data set) are ordered as follows: 1) Random; 2) Nagaraja; 3) DICE; 4) SADDEN; and 5) NEURAL [as shown in (c)]. (e) Composite performance of each competing method based on every evaluation metric averaged over all the data sets.

data sets. We observe that NEURAL outperforms others with a significant margin—NEURAL achieves a composite score of 5.53 (averaged over all the evaluation metrics and data sets), outperforming Random, Nagaraja, DICE, and SADDEN by 376.72%, 286.71%, 128.51%, and 26.54%, respectively (see the Supplementary Material for the raw accuracy scores over all the data sets). Note that SADDEN turns out to be highly competitive, sometimes showing marginal improvement over NEURAL. However, in general, NEURAL is better than others [see Fig. 5(e)]. We also consider hiding individual target nodes instead of communities (refer to the Supplementary Material for the same).

C. Nonuniform Budget for Edge Updates

Until now, we have reported the results with a unified budget β for all the edge update operations. In this section, we extend NEURAL with nonuniform budget in which separate budget constraints are applied to the two types of allowed edge updates (as elaborated in Section IV-D): 1) β_D for intracommunity edge deletion and 2) β_A for intercommunity edge addition. Such an analysis could be useful in situations in which the costs incurred while deleting an intracommunity edge and adding an intercommunity edge are different. We perform experiments under two different settings of β_D and β_A: 1) $\beta_D = 0.3\beta$ and $\beta_A = 0.7\beta$ and 2) $\beta_D = 0.3\beta$ and $\beta_A = 0.7\beta$ (we fix β as default, i.e., 30% of the size of the target community C). Tables IV and V show raw accuracy values (by averaging over all the communities) for NEURAL and SADDEN (the best baseline, extending it in a similar manner) on Karate real-world network (see the Supplementary Material for others) for the two settings mentioned earlier. We observe that NEURAL outperforms SADDEN in most cases.

TABLE IV
ACCURACY OF TWO COMPETING COMMUNITY DECEPTION METHODS: 1) S: SADDEN (BEST BASELINE) AND 2) N: NEURAL OVER KARATE, SUCH THAT $\beta_D = 0.3\beta$ AND $\beta_A = 0.7\beta$

Comm. Det.	NMI		MNMI		CommS		CommU	
Algo.	S	N	S	N	S	N	S	N
Louv	0.94	**0.75**	0.50	**0.30**	1.50	**2.50**	0.27	**0.64**
Walk	0.78	**0.74**	0.37	**0.30**	1.40	**2.20**	0.57	**0.96**
Greedy	0.77	**0.64**	0.35	**0.31**	2.00	**2.67**	0.61	**0.74**
Info	0.83	**0.69**	0.52	**0.45**	**1.33**	1.00	1.75	**2.84**
Labprop	0.84	**0.00**	0.09	**0.00**	1.50	**4.00**	**1.56**	0.40
Eig	0.95	**0.82**	0.41	**0.32**	**1.50**	**1.50**	1.53	**1.64**

TABLE V
ACCURACY OF TWO COMPETING COMMUNITY DECEPTION METHODS: 1) S: SADDEN (BEST BASELINE) AND 2) N: NEURAL OVER KARATE, SUCH THAT $\beta_D = 0.7\beta$ AND $\beta_A = 0.3\beta$

Comm. Det.	NMI		MNMI		CommS		CommU	
Algo.	S	N	S	N	S	N	S	N
Louv	0.94	**0.84**	0.54	**0.50**	1.50	**2.00**	0.79	**0.87**
Walk	0.79	**0.78**	0.49	**0.44**	1.40	**2.00**	0.39	**0.86**
Greedy	0.81	**0.66**	0.34	**0.31**	2.10	**2.67**	0.66	**0.74**
Info	**0.80**	0.93	0.44	**0.42**	**2.00**	1.00	0.85	**2.39**
Labprop	0.68	**0.65**	0.43	**0.13**	1.33	**1.67**	**1.76**	1.45
Eig	0.92	**0.89**	0.45	**0.33**	1.25	**1.75**	0.95	**2.33**

VII. Qualitative Evaluation

To interpret the rewiring suggested by NEURAL and SADDEN (top two methods), we further take three real-world attributed networks. Unless otherwise stated, we only consider deletion of edges (as addition of a new edge does not make any sense for these networks). The Louvain algorithm is used for community detection, and the largest community is considered as the target community.

A. Citation Network

We consider 6320 papers published in *Physical Review* Journal as nodes and 10 000 citation interactions (we ignore directionality) among them as edges.[6] After hiding the target community (largest community), we observe that NEURAL tends to pick up those citation interactions (or edges) whose *age* (defined by the difference between the publication years of citing and cited papers) is relatively high (we believe that these edges have much more important in terms of keeping

[6]https://journals.aps.org/datasets

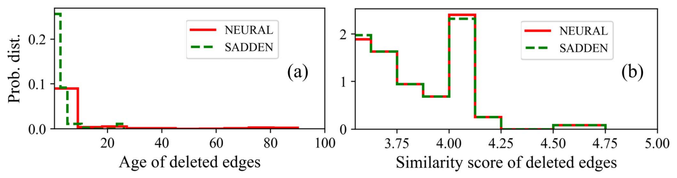

Fig. 6. (a) Age and (b) similarity score distribution of edges selected by NEURAL and SADDEN from citation and terrorist networks, respectively.

the identity of the target community intact, being connected to papers (or nodes) published earlier than most in the literature). NEURAL performs better than SADDEN in terms of updating more edges of such kind [see Fig. 6(a)].

We further measure the correlation (Spearman's ρ and Kendall's τ) of 138 edges selected and ranked by NEURAL and those ranked by their age (ground truth) (similar correlation for 138 edges returned by SADDEN). Table VI shows that NEURAL outperforms SADDEN. Moreover, NEURAL returns the top three edges based on their age present in the target community within top 20 of the rank list, whereas SADDEN is unable to return a single such edge within the 138 edges returned.

B. Terrorist Network

We use the Global Terrorism Database[7] to create a network of terrorist group associations. This data set consists of 191 465 terrorist events around the world between 1970 and 2018. In order to create the network, we compute the similarity between two terrorist groups based on their activities. To quantify the similarity between two groups, we use five attributes: 1) severity of the attack (number of casualties); 2) attacking strategy used in majority events; (3) type of weapon used in majority events; 4) peak year of attacks; and 5) the target type in majority events. Thus, two terrorist groups are associated with a link if the similarity score is greater than or equal to 2.5 (out of 5). This gives rise to a network having 3616 nodes as terrorist groups and 22 141 unweighted edges as association links among these groups.

After hiding the target community (largest community), we observe that NEURAL first picks up those edges that have higher similarity scores, removing a link between two highly similar terrorist groups. NEURAL performs better compared to SADDEN in terms of providing more edges with high similarity scores [see Fig. 6(b)]. We further measure the rank correlation between 93 edges returned and ranked by NEURAL with those ranked by their similarity scores (ground truth) (similarly for 93 edges returned by SADDEN). Table VI shows that NEURAL once again outperforms SADDEN in terms of returning edges whose similarity score is high.

C. Breast Cancer Network

Breast cancer[8] is considered a leading cause of morbidity and mortality among women worldwide. Above 12% of the women in the United States are diagnosed with breast cancer during their lifetime [32]. Alteration of gene regulation has been widely studied in this context [33], with a special focus on dynamic changes in gene co-expression modules. Under the Cancer Genome Atlas (TCGA) program, a community-scale effort has been directed toward multiomic molecular profiling of breast tumors in hundreds of patients [34]. We use Fragments Per Kilobase of transcript per Million mapped reads (FPKM) normalized gene expression data from TCGA to understand whether disguising community affiliation plays a role in the pathogenesis of critical diseases such as cancers. To achieve this, we construct a control and a cancer-specific co-expression network based on transcriptomic profiles of 1097 normal (as controls) and 113 tumor samples obtained from the TCGA repository. Both networks spanned the same set of 1000 genes (1000 nodes). Two nodes are connected by an edge when Pearson's correlation coefficient computed across the entire spectrum of control/tumor samples qualifies a cutoff value of 0.6 (12 161 edges). Deleterious mutations in cancer cause widespread loss-of-function events, which are often manifested by changes in gene expression levels.

We employ NEURAL and SADDEN to retrieve co-expressions (edges), whose disappearance fosters community disintegration. NEURAL and SADDEN could pin-point 15 and 10 rewirings in the form of edge deletion, respectively, which could be cross-validated with respect to the cancer-specific network. Quite strikingly, 5 out of the 15 correctly predicted deletions by NEURAL, harbors BMP2 inducible kinase (BPMP2K) as one of the nodes. We find definitive studies implicating this molecule in breast cancer [35]. We fail to find any literature support for the novel gene Z97832.2 that was relatively enriched (three out of ten rewirings) among SADDEN predicted rewirings. We also measure how accurate NEURAL and SADDEN are to predict the ground-truth edges with respect to the cancer-specific network. Table VI shows that NEURAL outperforms SADDEN on four evaluation measures. To this end, we conclude that NEURAL-led investigation of genome-scale molecular networks holds significant promise in understanding genetic diseases such as cancers.

TABLE VI

RANK CORRELATION FOR CITATION AND TERRORIST NETWORKS, AND ACCURACY FOR BREAST CANCER NETWORK

Method	Citation		Terrorist	
	Spearman's ρ		Kendall's τ	
SADDEN	0.12	0.21	0.00	0.07
NEURAL	**0.16**	**0.41**	**0.06**	**0.29**
	Breast cancer			
	MAP	F1 score	nDCG	AUC
SADDEN	0.004	0.20	0.47	0.30
NEURAL	**0.006**	**0.29**	**0.54**	**0.39**

Rank correlations are statistically significant with p-value> 0.8

VIII. CONCLUSION

This article addressed the problem of *community deception*—outwitting CDAs from discovering the community affiliation of nodes in a target community. Our major contributions are as follows: 1) we formalized the problem and called it HSC; 2) we proposed a novel objective function (permanence loss), which has been analyzed theoretically; 3) we proposed NEURAL, a novel greedy strategy to optimize the permanence loss; 4) NEURAL turned out to be more efficient than the baselines; and 5) NEURAL unfolded different metainformation of edges, which would otherwise not have been possible to explain just by analyzing the network structure. In particular, NEURAL showed promise in the analysis of genome-scale molecular networks.

[7]https://www.start.umd.edu/gtd/

[8]This analysis was conducted by two professional biologists.

References

[1] S. Fortunato and D. Hric, "Community detection in networks: A user guide," *Phys. Rep.*, vol. 659, pp. 1–44, Nov. 2016.

[2] M. Waniek, T. P. Michalak, M. J. Wooldridge, and T. Rahwan, "Hiding individuals and communities in a social network," *Nature Hum. Behav.*, vol. 2, no. 2, pp. 139–147, Feb. 2018.

[3] A. Mislove, B. Viswanath, K. P. Gummadi, and P. Druschel, "You are who you know: Inferring user profiles in online social networks," in *Proc. 3rd ACM Int. Conf. Web Search Data Mining (WSDM)*, 2010, pp. 251–260.

[4] T. Ji, C. Luo, Y. Guo, Q. Wang, L. Yu, and P. Li, "Community detection in online social networks: A differentially private and parsimonious approach," *IEEE Trans. Comput. Social Syst.*, vol. 7, no. 1, pp. 151–163, Feb. 2020.

[5] V. Fionda and G. Pirro, "Community deception or: How to stop fearing community detection algorithms," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 4, pp. 660–673, Apr. 2018.

[6] S. Nagaraja, "The impact of unlinkability on adversarial community detection: Effects and countermeasures," in *Proc. PETS*, Jul. 2010, pp. 253–272.

[7] Y. Liu, J. Liu, Z. Zhang, L. Zhu, and A. Li, "REM: From structural entropy to community structure deception," in *Proc. NIPS*, 2019, pp. 12918–12928.

[8] T. Chakraborty, S. Srinivasan, N. Ganguly, A. Mukherjee, and S. Bhowmick, "On the permanence of vertices in network communities," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2014, pp. 1396–1405.

[9] K. Berahmand, A. Bouyer, and M. Vasighi, "Community detection in complex networks by detecting and expanding core nodes through extended local similarity of nodes," *IEEE Trans. Comput. Social Syst.*, vol. 5, no. 4, pp. 1021–1033, Dec. 2018.

[10] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 69, no. 2, 2004, Art. no. 026113.

[11] M. Chen, K. Kuzmin, and B. K. Szymanski, "Community detection via maximization of modularity and its variants," *IEEE Trans. Comput. Social Syst.*, vol. 1, no. 1, pp. 46–65, Mar. 2014.

[12] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney, "Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters," *Internet Math.*, vol. 6, no. 1, pp. 29–123, Jan. 2009, doi: 10.1080/15427951.2009.10129177.

[13] J. Leskovec, K. J. Lang, and M. Mahoney, "Empirical comparison of algorithms for network community detection," in *Proc. 19th Int. Conf. World Wide Web (WWW)*, 2010, pp. 631–640.

[14] P. Pons and M. Latapy, "Computing communities in large networks using random walks," in *Computer and Information Sciences*, P. Yolum, T. Güngör, F. Gürgen, and C. Özturan, Eds. Berlin, Germany: Springer, 2005, pp. 284–293.

[15] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proc. Nat. Acad. Sci. USA*, vol. 105, no. 4, pp. 1118–1123, Jan. 2008.

[16] M. Rosvall and C. T. Bergstrom, "An information-theoretic framework for resolving community structure in complex networks," *Proc. Nat. Acad. Sci. USA*, vol. 104, no. 18, pp. 7327–7331, May 2007.

[17] L. Donetti and M. A. Muñoz, "Detecting network communities: A new systematic and efficient algorithm," *J. Stat. Mech., Theory Exp.*, vol. 2004, no. 10, Oct. 2004, Art. no. P10012.

[18] A. Capocci, V. D. P. Servedio, G. Caldarelli, and F. Colaiori, "Detecting communities in large networks," *Phys. A, Stat. Mech. Appl.*, vol. 352, nos. 2–4, pp. 669–676, Jul. 2005.

[19] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, pp. 814–818, Jun. 2005.

[20] N. Alduaiji, A. Datta, and J. Li, "Influence propagation model for clique-based community detection in social networks," *IEEE Trans. Comput. Social Syst.*, vol. 5, no. 2, pp. 563–575, Jun. 2018.

[21] S. Fortunato, "Community detection in graphs," *Phys. Rep.*, vol. 486, nos. 3–5, pp. 75–174, Feb. 2010.

[22] T. Chakraborty, A. Dalmia, A. Mukherjee, and N. Ganguly, "Metrics for community analysis: A survey," *ACM Compt. Surv.*, vol. 50, no. 4, pp. 1–37, 2017.

[23] T. Chakraborty, S. Srinivasan, N. Ganguly, A. Mukherjee, and S. Bhowmick, "Permanence and community structure in complex networks," *ACM Trans. Knowl. Discovery Data*, vol. 11, no. 2, pp. 1–34, Dec. 2016.

[24] M. E. J. Newman, "Modularity and community structure in networks," *Proc. Nat. Acad. Sci. USA*, vol. 103, no. 23, pp. 8577–8582, Jun. 2006.

[25] S. Muff, F. Rao, and A. Caflisch, "Local modularity measure for network clusterizations," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 72, no. 5, Nov. 2005, Art. no. 056107.

[26] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 78, no. 4, Oct. 2008, Art. no. 046110.

[27] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech., Theory Exp.*, vol. 2008, no. 10, Oct. 2008, Art. no. P10008.

[28] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 70, no. 6, Dec. 2004, Art. no. 066111.

[29] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 76, no. 3, Sep. 2007, Art. no. 036106.

[30] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 74, no. 3, Sep. 2006, Art. no. 036104.

[31] L. Danon, A. Díaz-Guilera, J. Duch, and A. Arenas, "Comparing community structure identification," *J. Stat. Mech., Theory Exp.*, vol. 2005, no. 9, Sep. 2005, Art. no. P09008.

[32] A. G. Waks and E. P. Winer, "Breast cancer treatment: A review," *Jama*, vol. 321, no. 3, pp. 288–300, 2019.

[33] D. Sengupta and S. Bandyopadhyay, "Topological patterns in microrna–gene regulatory network: Studies in colorectal and breast cancer," *Mol. Biosyst.*, vol. 9, no. 6, pp. 1360–1371, 2013.

[34] C. G. A. Network *et al.*, "Comprehensive molecular portraits of human breast tumours," *Nature*, vol. 490, no. 7418, p. 61, 2012.

[35] S. Buraschi *et al.*, "Decorin protein core affects the global gene expression profile of the tumor microenvironment in a triple-negative orthotopic breast carcinoma xenograft model," *PLoS ONE*, vol. 7, no. 9, Sep. 2012, Art. no. e45559.

Shravika Mittal is currently pursuing the bachelor's degree in computer science and engineering with the Indraprastha Institute of Information Technology Delhi, New Delhi, India.

Her research interests include social network analysis, network science, and natural language processing.

Ms. Mittal has received the Dean's list for Excellence in Academics, and Innovation in Research and Development.

Debarka Sengupta received the Ph.D. degree from Jadavpur University, Kolkata, India, in 2013.

Before joining the Indraprastha Institute of Information Technology Delhi, New Delhi, India, he worked as an INSPIRE Faculty at the Indian Statistical Institute, Kolkata. He consulted and advised a number of technology and service-based firms, including IPsoft, Datanomers, CoreCompete, and Applied Research Works on various data science and business analytics projects.

Dr. Sengupta has twice been nominated for the prestigious INSPIRE Faculty Award in 2014 and 2016.

Tanmoy Chakraborty (Member, IEEE) is currently an Assistant Professor and a Ramanujan Fellow with the Department of Computer Science and Engineering, Indraprastha Institute of Information Technology Delhi, New Delhi, India, where he leads a research group, called LCS2. His primary research interests include social network analysis, data mining, and natural language processing.

Prof. Chakraborty has received several awards, including the Google Indian Faculty Award, the Early Career Research Award, and the DAAD Faculty award. More details at http://faculty.iiitd.ac.in/ tanmoy/.