

Hide and Seek: Outwitting Community Detection Algorithms

Shravika Mittal*, Debarka Sengupta*[†], Tanmoy Chakraborty*

*Dept. of CSE, [†]Dept. of Comp. Biology, IIT-Delhi, India
 {shravika16093, debarka, tanmoy}@iitd.ac.in

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 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. We formalize the problem of community deception and introduce 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 7 real-world networks, NEURAL is able to deceive 6 widely used community detection algorithms. We benchmark its performance with respect to 4 state-of-the-art methods on 4 evaluation metrics. Additionally, our qualitative analysis on 3 other attributed real-world networks reveals that NEURAL, quite strikingly, captures important meta-information about edges that otherwise could not be inferred by observing only their topological structures.

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

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 can be deceived by minimal rewiring of nodes.

In this paper, we ask a fundamental question: *How do we hide a target community from being exposed to a community detection algorithm, 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 surveillance¹ [4]. It also helps law-enforcement

¹Mislove 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.

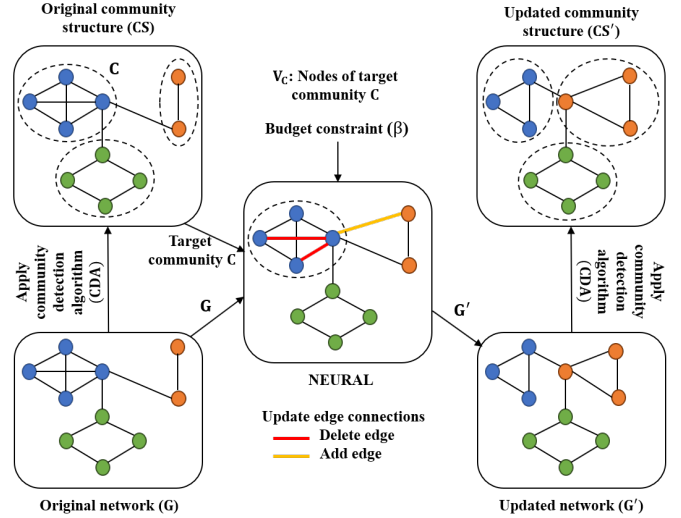


Fig. 1. Flow diagram showing the procedure of NEURAL.

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 of 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 the 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 towards 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 community detection algorithms. 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 (inter-community edge addition and intra-community edge deletion) are useful for rewiring nodes to optimize our proposed objective function. We further show that the objective function is *submodular* and *monotone* w.r.t. the required edge updates. Therefore, we propose **NEURAL** (Network deception using permanence loss), a greedy optimization algorithm to optimize the objective function. Given a network G , its community structure CS obtained from a community detection algorithm 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 7 real-world networks. Six widely used community detection algorithms are considered for deception. We compare NEURAL with 4 state-of-the-art community deception methods. The performance is measured based on 4 evaluation metrics (two of them are proposed by us). Our quantitative analysis shows that NEURAL significantly outperforms others across all the datasets 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 meta-information of edges that otherwise could not be inferred just by observing the topological structure of networks.

In short, our major contributions are four-fold:

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

Reproducibility: The codes and datasets are available at: <https://github.com/mittalshravika/HideAndSeek-NEURAL>.

II. RELATED WORK

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

TABLE I
COMPARISON OF NEURAL WITH EXISTING METHODS.
ABBREVIATIONS: $E+$, $-$: EDGE ADDITION AND DELETION, NC: NODE CENTRALITY, $E \rightarrow C$: 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

clustering, which group nodes together based on a similarity metric [9]. Another class of community detection algorithms revolves around the optimization of metrics that define the quality of a network partition, such as modularity [10], [11], conductance [12], cut-ratio [13], etc. 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 community detection algorithms can be found in [21], [22].

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 community detection algorithms. 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 intra-community edges (*disconnect internal*) and adds inter-community edges (*connect external*), inspired by the functioning of modularity. The authors 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 community detection algorithms. 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, SADDEN) as baselines² 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

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

method (community-based structural entropy) requires entire community information.

How NEURAL is different from others? Table II 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 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 3 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 community detection algorithm on G , we get $CS = (C_1, C_2, \dots, C_k)$ as the community structure. We only consider communities that are non-overlapping. For community $C \in CS$, an intra-community edge $\langle u, v \rangle$ is defined such that $u, v \in C$, and an inter-community 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 (HSC)

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 community detection algorithm that we intend to fool.

To avoid this, we introduce the problem, called *Hide and Seek Community* to camouflage a target community C from a community detection method.

Definition 3.1: (Hide and Seek Community) For a network $G = (V, E)$, the problem of *Hide and Seek Community* (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:

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

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 metric³ 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 w.r.t. 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 - (i) the internal pull $I(v)$, denoted by the internal connections of a node v within its own community, (ii) maximum external pull $E_{\text{max}}(v)$, denoted by the maximum connections of v to its neighboring communities, and (iii) 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,

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

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 $\text{Perm}(G) = \frac{\sum_{v \in V} \text{Perm}(v)}{|V|}$. The reasons behind choosing Permanence instead of other community scoring metrics such as (local) modularity [24], [25], conductance, cut-ratio [1] are two-fold: (i) 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 (ii) Permanence has been shown to be superior to other local and global scoring metrics for community detection [22].

B. Proposed Objective Function: Permanence Loss

Our proposed community deception method NEURAL (discussed in Section IV-D) aims to reduce Permanence of the network for a target community C to be hidden from community detection algorithms. 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

³Permanence can be also computed for an entire network.

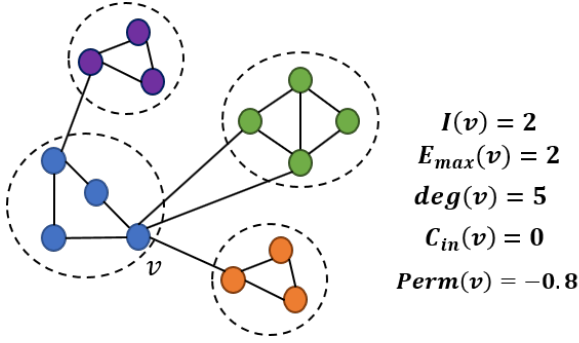


Fig. 2. A toy example demonstrating the calculation of Permanence for a node, given the network and the community structure.

(addition/deletion of edges) by maximizing the **Permanence loss** at every iteration, defined as,

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

where G represents the original network, and G' represents the modified network after updating edges (see Fig. 1) w.r.t. the target community C , elaborated in Sections IV-C and IV-D.

In Section IV-C, we will show that Permanence loss will be affected (positively) only due to the (i) intra-community edge deletion, and (ii) inter-community edge addition. Readers are encouraged to see supplementary where we show that **Permanence loss is submodular and monotone w.r.t. each of the edge updates stated above.**

C. Edge Updates

In this section, we describe four possible edge update operations to maximise Permanence loss \mathcal{P}_l in NEURAL — inter- and intra-community edge deletion, and inter- and intra-community edge addition.

1) Inter-community Edge Deletion:

Theorem 4.1: Deleting an inter-community 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 inter-community edge does not amount to Permanence loss. An inter-community 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:

(i) **$E_{max}(u)$ 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 $C_{in}(u)$. It would only decrease 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,

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

Therefore, no Permanence loss is possible in this case.

(ii) **$E_{max}(u)$ 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 $C_{in}(u)$. Therefore, for Permanence loss, we need to see whether $\mathcal{P}_l = Perm(u, G) - Perm(u, G') \geq 0$. This reduces to,

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

Therefore, no Permanence loss is possible in the case of deleting an inter-community edge. ■

2) Intra-community Edge Deletion:

Theorem 4.2: Deleting an intra-community edge $\langle u, v \rangle$ where $u, v \in C$, always results in Permanence loss.

Proof: Here we show that deleting an intra-community edge always results in Permanence loss. We will only show the change in Permanence for node u (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, $C_{in}(u)$ decreases after the deletion of $\langle u, v \rangle$. Therefore, for Permanence loss, we need to see whether $\mathcal{P}_l = Perm(u, G) - Perm(u, G') \geq 0$. This reduces to,

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

Therefore, deleting an intra-community edge $\langle u, v \rangle$ would bring in Permanence loss in terms of nodes u and v .

The intra-community 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 $\mathcal{P}_l = Perm(w, G) - Perm(w, G') \geq 0$. This reduces to,

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

where $C'_{in}(w)$ represents the updated internal clustering coefficient of w . In the above equation, $\mathcal{P}_l > 0$ since $C_{in}(w) > C'_{in}(w)$. 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) Inter-community Edge Addition:

Theorem 4.3: Adding an inter-community 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 inter-community edge always causes Permanence loss. An inter-community 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:

(i) $\mathbf{E_{max}(u)}$ **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 $C_{in}(u)$. It would only increase its degree by 1. So, 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 inter-community edge such that $E_{max}(u)$ does not change after edge addition.

(ii) $\mathbf{E_{max}(u)}$ **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 $C_{in}(u)$. Therefore, for Permanence loss, we need to see whether $\mathcal{P}_l = Perm(u, G) - Perm(u, G') \geq 0$. This reduces to,

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

Therefore, there is Permanence loss in the case of adding an inter-community 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) as compared to (5).

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

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

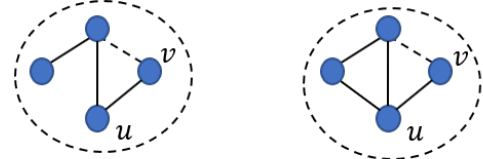
4) Intra-community Edge Addition:

Theorem 4.5: Adding an intra-community 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 intra-community edge does not always result in Permanence loss. We will only show the change in Permanence for node u (same applies to v).

For this, we consider two parts of Permanence separately - (i) ratio of internal-external pull, denoted by $Perm(G)_1$, and (ii) cohesiveness of internal neighbors, denoted by $Perm(G)_2$.

(i) **Impact on the ratio of internal-external pull:** In this, we



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

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

consider the effect of adding an intra-community 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 $\mathcal{P}_{l1} = Perm(u, G)_1 - Perm(u, G')_1 \geq 0$. This reduces to,

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

Therefore, there is no Permanence loss w.r.t. the internal-external pull (first part of (2)).

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

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

where $C'_{in}(u)$ represents the updated internal clustering coefficient of u in G . \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 1st case, $\mathcal{P}_{l2} < 0$, while in the second case, $\mathcal{P}_{l2} > 0$.

By combining (i) and (ii), we conclude that intra-community edge addition does not always ensure Permanence loss. ■

D. Proposed Algorithm: NEURAL

Since our objective function is submodular and monotone w.r.t. 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 the previous section to rewire the network structure such that the community detection algorithms 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 pseudo-code of NEURAL is shown in Algorithm 1 (flow diagram in Fig. 1). At every iteration, it considers an edge update which contributes towards the maximum loss in Permanence for the network, hence greedily updating the original network. For an edge addition, we only consider adding inter-community edges following Theorems 4.3 and 4.5 as it has been shown

that adding an intra-community 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 intra-community edges following Theorems 4.1 and 4.2 (lines 7-9 of Algorithm 1). Deleting an inter-community 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 inter-community edges, the one which has the highest Permanence loss for the network is considered. The same approach is followed for selecting the best intra-community edge for deletion. In the end, a choice between the best inter-community edge to be added and the best intra-community 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 intra-community and inter-community) in the target community C , respectively. This is because, in order to search for edge updates that best contribute towards 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 supplementary.

Algorithm 1 NEURAL: Network Deception using Permanence Loss

Input: (i) Network G , (ii) target community C , (iii) budget β

Output: Updated Network G'

```

1:  $\mathcal{P}_{1,\text{add}} = 0$ 
2:  $\mathcal{P}_{1,\text{del}} = 0$ 
3: while  $\beta > 0$  do
4:    $\text{add}_u, \text{maxComm}_u = \text{getNodeForAddition}(C)$  (6)
5:    $\text{add}_v = \text{getBestExternalNodeForAddition}(\text{maxComm}_u)$ 
6:    $\mathcal{P}_{1,\text{add}} = \text{getEdgeAdditionLoss}(\text{add}_u, \text{add}_v)$ 
7:    $\text{intraEdge} \leftarrow \text{getConnectingEdges}(C)$ 
8:    $\text{del}_u, \text{del}_v = \text{getBestEdgeForDeletion}(\text{intraEdge})$  (4)
9:    $\mathcal{P}_{1,\text{del}} = \text{getEdgeDeletionLoss}(\text{del}_u, \text{del}_v, C)$ 
10:  if  $\mathcal{P}_{1,\text{add}} \geq \mathcal{P}_{1,\text{del}}$  and  $\mathcal{P}_{1,\text{add}} > 0$  then
11:     $G \leftarrow (V, E \cup \{\text{add}_u, \text{add}_v\})$ 
12:  else if  $\mathcal{P}_{1,\text{del}} > 0$  then
13:     $G \leftarrow (V, E \setminus \{\text{del}_u, \text{del}_v\})$ 
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 datasets, baseline methods, community detection methods we considered for deception, and the evaluation metrics. We then elaborate on the experimental results and the case studies.

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

A. Synthetic and Real-world Networks

We conduct experiments on two types of networks:

- (i) **Synthetic networks:** We use LFR Benchmark [26] and vary the following parameters to generate synthetic networks: N , number of nodes and μ , 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: $N = 10,000$, $\mu = 0.4$ (as suggested in [8]).
- (ii) **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 US Politics (Polbook)⁴, (5) Word adjacencies (Adjn)⁴, (6) US Power Grid (Power)⁴ and (7) DBLP collaboration network (Dblp)⁵. Table II summarises the statistics of the networks.

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

B. Baseline Methods

We compare NEURAL with four baseline methods:

- 1) **Random algorithm** updates the network by randomly selecting the type of edge update (edge addition/deletion), along with the end nodes.
- 2) **Nagaraja** algorithm [6] updates the network by adding edges between nodes selected on the basis of vertex-centrality measures.
- 3) **DICE** [2] updates the network by randomly adding inter-community edges or deleting intra-community edges.
- 4) **SADDEN** [5] 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 community detection algorithms: 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.

⁴<http://www-personal.umich.edu/~mejn/netdata/>

⁵<http://snap.stanford.edu/data/>

TABLE III
COMPARISON ON THE DEFAULT LFR NETWORK, KEEPING $\beta = 0.3|V_C|$,
WHERE V_C IS THE SIZE 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

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.

(i) Normalized Mutual Information (NMI) \downarrow [31]: To check how much the deception methods are able to hide a particular target community C in the network, we calculate the NMI score between the original community structure of the network, $CS = (C_1, C_2, \dots, C_k)$ and the new community structure obtained from a community detection algorithm on the updated network, $CS' = (C'_1, C'_2, \dots, C'_k)$. The metric ranges from 0 (suggesting no overlap between CS and CS') to 1 (suggesting a complete overlap between CS and CS').

(ii) Modified Normalized Mutual Information (MNMI) \downarrow : For large networks, hiding a target community C may not have a major effect on the *other communities* which 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 same as that of NMI.

(iii) 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 $C'_i \in CS'$.

(iv) 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: $\text{CommU} = \sum_{C'_i \in CS'} -\frac{|V_{C, C'_i}|}{|V_C|} \log \frac{|V_{C, C'_i}|}{|V_C|}$, 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 $\log |CS'|$ (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.

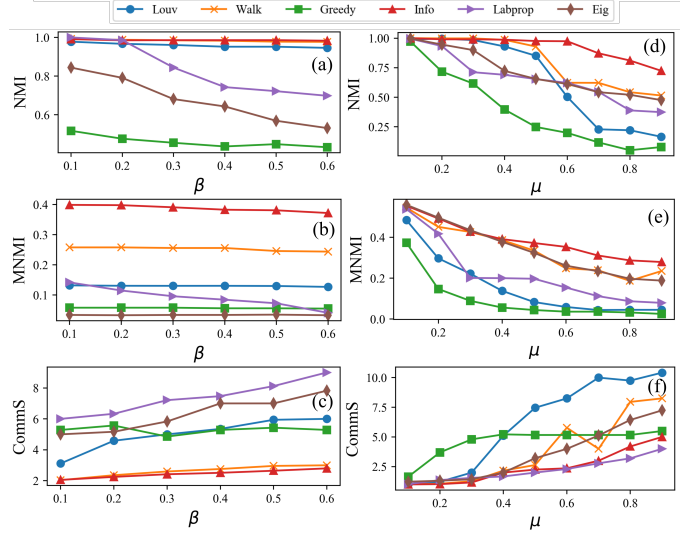


Fig. 4. NMI, MNMI and CommS on the default synthetic network by varying β in (a)-(c) and μ in (d)-(f), keeping $\beta = 0.3|V_C|$, where $|V_C|$ is the number of nodes in the target community.

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, 5 randomly selected target communities and 10 runs for each target community. Figs. 4(a)-(c) show 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.

We further conduct experiments by varying the parameter μ of LFR network from 0.1 to 0.9. Figs. 4(d)-(f) show 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 intra-community connections than the inter-community 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 10 runs for each target community.

For a compact visualization, we rank five competing community deception methods as follows: for each evaluation metric and each community detection algorithm, 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 community detection algorithms w.r.t. that evaluation metric, it will secure a composite score of 6.

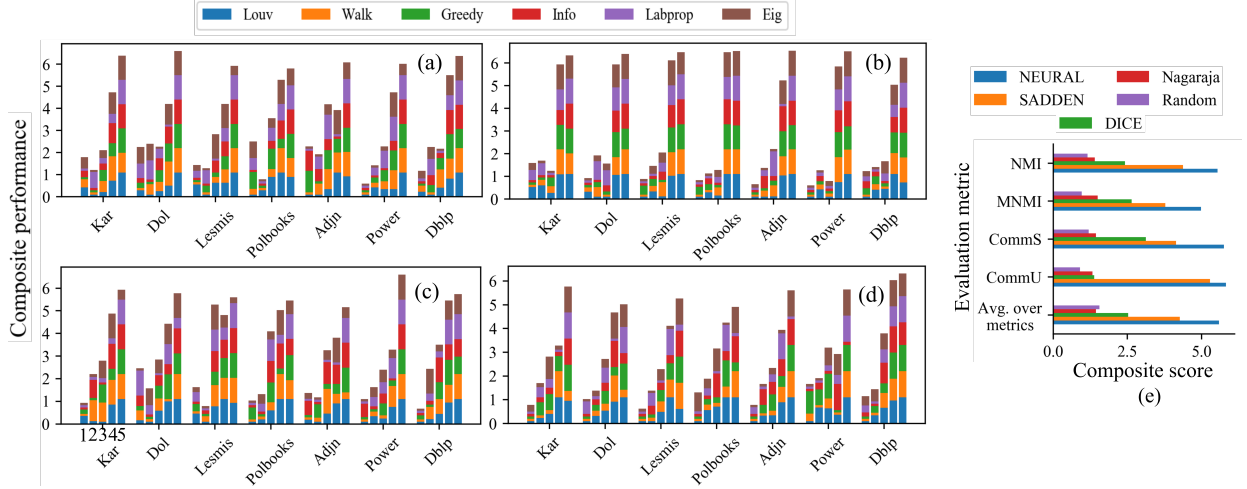


Fig. 5. (Color online) Composite performance of the five competing community deception methods based on (a) NMI, (b) MNMI, (c) CommS, and (d) CommU. Bars in each group (under each dataset) are ordered as follows: (1) Random, (2) Nagaraja, (3) DICE, (4) SADDEN, and (5) NEURAL (as shown in Fig. (e)). Fig.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

Figs. 5(a)-(d) show the composite performance across all the evaluation metrics. Fig. 5(e) shows the composite performance of individual competing methods averaged over all the datasets. 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 datasets), outperforming Random, Nagaraja, DICE and SADDEN by 376.72%, 286.71%, 128.51%, and 26.54%, respectively (see supplementary for the raw accuracy scores over all the datasets). Note that SADDEN turns out to be highly competitive, sometimes showing marginal improvement over NEURAL. However, in general NEURAL is better than others (Fig. 5(e)). We also consider hiding individual target nodes instead of communities. Refer supplementary for the same.

C. Non-uniform Budget for Edge Updates

Till now, we have reported the results with a unified budget β for all the edge update operations. In this section, we extend NEURAL with non-uniform budget wherein separate budget constraints are applied to the two types of allowed edge updates (as elaborated in Section IV-D): (i) β_D for intra-community edge deletion, and (ii) β_A for inter-community edge addition. Such an analysis could be useful in situations wherein the costs incurred while deleting an intra-community edge and adding an inter-community edge are different. We perform experiments under two different settings of β_D , β_A : (i) $\beta_D = 0.3\beta$, $\beta_A = 0.7\beta$, and (ii) $\beta_D = 0.3\beta$, $\beta_A = 0.7\beta$ (we fix β as default, i.e., 30% of the size of the target community C). Tables IV and V provide 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 supplementary for others) for the two settings mentioned above. We observe that NEURAL outperforms SADDEN in most cases.

VII. QUALITATIVE EVALUATION

To interpret the rewiring suggested by NEURAL and SADDEN (top two methods), we further take three real-world

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

Comm. Det. Algo.	NMI		MNMI		CommS		CommU	
	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 $\beta_D = 0.7\beta$; $\beta_A = 0.3\beta$

Comm. Det. Algo.	NMI		MNMI		CommS		CommU	
	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

attributed networks. Unless otherwise state, we only consider deletion of edges (as addition of a new edge does not make any sense for these networks). Louvain algorithm is used for community detection, and the largest community is considered as the target community.

A. Citation Network

We consider 6,320 papers published in Physical Review Journals as nodes and 10,000 citation interactions (we ignore directionality) among them as edges⁶. 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 importance in terms of keeping the identity of the target community intact, being connected to papers (or nodes) published earlier than most in literature).

⁶<https://journals.aps.org/datasets>

NEURAL performs better than SADDEN in terms of updating more edges of such kind (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.

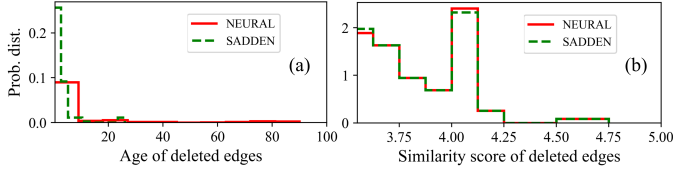


Fig. 6. (a) Age and (b) similarity score distribution of edges selected by NEURAL and SADDEN from Citation and Terrorist networks, respectively.

B. Terrorist Network

We use the *Global Terrorism Database*⁷ to create a network of terrorist group associations. This dataset consists of 191,465 terrorist events around the world between 1970-2018. In order to create the network, we compute the similarity between two terrorist groups based on their activities. To quantify similarity between two groups, we use five attributes: (i) severity of the attack (number of casualties), (ii) attacking strategy used in majority events, (iii) type of weapon used in majority events, (iv) peak year of attacks, and (v) 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 3,616 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 which 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 (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 cancer⁸ 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

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

the Cancer Genome Atlas (TCGA) program, a community-scale effort has been directed towards multi-omic 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 if 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 1,097 normal (as controls) and 113 tumor samples obtained from the TCGA repository. Both networks spanned the same set of 1,000 genes (1,000 nodes). Two nodes are connected by an edge when the Pearson’s correlation coefficient computed across the entire spectrum of control/tumor samples qualifies a cut-off value of 0.6 (12,161 edges). Deleterious mutations in cancer cause wide-spread 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 w.r.t. 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 (3 out 10 rewirings) among SADDEN predicted rewirings. We also measure how accurate NEURAL and SADDEN are to predict the ground-truth edges w.r.t. 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.

VIII. CONCLUSION

This paper addressed the problem of *community deception* – outwitting community detection algorithms from discovering the community affiliation of nodes in a target community. Our major contributions are as follows: (i) we **formalized** the problem and called it *Hide and Seek Community* (HSC); (ii) we proposed a **novel objective function** (Permanence loss) which has been analyzed theoretically; (iii) we proposed NEURAL, a **novel greedy strategy** to optimize Permanence loss; (iv) NEURAL turned out to be **more efficient** than the baselines; and (v) NEURAL unfolded different **meta-information of edges** which would otherwise not have been

⁷<https://www.start.umd.edu/gtd/>

⁸This analysis was conducted by two professional biologists.

possible to explain just by analyzing the network structure. In particular, NEURAL showed promise in the analysis of genome-scale molecular networks.

ACKNOWLEDGEMENT

The work was partially supported by the Ramanujan Fellowship and DST (ECR/2017/001691). T. Chakraborty would like to acknowledge the support of CAI, IIIT-Delhi.

REFERENCES

- [1] S. Fortunato and D. Hric, "Community detection in networks: A user guide," *Physics reports*, vol. 659, pp. 1–44, 2016.
- [2] 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, pp. 139–147, 2018.
- [3] A. Misllove, B. Viswanath, K. P. Gummadi, and P. Druschel, "You are who you know: inferring user profiles in online social networks," in *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 Transactions on Computational Social Systems*, vol. 7, no. 1, pp. 151–163, 2020.
- [5] V. Fionda and G. Pirrò, "Community deception or: How to stop fearing community detection algorithms," *IEEE TKDE*, vol. 30, no. 4, pp. 660–673, 2018.
- [6] S. Nagaraja, "The impact of unlinkability on adversarial community detection: Effects and countermeasures," in *PETS*, 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 *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 *SIGKDD*, 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 Transactions on Computational Social Systems*, vol. 5, no. 4, pp. 1021–1033, 2018.
- [10] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *PRE*, vol. 69, no. 2, 2004.
- [11] M. Chen, K. Kuzmin, and B. K. Szymanski, "Community detection via maximization of modularity and its variants," *IEEE Transactions on Computational Social Systems*, vol. 1, no. 1, pp. 46–65, 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 Mathematics*, vol. 6, no. 1, pp. 29–123, 2009.
- [13] J. Leskovec, K. J. Lang, and M. Mahoney, "Empirical comparison of algorithms for network community detection," in *WWW*, 2010, pp. 631–640.
- [14] P. Pons and M. Latapy, "Computing communities in large networks using random walks," in *Computer and Information Sciences - ISCIS*, p. Yolum, T. Güngör, F. Gürgeç, and C. Özturan, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 284–293.
- [15] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *PNAS*, vol. 105, no. 4, pp. 1118–1123, 2008.
- [16] —, "An information-theoretic framework for resolving community structure in complex networks," *PNAS*, vol. 104, no. 18, pp. 7327–7331, 2007.
- [17] L. Donetti and M. A. Muñoz, "Detecting network communities: a new systematic and efficient algorithm," *JSTAT*, vol. 2004, no. 10, p. P10012, 2004.
- [18] A. Capocci, V. Servedio, G. Caldarelli, and F. Colaiori, "Detecting communities in large networks," *Physica A*, vol. 352, no. 2, pp. 669–676, 2005.
- [19] G. Palla, I. Derenyi, 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, 2005.
- [20] N. Alduaiji, A. Datta, and J. Li, "Influence propagation model for clique-based community detection in social networks," *IEEE Transactions on Computational Social Systems*, vol. 5, no. 2, pp. 563–575, 2018.
- [21] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3–5, pp. 75–174, 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 TKDD*, vol. 11, no. 2, pp. 1–34, 2016.
- [24] M. E. J. Newman, "Modularity and community structure in networks," *PNAS*, vol. 103, no. 23, pp. 8577–8582, 2006.
- [25] S. Muff, F. Rao, and A. Cafilisch, "Local modularity measure for network clusterizations," *PRE*, vol. 72, no. 5, p. 056107, 2005.
- [26] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Physical review E*, vol. 78, no. 4, p. 046110, 2008.
- [27] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *JSTAT*, vol. 2008, no. 10, 2008.
- [28] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *PRE*, vol. 70, p. 066111, 2004.
- [29] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *PRE*, vol. 76, no. 3, p. 036106, 2007.
- [30] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *PRE*, vol. 74, p. 036104, 2006.
- [31] L. Danon, A. Diaz-Guilera, J. Duch, and A. Arenas, "Comparing community structure identification," *JSTAT*, vol. 2005, no. 09, 2005.
- [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," *Molecular bioSystems*, 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, T. Neill, R. T. Owens, L. A. Iniguez, G. Purkins, R. Vadigepalli, B. Evans, L. Schaefer, S. C. Peiper, Z.-X. Wang *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, 2012.



Shravika Mittal is a senior undergraduate student in Computer Science and Engineering at IIIT-Delhi. Her research interests include Social Network Analysis, Network Science, and Natural Language Processing. She has received the Dean's list for Excellence in Academics, and Innovation in Research and Development.



Debarka Sengupta received his Ph.D. from Jadavpur University. Before joining IIIT-D, he worked as an INSPIRE Faculty at Indian Statistical Institute. 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. He has twice been nominated for the prestigious INSPIRE Faculty award - in 2014 and 2016.



Tanmoy Chakraborty is an Assistant Professor and a Ramanujan Fellow at the Dept. of Computer Science and Engineering, IIIT-Delhi, India, where he leads a research group, called LCS2 (<http://lcs2.iiitd.edu.in/>). His primary research interests include Social Network Analysis, Data Mining, and Natural Language Processing. He has received several awards including Google Indian Faculty Award, Early Career Research Award, DAAD Faculty award. More details at <http://faculty.iiitd.ac.in/~tanmoy/>.

Hide and Seek: Outwitting Community Detection Algorithms (Supplementary Materials)

Shravika Mittal, Debarka Sengupta, Tanmoy Chakraborty
{shravika16093, debarka, tanmoy}@iiitd.ac.in

IX. PERMANENCE LOSS IS SUBMODULAR AND MONOTONE

In this section, we prove that the proposed objective function is submodular and monotone w.r.t. the number of edge updates.

Theorem 9.1: Permanence loss \mathcal{P}_l is submodular w.r.t. the addition of an inter-community edge.

Proof: Let A be the set of inter-community edges that are being considered to be added into the network G using Theorem 4.3. Let x_1 and x_2 be two other such inter-community edges where $x_1, x_2 \notin A$. $\mathcal{P}_l(A \cup \{x_1\})$ (resp. $\mathcal{P}_l(A \cup \{x_2\})$) represents the permanence loss due to the addition of the inter-community edge set $A \cup \{x_1\}$ (resp. $A \cup \{x_2\}$) (given by (6)). Therefore,

$$\begin{aligned} \mathcal{P}_l(A \cup \{x_1\}) + \mathcal{P}_l(A \cup \{x_2\}) &= 2 \sum_{i \in A} \mathcal{P}_l(i) + \mathcal{P}_l(x_1) + \mathcal{P}_l(x_2) \\ &= \mathcal{P}_l(A \cup \{x_1, x_2\}) + \mathcal{P}_l(A) \end{aligned}$$

This shows that Permanence loss is submodular w.r.t. the addition of an inter-community edge. ■

Theorem 9.2: Permanence loss \mathcal{P}_l is submodular w.r.t. the deletion of an intra-community edge.

Proof: Let A be the set of intra-community edges that are being considered to be added into the network G using Theorem 4.2. Let x_1 and x_2 be two other such intra-community edges such that $x_1, x_2 \notin A$. $\mathcal{P}_l(A \cup \{x_1\})$ (resp. $\mathcal{P}_l(A \cup \{x_2\})$) represents the permanence loss due to the deletion of the set of intra-community edges $A \cup \{x_1\}$ (resp. $A \cup \{x_2\}$) (given by (4)). Therefore,

$$\begin{aligned} \mathcal{P}_l(A \cup \{x_1\}) + \mathcal{P}_l(A \cup \{x_2\}) &= 2 \sum_{i \in A} \mathcal{P}_l(i) + \mathcal{P}_l(x_1) + \mathcal{P}_l(x_2) \\ &= \mathcal{P}_l(A \cup \{x_1, x_2\}) + \mathcal{P}_l(A) \end{aligned}$$

This shows that Permanence loss is submodular w.r.t. the deletion of an intra-community edge. ■

Theorem 9.3: Permanence loss \mathcal{P}_l is monotone w.r.t. the addition of an inter-community edge.

Proof: Let A be the set of inter-community edges obtained using Theorem 4.3. Let B be the set of inter-community edges such that $B = A \cup \{x_1\}$, where x_1 is another inter-community edge considered to update the network for community deception. Since $A \subseteq B$, $\mathcal{P}_l(B) = \mathcal{P}_l(A) + \mathcal{P}_l(x_1) \geq \mathcal{P}_l(A)$ using (6), which proves that the Permanence loss is monotone w.r.t. to the addition of an inter-community edge. ■

Theorem 9.4: Permanence loss \mathcal{P}_l is monotone w.r.t. the deletion of an intra-community edge.

Proof: Let A be the set of intra-community edges obtained using Theorem 4.2. Let B be the set of intra-community edges such that $B = A \cup \{x_1\}$, where x_1 is another intra-community edge considered to update the network for community deception. Since $A \subseteq B$, $\mathcal{P}_l(B) = \mathcal{P}_l(A) + \mathcal{P}_l(x_1) \geq$

TABLE VII
SCORES OF TWO METHODS: (1) SADDEN, AND (2) NEURAL FOR HIDING INDIVIDUAL TARGET NODES, AVERAGED ACROSS 20 DIFFERENT RUNS.

Network	Score (SADDEN)	Score (NEURAL)
Kar	0.62	0.69
Dol	0.59	0.71
Lesmis	0.64	0.70
Polbook	0.71	0.75
Adjnoun	0.68	0.79
Power	0.54	0.68
Dblp	0.64	0.71

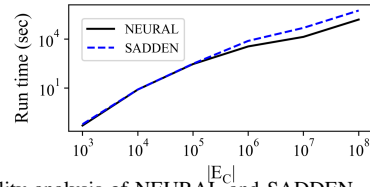


Fig. 7. Scalability analysis of NEURAL and SADDEN.

$\mathcal{P}_l(A)$ using (4), which proves that the Permanence loss is monotone w.r.t. the deletion of an intra-community edge. ■

X. HIDING NODES RATHER THAN COMMUNITIES

In this section, we address a modified version of our problem statement to hide individual target nodes instead of communities. To hide nodes, we employ our proposed NEURAL algorithm by treating nodes as singleton communities. Since deleting an intra-community edge would be redundant in case of a singleton community, NEURAL only focuses on adding inter-community edges to hide target nodes. We evaluate our methodology by selecting $0.3|V|$ nodes randomly as targets for 7 real-world networks. Louvain algorithm is used to extract community assignment. Table VII summarises the probability scores of hiding the target nodes for NEURAL and SADDEN (the best baseline). We observe that NEURAL outperforms SADDEN by assigning a different community label for more target nodes.

XI. SCALABILITY ANALYSIS

The discussion in Section IV-D has established that the time complexity of NEURAL is $\mathcal{O}(|V_C| + |E_C|)$. To show this empirically, we use the LFR Benchmark [26] for generating synthetic networks with $\mu = 0.4$. Louvain algorithm is used for community detection, and the largest community is considered as the target community such that $|E_C|$ lies within the range $10^3 - 10^8$. The results are averaged over 10 such synthetic networks. We record the run times for two deception strategies (based on greedy optimization): (i) SADDEN and (ii) NEURAL (we fix β as default i.e., $\beta = 0.3|V_C|$). Fig. 7 shows that the run time for NEURAL increases linearly with $|E_C|$ thereby verifying the analytical time complexity shown in Section IV-D ($|E_C| \gg |V_C|$; $\mathcal{O}(|V_C| + |E_C|) \approx \mathcal{O}(|E_C|)$). We also observe that with an increase in $|E_C|$, NEURAL outperforms SADDEN in terms of its run time.

TABLE VIII

ACCURACY OF THE FIVE COMPETING COMMUNITY DECEPTION METHODS: (1) NAG: NAGARAJA, (2) R: RANDOM, (3) D: DICE, (4) S: SADDEN, AND (5) N: NEURAL OVER 7 REAL WORLD NETWORKS: (A) KARATE, (B) DOLPHIN, (C) LESMIS, (D) POLBOOKS, (E) ADJN, (F) POWER AND (G) DBLP.

(A) Karate																				
Comm. Det.	NMI					MNMI					CommS					CommU				
	Nag	R	D	S	N	Nag	R	D	S	N	Nag	R	D	S	N	Nag	R	D	S	N
Louv	0.94	0.88	0.92	0.82	0.75	0.33	0.65	0.95	0.27	0.26	1.27	1.50	1.25	2.00	2.25	0.55	0.61	0.72	0.98	0.93
Walk	0.82	0.78	0.76	0.67	0.70	0.46	0.70	0.61	0.17	0.27	1.33	1.00	1.30	1.40	1.41	0.82	1.52	1.54	1.81	1.61
Greedy	0.82	0.82	0.78	0.77	0.72	0.25	0.85	0.73	0.25	0.24	1.58	1.67	2.33	2.00	2.67	0.53	0.43	0.59	0.58	0.71
Info	0.89	0.85	0.85	0.72	0.68	0.47	0.81	0.63	0.46	0.17	1.23	1.00	1.00	1.33	1.34	1.87	1.84	1.91	1.87	2.25
Labprop	0.29	0.76	0.64	0.52	0.00	0.33	0.35	0.54	0.10	0.00	1.49	1.33	1.75	2.00	6.00	0.97	0.57	0.64	0.41	1.31
Eig	0.94	0.89	0.91	0.83	0.82	0.38	0.64	0.73	0.30	0.33	1.02	1.00	1.50	1.75	1.25	2.12	0.98	1.96	1.07	2.14
(B) Dolphin																				
Comm. Det.	NMI					MNMI					CommS					CommU				
	Nag	R	D	S	N	Nag	R	D	S	N	Nag	R	D	S	N	Nag	R	D	S	N
Louv	0.89	0.84	0.85	0.79	0.64	0.21	0.65	0.78	0.20	0.18	1.31	1.40	2.00	2.60	2.75	0.64	0.41	0.56	0.69	0.75
Walk	0.77	0.83	0.78	0.67	0.67	0.35	0.61	0.65	0.37	0.24	2.04	2.75	3.00	2.00	4.25	0.88	0.79	0.98	1.77	1.01
Greedy	0.91	0.89	0.85	0.81	0.77	0.31	0.55	0.54	0.20	0.27	1.36	1.50	2.00	2.75	2.50	0.78	0.54	0.76	0.67	1.04
Info	0.92	0.91	0.87	0.85	0.82	0.27	0.66	0.59	0.31	0.25	1.29	2.20	2.20	2.16	3.80	0.89	0.63	0.81	1.34	0.87
Labprop	0.69	0.71	0.73	0.76	0.67	0.11	0.74	0.36	0.12	0.07	1.69	2.50	1.33	2.25	1.75	1.60	1.10	1.25	0.41	2.00
Eig	0.84	0.84	0.88	0.83	0.82	0.21	0.86	0.73	0.23	0.16	1.65	1.00	1.50	1.80	2.00	0.66	0.64	0.87	1.48	1.03
(C) Lesmis																				
Comm. Det.	NMI					MNMI					CommS					CommU				
	Nag	R	D	S	N	Nag	R	D	S	N	Nag	R	D	S	N	Nag	R	D	S	N
Louv	0.96	0.90	0.89	0.89	0.83	0.32	0.85	0.84	0.33	0.29	1.30	1.67	2.00	2.33	2.16	0.69	0.54	0.88	1.41	0.98
Walk	0.95	0.95	0.94	0.92	0.89	0.55	0.94	0.65	0.43	0.39	1.30	1.50	1.75	1.88	2.00	1.33	0.41	0.87	1.30	1.77
Greedy	0.90	0.92	0.91	0.87	0.82	0.37	0.65	0.78	0.33	0.32	1.30	1.50	1.75	2.00	2.20	0.77	0.66	0.78	0.85	0.98
Info	0.98	0.98	0.95	0.94	0.91	0.44	0.86	0.71	0.35	0.29	1.17	1.11	2.13	2.00	2.38	0.81	0.89	1.02	1.05	1.55
Labprop	0.69	0.74	0.78	0.69	0.60	0.36	0.68	0.53	0.28	0.16	1.47	1.86	2.50	2.00	2.67	0.14	0.44	0.41	1.37	0.55
Eig	0.97	0.96	0.91	0.91	0.95	0.95	0.94	0.73	0.34	0.41	0.96	1.17	2.70	1.75	1.25	1.49	0.45	0.97	0.49	1.82
(D) Polbooks																				
Comm. Det.	NMI					MNMI					CommS					CommU				
	Nag	R	D	S	N	Nag	R	D	S	N	Nag	R	D	S	N	Nag	R	D	S	N
Louv	0.98	0.99	0.95	0.94	0.95	0.24	0.79	0.76	0.22	0.23	1.30	1.25	1.50	1.74	1.75	0.44	0.43	0.62	0.74	0.75
Walk	0.97	0.95	0.94	0.85	0.94	0.94	0.92	0.76	0.32	0.31	1.30	1.25	1.64	2.75	1.75	0.88	0.79	0.82	0.96	1.26
Greedy	0.97	0.93	0.92	0.95	0.91	0.73	0.70	0.83	0.28	0.31	1.12	1.50	1.80	1.25	2.00	0.23	0.34	0.63	2.56	0.95
Info	0.99	0.99	0.98	0.96	0.95	0.34	0.74	0.74	0.32	0.33	1.02	1.00	2.13	2.33	1.83	0.54	0.39	0.98	0.58	1.04
Labprop	0.87	0.81	0.80	0.78	0.73	0.21	0.67	0.59	0.23	0.17	1.50	1.00	3.00	1.67	3.33	0.47	0.48	0.43	2.88	0.52
Eig	0.94	0.92	0.93	0.91	0.92	0.26	0.74	0.63	0.19	0.18	1.57	1.25	1.50	2.25	1.75	0.61	0.66	0.76	0.26	0.84
(E) Adjn																				
Comm. Det.	NMI					MNMI					CommS					CommU				
	Nag	R	D	S	N	Nag	R	D	S	N	Nag	R	D	S	N	Nag	R	D	S	N
Louv	0.74	0.74	0.71	0.62	0.64	0.23	0.73	0.72	0.25	0.22	2.73	2.86	3.23	3.86	4.14	0.89	0.77	0.85	1.03	1.10
Walk	0.97	0.99	0.95	0.95	0.94	0.73	0.85	0.79	0.73	0.69	1.22	1.16	2.00	1.44	1.32	0.60	0.41	0.45	0.64	0.83
Greedy	0.75	0.60	0.62	0.66	0.58	0.41	0.69	0.64	0.22	0.21	1.00	2.50	3.50	3.14	3.86	0.89	0.68	0.97	1.10	1.37
Info	0.50	0.45	0.51	0.57	0.42	0.24	0.73	0.65	0.02	0.05	1.75	1.00	1.16	1.50	2.50	0.92	0.62	0.89	0.78	1.08
Labprop	1.00	1.00	0.00	1.00	0.00	1.00	1.00	0.00	1.00	0.00	1.00	1.00	1.00	1.00	6.00	4.71	1.10	1.23	4.72	0.25
Eig	0.74	0.85	0.77	0.64	0.71	0.42	0.68	0.70	0.40	0.38	1.04	1.30	1.60	2.80	1.70	1.03	0.66	0.79	0.69	1.42
(F) Power																				
Comm. Det.	NMI					MNMI					CommS					CommU				
	Nag	R	D	S	N	Nag	R	D	S	N	Nag	R	D	S	N	Nag	R	D	S	N
Louv	0.97	0.98	0.99	0.98	0.95	0.62	0.74	0.74	0.29	0.05	2.10	1.10	1.50	3.00	4.00	0.14	0.11	0.15	0.13	0.18
Walk	0.98	0.99	0.97	0.93	0.91	0.78	0.83	0.81	0.38	0.39	4.00	2.00	3.00	20.00	24.00	0.34	0.48	0.51	0.41	0.75
Greedy	0.97	0.98	0.96	0.94	0.93	0.86	0.83	0.81	0.03	0.07	2.50	2.50	4.00	3.00	6.00	0.63	0.98	1.10	0.12	1.15
Info	0.98	0.98	0.98	0.97	0.95	0.89	0.92	0.91	0.22	0.10	1.94	2.50	2.00	1.00	4.00	0.74	0.55	0.61	3.89	0.68
Labprop	0.94	0.97	0.92	0.91	0.90	0.84	0.87	0.86	0.38	0.27	4.00	3.00	5.00	7.00	8.00	1.05	1.05	1.14	1.24	1.74
Eig	0.95	0.95	0.93	0.76	0.87	0.54	0.54	0.48	0.08	0.04	3.22	1.50	4.00	3.00	7.00	0.45	0.45	0.62	0.66	0.72
(G) Dblp																				
Comm. Det.	NMI					MNMI					CommS					CommU				
	Nag	R	D	S	N	Nag	R	D	S	N	Nag	R	D	S	N	Nag	R	D	S	N
Louv	1.00	0.99	0.99	0.98	0.97	0.39	0.42	0.35	0.28	0.31	3.75	3.50	4.25	5.00	5.33	0.08	0.05	0.10	0.15	0.16
Walk	1.00	0.99	0.97	0.97	0.94	0.40	0.53	0.49	0.37	0.36	14.25	10.00	16.00	20.00	24.25	0.28	0.28	0.35	0.41	0.47
Greedy	0.99	1.00	0.99	0.98	0.99	0.35	0.35	0.39	0.24	0.23	1.25	1.75	2.50	4.00	3.07	0.08	0.04	0.08	0.12	0.13
Info	0.99	1.00	0.99	0.99	0.98	0.32	0.29	0.35	0.22	0.19	1.25	1.00	5.25	2.00	4.00	0.98	1.11	2.01	2.15	2.09
Labprop	0.85	0.93	0.96	0.89	0.79	0.49	0.44	0.40	0.38	0.29	5.00	3.50	5.00	7.00	7.20	0.89	0.95	1.25	1.50	1.82
Eig	0.93	0.97	0.98	0.92	0.89	0.21	0.17	0.11	0.08	0.07	5.00	1.25	1.50	3.00	3.34	0.68	0.64	0.70	0.73	0.75

TABLE IX

ACCURACY OF TWO COMPETING COMMUNITY DECEPTION METHODS: (1) S: SADDEN (BEST BASELINE), AND (2) N: NEURAL OVER 4 REAL WORLD NETWORKS: (A) DOLPHIN, (B) LESMIS, (C) POLBOOKS, (D) ADJN, AND (E) POWER, SUCH THAT $\beta_D = 0.3\beta$; $\beta_A = 0.7\beta$, $\beta = 0.3|V_C|$

(A) Dolphin								
Comm. Det. Algo.	NMI		MNMI		CommS		CommU	
	S	N	S	N	S	N	S	N
Louv	0.81	0.79	0.25	0.23	2.60	3.00	0.81	0.68
Walk	0.72	0.65	0.37	0.11	2.50	4.00	0.81	1.04
Greedy	0.86	0.78	0.32	0.27	2.25	2.75	0.66	1.11
Info	0.84	0.82	0.24	0.27	2.50	3.80	0.84	0.89
Labprop	0.75	0.69	0.15	0.10	1.80	2.50	0.92	1.56
Eig	0.81	0.78	0.22	0.16	2.00	2.20	1.04	1.13

(B) Lesmis								
Comm. Det. Algo.	NMI		MNMI		CommS		CommU	
	S	N	S	N	S	N	S	N
Louv	0.89	0.87	0.33	0.31	2.13	2.17	0.83	1.43
Walk	0.92	0.90	0.36	0.33	1.75	1.75	1.72	1.58
Greedy	0.88	0.81	0.33	0.32	2.20	2.60	1.05	1.09
Info	0.95	0.90	0.82	0.13	2.00	2.50	1.42	0.98
Labprop	0.73	0.61	0.82	0.24	2.17	2.60	1.02	1.07
Eig	0.95	0.90	0.72	0.13	1.38	1.88	1.42	1.46

(C) Polbooks								
Comm. Det. Algo.	NMI		MNMI		CommS		CommU	
	S	N	S	N	S	N	S	N
Louv	0.95	0.94	0.30	0.28	1.75	1.75	0.63	0.74
Walk	0.91	0.92	0.31	0.31	1.75	2.00	1.25	1.30
Greedy	0.92	0.91	0.32	0.31	1.75	2.00	0.76	0.95
Info	0.94	0.92	0.42	0.36	2.50	2.00	0.47	0.85
Labprop	0.89	0.81	0.35	0.23	2.00	2.10	0.23	0.26
Eig	0.92	0.90	0.20	0.18	1.75	1.50	1.16	1.80

(D) Adjn								
Comm. Det. Algo.	NMI		MNMI		CommS		CommU	
	S	N	S	N	S	N	S	N
Louv	0.63	0.62	0.25	0.23	3.71	4.00	0.95	1.09
Walk	0.96	0.94	0.37	0.31	1.60	1.48	0.55	0.88
Greedy	0.67	0.56	0.24	0.22	3.28	4.00	1.16	1.18
Info	0.57	0.48	0.02	0.06	1.50	2.50	0.78	0.94
Labprop	1.00	0.00	1.00	0.00	1.00	4.00	0.02	0.15
Eig	0.68	0.69	0.28	0.24	2.00	2.10	0.57	1.36

(E) Power								
Comm. Det. Algo.	NMI		MNMI		CommS		CommU	
	S	N	S	N	S	N	S	N
Louv	0.98	0.96	0.33	0.06	3.00	3.20	0.45	0.76
Walk	0.94	0.93	0.39	0.34	14.00	18.00	1.01	1.51
Greedy	0.95	0.92	0.05	0.10	3.00	5.00	0.62	0.89
Info	0.98	0.97	0.24	0.14	1.14	3.00	1.21	0.75
Labprop	0.92	0.89	0.35	0.26	6.50	5.00	0.81	1.02
Eig	0.81	0.88	0.12	0.06	3.50	4.00	0.61	0.77

TABLE X

ACCURACY OF TWO COMPETING COMMUNITY DECEPTION METHODS: (1) S: SADDEN (BEST BASELINE), AND (2) N: NEURAL OVER 4 REAL WORLD NETWORKS: (A) DOLPHIN, (B) LESMIS, (C) POLBOOKS, (D) ADJN, AND (E) POWER, SUCH THAT $\beta_D = 0.7\beta$; $\beta_A = 0.3\beta$, $\beta = 0.3|V_C|$

(A) Dolphin								
Comm. Det. Algo.	NMI		MNMI		CommS		CommU	
	S	N	S	N	S	N	S	N
Louv	0.91	0.80	0.22	0.15	2.40	3.00	0.67	0.69
Walk	0.71	0.66	0.34	0.09	3.00	4.00	0.89	0.97
Greedy	0.80	0.86	0.30	0.24	2.25	2.50	1.09	1.15
Info	0.87	0.85	0.24	0.23	2.67	3.00	1.02	1.14
Labprop	0.72	0.66	0.24	0.08	2.50	2.00	0.57	0.95
Eig	0.76	0.78	0.28	0.18	2.00	2.20	1.05	1.08

(B) Lesmis								
Comm. Det. Algo.	NMI		MNMI		CommS		CommU	
	S	N	S	N	S	N	S	N
Louv	0.90	0.86	0.32	0.30	2.33	2.33	0.90	1.04
Walk	0.95	0.94	0.31	0.24	1.75	1.88	1.70	1.52
Greedy	0.84	0.83	0.32	0.29	2.20	2.40	0.96	1.05
Info	0.95	0.94	0.68	0.51	2.00	2.00	1.15	1.19
Labprop	0.75	0.73	0.68	0.22	1.20	2.00	1.09	1.59
Eig	0.92	0.91	0.16	0.15	1.63	1.34	0.87	1.62

(C) Polbooks								
Comm. Det. Algo.	NMI		MNMI		CommS		CommU	
	S	N	S	N	S	N	S	N
Louv	0.90	0.90	0.29	0.27	2.00	2.00	0.88	0.89
Walk	0.90	0.92	0.35	0.34	2.00	2.10	0.66	0.73
Greedy	0.95	0.92	0.33	0.30	1.50	1.75	1.02	1.14
Info	0.96	0.91	0.38	0.27	2.17	2.19	0.81	0.88
Labprop	0.82	0.78	0.29	0.24	2.00	2.33	1.59	1.76
Eig	0.91	0.89	0.19	0.18	2.00	1.75	1.08	0.83

(D) Adjn								
Comm. Det. Algo.	NMI		MNMI		CommS		CommU	
	S	N	S	N	S	N	S	N
Louv	0.54	0.62	0.23	0.21	4.00	4.43	0.99	1.16
Walk	0.95	0.94	0.36	0.38	1.68	1.69	0.71	0.84
Greedy	0.65	0.63	0.26	0.24	3.57	3.86	1.06	1.21
Info	0.68	0.56	0.12	0.05	1.50	2.50	0.74	0.37
Labprop	1.00	0.00	1.00	0.00	1.00	3.00	0.04	0.10
Eig	0.69	0.73	0.27	0.21	2.30	2.70	0.96	1.38

(E) Power								
Comm. Det. Algo.	NMI		MNMI		CommS		CommU	
	S	N	S	N	S	N	S	N
Louv	0.98	0.96	0.34	0.09	3.50	5.00	0.65	1.21
Walk	0.93	0.96	0.38	0.34	10.00	24.00	1.32	2.44
Greedy	0.95	0.91	0.15	0.12	3.10	7.00	0.67	1.56
Info	0.98	0.96	0.29	0.12	1.10	2.00	1.03	0.55
Labprop	0.93	0.89	0.37	0.26	3.50	4.00	0.90	1.31
Eig	0.80	0.82	0.14	0.04	3.50	4.00	0.66	0.76