


Robustness of community structure under edge addition

Moyi Tian ^{*}*Division of Applied Mathematics, Brown University, Providence, Rhode Island 02912, USA*Pablo Moriano [†]*Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, Tennessee 37830, USA*

(Received 17 April 2023; revised 22 August 2023; accepted 8 September 2023; published 1 November 2023)

Communities often represent key structural and functional clusters in networks. To preserve such communities, it is important to understand their robustness under network perturbations. Previous work in community robustness analysis has focused on studying changes in the community structure as a response of edge rewiring and node or edge removal. However, the impact of increasing connectivity on the robustness of communities in networked systems is relatively unexplored. Studying the limits of community robustness under edge addition is crucial to better understanding the cases in which density expands or false edges erroneously appear. In this paper, we analyze the effect of edge addition on community robustness in synthetic and empirical temporal networks. We study two scenarios of edge addition: random and targeted. We use four community detection algorithms, Infomap, Label Propagation, Leiden, and Louvain, and demonstrate the results in community similarity metrics. The experiments on synthetic networks show that communities are more robust when the initial partition is stronger or the edge addition is random, and the experiments on empirical data also indicate that robustness performance can be affected by the community similarity metric. Overall, our results suggest that the communities identified by the different types of community detection algorithms exhibit different levels of robustness, and so the robustness of communities depends strongly on the choice of detection method.

DOI: [10.1103/PhysRevE.108.054302](https://doi.org/10.1103/PhysRevE.108.054302)

I. INTRODUCTION

Many complex systems, such as critical infrastructures, biological networks, and social groups, exhibit network structures consisting of nodes and edges that capture their connectivity [1]. Extracting different features from these networks is useful to better understand their structure and function [2,3]. Among different properties of networks, many real networked systems demonstrate community structures, or clusters, which are groups of nodes that have higher probability of sharing links with other nodes within the same group than with nodes in different groups. These communities typically represent essential functional or behavioral units in the networks [3–5]. Therefore, it is important to understand the conditions under which they persist. Community *robustness* describes how much network perturbation a community structure can tolerate while still being able to recover its original structure [4,6]. The community robustness problem is of practical interest because we want to understand how well the components in networks can sustain their basic functionalities when facing errors or attacks [7,

Chapter 8]. Usually, network perturbations are used to model either random failures or deliberate attacks in real networked systems [8].

Studying the robustness of communities in networks confronts two main challenges. First, although community structure in networks can provide insights into the organization of nodes based on their connectivity, there is no universal definition of community. This makes the detection of communities itself an ill-posed problem, and hence, in practice, discovering community structure often depends on the methods and application [5]. Second, community robustness involves network perturbations, and there are various ways a network can be modified. Studies so far have focused on edge rewiring and node or edge removal as the network perturbation schemes for studying a system's ability to sustain basic functionality when some components fail [6,9–14].

However, real networked systems, such as communication, citation, or social networks, often increase connectivity over time, so it is important to consider the scenario of an increasing number of edges [15]. Added edges can also simulate errors or attacks [10,16,17]. For example, when simulating errors, added edges can simulate false-positive edges (i.e., edges that are present in data erroneously but do not actually represent the real relations). This is one type of measurement error commonly found, such as in online community, communication, and collaboration network data [18]. Similarly, added edges can also simulate deliberate attacks intended to destroy the established community structure to quickly disrupt the functionality of the system as in the case

^{*}moyi_tian@brown.edu; <https://moyi-tian.github.io/moyi-tian>

[†]moriano@ornl.gov; <https://pmoriano.com>

TABLE I. Synthetic experiment parameters.

Parameter	Description	Value
N	Number of nodes	1000, 10 000
maxk	Max degree for LFR	$0.1N$
$\langle k \rangle$	Average degree for LFR	25
minc	Min community size for LFR	50
maxc	Max community size for LFR	$0.1N$
α	Degree distribution exponent for LFR	-2
β	Community size distribution exponent for LFR	-1
μ	Mixing parameter for LFR	0.01, 0.1, 0.2, 0.3, 0.4, 0.5
h	Times of initial number of edges added up to	10
s	Number of steps to add edges	50
r	Number of realizations per step	50

of a Distributed Denial of Service attack [19,20]. Therefore, there is a need to understand the effect of network densification on community robustness.

Here we conduct a systematic study to understand the impact of edge addition on the robustness of communities. We focus on both the synthetic Lancichinetti-Fortunato-Radicchi (LFR) benchmark graphs [21] and empirical networks to investigate the limits of the robustness of the initial community structure as the network is perturbed through edge addition. We study two scenarios of edge addition, namely random and targeted addition. Random addition selects from the set of all nonexistent edges, and this process is analogous to a random error in the system [17,18]. Targeted addition selects only from the nonexistent, cross-community edges, which we propose to use to simulate attacks. We compute the effects on community robustness by using Normalized Mutual Information (NMI) [22], which is a community similarity measure often used in network community analysis, and by using element-centric clustering similarity [23] to control for biases when comparing clusters. We specifically select four community detection algorithms commonly used in community detection benchmarking studies [24,25]: Infomap, Label Propagation, Leiden, and Louvain. We also demonstrate and compare community robustness results by using the same set of community detection algorithms on empirical email network data, in which the edge density increases over time.

Our results suggest that the chosen clustering algorithm strongly affects community robustness under edge addition. Specifically, in both synthetic and empirical networks, we observe that for different types of community detection algorithms, the similarity measures between communities in the original and the perturbed networks decay with distinct rates while more and more edges are being added. Additionally, in synthetic experiments on LFR benchmark graphs, we observe that with a smaller mixing parameter, which means that initial communities are more loosely connected to each other, the communities tend to be more robust. Synthetic experimental results also align with the expectation that targeted edge addition tends to destroy the original community structure more rapidly compared with random addition, so communities are less robust with targeted addition. In experiments that use empirical data, we observe that the choice of community similarity metrics, NMI or element-centric clustering similarity in particular, also affects the results on community robustness.

II. METHODS

A. LFR benchmark

The LFR benchmark graphs [21,26] are commonly used in network community studies to create graphs with ground-truth partitions. The advantage of using an LFR benchmark is that the degree and the community size both follow power-law distributions, which more closely resemble the properties observed in many real-world networks [2,26–29]. The exponents of degree distributions are controlled by parameter α , and the exponents of community size distributions are controlled by parameter β . We take the typical values of the exponents observed in real networks, i.e., $\alpha = -2$ and $\beta = -1$ [26]. Other parameters required for generating LFR benchmark graphs include the average degree $\langle k \rangle$, maximum degree maxk, minimum community size minc, maximum community size maxc, and the mixing parameter μ , which represents the fraction of nodes that each node shares edges with across different communities. The lower the μ , the higher the ratio between the number of internal and external connections. This leads to higher modularity, which is a quality function commonly used to express the strength of communities in studies of communities [4]. Thus, the smaller the μ , the stronger the partition.

Although we use LFR benchmark graphs for the synthetic experiments, our method is generally applicable to any type of benchmark graph that provides ground-truth community labels, such as the Girvan-Newman benchmark [3] and the stochastic block model [30]. We generate the LFR benchmark graphs by using the publicly available implementation [31] of the algorithm described in Ref. [21]. We conduct experiments on 1000 nodes and then on 10 000 nodes to study the effect of perturbations at different scales. We also examine the effect of the strength of the initial community by varying the

TABLE II. Empirical experiment parameters.

Parameter	Description	Value
np	Number of partitions (fast consensus)	20
s	Number of steps	50
r	Number of realizations per step	10

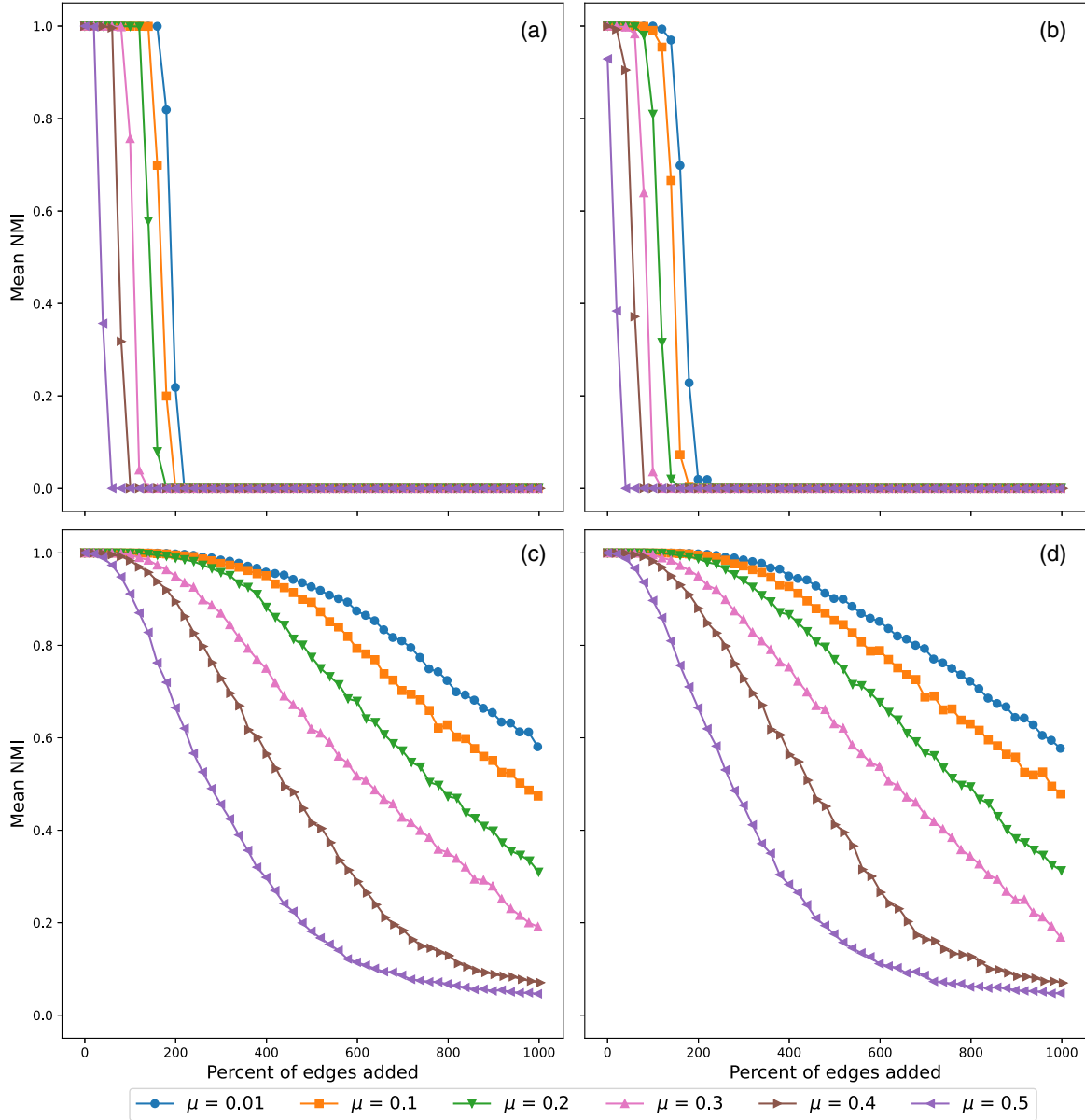


FIG. 1. Mean NMI over the percentage of edges added uniformly at random on LFR benchmark graphs with 1000 nodes. Communities detected by (a) Infomap, (b) Label Propagation, (c) Leiden, and (d) Louvain. Average degree is 25. Edges are added up to $10\times$ the original number over 50 independent steps and are selected without producing multiedges. For each algorithm, we average NMI between the ground-truth partition and new partitions of the perturbed network over 50 independent runs at each step. The maximum amount of edges we add is about 25% of all nonexistent edges.

mixing parameter, μ . The specific parameter values used for generating the LFR benchmark graphs are listed in Table I.

B. Community detection

Community detection is the task of assigning nodes in a network into clusters based on topological similarity [4]. Nodes within the same community are more densely connected compared with the ones across distinct communities. Various community detection algorithms have been developed to find clusters in networks. These algorithms are based on different methodologies to achieve their optimal clustering solutions. In this work, we use four algorithms

based on three popular methods of community detection: information-theoretic-based algorithm Infomap [32], message-passing-based algorithm Label Propagation [33], and modularity-based algorithms Leiden (partly based on smart local move algorithm and improved from Louvain) [34] and Louvain [35]. These chosen algorithms have low-enough computational complexity [34,36] to accomplish our experiments in reasonable run time. For Infomap, Label Propagation, and Louvain, we use the publicly available package [31] implemented by Lancichinetti and Fortunato [37]. The package for Leiden is publicly available on GitHub [38] and described in the original paper [34]. We use the undirected and unweighted implementations for all four algorithms to

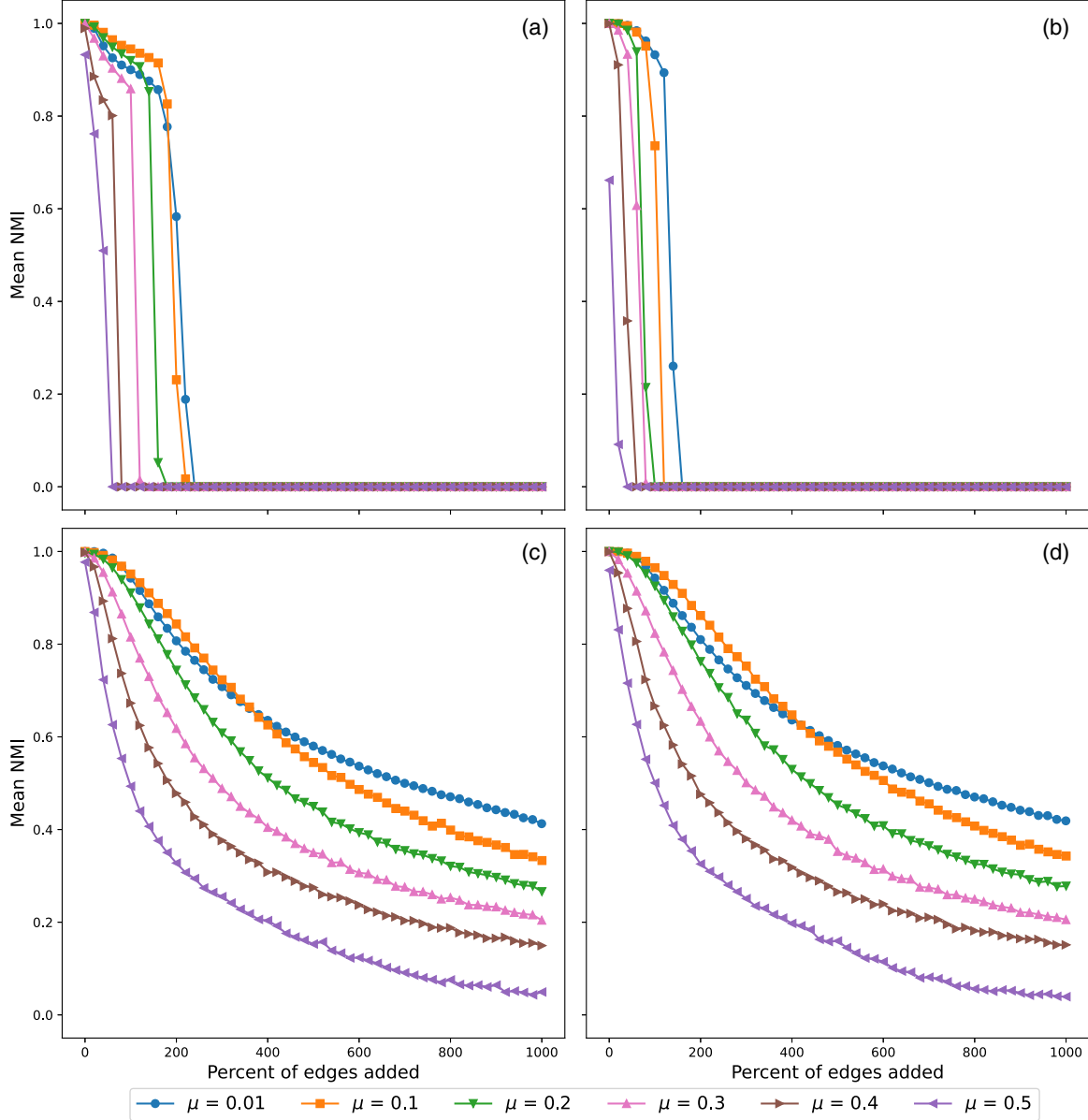


FIG. 2. Mean NMI over the percentage of edges added uniformly at random on LFR benchmark graphs with 10 000 nodes. Communities detected by (a) Infomap, (b) Label Propagation, (c) Leiden, and (d) Louvain. Same parameter values are used as with the 1000-node case. The maximum amount of edges we add is about 2.5% of all nonexistent edges.

ensure consistency with the LFR benchmark graphs we generate. Rigorous analysis of these four clustering algorithms is presented in previous work [25].

The community detection algorithm is an essential component for studying communities because, although the benchmark graphs have ground-truth labels of the communities, few empirical networks have ground-truth partitions. Moreover, the temporal evolution of the networks makes it more difficult to obtain ground-truth clustering information at all times. Notably, although graph embeddings have become popular for downstream tasks, they have not been developed thoroughly enough to efficiently achieve good graph clustering results; their hyperparameter tuning is cumbersome when attempting good performance [39]. By comparison, traditional community detection algorithms require no parameter tuning but can provide relatively good clustering results in

a reasonable run time. For these reasons, we use the four well-developed community detection algorithms for our experiments.

C. Community similarity

To measure similarity between communities, we use NMI [22] and element-centric clustering similarity [23] as the metrics. Without loss of generality, suppose that C_1 and C_2 are partitions on the same set of N nodes. The NMI score of the two partitions is defined as

$$\begin{aligned} \text{NMI}(C_1, C_2) &= \frac{-2 \sum_{i=1}^{|C_1|} \sum_{j=1}^{|C_2|} \mathbb{P}(i, j) \log \left[\frac{\mathbb{P}(i, j)}{\mathbb{P}_1(i) \mathbb{P}_2(j)} \right]}{\sum_{i=1}^{|C_1|} \mathbb{P}_1(i) \log \mathbb{P}_1(i) + \sum_{j=1}^{|C_2|} \mathbb{P}_2(j) \log \mathbb{P}_2(j)}, \quad (1) \end{aligned}$$

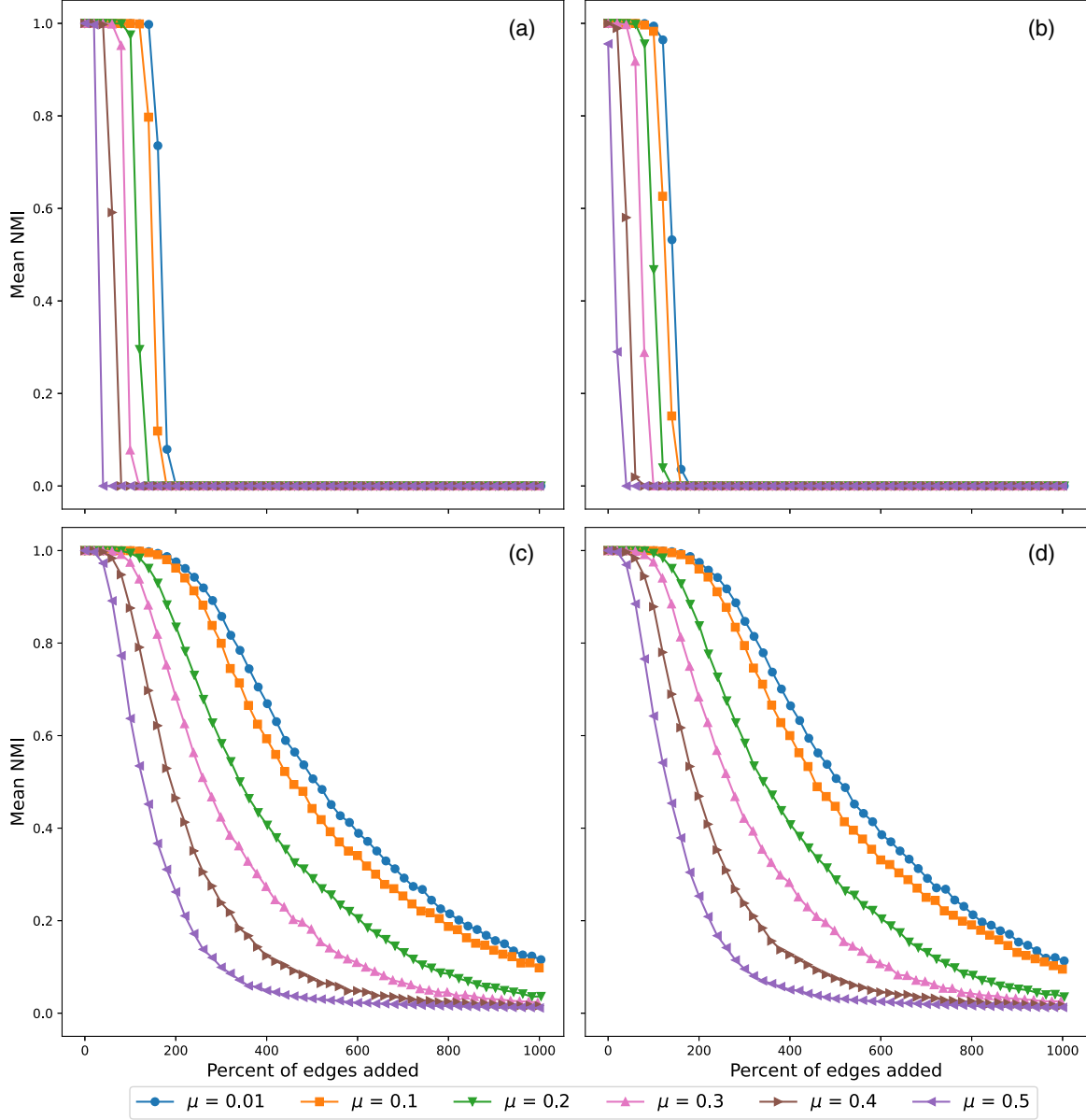


FIG. 3. Mean NMI over the percentage of edges added that are selected uniformly at random across different communities on LFR benchmark graphs with 1000 nodes. Communities detected by (a) Infomap, (b) Label Propagation, (c) Leiden, and (d) Louvain. Same parameter values are used as in previous experiments. The maximum numbers of edges we add are between 26% and 28% of all nonexistent cross-community edges for all μ .

where $\mathbb{P}_1(i) = \frac{|C_{1i}|}{N}$, $\mathbb{P}_2(j) = \frac{|C_{2j}|}{N}$, and $\mathbb{P}(i, j) = \frac{|C_{1i} \cap C_{2j}|}{N}$ for $C_{1i} \in C_1$, $C_{2j} \in C_2$ as the clusters of partition C_1 and C_2 , respectively. We use the scikit-learn implementation of NMI [40] in our experiments.

In addition to NMI, we also compute the similarity between communities by using element-centric clustering similarity, which better copes with issues such as bias in randomized membership, bias in skewed cluster sizes, and the problem of matching [23]. Although NMI tends to favor more clusters, element-centric clustering similarity overcomes such bias in the number of clusters. We report the experimental results computed in this metric using the CluSim package [41] along with the default parameter. Both NMI and element-centric clustering similarity range from 0

to 1, where a higher value means more similarity between partitions.

D. Experimental procedure

We experiment on several computer-generated networks and empirical temporal networks to examine the robustness of their community structures. The empirical networks are real-world data, and we have no control over their intrinsic network properties. The ground-truth communities and the interpretability of the clusterings found by detection algorithms on these data are often unclear [42], so in the studies on clusterings in networks, synthetic networks often serve as handy examples for tests. Here we illustrate the details

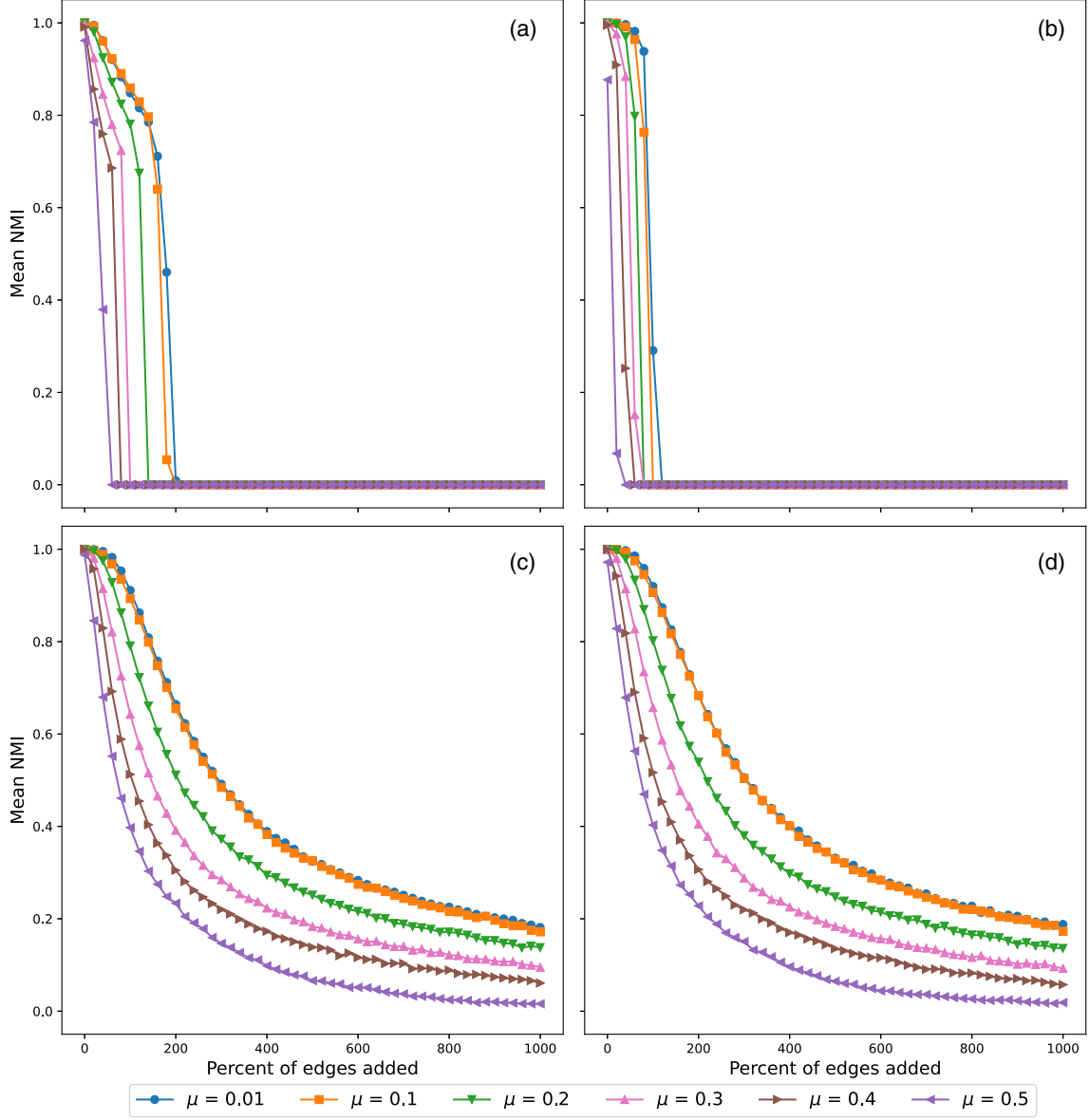


FIG. 4. Mean NMI over the percentage of edges added that are selected uniformly at random across different communities on LFR benchmark graphs with 10000 nodes. Communities detected by (a) Infomap, (b) Label Propagation, (c) Leiden, and (d) Louvain. Same parameter values are used as in previous experiments. The maximum numbers of edges we add are in the range from 2.4% and 2.7% of all nonexistent cross-community edges for all μ .

of the experimental procedures for synthetic and empirical networks. The tests on synthetic and empirical networks are designed differently because the synthetic network is stationary without natural perturbations, whereas the empirical data does not have ground-truth community labels and requires further cleaning to work as comparable examples [43].

1. Experiments on synthetic networks

Suppose the initial network is $G = (V, E)$, where $V = \{1, 2, \dots, N\}$ is the set of N nodes, and $E = \{e_{ij} : i, j \in V\}$ is the set of M edges. Let E^c be the set of edges in the complement graph of G . The benchmark graph provides each node a community label denoted by c_i for $i \in V$. The graph partition is then $C = \bigcup_{k \in \bigcup_{i \in V} \{c_i\}} \{\bigcup_{j \in V} \{j : c_j = k\}\}$. We also

define a set for nonexistent edges across different communities denoted by $E_{\text{inter}}^c = \{e_{ij} \in E^c : c_i \neq c_j, i, j \in V\}$. To start, we choose a community detection algorithm and specify parameters h , s , and $r \in \mathbb{Z}_+$, where h is how many times the initial number of edges is added up, s is the number of steps to add edges, and r is the number of realizations per step. The parameters we use for the synthetic experiments are listed in Table I.

We illustrate the effects of edge addition on community structures through two different network perturbations: random addition and targeted addition. Let $t \in \{0, 1, 2, \dots, s\}$. For the random addition approach, we select $E_v \subseteq E^c$ uniformly at random. For targeted addition, we choose $E_v \subseteq E_{\text{inter}}^c$ uniformly at random, where $|E_v| = \lfloor \frac{hM}{s} \rfloor t$. Notably, the random addition is analogous to the case in which random

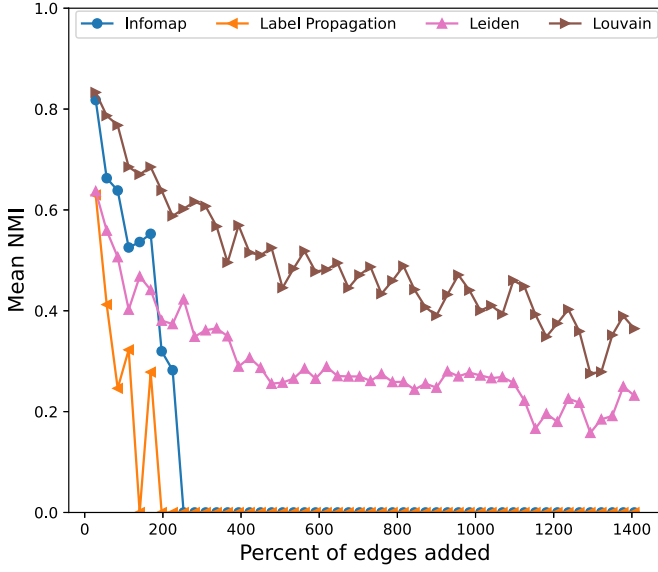


FIG. 5. Mean NMI over the percentage of edges added for the ia-radoslaw-email subnetwork with 74 nodes. We use fast consensus to obtain 20 initial community partitions. Edges are added over 50 steps following time stamps from the dataset. The average NMI is computed over all pairs of the initial consensus partitions and the partitions from 10 independent realizations at each time step.

errors or random additional connections appear. Conversely, targeted addition simulates the case in which much of the information about the community structure is known, and there is an intention to break the current partitions through increasing connectivity. In each of these edge-addition configurations, we create a new graph, $G' = (V, E')$, such that $E' = E \cup E_v$.

We then apply the specific community detection algorithm on G' to yield an associated graph partition, C' . We repeat the previous steps for r -independent times at each t : specifically

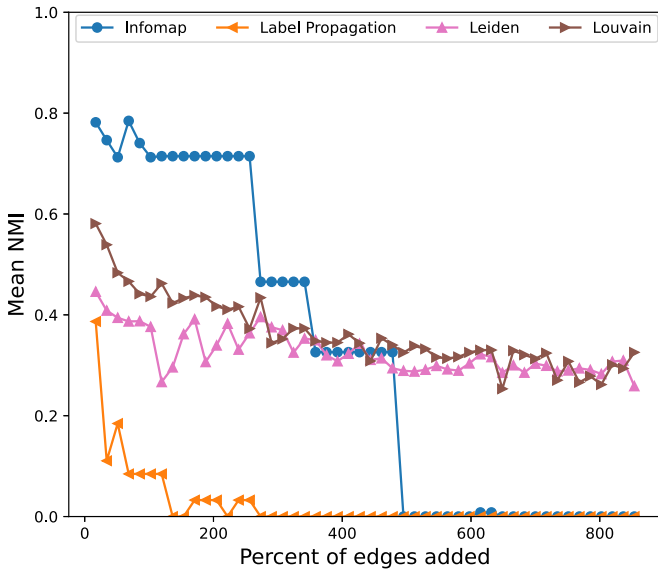


FIG. 6. Mean NMI over the percentage of edges added for the Enron subnetwork with 120 nodes.

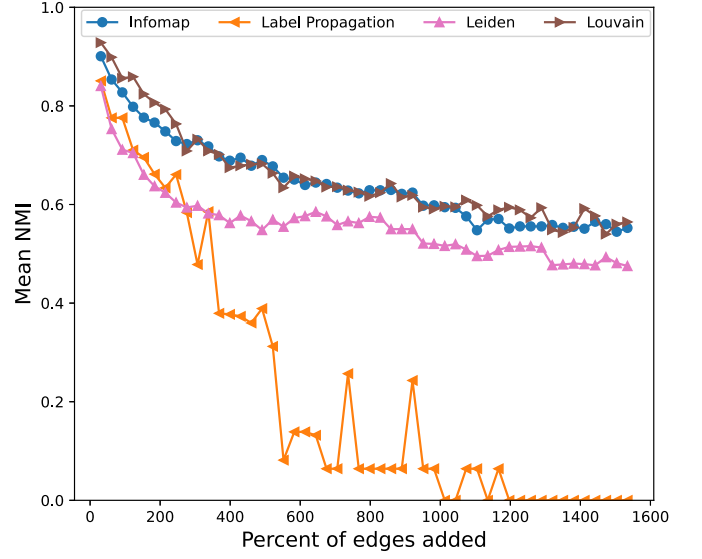


FIG. 7. Mean NMI over the percentage of edges added for the email-Eu-core-temporal subnetwork with 282 nodes.

r realizations at every single step t over the total s steps. Suppose the chosen community similarity metric is \mathcal{S} . We then calculate the metric score $\mathcal{S}_k(C, C'_k)$ for each realization, k , where C'_k is the associated graph partition, and report the average $\mathcal{S}_{\text{avg}} = \frac{1}{r} \sum_{k=1}^r \mathcal{S}_k(C, C'_k)$.

2. Experiments on empirical temporal networks

Temporal email networks are natural candidates for empirical experiments and are comparable to the synthetic experiments. The email conversations between users in these networks naturally emerge at their sent time, which can be directly considered as additional edges over time steps.

Unlike the benchmark graphs, empirical networks do not have ground-truth community labels, so we must first identify

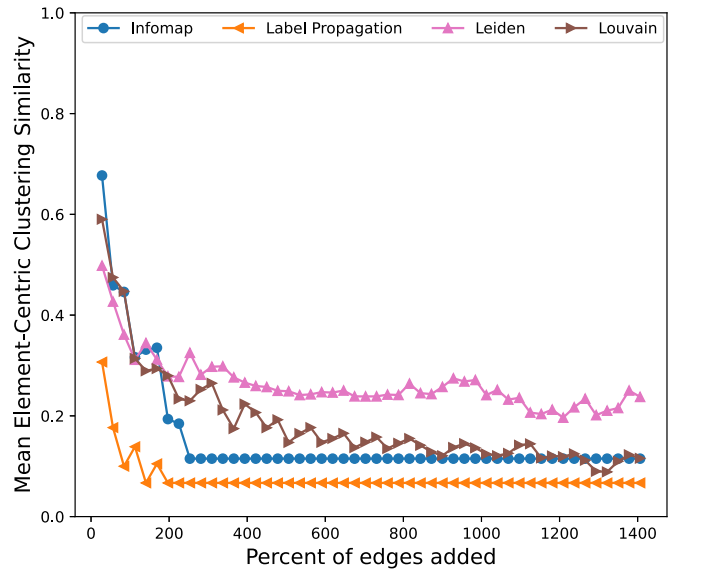


FIG. 8. Mean element-centric clustering similarity over the percentage of edges added for the ia-radoslaw-email subnetwork.

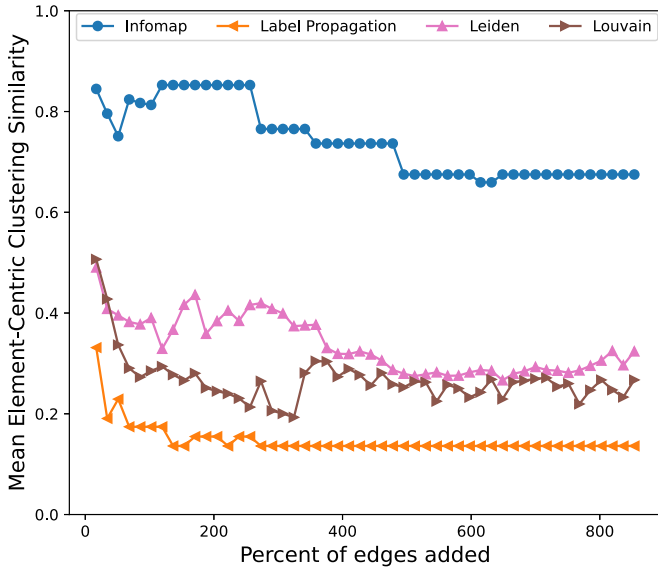


FIG. 9. Mean element-centric clustering similarity over the percentage of edges added for the Enron subnetwork.

a reliable community partition on the initial graph before the start of network perturbation. In doing so, we apply the fast consensus algorithm [44], which is based on the idea of consensus clustering [37]. Because established community detection algorithms produce nondeterministic partitions, consensus clustering was proposed for more stable and accurate results after iterations among multiple clustering results given a prespecified clustering method. There are two parameters we must specify as inputs for the fast consensus algorithm: the number of partitions, np , and the clustering method. The default value for np in the fast consensus algorithm is set to be 20, which is also the value used for tests demonstrated in the original paper [44]. We also found that $np = 20$ balances performance and run time. This choice of the value is specified in Table II. For the clustering method,

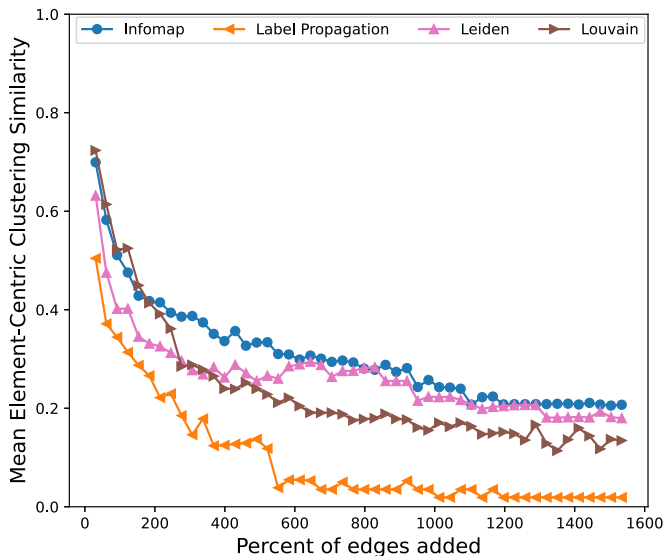


FIG. 10. Mean element-centric clustering similarity over the percentage of edges added for the email-Eu-core-temporal subnetwork.

we choose it to be consistent with the one used for community detection in the later perturbed networks.

In addition, there are other issues that must be addressed before testing on email networks with time stamps associated with edge emergence.

First, the problem is finding appropriate empirical network examples on which we can perform experiments comparable to the synthetic case. Specifically, the empirical network should have a temporally growing number of edges on a fixed set of nodes, and the growth within the recorded time frame should be on the comparable scale as the synthetic experiment, where we add edges up to $10\times$. The problem is that temporal email networks always expand in the number of edges *and* in the number of nodes. If we look at the graph on the set of all users who appear within the given time frame, then we usually find that only a tiny fraction of edges, or sometimes none, are added until the end of the recorded time. This is because there are always new users joining the networks, occasionally even at the very end. For some networks, there is also a co-occurring issue that many users are not very active and do not contribute many new communications (i.e., edges) over time. This is why, rather than using the entire network datasets as obtained, we instead first identify appropriate subnetworks extracted from the original email data and then use them as examples for our empirical experiments. There are different ways to select subnetworks from the original ones. When the entire network is small enough, choosing subnetworks based on an exhaustive search may be possible. However, due to the size of our original empirical datasets, checking all possible combinations of nodes and the growth in density of their induced subnetworks will be computationally expensive. Therefore, our approach is to search over a family of subnetworks induced by the first n nodes showing up in time with n swept from 1 to N where N is the total number of users present in the full dataset. We then look at the growth in density for each of these subnetworks and select an appropriate one as the example to use. More details of our subnetwork selection and the corresponding network properties are described in Sec. III B.

Second, email networks usually have multiedges emerging in time because several emails can be sent between the same pair of users, but the synthetic networks we test on have no multiedges. Here we align our empirical experiment with the synthetic case by preprocessing the network data so that there are no repeating edges. We do so by removing the edges that arrived later and already showed up once at a previous time from the edge list. In this way, we consider an edge to represent an existing relationship between users, thereby omitting the number of conversations that occurred.

Third, we must also identify which network should be treated as the initial network. The empirical networks always start with zero edges, but it is not meaningful to use the null graph because then no communities will ever exist at the initial time. Also, the community detection algorithms take in only the edge lists and so only the nodes incident to the edges present in the list are assigned with community labels. So, to use the algorithms, we must ensure all nodes appear in the edge lists. Therefore, in our empirical experiment, we choose the initial network by picking the first one without any isolated nodes when growing the network in time. This is achieved

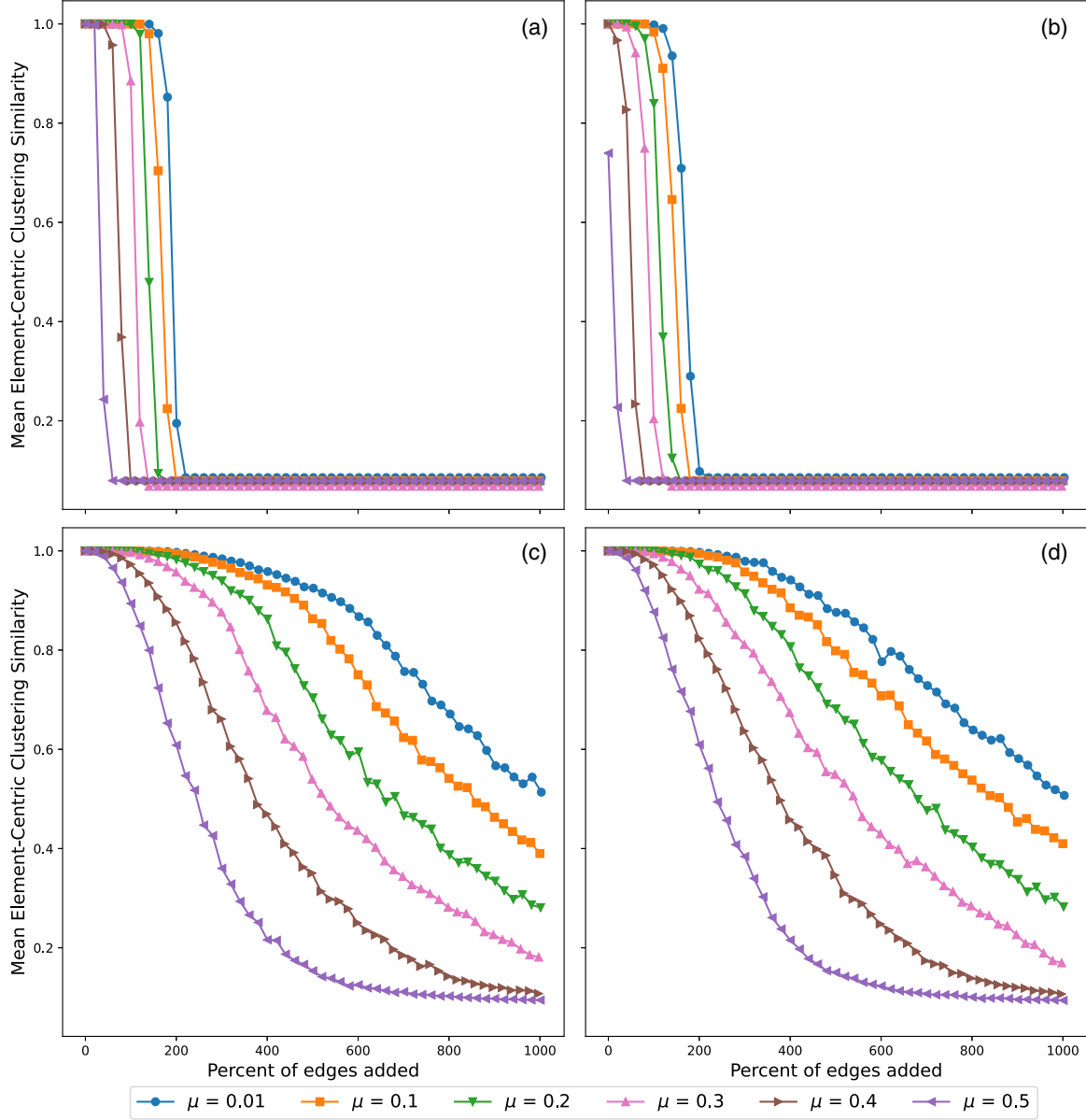


FIG. 11. Mean element-centric clustering similarity over the percentage of edges added uniformly at random on LFR benchmark graphs with 1000 nodes. Communities detected by (a) Infomap, (b) Label Propagation, (c) Leiden, and (d) Louvain.

by adding edges one at a time according to their associated time stamp and checking at which time every node is at least incident to one edge.

In our empirical experiments, we use np for fast consensus, and we also have parameters s and r . Here s refers to the number of steps with respect to time, and r is the number of realizations per step. Table II lists the parameter values for the empirical experiments.

As previously mentioned, we preprocess and identify suitable empirical networks for our experiment. The experimental procedure is as follows. Suppose that we have a pre-cleaned empirical network dataset (without multiedge) with $V = \{1, 2, \dots, N\}$ to be the set of nodes and M to be the number of edges. Recall that edges appear in time one by one, so there are M associated time stamps. At time stamp $i \in \{1, 2, \dots, M\}$, we denote the emergent edge pointing from

node $v_s(i)$ to node $v_t(i)$ by $e_{v_s(i), v_t(i)}$, where $v_s(i), v_t(i) \in V$. Using these notations, the dataset can be presented as an edge list $\{e_{v_s(1), v_t(1)}, e_{v_s(2), v_t(2)}, \dots, e_{v_s(i), v_t(i)}, \dots, e_{v_s(M), v_t(M)}\}$. Then we grow the network until the time stamp $t_0 = \min \{t_s : \bigcup_{1 \leq i \leq t_s} \{v_s(i), v_t(i)\} = V, 1 \leq t_s \leq M\}$, which is the first time when there are no isolated nodes. The initial network is chosen to be $G_0 = (V, E_0)$, where $E_0 = \{e_{v_s(i), v_t(i)} : 1 \leq i \leq t_0\}$. Before each experiment, we specify a clustering method and parameters, $np, s, r \in \mathbb{Z}_+$ (s is upper bounded by $M - t_0$, and our preprocessing should provide an appropriate dataset that $M \gg t_0$). We first apply fast consensus on G_0 using np as the hyperparameter for the number of partitions and the chosen method for the clustering algorithm. The consensus algorithm yields np initial partitions $C_{0,w}$ for $1 \leq w \leq np$. Then we simulate on the evolved networks over time. Specifically, for $1 \leq p \leq s$, let $t_p = t_0 + \lfloor \frac{M-t_0}{s} \rfloor p$. The new

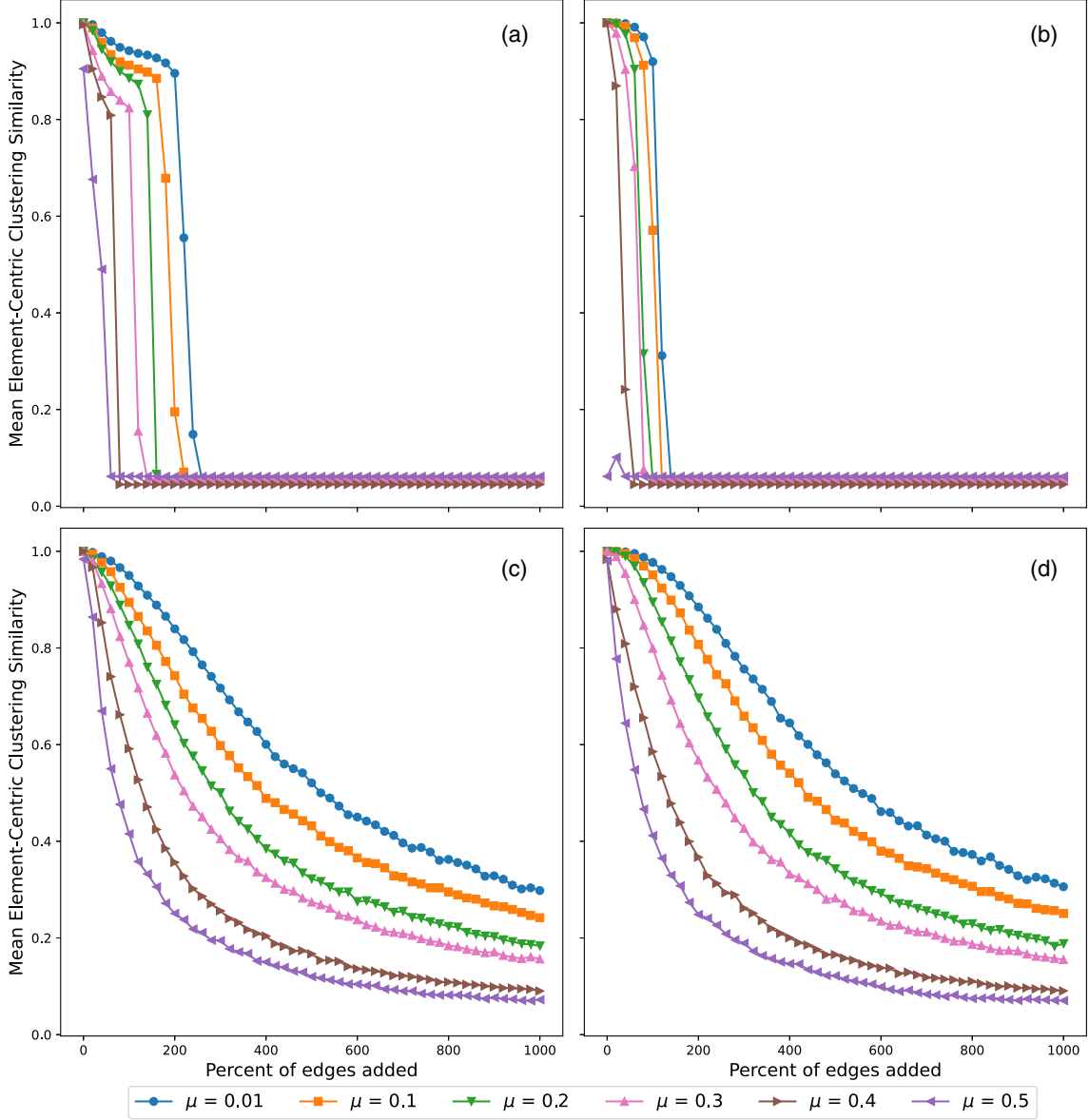


FIG. 12. Mean element-centric clustering similarity over the percentage of edges added uniformly at random on LFR benchmark graphs with 10 000 nodes. Communities detected by (a) Infomap, (b) Label Propagation, (c) Leiden, and (d) Louvain.

graph $G_p = (V, E_p)$ at time step p is constructed such that $E_p = \{e_{v_s(i), v_t(i)} : 1 \leq i \leq t_p\}$. Use the chosen clustering method to find partitions of G_p for r times independently and denote the corresponding clustering results as $C_{p,q}$ for $1 \leq q \leq r$. For each $1 \leq p \leq s$, we report the average community similarity metric score: $\mathcal{S}_{\text{avg},p} = \frac{1}{np \cdot r} \sum_{w=1}^{np} \sum_{q=1}^r \mathcal{S}(C_{0,w}, C_{p,q})$.

III. RESULTS AND DISCUSSION

This section describes the computational results from the synthetic experiments on LFR benchmark graphs and the empirical experiments on subnetworks obtained from temporal email networks.

Note that while more edges are added to the initial network, it is likely that the community structure evolves over perturbation. However, our focus is not on tracking changes in

the community structure itself but on understanding the limits of the robustness of the initial community structure under edge-addition perturbation.

A. Synthetic networks

We test on LFR benchmark graphs with 1000 and 10 000 nodes and report the results calculated with the NMI metric. The results for the same series of experiments using the element-centric clustering similarity metric are presented in Appendix A. We also include the associated plots of standard deviation for all synthetic experiments in Appendix C.

Figures 1 and 2 show how the four different community detection algorithms—Infomap, Label Propagation, Leiden, and Louvain—perform when the LFR benchmarks with 1000 and 10 000 nodes are perturbed under uniformly random edge addition. Recall that in this setting, the edges added at each step are selected uniformly at random from all the

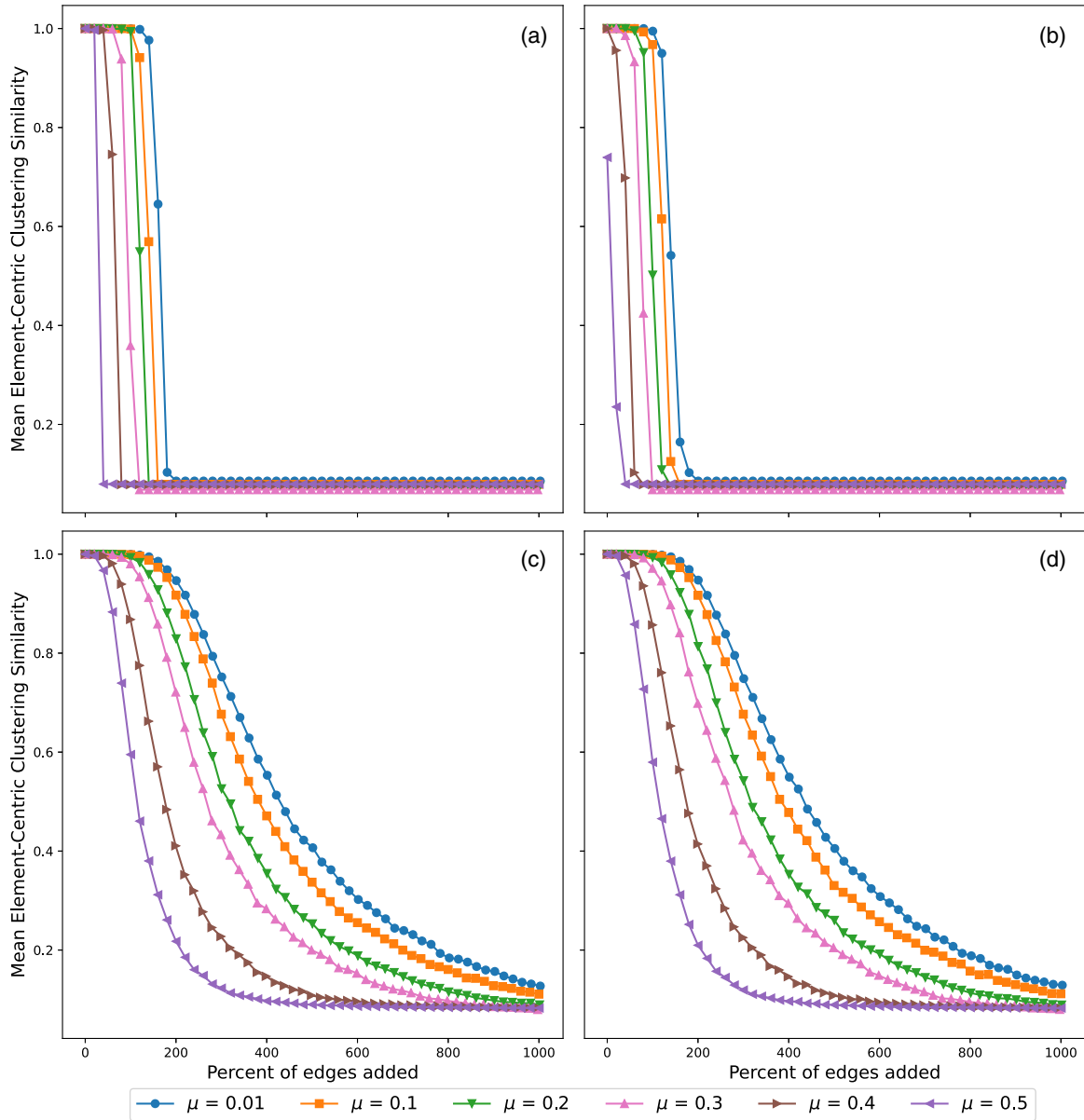


FIG. 13. Mean element-centric clustering similarity over the percentage of edges added that are selected uniformly at random across different communities on LFR benchmark graphs with 1000 nodes. Communities detected by (a) Infomap, (b) Label Propagation, (c) Leiden, and (d) Louvain.

nonexistent edges in the initial network, while multiedges are prohibited. This is analogous to random errors in real-world networks.

LFR benchmark graphs with lower μ values (i.e., stronger initial partitions) have higher NMI values compared with the ones with higher μ after $10\times$ the original number of edges are added. Note that adding edges is essentially a process of balancing the fraction of intercommunity edges with that of intracommunity edges. Hence, this observation aligns with our intuition because when a community partition is stronger, it requires more edges to be added until the established community structure becomes less clear, meaning that the community structure is more robust.

The modularity-based algorithms, Louvain and Leiden, have relatively higher NMI scores vs Infomap and Label Propagation. For example, if we look at the $\mu = 0.3$ curves

in Fig. 1, then Louvain and Leiden need about $6.3\times$ the original number of edges to be added until the similarity scores drop 50%, whereas Infomap and Label Propagation only need about $1.1\times$ and $0.9\times$, respectively. This indicates that Louvain and Leiden are better at detecting community structures that are similar to those initial ones in the LFR benchmark graphs after a large number of edges is added. The exact reasons are unclear, but a plausible explanation is that the algorithms have different behaviors due to their assumptions and methodologies when performed on graphs with different intrinsic network properties. Specifically, we observe that our method of appending edges shifts the degree distribution from the initial power-law distribution to a distribution closer to the binomial.

In our experiments, Infomap and Label Propagation end up only finding one giant community for the entire network

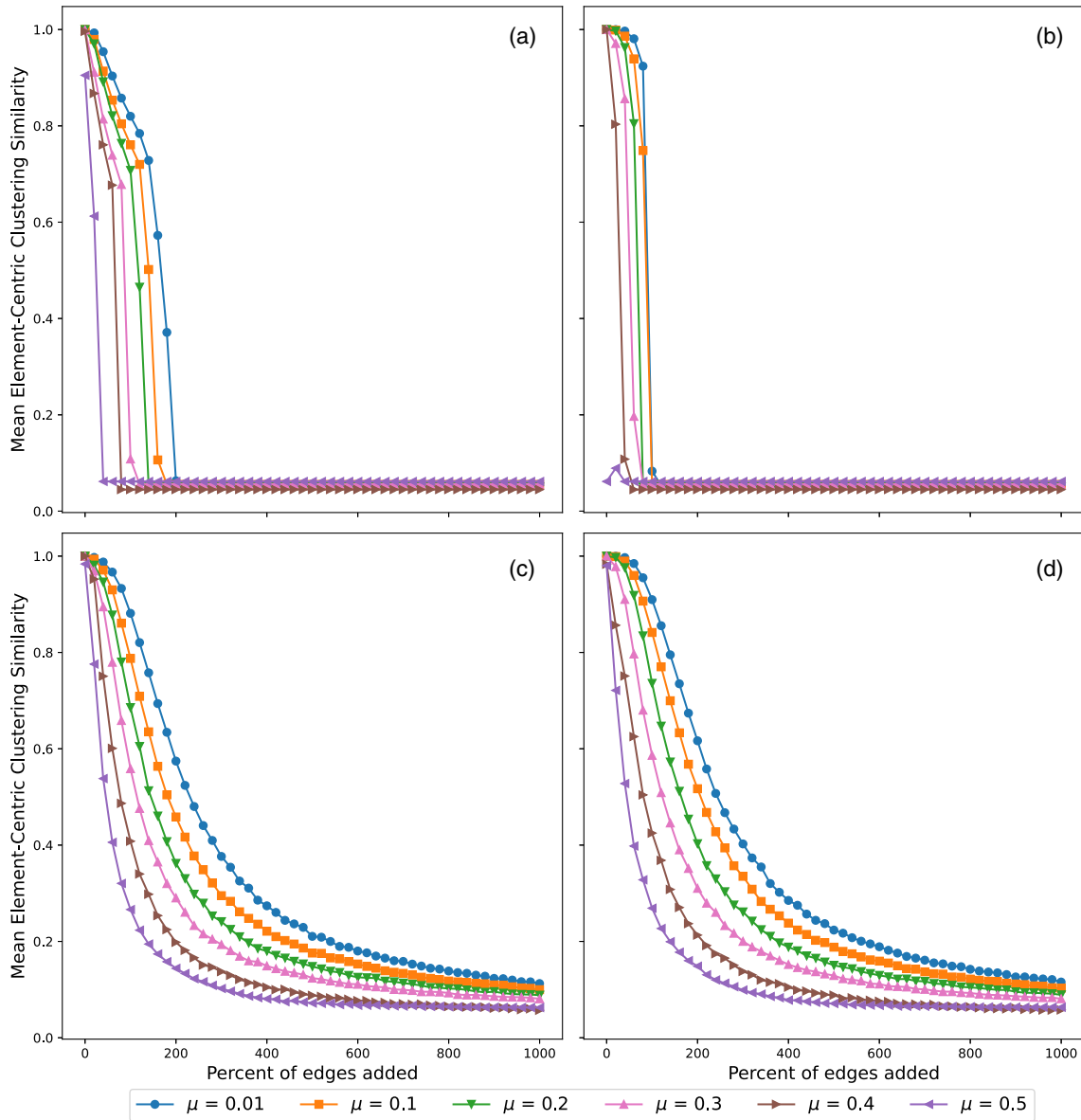


FIG. 14. Mean element-centric clustering similarity over the percentage of edges added that are selected uniformly at random across different communities on LFR benchmark graphs with 10 000 nodes. Communities detected by (a) Infomap, (b) Label Propagation, (c) Leiden, and (d) Louvain.

after about $1 \times$ or $2 \times$ the original number of edges are added. According to the original Infomap paper [32], this flow-based method excels at identifying movement patterns, whereas the modularity-based method is better at detecting structure in networks with pairwise relations but not many flows. The rapid drop in NMI for Infomap could result from our perturbation methods focusing on adding connections between nodes because this focus more directly alters the topological structure of the graph rather than representing any flow of patterns. For Label Propagation [33], the authors explicitly state that their method only detects a single community for the giant connected component in those homogeneous networks without community structures, such as the Erdős-Rényi model. A reason for the rapid drop in NMI for Label Propagation could be that as we fix the number of nodes and keep adding edges selected uniformly at random among the

nonexistent edges (with restriction to the cross-community ones for the targeted case), the perturbed graph gradually becomes more and more homogeneous, which gets closer to the structure of an Erdős-Rényi random graph while growing in its density.

Figures 3 and 4 demonstrate the results on LFR benchmark graphs under targeted edge addition for 1000 nodes and 10 000 nodes, respectively, which means that we restrict the new edges to be across distinct communities in the initial networks. Because the appended edges are forced to connect different communities, the targeted addition should be able to destroy the original community structure quicker than random addition, which is similar to the purpose of attacks on real networks. As expected, the targeted edge addition has relatively lower NMI values for all four algorithms vs the previous results with random edge addition. For example, if

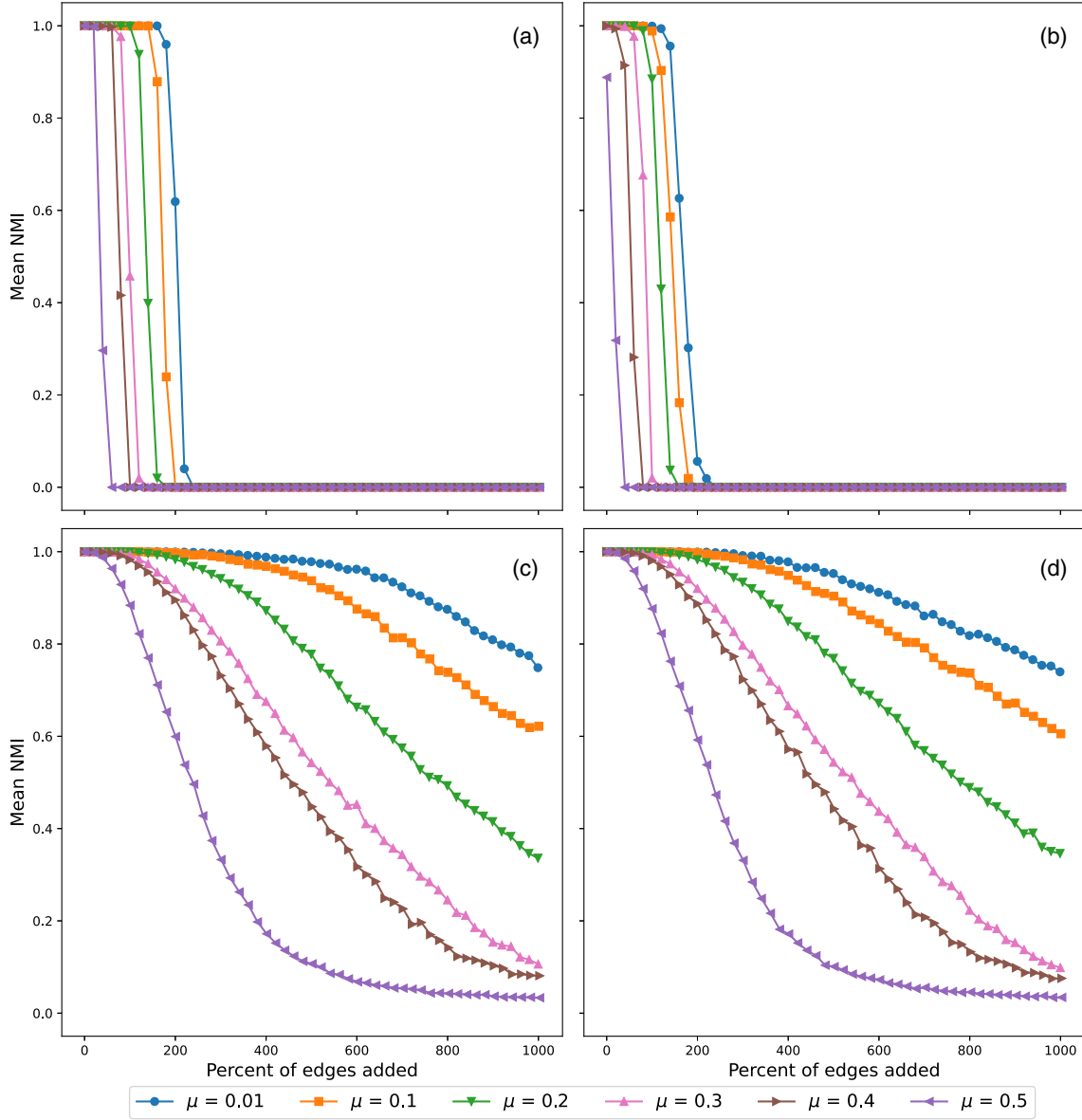


FIG. 15. Mean NMI over the percentage of edges added on LFR benchmark graphs with 1000 nodes. The ratio of intercommunity edges added is 94%, which matches the one in the ia-radoslaw-email subnetwork. Communities detected by (a) Infomap, (b) Label Propagation, (c) Leiden, and (d) Louvain.

we again look at the $\mu = 0.3$ curves with 1000 nodes in Fig. 3 but for targeted addition, Louvain and Leiden need to add about $2.7\times$ (vs $6.3\times$ for random addition), whereas Infomap and Label Propagation need to add about $0.9\times$ and $0.7\times$ (vs $1.1\times$ and $0.9\times$ for random addition) the original number of edges, respectively, to drop the similarity scores below 0.5. In targeted addition, we also observe that networks with stronger initial community structures tend to be more robust, and this is the same trend we see in random addition. Moreover, we again observe that Louvain and Leiden are better at detecting clusters similar to the originals.

Notably, in experiments with larger networks, specifically in Fig. 2, the curves for $\mu = 0.01$ and $\mu = 0.1$ with Louvain and Leiden cross each other when about $4\times$ of the original number of edges is added. Also, in Fig. 4, the curves with Louvain and Leiden for $\mu = 0.01$ and $\mu = 0.1$ are almost

superimposed on each other. Recall that, as we previously discuss in Sec. II C, there are limitations with the NMI metric. For this reason, we also compute results with the element-centric clustering similarity, for which there is a clear separation between curves. We show these results in Figs. 12 and 14.

B. Empirical networks

For the empirical experiments, we use three empirical email networks with time stamps provided for all edges and test on their subnetworks. The specific procedure for these experiments is described in Sec. IID 2. The first network is the ia-radoslaw-email network [45]. The entire dataset is email network activity over the course of 6 months among 167 employee email addresses at a mid-sized manufacturing

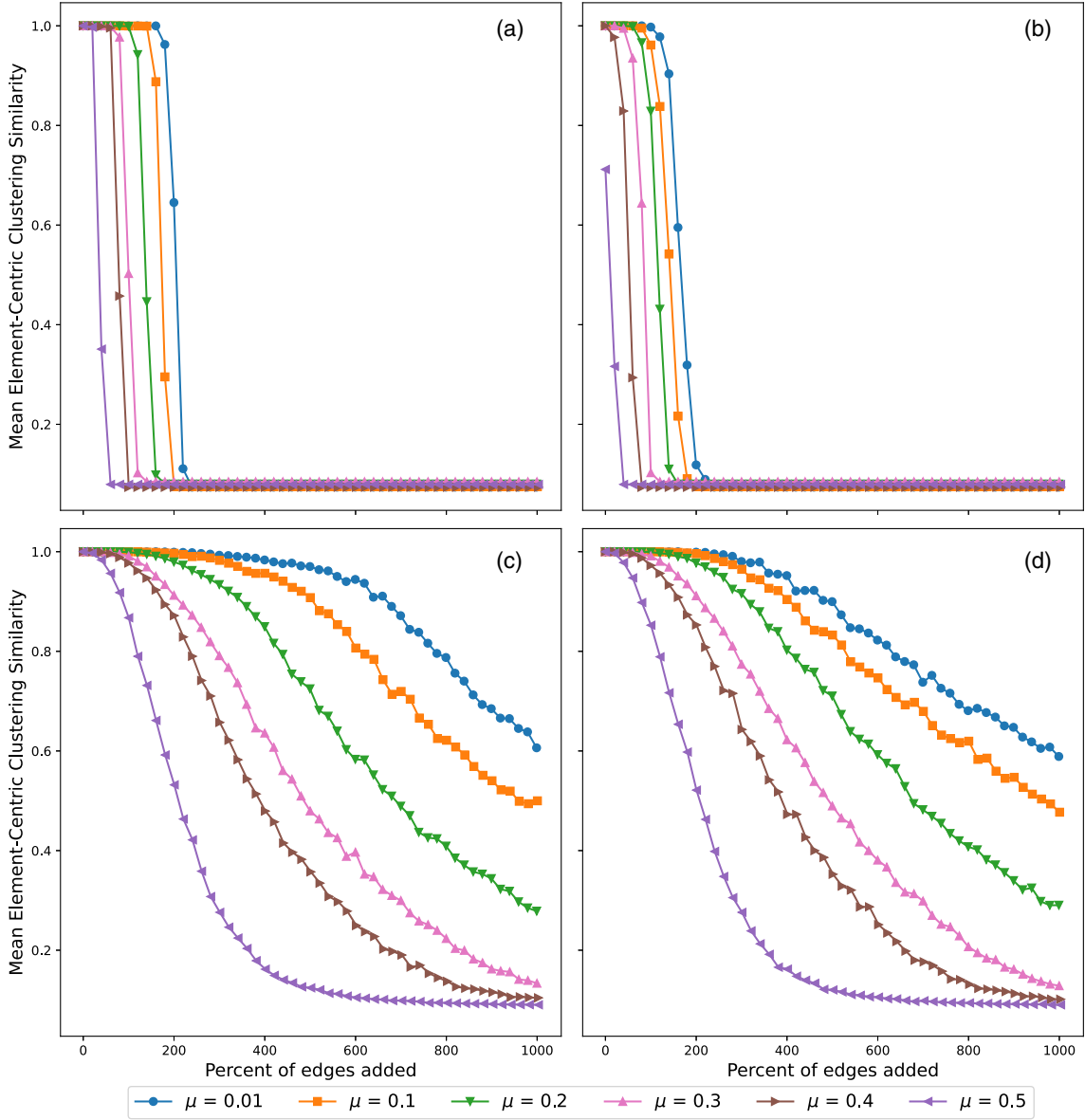


FIG. 16. Mean element-centric clustering similarity over the percentage of edges added on LFR benchmark graphs with 1000 nodes. The ratio of intercommunity edges added is 94%. Communities detected by (a) Infomap, (b) Label Propagation, (c) Leiden, and (d) Louvain.

company. The second network is the Enron network as part of the Koblenz Network Collection [46]. The original network consists of more than 80 000 users and 1 million emails between Enron employees from 1999 to 2003. The last network is the email-Eu-core-temporal network [47] generated from email data among 986 members of a large European research institution between 2003 and 2005.

To select appropriate subnetworks, we look at all subnetworks induced by subsets of nodes $\{1, 2, \dots, n\}$ for every $1 \leq n \leq N$ in the entire graph $G = (V, E)$, where $|V| = N$, and the nodes' *ids* are assigned according to their first appearance in time. We then extract the subnetwork that has a significant increase in density in time, so the demonstration can be comparable to the synthetic results for how many multiples of edges are added by the end. The number of nodes represented in the following network examples refers to the subnetworks on the email users who show up first in

time. Specifically, the ia-radoslaw-email subnetwork has the first 74 nodes in time with their corresponding 1457 edges, the Enron subnetwork is obtained by first trimming down to a 1999–2002 time frame and then selecting the first 120 nodes in time with their associated 1603 edges, and the email-Eu-core-temporal subnetwork is the first 282 nodes with 4544 edges.

Figures 5–7 show the mean NMI over the percentage of added edges in the subnetworks of the ia-radoslaw-email, Enron, and email-Eu-core-temporal networks, respectively. The corresponding results for standard deviation are included in Figs. 25–27 in Appendix D.

Notably, empirical networks are more complex to analyze because they lack ground-truth community structure, the degree distribution might not belong to a certain family of distributions, and the set of edges added in time may be governed by different unknown factors that we cannot control.

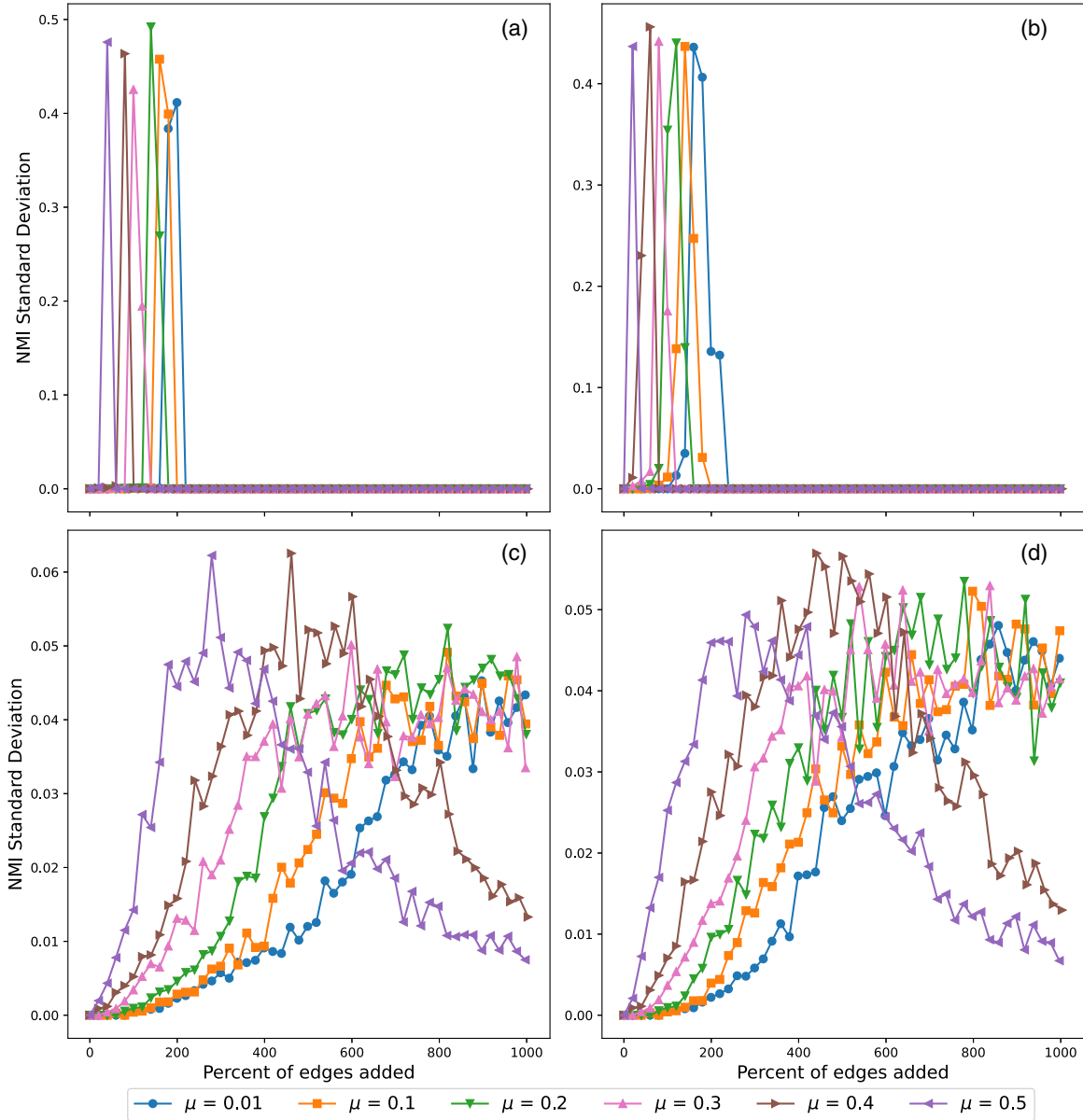


FIG. 17. Standard deviation of NMI over the percentage of edges added uniformly at random on LFR benchmark graphs with 1000 nodes. Communities detected by (a) Infomap, (b) Label Propagation, (c) Leiden, and (d) Louvain.

In considering the effect of edge-addition rules in the empirical compared with the synthetic cases, we look at the ratio of intercommunity edges among all added edges over time for each empirical network. For ia-radoslaw-email, Enron, and email-Eu-core-temporal subnetworks, the intercommunity ratios are 94%, 90%, and 59%, respectively. Note that the LFR benchmark graphs are sparse and so there is a lower fraction of intracommunity edges available to be added. Specifically, for the LFR benchmark on 1000 nodes, the number of intracommunity nonexistent edges is only about twice the original number of edges but the number of intercommunity nonexistent edges can go to about $37\times$. This means that for our random addition on LFR benchmark with 1000 nodes, the fraction of intercommunity edges added is around 95%, and due to the limited number of intracommunity edges, it is impossible to have intracommunity edges taken more than

20% of the $10\times$ additional edges. The ia-radoslaw-email subnetwork has the ratio of added intercommunity edges most comparable with our previous synthetic experiments on 1000 nodes. To draw a direct comparison between the empirical and the synthetic cases, we experiment on synthetic networks where the ratio of added intercommunity edges is controlled to match the one in ia-radoslaw-email subnetwork, namely 94%. The results show similar behaviors as the previous synthetic ones and we include them in Appendix B.

In addition, we acknowledge that the chosen subnetworks described here are not as large as the synthetic benchmark graphs: The benchmark graphs have thousands of nodes, whereas the empirical network examples only have hundreds.

Among these results in NMI, the experiments on the ia-radoslaw-email and Enron subnetworks reveal similar behaviors as the synthetic networks (i.e., Leiden and Louvain

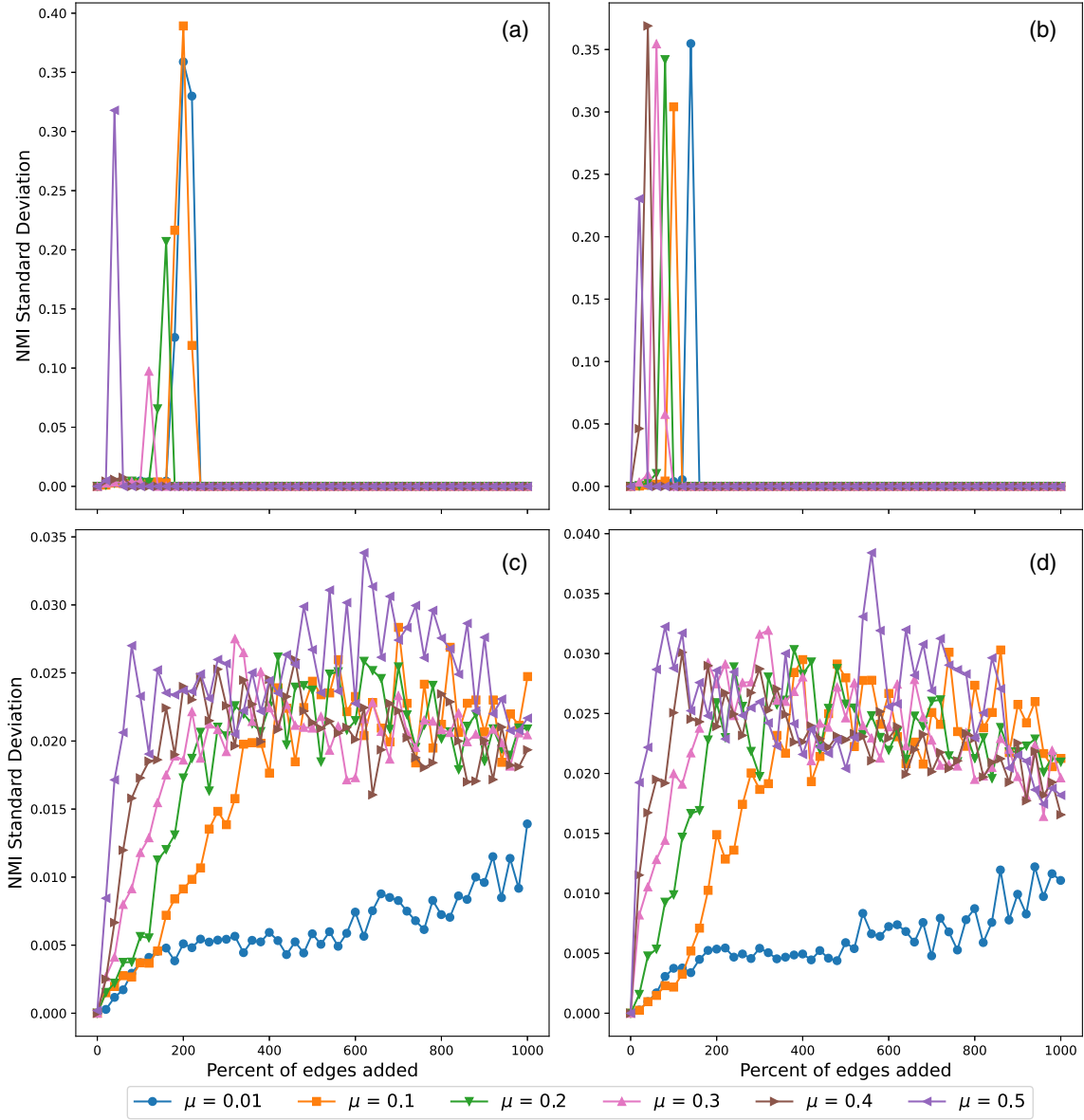


FIG. 18. Standard deviation of NMI over the percentage of edges added uniformly at random on LFR benchmark graphs with 10 000 nodes. Communities detected by (a) Infomap, (b) Label Propagation, (c) Leiden, and (d) Louvain.

appear to detect more sustainable community structure than Infomap or Label Propagation as more edges are added). However, in the experiments on the email-Eu-core-temporal subnetwork, Infomap has performance similar to Leiden and Louvain, whereas Label Propagation has the lowest NMI score almost throughout all time steps. Although there may be intertwining reasons for these results, one shared phenomenon we observe for all subnetworks is that with Infomap or Label Propagation, the mean NMI drops to almost 0, whenever the detection algorithm begins to detect only one community in the perturbed networks.

While community detection algorithms have different performance, we also observe effects from using different community similarity metrics. Specifically, we repeat the same sets of experiments but replace the NMI metric with the element-centric clustering similarity. Figures 8–10 illustrate these results. The corresponding standard deviations are

provided in Figs. 28–30 in Appendix D. Using this different metric, we find that Infomap, Leiden, and Louvain do not show consistent advantages over each other, but Label Propagation, although not necessarily dropping to 0 by the end of time, always has the lowest metric value.

One noticeable difference in these element-centric clustering similarity results from the NMI results is the curve for Infomap in the Enron subnetwork. In the NMI plot (Fig. 6), the value eventually drops to 0, which shows that the community structure detected by Infomap is not as robust as that by Leiden or Louvain in terms of NMI. However, in the element-centric clustering similarity metric, the clustering similarity value is the highest for Infomap throughout time among the four algorithms, and this suggests that the community structure found by Infomap is the most robust in our experiments according to this alternative metric. To determine what is happening, we look at the mean number of

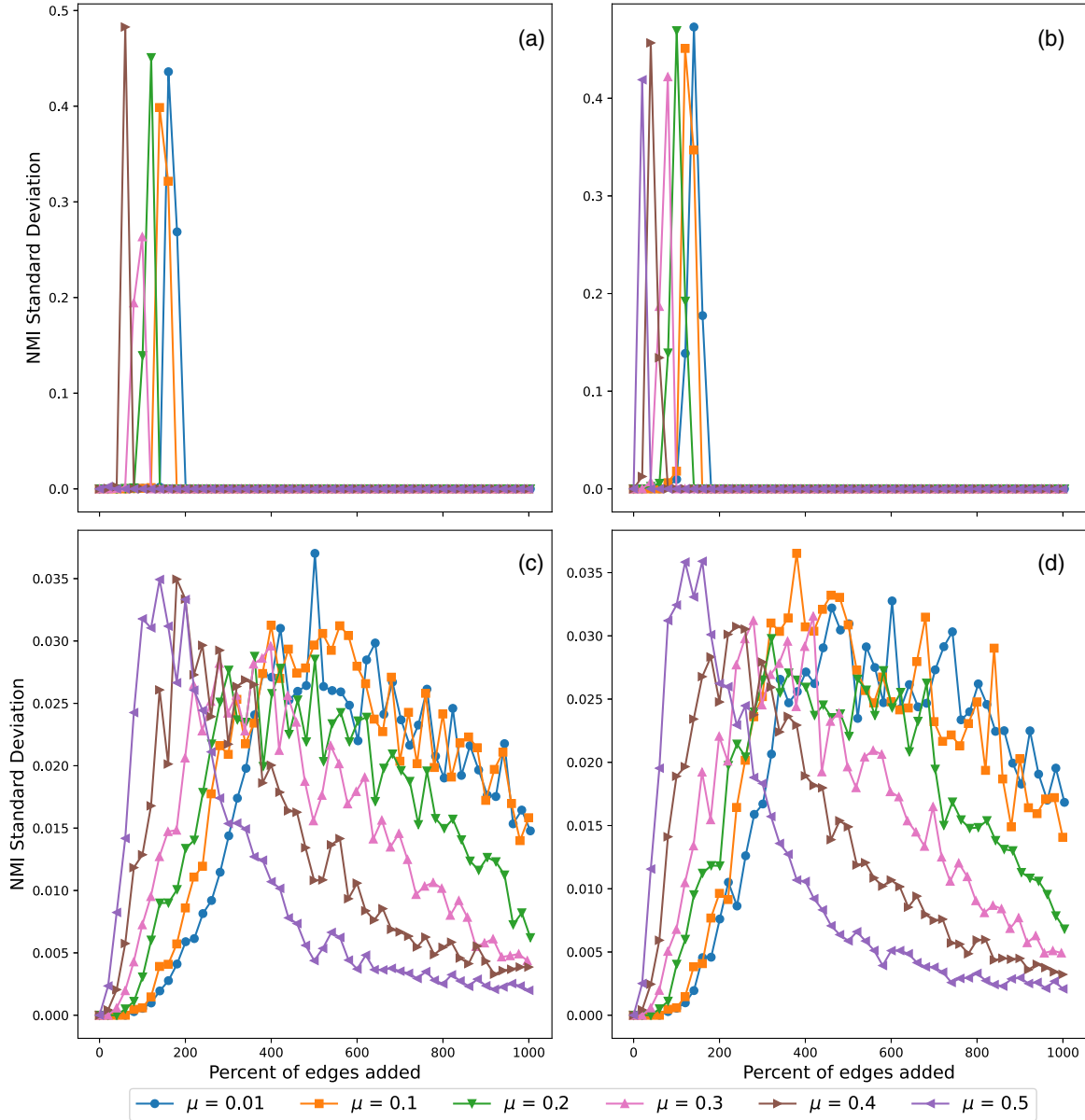


FIG. 19. Standard deviation of NMI over the percentage of edges added that are selected uniformly at random across different communities on LFR benchmark graphs with 1000 nodes. Communities detected by (a) Infomap, (b) Label Propagation, (c) Leiden, and (d) Louvain.

communities detected by the four algorithms at each step and find that it drops from higher numbers to 1 for both Infomap and Label Propagation but maintains around 5 or 6 for Leiden and Louvain by the end of time. Note that the NMI always outputs 0 whenever one of the two partitions being compared has only one community due to the formulation of this metric, so we suspect that this property might have hindered NMI from capturing some similarities between the communities in the initial and the highly perturbed networks, especially after the number of detected communities drops to 1.

Nonetheless, when networks are perturbed under edge addition, and the density is increased by a significant amount, different community detection algorithms start to show clear discrepancies in whether they can find a community structure similar to the initial one. Hence, the chosen community detection algorithm plays an important role in detecting robust community structures over time.

IV. CONCLUSION

We design synthetic and empirical experiments to test the robustness of community structure under the perturbation of edge addition by using different community detection algorithms. Overall, we found that community robustness strongly depends on the community detection algorithm selected. In the synthetic experiments, we use LFR benchmark graphs and control the mixing parameter, μ . To mimic how edges may be added in different scenarios in real networks and to illustrate the difference in the outputs, we add random edges in two different ways. Both ways select additional edges from nonexistent edges. One is a completely uniformly random selection, which is analogous to random errors, and the other is a targeted selection from edges across different communities, which is analogous to attacks in real-world networks.

We demonstrate results for six different mixing parameter values and the two different edge-addition methods described

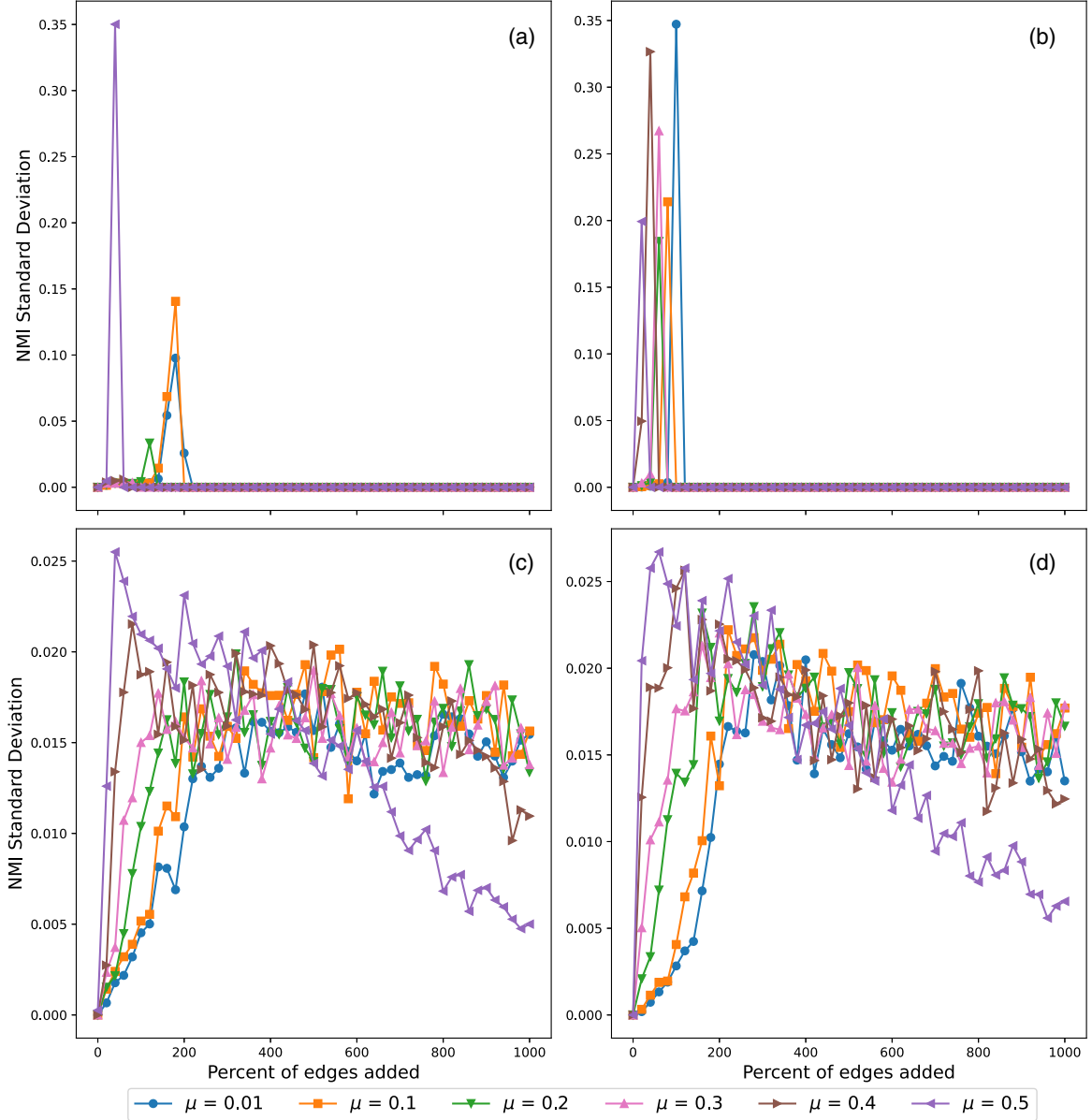


FIG. 20. Standard deviation of NMI over the percentage of edges added that are selected uniformly at random across different communities on LFR benchmark graphs with 10 000 nodes. Communities detected by (a) Infomap, (b) Label Propagation, (c) Leiden, and (d) Louvain.

above. The clustering similarity scores computed in NMI indicate that networks with lower mixing parameters (i.e. stronger partitions) have more robust community structures under the perturbation of edge addition. Our targeted edge-addition method can more efficiently alter the initial communities compared with the random addition. As expected, the NMI values drop faster in the targeted case among all chosen community detection algorithms.

In these synthetic experiments, modularity-based algorithms, Leiden and Louvain, show better performance in detecting more similar communities to the initial ones vs Infomap and Label Propagation, which cannot detect any community structure when graphs become too dense. In other words, Leiden and Louvain excel at finding more robust communities in networks that can withstand more severe network perturbations. We also observe effects caused by community

similarity metrics, NMI and element-centric clustering similarity specifically, but the takeaways in the two metrics are not significantly different in the qualitative sense. The overall impacts of the initial partition strength, the edge-addition method, and the selected community detection algorithm are similar with either metric.

We acknowledge that empirical temporal networks introduce more complexity, and we see different community similarity metrics demonstrate significantly different results when determining whether the detected communities are similar to the initial ones (i.e., whether the community detection algorithm can find a robust community structure). In the empirical experiments, we again find that community detection algorithms play an important role in the robustness of communities. Specifically, for all three empirical subnetworks tested, we observe that Label Propagation performs the worst

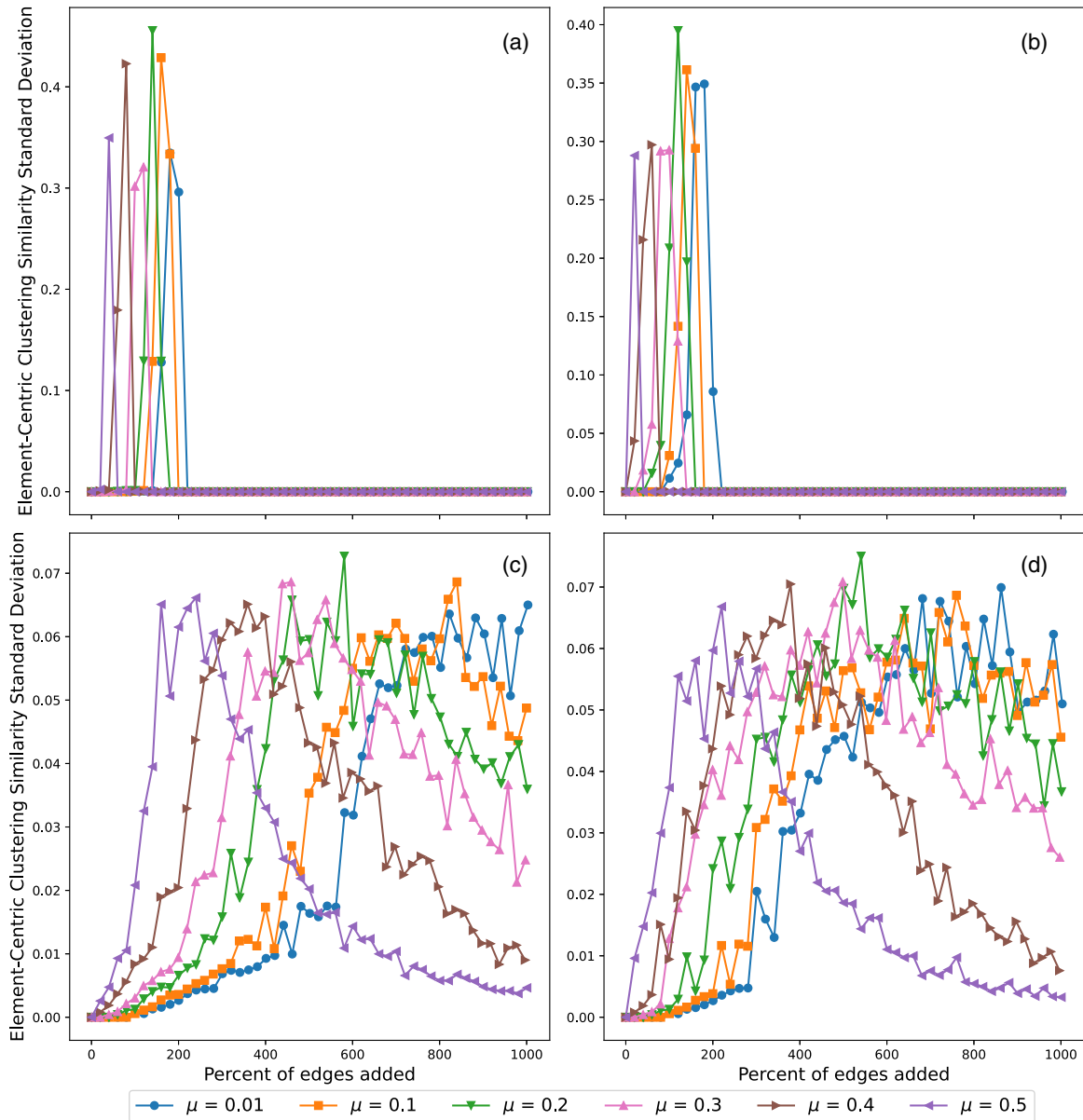


FIG. 21. Standard deviation of element-centric clustering similarity over the percentage of edges added uniformly at random on LFR benchmark graphs with 1000 nodes. Communities detected by (a) Infomap, (b) Label Propagation, (c) Leiden, and (d) Louvain.

among the four algorithms in detecting robust communities over time using either NMI or element-centric clustering similarity.

When considering future research directions, we note that the different metrics and community detection algorithms show different outcomes on the empirical temporal networks. To understand more about their effects on the community robustness performance, we need many more network examples in which edge densities expand over time. Adequate network candidates with properties in different families—including scales, densities, degree distributions, and edge-addition rules—are needed for comparison tests in order to distinguish the effects from each individual aspect. Another direction for future research is to explore different types of generative models for benchmarking and include community detection algorithms based on other methods for comparison.

The code for reproducing our experiments can be found on GitHub [43].

ACKNOWLEDGMENTS

This paper has been authored by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. The publisher, by accepting the article for publication, acknowledges that the U.S. government retains a nonexclusive, paid up, irrevocable, world-wide license to publish or reproduce the published form of the manuscript, or allow others to do so, for U.S. government purposes. The DOE will provide public access to these results in accordance with the DOE Public Access Plan.

This research was sponsored in part by Oak Ridge National Laboratory's (ORNL's) Laboratory Directed Research and Development program and by the U.S. Department of Energy.

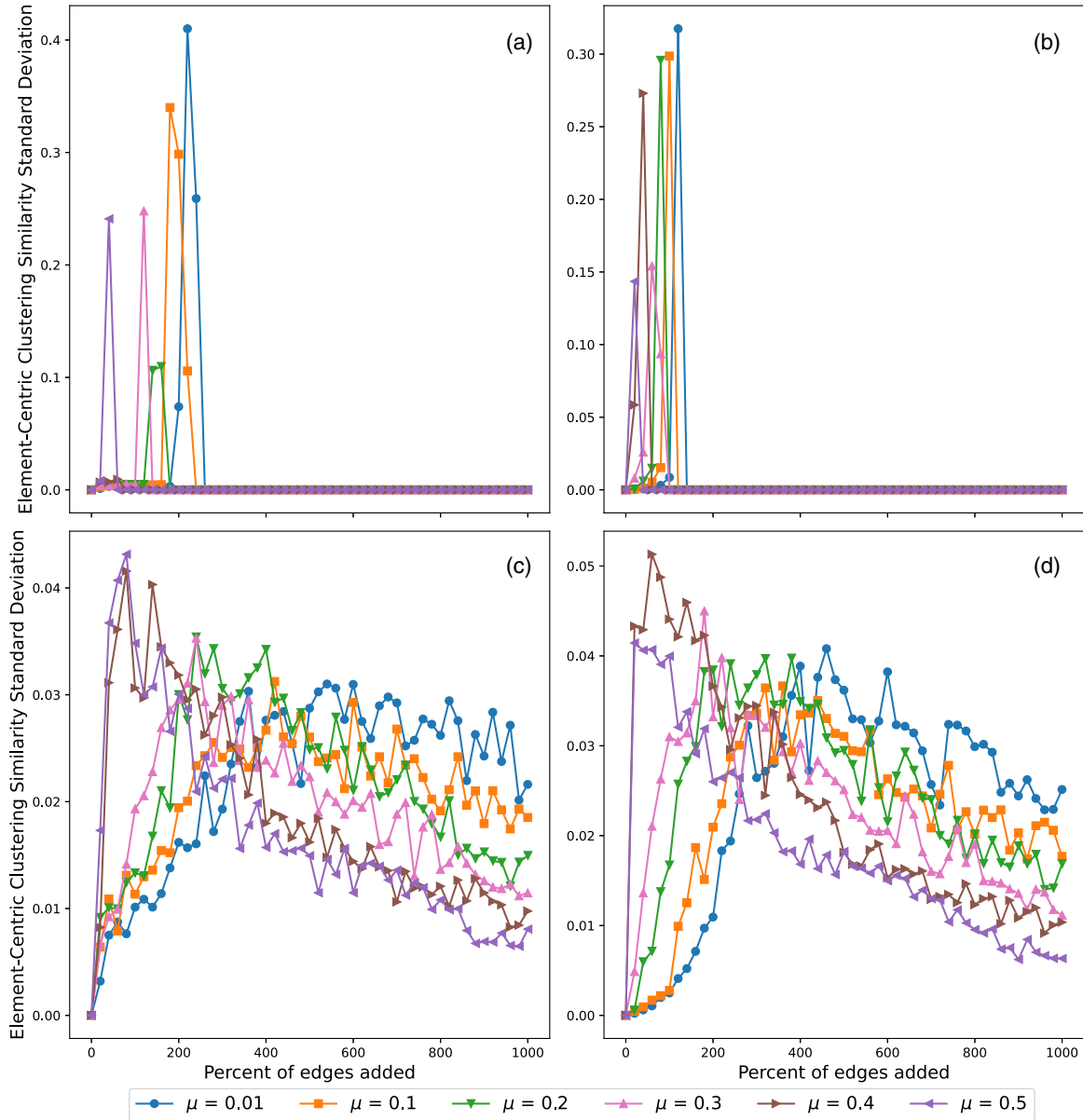


FIG. 22. Standard deviation of element-centric clustering similarity over the percentage of edges added uniformly at random on LFR benchmark graphs with 10 000 nodes. Communities detected by (a) Infomap, (b) Label Propagation, (c) Leiden, and (d) Louvain.

M.T. acknowledges support from the National Science Foundation Mathematical Sciences Graduate Internship program. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript. We also thank Björn Sandstedt for providing critical feedback on the manuscript and Matthew T. Harrison and Ramakrishnan Kannan for providing inspirational suggestions to the project over our conversations.

APPENDIX A: SYNTHETIC RESULTS USING ELEMENT-CENTRIC CLUSTERING SIMILARITY METRIC

Figures 11 and 12 show the results of using the element-centric clustering similarity metric for the LFR benchmark graphs with 1000 nodes and 10 000 nodes, respectively, with

added edges selected uniformly at random from all non-existent edges while multiedges are prohibited. Notably, the parameter values are chosen the same as in experiments that use NMI, and the experimental procedure follows Sec. IID 1. Figures 13 and 14 show the LFR benchmark graphs with 1000 nodes and 10 000 nodes, respectively, with additional edges selected uniformly at random but restricted to ones that cross different communities.

Results in element-centric clustering similarity agree with the observations from results in NMI. Networks with stronger initial partitions, specifically those with lower μ values, are relatively more robust in the community structure. Targeted edge addition can more quickly destroy the original community structure. Also, community robustness is highly dependent on the chosen community detection algorithms. In particular, Leiden and Louvain can detect communities

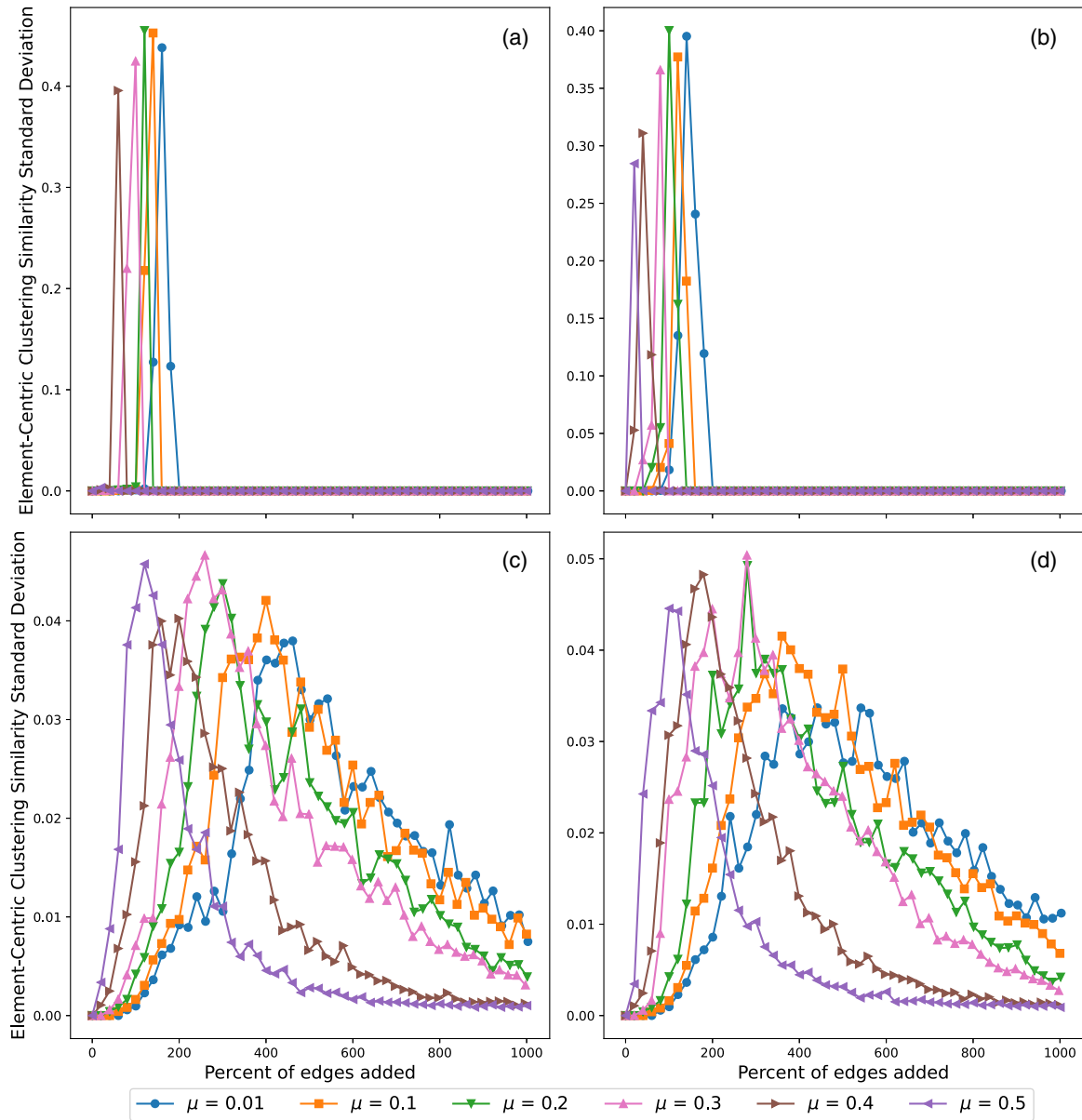


FIG. 23. Standard deviation of element-centric clustering similarity over the percentage of edges added that are selected uniformly at random across different communities on LFR benchmark graphs with 1000 nodes. Communities detected by (a) Infomap, (b) Label Propagation, (c) Leiden, and (d) Louvain.

similar to the initial ground truth as the graphs increase the number of edges significantly.

APPENDIX B: SYNTHETIC RESULTS MATCHING RATIO OF INTERCOMMUNITY EDGES IN EMPIRICAL EXPERIMENTS

Figures 15 and 16 show mean community similarity metrics, NMI and element-centric clustering similarity respectively, over percentage of edges added on LFR benchmark graphs with 1000 nodes. Here, at each step, we force 94% of the added edges to be randomly selected from the pool of intercommunity nonexistent edges. This ratio matches the proportion we found for the intercommunity edges added of the total additional edges in the ia-radoslaw-email subnetwork. Specifically, since we have four community detection

methods and each of them has $np = 20$ initial communities found by the fast consensus algorithm, we have 80 corresponding ratios and then we take the mean as the estimate.

These results show similar behaviors as those observed in the synthetic experiments: A lower mixing parameter, μ , tends to have a more robust community structure, and among the four community detection algorithms, Leiden and Louvain can detect communities more similar to the initial ground truth.

APPENDIX C: STANDARD DEVIATION FOR SYNTHETIC RESULTS

Figures 17–24 show the standard deviation of NMI and element-centric clustering similarity metric for the syn-

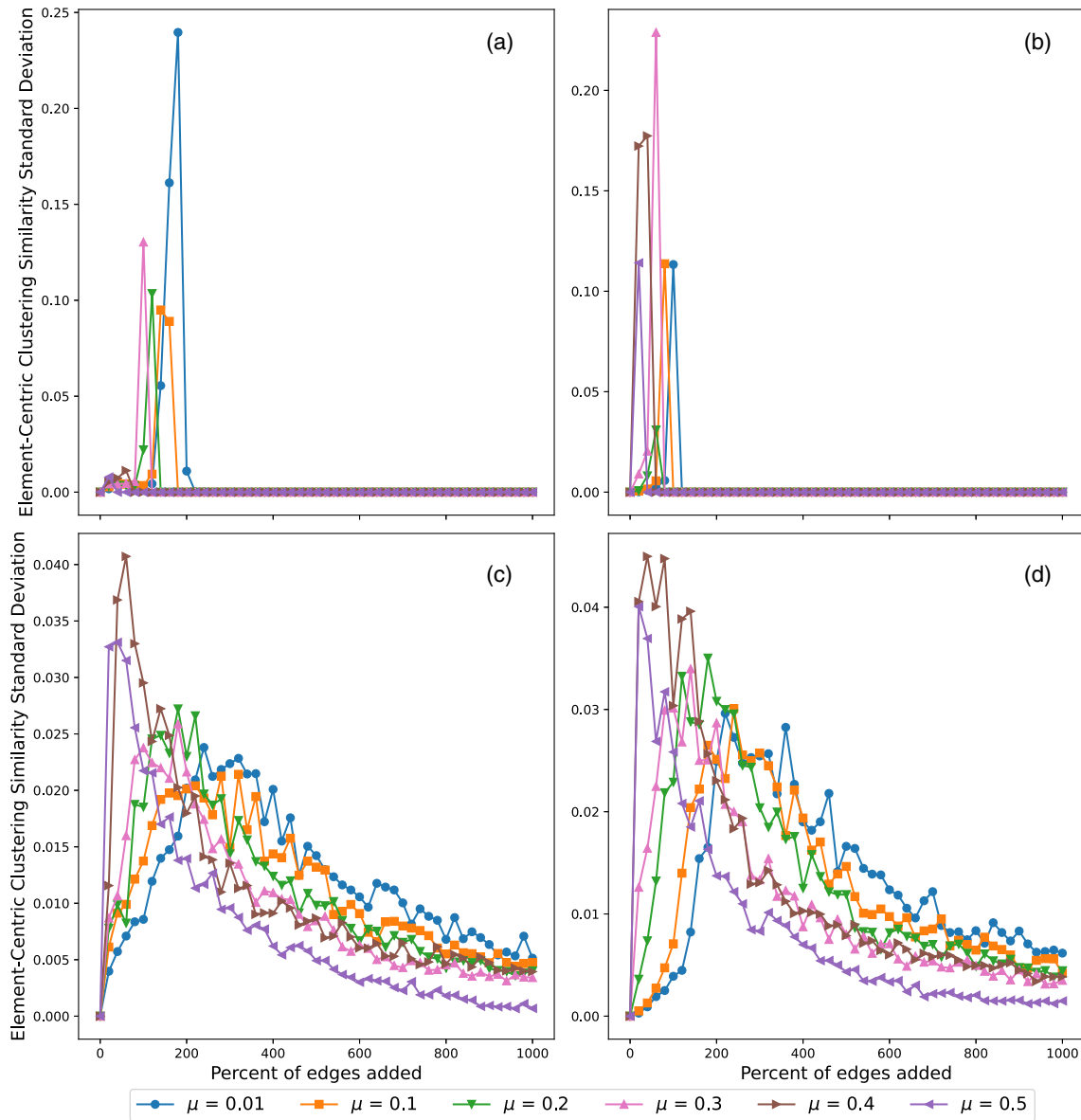


FIG. 24. Standard deviation of element-centric clustering similarity over the percentage of edges added that are selected uniformly at random across different communities on LFR benchmark graphs with 10 000 nodes. Communities detected by (a) Infomap, (b) Label Propagation, (c) Leiden, and (d) Louvain.

thetic results. Note that the overall changes in the standard deviation are generally restricted to a region with negligible scale compared with the mean. However, the only exception is for Infomap (a) and Label Propagation (b) where we observe a spike for almost every curve within the region when less than $2\times$ the original number of edges are added. Further investigation of the mean and the distributions reveals that these locations with large standard deviations correspond to the steps where the rapid drops in the means happen. Specifically, values of the community similarity metric at these steps are split into two families with comparable size—one with 0s and the other with values very close to 1. Recall that at each step,

there are 50 independent realizations, meaning 50 different perturbed graphs, so we suspect this is the uncertainty from the edge-addition stochastic process.

APPENDIX D: STANDARD DEVIATION FOR EMPIRICAL RESULTS

Figures 25–30 show the standard deviation of NMI and element-centric clustering similarity metric for the empirical results. Note that the changes in standard deviation over steps are always restricted to a small region and the values are kept in scales that are negligible compared to the mean.

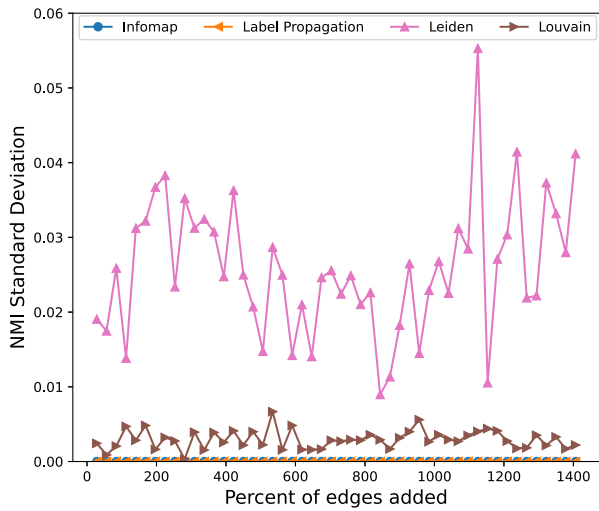


FIG. 25. Standard deviation of NMI over the percentage of edges added for the ia-radoslaw-email subnetwork.

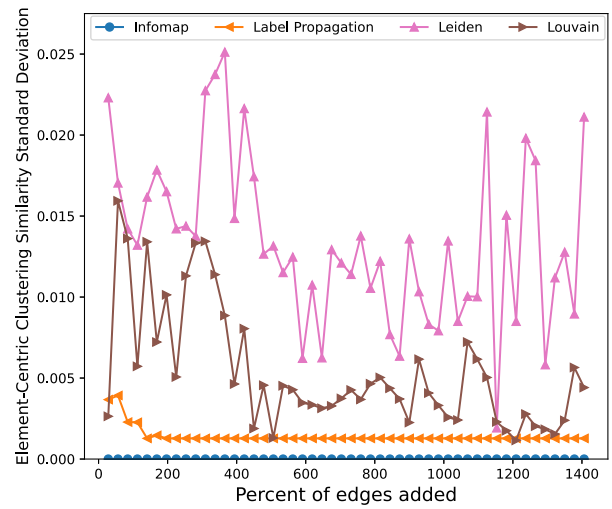


FIG. 28. Standard deviation of element-centric clustering similarity over the percentage of edges added for the ia-radoslaw-email subnetwork.

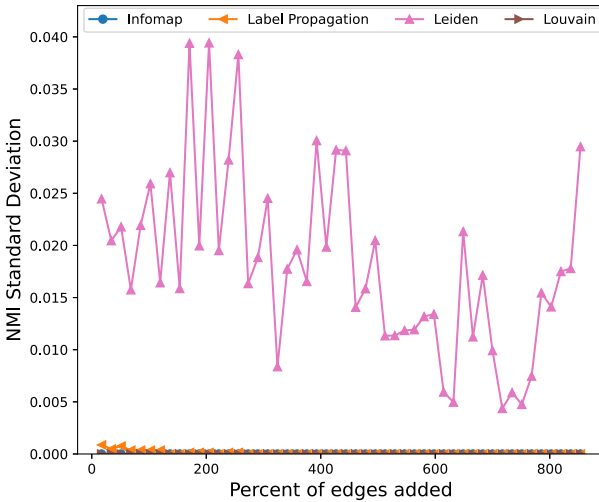


FIG. 26. Standard deviation of NMI over the percentage of edges added for the Enron subnetwork.

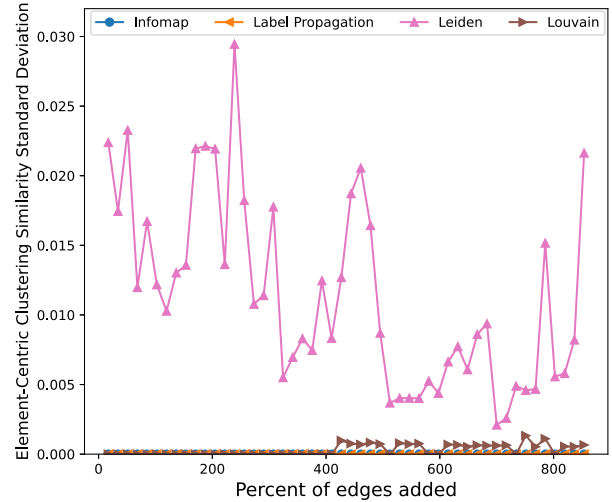


FIG. 29. Standard deviation of element-centric clustering similarity over the percentage of edges added for the Enron subnetwork.

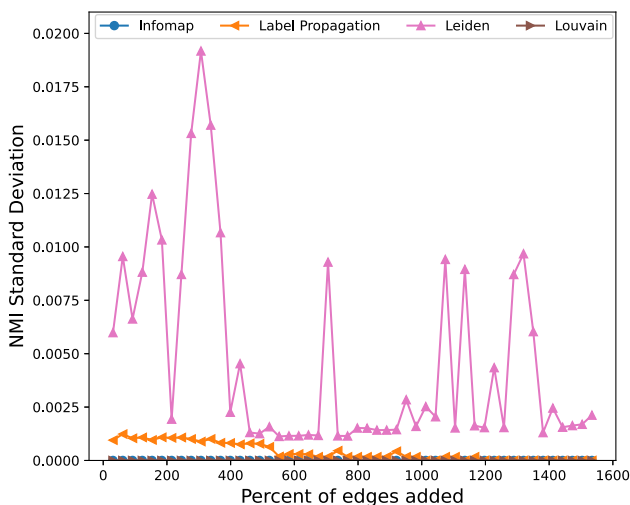


FIG. 27. Standard deviation of NMI over the percentage of edges added for the email-Eu-core-temporal subnetwork.

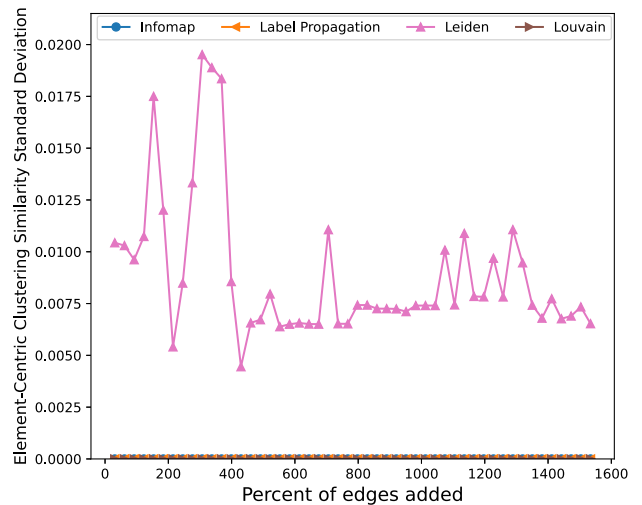


FIG. 30. Standard deviation of element-centric clustering similarity over the percentage of edges added for the email-Eu-core-temporal subnetwork.

- [1] L. A. N. Amaral and J. M. Ottino, *Eur. Phys. J. B* **38**, 147 (2004).
- [2] M. E. J. Newman, *SIAM Rev.* **45**, 167 (2003).
- [3] M. Girvan and M. E. J. Newman, *Proc. Natl. Acad. Sci. USA* **99**, 7821 (2002).
- [4] S. Fortunato, *Phys. Rep.* **486**, 75 (2010).
- [5] S. Fortunato and D. Hric, *Phys. Rep.* **659**, 1 (2016).
- [6] B. Karrer, E. Levina, and M. E. J. Newman, *Phys. Rev. E* **77**, 046119 (2008).
- [7] A.-L. Barabási and M. Pósfai, *Network Science* (Cambridge University Press, Cambridge, UK, 2016).
- [8] S. E. Schaeffer, V. Valdés, J. Figols, I. Bachmann, F. Morales, J. Bustos-Jiménez, and E. Estrada, *J. Complex Netw.* **9**, cnab018 (2021).
- [9] S. Wang and J. Liu, *IEEE Syst. J.* **13**, 582 (2019).
- [10] R. Albert, H. Jeong, and A.-L. Barabási, *Nature (Lond.)* **406**, 378 (2000).
- [11] D. R. Amancio, O. N. Oliveira, and L. da F Costa, *J. Stat. Mech.: Theory Exp.* (2015) P03003.
- [12] S. Wang, J. Liu, and X. Wang, *J. Stat. Mech.: Theory Exp.* (2017) 043405.
- [13] A. Carissimo, L. Cuttillo, and I. De Feis, *Comput. Stat. Data Anal.* **120**, 1 (2018).
- [14] M. Mozafari and M. Khansari, *J. Complex Netw.* **7**, 838 (2019).
- [15] J. Leskovec, J. M. Kleinberg, and C. Faloutsos, in *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, KDD '05 (ACM, New York, 2005), pp. 177–187.
- [16] P. Holme, B. J. Kim, C. N. Yoon, and S. K. Han, *Phys. Rev. E* **65**, 056109 (2002).
- [17] S. P. Borgatti, K. M. Carley, and D. Krackhardt, *Soc. Netw.* **28**, 124 (2006).
- [18] D. J. Wang, X. Shi, D. A. McFarland, and J. Leskovec, *Soc. Netw.* **34**, 396 (2012).
- [19] S. T. Zargar, J. Joshi, and D. Tipper, *IEEE Commun. Surv. Tutor.* **15**, 2046 (2013).
- [20] O. Osanaiye, K.-K. R. Choo, and M. Dlodlo, *J. Netw. Comput. Appl.* **67**, 147 (2016).
- [21] A. Lancichinetti and S. Fortunato, *Phys. Rev. E* **80**, 016118 (2009).
- [22] L. N. F. Ana and A. K. Jain, in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003* (IEEE, Piscataway, NJ, 2003), p. II.
- [23] A. J. Gates, I. B. Wood, W. P. Hetrick, and Y.-Y. Ahn, *Sci. Rep.* **9**, 8574 (2019).
- [24] A. Lancichinetti and S. Fortunato, *Phys. Rev. E* **80**, 056117 (2009).
- [25] S. Emmons, S. Kobourov, M. Gallant, and K. Börner, *PLoS One* **11**, e0159161 (2016).
- [26] A. Lancichinetti, S. Fortunato, and F. Radicchi, *Phys. Rev. E* **78**, 046110 (2008).
- [27] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, *Nature (Lond.)* **435**, 814 (2005).
- [28] A. Clauset, M. E. J. Newman, and C. Moore, *Phys. Rev. E* **70**, 066111 (2004).
- [29] R. Guimerà, L. Danon, A. Díaz-Guilera, F. Giralt, and A. Arenas, *Phys. Rev. E* **68**, 065103(R) (2003).
- [30] P. W. Holland, K. B. Laskey, and S. Leinhardt, *Soc. Netw.* **5**, 109 (1983).
- [31] S. Fortunato, Santo Fortunato's Website: resources, <https://www.santofortunato.net/resources>.
- [32] M. Rosvall and C. T. Bergstrom, *Proc. Natl. Acad. Sci. USA* **105**, 1118 (2008).
- [33] U. N. Raghavan, R. Albert, and S. Kumara, *Phys. Rev. E* **76**, 036106 (2007).
- [34] V. A. Traag, L. Waltman, and N. J. van Eck, *Sci. Rep.* **9**, 5233 (2019).
- [35] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, *J. Stat. Mech.: Theory Exp.* (2008) P10008.
- [36] Z. Yang, R. Algesheimer, and C. J. Tessone, *Sci. Rep.* **6**, 30750 (2016).
- [37] A. Lancichinetti and S. Fortunato, *Sci. Rep.* **2**, 336 (2012).
- [38] V. A. Traag, Github repository: Networkanalysis, <https://github.com/CWTSLeiden/networkanalysis>.
- [39] A. Tandon, A. Albeshri, V. Thayanathan, W. Alhalabi, F. Radicchi, and S. Fortunato, *Phys. Rev. E* **103**, 022316 (2021).
- [40] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, *J. Mach. Learn. Res.* **12**, 2825 (2011).
- [41] A. J. Gates and Y.-Y. Ahn, *J. Open Source Softw.* **4**, 1264 (2019).
- [42] L. Peel, D. B. Larremore, and A. Clauset, *Sci. Adv.* **3**, e1602548 (2017).
- [43] The code for our experiments is publicly available at <http://github.com/Moyi-Tian/CommunityRobustness>.
- [44] A. Tandon, A. Albeshri, V. Thayanathan, W. Alhalabi, and S. Fortunato, *Phys. Rev. E* **99**, 042301 (2019).
- [45] R. Rossi and N. Ahmed, Ia-radoslaw-email, <https://networkrepository.com/ia-radoslaw-email.php>.
- [46] J. Kunegis, Enron, <http://konect.cc/networks/enron>.
- [47] J. Leskovec, email-eu-core temporal network, <http://snap.stanford.edu/data/email-Eu-core-temporal.html>.