Journal of the Franklin Institute 362 (2025) 107404

Contents lists available at ScienceDirect

Journal of the Franklin Institute

journal homepage: www.elsevier.com/locate/fi

Generic network sparsification via hybrid edge sampling

Zhen Su [a,b], Jürgen Kurths [a,c], Henning Meyerhenke [b,*]

[a] *Potsdam Institute for Climate Impact Research, Potsdam, Germany*
[b] *Department of Computer Science, Humboldt-Universität zu Berlin, Berlin, Germany*
[c] *Department of Physics, Humboldt-Universität zu Berlin, Berlin, Germany*

ARTICLE INFO

Keywords:
Graph sparsification
Edge sampling
Hybrid sampling
Triads

ABSTRACT

Network (or graph) sparsification benefits downstream graph mining tasks. Finding a sparsified subgraph $\hat{G}$ similar to the original graph $\mathcal{G}$ is, however, challenging due to the requirement of preserving various (or at least representative) network properties. In this paper, we propose a general hybrid edge sampling scheme named LOGA, as the combination of the Local-filtering-based Random Edge sampling (LRE) (Hamann et al., 2016) and the Game-theoretic Sparsification with Tolerance (GST) (Su et al., 2022). LOGA fully utilizes the advantages of GST — in preserving complex structural properties by preserving local node properties in expectation – and LRE – in preserving the connectivity of a given network. Specifically, we first prove the existence of multiple equilibria in GST. This insight leads us to propose LOGA and its variant LOGA^{sc} by refining GST. LOGA is obtained by regarding LRE as an empirically good initializer for GST, while LOGA^{sc} is obtained by further including a constrained update for GST. In this way, LOGA/LOGA^{sc} generalize the work on GST to graphs with weights and different densities, without increasing the asymptotic time complexity. Extensive experiments on 26 weighted and unweighted networks with different densities demonstrate that LOGA^{sc} performs best for all 26 instances, i.e., they preserve representative network properties better than state-of-the-art sampling methods alone.

1. Introduction

Networks $\mathcal{G} = (V, E, W)$ (= graphs, we use both terms interchangeably) have been a prevalent data representation form. In practice, it can be computationally demanding to analyze large networks with an average degree in the order of hundreds or thousands. One common solution to speed up graph analyses is to use sparsification — removing a significant proportion of possibly redundant edges of $\mathcal{G}$ without the aggregation of nodes. $\mathcal{G}$ is therefore compressed into a sparser graph $\hat{G}$ by sparsification. To obtain a meaningful $\hat{G}$, sparsification requires the preservation of structural properties of $\mathcal{G}$ in $\hat{G}$ in a scaled manner. When doing so, downstream graph mining tasks can benefit from sparsification, both regarding speed and quality [1–3]. For example, by sparsification, important edges can be identified and used for graph representation learning [4]; other relevant applications include visualization [2] and influence maximization [5].

Ideally, $\hat{G}$ should be sufficiently similar (in a scaled manner) to $\mathcal{G}$, so that $\hat{G}$ can be used in place of $\mathcal{G}$ for various applications as mentioned above. However, this is a non-trivial problem due to the need to preserve various structural properties of $\mathcal{G}$. A pragmatic solution is to preserve *representative* ones (see Fig. 1). By doing so, we can expect other properties to be preserved to some extent, due to the correlations between different properties [6–9].

[*] Corresponding author at: Department of Computer Science, Humboldt-Universität zu Berlin, Berlin, Germany.
E-mail address: meyerhenke@hu-berlin.de (H. Meyerhenke).

https://doi.org/10.1016/j.jfranklin.2024.107404
Received 19 June 2024; Received in revised form 28 September 2024; Accepted 10 November 2024

Available online 16 November 2024

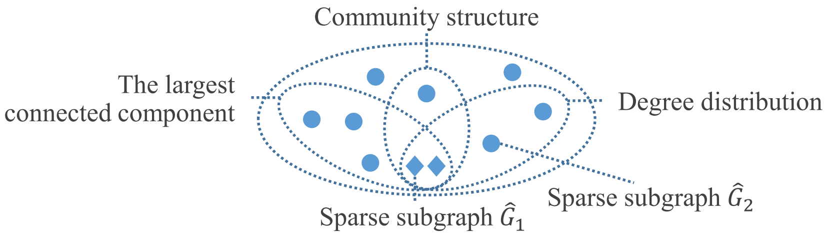

Community structure
The largest connected component
Degree distribution
Sparse subgraph $\hat{G}_1$
Sparse subgraph $\hat{G}_2$

Fig. 1. A schematic view of sparse subgraphs $\hat{G}$ preserving structural properties of a given graph $\mathcal{G}$. Sparse subgraphs represented in shapes have the same number of edges. Assume that the largest connected component, community structure, and degree distribution are *representative* structural properties to be preserved. Some $\hat{G}$ share a relatively higher similarity to $\mathcal{G}$ in terms of community structure, while others preserve well the degree distribution. Compared with all others, $\hat{G}_1$ and $\hat{G}_2$ in diamond are ideal choices for well-preserving the selected *representative* properties.

When doing sparsification, time consumption is an important aspect to consider. For this, edge sampling methods using local structural information are often preferred [10–12]. Still, preserving a set of representative properties by edge sampling is also non-trivial, because it is hard to define an appropriate sampling objective characterizing well the selected representative properties. A practical option is to combine different edge sampling methods.

Therefore, we propose a hybrid sampling scheme called LOGA, as a combination of a well-known edge-focused sampling, i.e., the local-filtering-based random edge sampling (LRE) [10], and the state-of-the-art node-focused sampling, i.e., the game-theoretic sparsification with tolerance (GST) [13]. LRE applies a local filtering post-processing step to random sampling, emphasizing the preservation of the largest connected component; GST, in turn, is motivated by the fact that local structural characteristics enable defining the basic and global organization of a network [14,15]. This combination is motivated by two drawbacks of GST: *initialization dependency* and *unconstrained update*. The initialization dependency arises due to the possible existence of multiple Nash equilibria in an exact network potential game [16–18]; this is also true for GST under assumptions (see Lemma 1). The unconstrained update means that GST proceeds based solely on the optimization objective. Consequently, representative properties not characterized by the optimization objective or not specified in the sparsification process will not be properly preserved. Mainly due to the two drawbacks, GST has limited sparsification performance in networks with weights and different densities (see preliminary studies in Figs. 5(a) and 6(a)).

The paper is organized as follows. Section 2 reviews the related work on edge sampling and exact potential games. The proposed hybrid edge sampling scheme is explained in Section 3. Section 4 presents the experimental evaluation, and Section 5 concludes this paper.

1.1. Contributions

Thus, our contributions are as follows:

- We propose a hybrid edge sampling scheme LOGA and its variant LOGA^{sc} for graph sparsification. Specifically, LOGA improves the initialization of GST by providing GST with an empirically good sparse G^0 subgraph using LRE. LOGA^{sc} improves GST further by including a constrained update for GST, i.e., preserving the largest connected component and the weighted average clustering coefficient based on G^0.
- LOGA^{sc} preserves representative properties better than the state-of-the-art sampling methods for functional climate, real-world, and synthetic networks (on average).
- We recommend using LOGA$^{sc}_{2,3,w}$ in practice, in which subscripts '2', '3', and 'w' represent that GST preserves the expected degrees of nodes, the expected number of triangles (i.e., closed wedges), and the expected number of non-closed wedges associated with nodes.

2. Related work

Edge sampling methods for graph sparsification can be classified into two categories: edge-focused and node-focused ones. We review both of them. We also provide the necessary background on exact potential games, as our sampling scheme is based on GST which adapts a game-theoretic sampling [19].

Edge-focused sampling. Typical edge sampling methods (the probability-based and filtering-based ones) are edge-focused because they use properties associated with edges for sampling.

- *Probability-based sampling* samples edges based on a given edge probability distribution. For example, uniform sampling samples edges uniformly and independently at random [20]; despite simplicity, it preserves spectral properties with high probability [21]. When sampling for graph sparsification, spectral properties such as eigenvalue spectra are of high interest since they contain global information on both graph topology and dynamical properties [22]. Le [11] proposed a non-uniform sampling which samples edge $e = \{u, v\}$ with probability inversely proportional to the number of common neighbors of u and v.

Table 1
List of symbols.

Symbol	Definition
$\mathcal{G} = (V, E, W)$	An undirected and weighted graph with V, E, and W as vertex, edge, and weight sets, respectively
$\mathbf{A}, \mathbf{W}$	The unweighted and weighted adjacency matrices of $\mathcal{G}$
$l \in \{2, 3, w\}$	The basic local properties associated with each node to be preserved, i.e., $l \in \{2\}$ for the degree, $l \in \{3\}$ for triangles (closed wedges), and $l \in \{w\}$ for non-closed wedges; by following Ref. [13], we particularly discuss the combinations of them, i.e., $l \in \{2, 3\}$ and $l \in \{2, 3, w\}$
p	The uniform and independent sampling probability $p \in (0, 1]$
$\mathbb{E}_l^i$	The expected degree ($\mathbb{E}_2^i$) of node i, the expected number of triangles ($\mathbb{E}_3^i$), and the expected number of non-closed wedges ($\mathbb{E}_w^i$) associated with the node i, based on $\mathcal{G}$ and p
$G^0 = (V, E^0, W^0)$	The initialized sparse subgraph by LRE, based on $\mathcal{G}$ and p
$G' = (V, E', W')$	The current subgraph during edge sampling with V, $E' \subseteq E$, and $W' \subseteq W$ as vertex, edge, and weight sets, respectively
$\hat{G} = (V, \hat{E}, \hat{W})$	The desired sparse subgraph after edge sampling with V, $\hat{E} \subseteq E$, and $\hat{W} \subseteq W$ as vertex, edge, and weight sets, respectively
$m_l^i(\cdot)$	The degree ($m_2^i(\cdot)$) of node i, the number of triangles ($m_3^i(\cdot)$), and the number of non-closed wedges ($m_w^i(\cdot)$) associated with the node i, based on a given graph; for example, $m_2^i(\mathcal{G})$, $m_3^i(\mathcal{G})$, and $m_w^i(\mathcal{G})$ are for $\mathcal{G}$, while $m_2^i(G')$, $m_3^i(G')$, and $m_w^i(G')$ are for G'
$s(\cdot), \bar{c}(\cdot)$	The size of the largest connected component ($s(\cdot)$) and the weighted average clustering coefficient ($\bar{c}(\cdot)$) of a given graph; for example, $s(G^0)$ and $\bar{c}(G^0)$ are for G^0, while $s(G')$ and $\bar{c}(G')$ are for G'
$\text{LOGA}_{2,3,w}$	The detailed algorithm of the proposed hybrid sampling scheme LOGA, as the combination of the Local-filtering-based Random Edge sampling (LRE) [10] and the Game-theoretic Sparsification with Tolerance (GST) [13]. By default, it preserves degrees, triangles, and non-closed wedges in expectation, i.e., $l \in \{2, 3, w\}$; the preservation of subset properties leads to $\text{LOGA}_{2,3}$, i.e., $l \in \{2, 3\}$
$\text{LOGA}_{2,3,w}^{sc}$	The detailed algorithm of the variant LOGA^{sc}. By default, it preserves degrees, triangles, and non-closed wedges in expectation, i.e., $l \in \{2, 3, w\}$; the preservation of subset properties leads to $\text{LOGA}_{2,3}^{sc}$, i.e., $l \in \{2, 3\}$

- *Filtering-based sampling*, as in Ref. [10], includes two primary steps: edge scoring and filtering. Specifically, according to a pre-defined scoring method based on some network properties, edge scoring assigns each edge a value to describe its importance; filtering then removes all edges with scores below a certain threshold until the desired sparsification ratio is satisfied. Hamann et al. [10] have compared systematically typical well-known filtering-based sampling methods, including random edge, local Jaccard similarity [23], edge forest fire [24], Simmelian backbones [25], and algebraic distance [26]. By including an additional local-filtering step for these sampling methods, the preservation of properties considered in their work is improved. They also proposed the local-degree-based sampling emphasizing the preservation of the largest connected component.

Edge-focused sampling depends heavily on an appropriate way to define properties associated with edges. To relax such a dependency, a node-focused sampling is proposed.

Node-focused sampling. It is known that local structural characteristics can define the basic and global organization of a network [14,15]. Specifically, one can reproduce important properties in real-world networks, by constraining the averaged degree, degree distribution, degree correlations, and clustering in random graphs to values observed in those real-world networks. Motivated by this, node-focused sampling focuses on local properties associated with nodes, and formulates graph sparsification as an optimization problem [13,27–29]. In particular, based on Refs. [19,27,28,30], Su et al. [13,29] have recently generalized this idea for graph sparsification by proposing the game-theoretic sparsification with tolerance (GST, the foundation for Algorithm 1). As one of the state-of-the-art sparsification methods, GST always converges to a Nash equilibrium. This is because GST constitutes an exact potential game and the best-response dynamics over such a game ensures the convergence [13,19,31]. Still, two drawbacks of GST, i.e., initialization dependency and unconstrained update (see Section 1), to a large extent limit its performance of GST in networks with weights and different densities. We remedy the two drawbacks when proposing the hybrid sampling.

Background on exact potential games. A strategic game is a triplet $\langle P, \{S_p\}_{p \in P}, \{C_p(S_p, S_{-p})\}_{p \in P} \to \mathbb{R} \rangle$ consisting of players P, the strategy S_p of a player $p \in P$, and the individual cost C_p of the player p. The game proceeds in a round-robin fashion after assigning each player a strategy as the initialization. Based on the *best-response dynamics* [31], in every round, each player p minimizes its cost C_p based on all other players' strategies S_{-p}; when the gain $g(p)$ is positive, i.e., $g(p) = C_p(S_p, S_{-p}) - C_p(S'_p, S_{-p}) > 0$, the current strategy S_p is updated to a new one S'_p. If no player has an incentive to change the current strategy, then the strategic game reaches a *(pure) Nash equilibrium*. The strategic game is said to be a *potential game* if all players' incentives to change their strategies can be formulated using a single global function called the potential function Φ. Furthermore, if the gain in the cost function is reflected in the potential function, i.e., $C_p(S_p, S_{-p}) - C_p(S'_p, S_{-p}) = \Phi(S_p, S_{-p}) - \Phi(S'_p, S_{-p})$, the potential game is called *exact*. Most importantly, an exact potential game always converges to a Nash equilibrium due to the best-response dynamics, regardless of the initialization [31].

3. Proposed sparsification method

Given an undirected and weighted graph $\mathcal{G} = (V, E, W)$, graph sparsification aims to find a subgraph $\hat{G} = (V, \hat{E}, \hat{W})$ which preserves certain representative properties. The proposed hybrid sampling LOGA[1] improves GST via initialization improvement

[1] Code available at: https://github.com/Zsstarry/Network-Sparsification-via-Hybrid-Edge-Sampling.

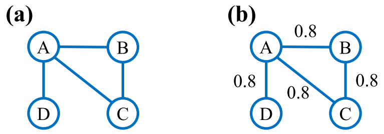


Fig. 2. An example of computing local node properties and their expectations associated with A. (a) The given graph $\mathcal{G}$. The degree of A, the number of triangles, and the number of non-closed wedges associated with A, are $m_2^A(\mathcal{G}) = 3$, $m_3^A(\mathcal{G}) = 1$ (i.e., $\{\{A,B\},\{A,C\},\{B,C\}\}$), and $m_w^A(\mathcal{G}) = 2$ (i.e., $\{\{A,B\},\{A,D\}\}$ and $\{\{A,C\},\{A,D\}\}$), respectively. (b) The given graph $\mathcal{G}$ with a uniform and independent sampling probability $p = 0.8$. The expected degree of A, the expected number of triangles, and the expected number of non-closed wedges associated with A, are $\mathbb{E}_2^A = 2.4$, $\mathbb{E}_3^A = 0.512$, and $\mathbb{E}_w^A = 1.408$, respectively.

and a constrained update. We thus describe GST first by following the same assumption – a uniform and independent sampling probability $p \in (0,1]$ for each edge – that applies to GST in Ref. [13]. Note that this probability p controls also the number of edges to be preserved indirectly. Section 3.2 highlights the contributions of this paper. The most common symbols of this work are listed in Table 1.

3.1. Game-theoretic Sparsification with Tolerance (GST)

Instead of sampling edges directly based on the sampling probability p, GST uses it to derive and preserve local node properties (*in expectation*) in the sparsified graph $\hat{G}$. Specifically, for a node i in $\mathcal{G}$, these local properties include the degree $m_2^i(\mathcal{G}) := \sum_{j=1}^{|V|} \mathbf{A}_{ij}$, the number of triangles (closed wedges) $m_3^i(\mathcal{G}) := \frac{1}{2}\sum_{j=1}^{|V|}\sum_{k=1}^{|V|} \mathbf{A}_{ij}\mathbf{A}_{ik}\mathbf{A}_{jk}$, and the number of non-closed wedges $m_w^i(\mathcal{G}) := \frac{1}{2} m_2^i(\mathcal{G})(m_2^i(\mathcal{G}) - 1) - m_3^i(\mathcal{G})$ [32]. Following Refs. [13,19,30], their expected values are defined based on p as: $\mathbb{E}_2^i := p m_2^i(\mathcal{G})$, $\mathbb{E}_3^i := p^3 m_3^i(\mathcal{G})$, and $\mathbb{E}_w^i := \frac{1}{2}p^2 m_2^i(\mathcal{G})(m_2^i(\mathcal{G}) - 1) - p^3 m_3^i(\mathcal{G})$. An example of computing these local node properties and their expectations is presented in Fig. 2. The graph sparsification is therefore formulated as the node-focused sampling:

Definition 1 (*Sparsification Via Scaled Local Node Properties [13]*)**.** Given an undirected and weighted network $\mathcal{G} = (V, E, W)$ and a uniform and independent sampling probability $p \in (0,1]$, find a sparsified subgraph $\hat{G} = (V, \hat{E}, \hat{W})$ such that:

$$\hat{G} := \operatorname*{argmin}_{G' \subseteq \mathcal{G}} \sum_{i \in V} \sum_{l \in \{2,3,w\}} \frac{1}{m_l^i(\mathcal{G})} |m_l^i(G') - \mathbb{E}_l^i|. \tag{1}$$

where the normalization factor $1/m_l^i(\mathcal{G})$ avoids the domination of any single one of these local properties. Eq. (1) preserves by default all three local node properties, i.e., the degree ($l \in 2$), the number of triangles ($l \in 3$), and the number of non-closed wedges ($l \in w$) a node belongs to. That is, each node in $\hat{G}$ has an incentive to make these local properties (in the current sparse subgraph G') as close as possible to their corresponding expectations. By following Ref. [13], we consider the two cases where l iterates (i) only over degrees and triangles ($l \in \{2,3\}$) and (ii) additionally over non-closed wedges ($l \in \{2,3,w\}$).

To find approximate solutions for Eq. (1), Ref. [13] proposed GST which constitutes an exact network potential game (see lines 7-18 in Algorithm 1). The game is a triplet $\langle E, \{S_e\}_{e\in E}, \{C_e(S_e, S_{-e})\}_{e\in E} \to \mathbb{R}\rangle$, where each edge $e = \{u, v\} \in E$ is a player sharing binary strategies: 0 for removal and 1 for preservation ($S_e = \{0,1\}$) and each edge minimizes its cost C_e based on the strategies S_{-e} of all other edges. Ref. [13] defines the cost function as $C_e := \sum_{i\in\mathcal{A}(e)} \sum_{l\in\{2,3,w\}} \frac{1}{m_l^i(\mathcal{G})} |m_l^i(G') - \mathbb{E}_l^i|$ where $\mathcal{A}(e) = \{u\} \cup \{v\} \cup \{z \in V : \{z,u\} \in E' \wedge \{z,v\} \in E'\}$ is the set of nodes *affected* by the strategy change of e. C_e considers only $\mathcal{A}(e)$ because only local subgraphs with up to 3 nodes are considered in Eq. (1). The game proceeds in a round-robin fashion based on the gain $g(e) := C_e(S_e, S_{-e}) - C_e(S'_e, S_{-e})$ in the cost function:

$$g(e) = \sum_{i\in\mathcal{A}(e)} \sum_{l\in\{2,3,w\}} \frac{|m_l^i(G') - \mathbb{E}_l^i| - |m_l^i(G'') - \mathbb{E}_l^i|}{m_l^i(\mathcal{G})}, \tag{2}$$

where e updates its strategy, e.g., from $S_e = 1$ to $S'_e = 0$ with the current subgraph G' correspondingly becoming $G'' = (V, E'') = (V, E' \setminus \{e\})$, only when $g(e) > 0$. The game has a *(pure) Nash equilibrium* when no edge has the incentive to change its current strategy. Meanwhile, a global function $\Phi = \sum_{i\in V}\sum_{l\in\{2,3,w\}} \frac{1}{m_l^i(\mathcal{G})} |m_l^i(G') - \mathbb{E}_l^i|$ (same as Eq. (1)) exists to ensure that GST is a *potential* game. The gain in the potential function is therefore:

$$\Phi(S_e, S_{-e}) - \Phi(S'_e, S_{-e}) := \sum_{i\in V} \sum_{l\in\{2,3,w\}} \frac{1}{m_l^i(\mathcal{G})} \left(|m_l^i(G') - \mathbb{E}_l^i| - |m_l^i(G'') - \mathbb{E}_l^i|\right).$$

In particular, $C_e(S_e, S_{-e}) - C_e(S'_e, S_{-e}) = \Phi(S_e, S_{-e}) - \Phi(S'_e, S_{-e})$ due to $\Phi(S_e, S_{-e}) - \Phi(S'_e, S_{-e}) = 0 \ \ \forall i \in V \setminus \mathcal{A}(e)$, ensuring that GST constitutes an *exact* potential game. The best-response dynamics in the exact potential game guarantees the convergence to a Nash equilibrium [31].

3.2. Hybrid edge sampling scheme

Our proposed hybrid sampling scheme LOGA and its variant LOGA^{sc} is based on the improvements over GST. The improvements include initialization via edge-focused sampling (i.e., LOGA – Algorithms 1 and 2), and a constrained update (i.e., the variant LOGA^{sc} – Algorithms 1, 2, and 3).

Algorithm 1: Hybrid sampling scheme (LOGA / LOGA^{sc}), based on the game-theoretic sparsification with tolerance (GST) [13]

```
Input: $\mathcal{G} = (V, E, W)$, $p \in (0, 1]$, and the tolerance threshold by default $T = 0.01$ (based on Ref. [13])
Output: $G' = (V, E', W')$
                                        /* Stage I (Precomputing expected basic properties) */
for $i \in V$ do in parallel
    Compute $\mathbb{E}^i_2$, $\mathbb{E}^i_3$, $\mathbb{E}^i_w$, $m^i_2(\mathcal{G})$, $m^i_3(\mathcal{G})$, and $m^i_w(\mathcal{G})$, based on $\mathcal{G}$ and $p$
                                        /* Stage II (Sparsification) */
$G', s(G^0), \bar{c}(G^0), \bar{c}(\mathcal{G}) \leftarrow$ Initialization$(\mathcal{G}, p)$
for $i \in V$ do in parallel
    Compute $m^i_2(G')$, $m^i_3(G')$, and $m^i_w(G')$, based on $G'$
$L' \leftarrow V$; $Gain[|V|] \leftarrow 0$; $r \leftarrow 0$
repeat
    $L \leftarrow L'$; $L' \leftarrow \emptyset$
    foreach $e = \{u, v\} \in E$ incident (in $\mathcal{G}$) to a node in $L$ do
        $\mathcal{A}(e) \leftarrow \{u\} \cup \{v\} \cup \{z \in V : \{z, u\} \in E' \wedge \{z, v\} \in E'\}$
        Compute $g(e)$ based on Eq. Eq. (2)
        $G', Flag \leftarrow$ Constrained_Update$(e, G', s(G^0), \bar{c}(G^0), \bar{c}(\mathcal{G}), g(e))$
        if $Flag$ is True then
            $L' \leftarrow L' \cup \mathcal{A}(e)$
            Update $m^i_2(G')$, $m^i_3(G')$, and $m^i_w(G')$, based on $G'$
    $r \leftarrow r + 1$
    $Gain[r] \leftarrow \sum_{i \in V} \sum_{l \in \{2,3,w\}} \frac{1}{m^i_l(\mathcal{G})} |m^i_l(G') - \mathbb{E}^i_l|$
until $r \geq 2 \wedge Gain[r-1] - Gain[r] \leq T$
return $G'$
```

Algorithm 2: Initialization

```
Input: $\mathcal{G} = (V, E, W)$ and $p \in (0, 1]$
Output: $G^0 = (V, E^0, W^0)$, $s(G^0)$, $\bar{c}(G^0)$
Generate a sparse subgraph $G^0 = (V, E^0, W^0)$ with $E^0 = p|E|$ using LRE [33]
Compute $s(G^0)$, $\bar{c}(G^0)$, and $\bar{c}(\mathcal{G})$
return $G^0$, $s(G^0)$, $\bar{c}(G^0)$, $\bar{c}(\mathcal{G})$
```

Algorithm 3: Constrained_Update

```
Input: An edge $e = \{u, v\}$ currently being visited, $G' = (V, E', W')$, $s(G^0)$, $\bar{c}(G^0)$, $\bar{c}(\mathcal{G})$, and $g(e)$
Output: Whether to update $G' = (V, E', W')$
if $e \in E'$ then  $G'' \leftarrow (V, E' \setminus \{e\})$
else $G'' \leftarrow (V, E' \cup \{e\})$
Compute $s(G'')$ and $\bar{c}(G'')$
if $g(e) > 0 \wedge s(G'') >= s(G^0) \wedge |\bar{c}(G^0) - \bar{c}(\mathcal{G})| >= |\bar{c}(G'') - \bar{c}(\mathcal{G})|$ then  return $G' \leftarrow G''$, True
else return $G'$, False
```

- **Initialization via edge-focused sampling — Algorithms 1 and 2.** The idea of improving the initialization of GST stems from the existence of multiple equilibria in GST, which is not indicated in Ref. [13]; as a result, the solution that GST converges to depends highly on initialization. Motivated by [30, Lemma 2], we prove the existence of multiple equilibria for the case $l \in \{2, 3\}$ of Eq. (1) as below.

Lemma 1. *For*

$$\hat{G} := \underset{G' \subseteq \mathcal{G}}{\operatorname{argmin}} \sum_{i \in V} \sum_{l \in \{2,3\}} \frac{1}{m^i_l(\mathcal{G})} |m^i_l(G') - \mathbb{E}^i_l|,$$

assume a globally optimal sparse subgraph G' exists which contains at least one edge $e = \{u, v\}$ with u and v satisfying: (a) $\{z \in V : \{z, u\} \in E' \wedge \{z, v\} \in E'\} = \emptyset$; (b) $m^u_2(G') > \mathbb{E}^u_2$ and $m^v_2(G') \leq \mathbb{E}^v_2$ with $\mathbb{E}^u_2, \mathbb{E}^v_2 \in \mathbb{N}$; (c) $m^u_2(\mathcal{G}) = m^v_2(\mathcal{G})$. Then, there exists a globally optimal sparse subgraph $\tilde{G} \neq G'$ representing a Nash equilibrium with $m^i_2(\tilde{G}) \leq \mathbb{E}^i_2$, $\forall i \in V$.

Proof. Since G' is a global optimum, it is a Nash equilibrium. We continue by first proving that $G'' = (V, E'') = (V, E' \backslash \{e\})$ is another global optimum. Condition (a) indicates the non-existence of common neighbors between u and v in G'. Therefore, $\mathcal{A}(e) = \{u, v\}$ is the node set affected by the removal of e from G'; more precisely, only the degrees of u and v, i.e., $m_2^u(G')$ and $m_2^v(G')$, get affected. Thus, the corresponding gain, as in Eq. (2), is simplified as:

$$g'(e) = \frac{|m_2^u(G') - \mathbb{E}_2^u| - |m_2^u(G'') - \mathbb{E}_2^u|}{m_2^u(\mathcal{G})} + \frac{|m_2^v(G') - \mathbb{E}_2^v| - |m_2^v(G'') - \mathbb{E}_2^v|}{m_2^v(\mathcal{G})}.$$

According to condition (b), $m_2^u(G') - \mathbb{E}_2^u > 0$ and $\mathbb{E}_2^u \in \mathbb{N}$; therefore, the removal of e from G' leads to G'' with $m_2^u(G'') - \mathbb{E}_2^u = m_2^u(G') - 1 - \mathbb{E}_2^u \geq 0$. Similarly, we have $m_2^v(G') - \mathbb{E}_2^v \leq 0$ and $m_2^v(G'') - \mathbb{E}_2^v = m_2^v(G') - 1 - \mathbb{E}_2^u < 0$. Therefore, $|m_2^u(G') - \mathbb{E}_2^u| - |m_2^u(G'') - \mathbb{E}_2^u| = 1$ and $|m_2^v(G') - \mathbb{E}_2^v| - |m_2^v(G'') - \mathbb{E}_2^v| = -1$. Further by condition (c), $g'(e) = 1/m_2^u(\mathcal{G}) - 1/m_2^v(\mathcal{G}) = 0$; that is, both G' and G'' are global optima representing different Nash equilibria. According to Ref. [30], by removing all edges satisfying conditions (a), (b), and (c), one constructs a globally optimal sparse subgraph $\tilde{G} \neq G'$ which satisfies $m_2^i(\tilde{G}) \leq \mathbb{E}_2^i$, $\forall i \in V$. Thus, the proof is complete.

Due to the (conditional) existence of more than one global optimum, it is natural to ask how to steer an algorithm to find a good optimum. One established strategy to increase the likelihood of convergence to a good local (or, better yet, global) optimum is to use a good starting solution. In our context, a starting solution that already preserves representative properties reasonably well should serve this purpose. Therefore, we propose LOGA first by using LRE as a good initializer in improving GST (see Algorithm 2), due to the best performance of LRE in Figs. 5(a) and 6(a). Our conjecture of convergence to a better optimum is empirically verified in Section 4.2.

- **Constrained update — Algorithms 1, 2, and 3.** For GST, initialization improvement is not sufficient to ensure that, the representative properties preserved by the initialized sparse subgraph G^0 can still be preserved after sparsification. This is because GST proceeds based only on its optimization objective (see Eq. (2)). As a result, the representative properties not characterized by the optimization objective cannot be preserved properly. Therefore, a constrained update is needed for the sparsification process. To this end, we pay attention to the preservation of the *largest connected component* $s(\cdot)$ and the *weighted average clustering coefficient* $\overline{c}(\cdot)$. These two structural properties characterize the global organization of a graph, and can be used for characterizing network resilience [34,35]. Besides, they are easy to compute. In particular, the largest connected component can be well-preserved by LRE [10] (also see LRE better than RE in Table 5), but not by GST. Thus, we further include the preservation of $s(G^0)$ and $\overline{c}(G^0)$ based on G^0 in Algorithm 3, leading to a variant LOGA^{sc}. Note that, for a given graph like $\mathcal{G}$, we compute the $\overline{c}(\mathcal{G}) = \frac{1}{|V|} \sum_{i=1}^{|V|} \left(\frac{1}{(\sum_{j=1}^{|V|} \mathbf{A}_{ij} - 1) \sum_{j=1}^{|V|} \mathbf{W}_{ij}} \sum_{k=1}^{|V|} \sum_{l \in 1}^{|V|} \frac{\mathbf{W}_{ik} + \mathbf{W}_{il}}{2} \mathbf{A}_{ik} \mathbf{A}_{il} \mathbf{A}_{kl} \right)$ based on Ref. [36].

Time complexity. We assume an adjacency array as the graph data structure. Stage I (see lines 1–2 in Algorithm 1) is dominated by the computation of the number of triangles $m_3^i(\mathcal{G})$ for each node. We use adjacency-marking-based triangle counting [37], which can be implemented to run in $\mathcal{O}(a(\mathcal{G})|E|)$ time [38], with $a(\mathcal{G})$ being the arboricity of $\mathcal{G}$ and being upper-bounded by the maximum degree d_{max}. The initialization (see Algorithm 2) takes $\mathcal{O}(\log(d_{max})|E|)$ for obtaining G^0 by LRE [10], $\mathcal{O}(|V| + |E|)$ for computing the largest connected component $s(\cdot)$, and $\mathcal{O}(a(\mathcal{G})|E|)$ for computing the weighted average clustering coefficient $\overline{c}(\cdot)$. For Stage II, computing the number of triangles $m_3^i(G')$ for each node based on the current subgraph G' dominates Lines 4–6. Lines 7–15 take $\mathcal{O}(rd_{max}|E|)$ time, where r is the number of iterations of the repeat-loop and $\mathcal{O}(d_{max})$ is required by a linear-time intersection operation to find $\mathcal{A}(e)$. Note that computing $\overline{c}(G'')$ takes $\mathcal{O}(d_{max})$ dominating Algorithm 3. Hence, in total, the (sequential) time complexity of LOGA/LOGA^{sc} equals that of GST with $\mathcal{O}(rd_{max}|E|)$, provided that the initialization step (i.e., Algorithm 2) takes at most $\mathcal{O}(rd_{max}|E|)$ time.

4. Experimental evaluation

In this section, we assess the performance of LOGA/LOGA^{sc} by answering:

Q1: How well do LOGA/LOGA^{sc} improve the state-of-the-art sampling methods in terms of preserving non-local/complex representative properties?
Q2: What is the empirical running time of LOGA/LOGA^{sc}, in particular in comparison to GST?

4.1. Experimental settings

Data sets. We consider 13 weighted networks from different domains in Table 2, including functional climate networks, observed real-world networks, and LFR networks. For the same data sets, another 13 networks in Table 3 are constructed to verify the usefulness of LOGA/LOGA^{sc} for unweighted and sparser networks.

- *Functional climate networks.* In Table 2, five functional networks are constructed based on Refs. [39–41]. By randomly choosing 50% edges from the respective networks in Table 2 and letting weights be 1, we obtain unweighted networks with reduced density in Table 3.
- *Observed real-world networks.* In Table 2, five selected real-world networks, from Squirrel to HepTh, describe social and biological relationships and are available publicly online.[2] We also construct unweighted networks with reduced density in Table 3, by randomly choosing 50% edges from the respective networks in Table 2 and letting weights be 1.

[2] http://snap.stanford.edu/ http://konect.cc/networks/, https://string-db.org/.

Table 2
Characteristics of data sets with weighted structure.

Type	Network	$\|V\|$	$\|E\|$	$\frac{\|E\|}{\|V\|}$	Description
Functional climate networks	Glo_ERA5SP	7320	1,559,513	213.05	Global surface pressure from ERA5
	Glo_ERA5ST	7320	2,022,285	276.27	Global surface temperature from ERA5
	Glo_ERA5GPH	7320	1,866,977	255.05	Global 250-hPa geopotential height from ERA5
	Glo_ERA5OLR	7320	845,465	115.50	Global outgoing long-wave radiation from ERA5
	Glo_TRMM	16,080	1,753,588	109.05	Precipitation from TRMM
Observed real-world networks	Squirrel	5201	198,353	38.14	Wikipedia articles on squirrels
	SC	6394	994,296	155.50	Protein network of Saccharomyces cerevisiae
	NIPS	13,875	746,316	53.79	Bipartite document–word dataset of NIPS full papers
	CE	18,387	4,481,664	243.74	Protein network of Caenorhabditis elegans
	HepTh	22,908	2,444,798	106.72	Co-citation network of arXiv's hep-th section
LFR networks	$\text{LFR}_{\mu=0.1}$	10,000	1,238,142	123.81	Synthetic benchmark
	$\text{LFR}_{\mu=0.2}$	10,000	1,255,220	125.52	Synthetic benchmark
	$\text{LFR}_{\mu=0.3}$	10,000	1,250,961	125.10	Synthetic benchmark

Table 3
Same data sets as Table 2 but with weights 1 and reduced density.

Type	Network	$\|V\|$	$\|E\|$	$\frac{\|E\|}{\|V\|}$	Description
Functional climate networks	Glo_ERA5SP	7320	779,756	106.52	Unweighted and reduced network density
	Glo_ERA5ST	7320	1,011,142	138.13	
	Glo_ERA5GPH	7320	933,488	127.53	
	Glo_ERA5OLR	7320	422,732	57.75	
	Glo_TRMM	16,080	876,794	54.53	
Observed real-world networks	Squirrel	5201	99,176	19.07	Unweighted and reduced network density
	SC	6394	497,148	77.75	
	NIPS	13,875	373,158	26.89	
	CE	18,387	2,240,832	121.87	
	HepTh	22,908	1,222,399	53.36	
LFR networks	$\text{LFR}_{\mu=0.1}$	10,000	252,923	25.29	Unweighted and reduced network density
	$\text{LFR}_{\mu=0.2}$	10,000	248,907	24.89	
	$\text{LFR}_{\mu=0.3}$	10,000	253,753	25.38	

- *LFR networks.* In Table 2, three synthetic networks are constructed based on the Lancichinetti–Fortunato–Radicchi (LFR) benchmark [42] implemented in NetworKit [33,43], a tool suite for scalable network analysis. The parameters are as follows: (i) power-law exponents for the degree distribution and the community size distribution: $\tau_1 = -2$ and $\tau_2 = -1$, respectively; (ii) fraction of inter-community edges: $\mu \in \{0.1, 0.2, 0.3\}$; (iii) desired average and maximum degrees: 250 and 1000 (50 and 250 for Table 3), respectively; (iv) minimum and maximum sizes of communities: 250 and 1000 (25 and 250 for Table 3), respectively.

Baselines. We compare LOGA/LOGA^{sc} with two state-of-the-art and four well-known sampling methods. Note that we compare LOGA/LOGA^{sc} and GST indirectly by comparing LOGA/LOGA^{sc} vs {LD, LJS, RE, LRE, and CN} and GST vs {LD, LJS, RE, LRE, and CN}, separately. This is because we want to compare graphs with similar densities, but both LOGA/LOGA^{sc} and GST cannot take directly a sparsification ratio as input while the other methods can. Both CN and LRE are additional competitors not used by Ref. [13].

- *Two state-of-the-art methods.* In Ref. [13], GST takes by default the original graph $\mathcal{G}$ as initialization. The necessity of preserving both degrees and the number of 3-node subgraphs in expectation for graph sparsification has also been confirmed. Therefore, by following Ref. [13], we consider also both $l \in \{2, 3\}$ and $l \in \{2, 3, w\}$. That is, the first comparison is between $\textbf{LOGA}_{2,3}$ / $\textbf{LOGA}_{2,3,w}$ / $\textbf{LOGA}^{sc}_{2,3}$ / $\textbf{LOGA}^{sc}_{2,3,w}$ and $\textbf{GST}_{2,3}$ / $\textbf{GST}_{2,3,w}$. The second competitor by Le [11] samples edges with probability inversely proportional to the number of common neighbors (**CN**) between the two nodes.
- *Four well-known methods.* Four well-known edge-focused sampling methods are: local degree (**LD**) [10], local Jaccard similarity (**LJS**) [23], random edge sampling (**RE**) [21], and the local-filtering based random edge sampling (**LRE**) [10]. Their empirical effectiveness in preserving the overall connectivity (by LD), community structure (by LJS), and eigenvalue distribution (by RE/LRE), are systematically compared in Ref. [10] and implemented in NetworKit [33,43].

Evaluation metrics and procedure. By default, $p \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$ is applied. For **Q1**, we use a three-step evaluation procedure, where the Mann–Whitney U test and shooting score are not used by Ref. [13]:

- *Step I — similarity estimation.* The similarity estimation between the original graph $\mathcal{G}$ and the obtained sparse subgraph G' considers the multi-level *representative* properties. The detailed methods for computing these properties and for similarity estimation are summarized in Table 4, with an example given in Fig. 3.

Table 4
Graph similarity estimation based on selected representative properties.

Representative property	Computation	Similarity estimation
The weighted average clustering coefficient	Ref. [36]	*Deviation* [10]
The size of the largest connected component	NetworKit [33,43]	Deviation
Community structure	PLM [44] in NetworKit	*Adjusted rand index* (ARI) [45]
Betweenness	EstimateBetweenness [46] in NetworKit	*Spearman's* ρ with $P < 0.05$ [10]
Degree	NetworKit	Spearman's ρ with $P < 0.05$
The weighted local clustering coefficient	Ref. [36]	Spearman's ρ with $P < 0.05$
Graph spectra (eigenvalue distribution)	SLAQ_NetLSD [47] using heat kernel [48]	*Euclidean distance* [47,48]
Graph spectra (eigenvalue distribution)	SLAQ_VNGE [47] using Von Neumann Graph Entropy [49]	Euclidean distance

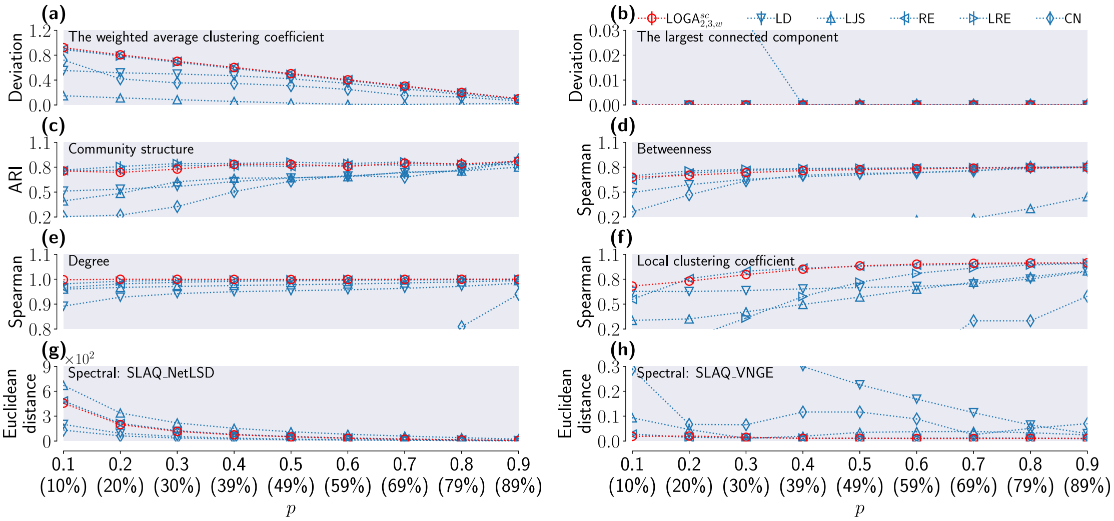


Fig. 3. An example of graph similarity estimation (see Step I of 'Evaluation metrics and procedure' in Section 4.1), in terms of six edge sampling methods (i.e., LOGA$^{sc}_{2,3,w}$, LD, LJS, RE, LRE, and CN) preserving representative structural properties, for Glo_ERA5SP in Table 2. Each sampling probability p on the x-axis is attached with the exact ratio of preserved edges in brackets. The LOGA$^{sc}_{2,3,w}$ is highlighted in red. This figure indicates that, there is no single method well-preserving all these considered representative structural properties, as in Ref. [13].

- *Step II — ranking comparison.* To obtain a conclusive summary, we summarize the performance of different sampling methods by *ranking distribution* and *Mann–Whitney U test*. This is partially motivated by the evaluation procedure of Ref. [24], where different algorithms are evaluated over different data sets and over different evaluation criteria. Specifically, given a sampling probability p, we rank from 1 to 6 for LOGA$_{2,3}$ / LOGA$_{2,3,w}$ / LOGA$^{sc}_{2,3}$ / LOGA$^{sc}_{2,3,w}$ / GST$_{2,3}$ / GST$_{2,3,w}$, LD, LJS, RE, LRE, and CN, in the similarity comparison of each representative property. We then summarize as a distribution all rankings, for each method over different p and over different similarity comparisons of representative properties. The ranking distribution exhibits a large variance since preserving well all selected representative properties is impossible. We, therefore, use the Mann–Whitney U test to classify the six sampling methods into two groups. Group I has better performance, Group II performs worse in comparison, and they satisfy: (1) methods in Group I share the same pair-wise cumulative distribution functions (CDFs) in terms of ranking distributions, given the null hypothesis that two CDFs to be compared are identical and the significance threshold of 0.1; (2) at least one method's CDF in Group I is larger than the CDF of any method in Group II, given the significance threshold of 0.05.
- *Step III — shooting score.* This step computes the number of data sets for which each method has better rankings in *Step II — ranking comparison*. The shooting score is summarized in Table 5.

For **Q2**, we compare LOGA/LOGA^{sc}, GST, LD, LJS, RE, LRE, and CN (see Section 4.2), using an unbiased single-threaded environment. The empirical running time is averaged (arithmetic mean) over 10 runs (sufficient due to small variance) for each given sampling probability × 9 sampling probabilities ($p \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$) × 2 cases ($l \in 2, 3$ and $l \in 2, 3, w$).

4.2. Discussion

An example of the detailed graph similarity estimation is presented in Fig. 3. The results for other graphs are similar. The proposed hybrid sampling method LOGA and its variant LOGA^{sc} share the same optimization objective as GST, and constitute also a network exact potential game. Therefore, LOGA$^{sc}_{2,3,w}$ preserves well the degree distribution (see Fig. 3(e)), just like GST. Meanwhile, LOGA$^{sc}_{2,3,w}$ shares similar preservation of the weighted average clustering coefficient and the approximated eigenvalue distribution,

Table 5
Summary of the performance of sampling methods, i.e., LOGA/LOGA^{sc} vs. {GST, LD, LJS, RE, LRE, and CN}, out of 26 networks in Tables 2 and 3. The shooting score counts the number of hatches for each method based on Figs. 5 and 6.

Method	GST$_{2,3}$	LD	LJS	RE	LRE	CN
Score (Figs. 5(a) and 6(a))	8	14	0	10	**25**	0
Method	GST$_{2,3,w}$	LD	LJS	RE	LRE	CN
Score (Figs. 5(a) and 6(a))	10	13	0	11	**23**	0
Method	LOGA$_{2,3}$	LD	LJS	RE	LRE	CN
Score (Figs. 5(b) and 6(b))	**21**	6	0	6	20	0
Method	LOGA$_{2,3,w}$	LD	LJS	RE	LRE	CN
Score (Figs. 5(b) and 6(b))	**22**	6	0	5	20	0
Method	LOGA$^{sc}_{2,3}$	LD	LJS	RE	LRE	CN
Score (Figs. 5(c) and 6(c))	**23**	4	0	5	18	0
Method	LOGA$^{sc}_{2,3,w}$	LD	LJS	RE	LRE	CN
Score (Figs. 5(c) and 6(c))	**26**	4	0	5	19	0

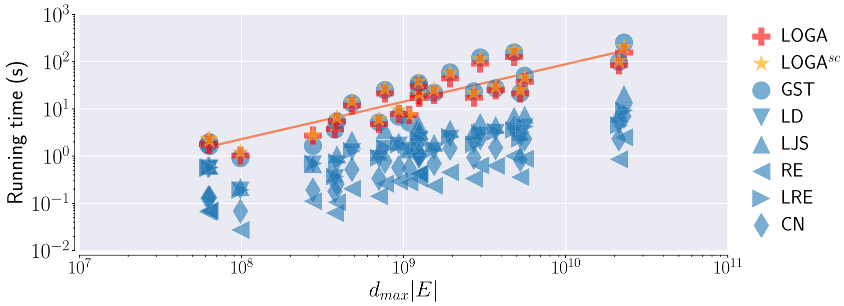


Fig. 4. The average running times of LOGA/LOGA^{sc}, GST, LD, LJS, RE, LRE, and CN for networks in Tables 2 and 3. LOGA/LOGA^{sc} takes a similar empirical running time as GST.

as LRE. This is mainly due to the inclusion of the constrained update in Section 3.2. Still, taking Fig. 3 as an example, comparing different graph similarity estimates one by one is not conclusive, as different sampling methods have diverse performances. We thus summarize Fig. 3 in Fig. 5(c)(b) by using their rankings, and apply the same summarization to all other graphs (see Step II of ‘Evaluation metrics and procedure’ in Section 4.1).

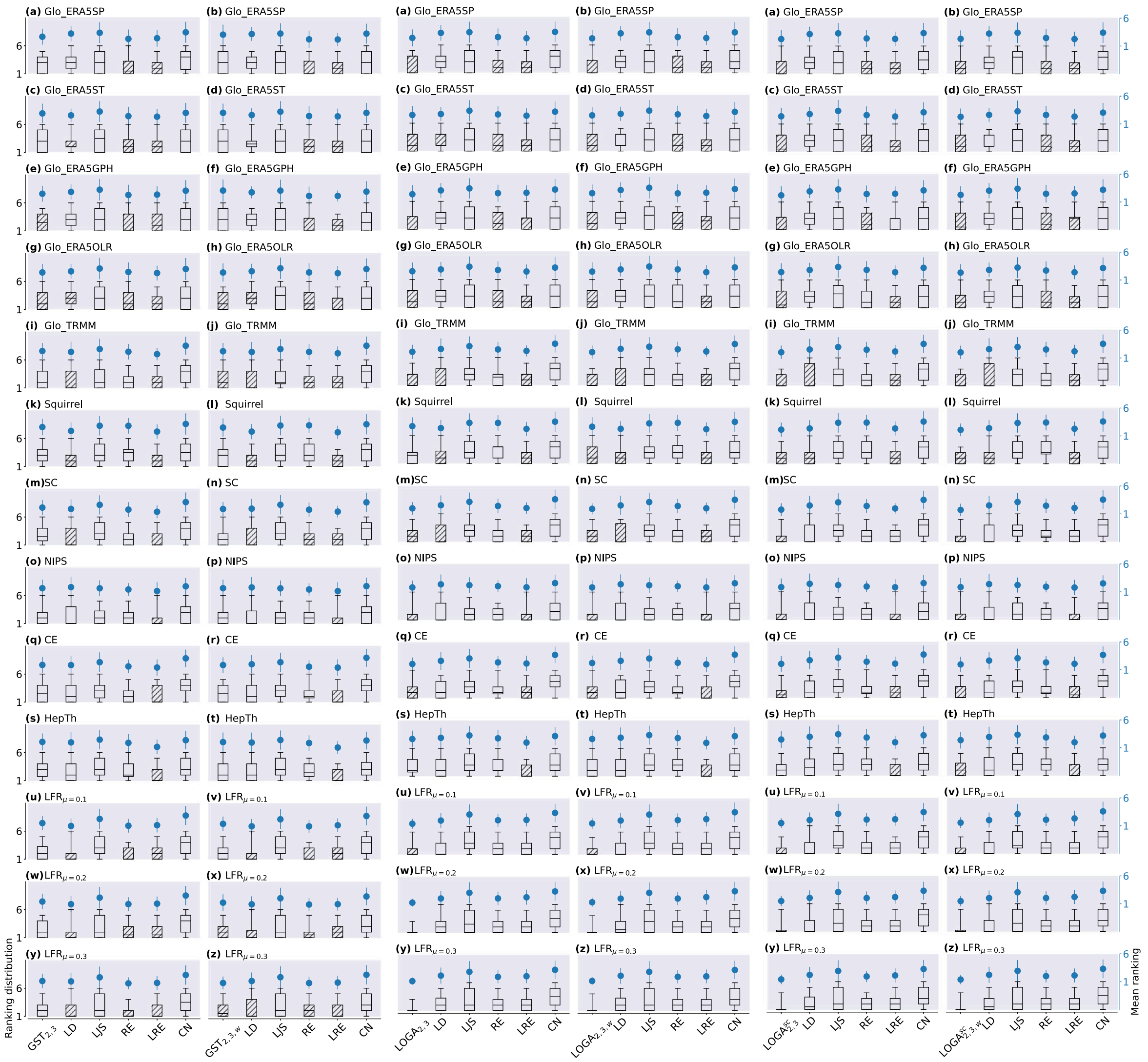
The ranking comparisons are given in Figs. 5 and 6. We summarize the final shooting score for each sampling method in Table 5. LRE performs well in practice compared with GST, LD, LJS, RE, and CN, in line with the conclusion in Ref. [10] that local filtering improves the preservation of representative properties. Yet, the sparse subgraph G' sampled by LOGA/LOGA^{sc} is even better than that by LRE, since LOGA$_{2,3,w}$ achieves the highest shooting score of $\frac{26}{26}$; that is, they better preserve representative properties on all network instances. The relative improvement of LOGA/LOGA^{sc} over GST, in terms of the shooting scores, is substantial with $\frac{LOGA_{2,3}+LOGA^{sc}_{2,3}-2GST_{2,3}}{4GST_{2,3}} + \frac{LOGA_{2,3,w}+LOGA^{sc}_{2,3,w}-2GST_{2,3,w}}{4GST_{2,3,w}} = 1.575$; similarly, an (albeit smaller) relative improvement of 0.2 is observed over LRE. This answers **Q1** and confirms that a good sparse subgraph as initialization can lead to better Nash equilibria for GST (see ‘Initialization via edge-focused sampling’ in Section 3.2).

To answer **Q2**, the average running times of LOGA/LOGA^{sc}, GST, LD, LJS, RE, LRE, and CN are compared in Fig. 4. LOGA/LOGA^{sc} take on average similar time as GST, confirming our analysis on ‘Time complexity’ in Section 3.2. Compared to the others, LOGA is 13, 8, 65, 12, and 24 times slower than LD, LJS, RE, LRE, and CN, respectively, on average. Despite taking longer than the simpler approaches, LOGA/LOGA^{sc} is scalable enough even for large-scale graph sparsification.

5. Conclusion

In summary, we proposed a hybrid sampling scheme LOGA for network sparsification. LOGA addresses the applicability of GST to graphs with weights and different densities, by providing GST with a good initialization and by including a constrained update. We verify the effectiveness of LOGA in producing even better (than the previous state of the art) sparse subgraphs G' similar to $\mathcal{G}$ in terms of preserving representative properties. According to extensive empirical studies on weighted graphs with different densities, we recommend LOGA$^{sc}_{2,3,w}$ in practice.

Regarding future work, one promising direction is to consider how to derive potentially suitable sparsification ratios in advance. Although this is situation-dependent, there should be a balance between the sparsification ratio and graph information loss. Furthermore, it remains to be answered whether this sparsification framework works well in directed graphs.



(A) GST vs {LD, LJS, RE, LRE, CN}. (B) LOGA vs {LD, LJS, RE, LRE, CN}. (C) LOGA^{sc} vs {LD, LJS, RE, LRE, CN}.

Fig. 5. Ranking comparison among LOGA/LOGA^{sc}, GST, LD, LJS, RE, LRE, and CN for 13 networks in Table 2. Methods with better rankings are hatched based on the Mann–Whitney U test (see Step II of ‘Evaluation metrics and procedure’ in Section 4).

CRediT authorship contribution statement

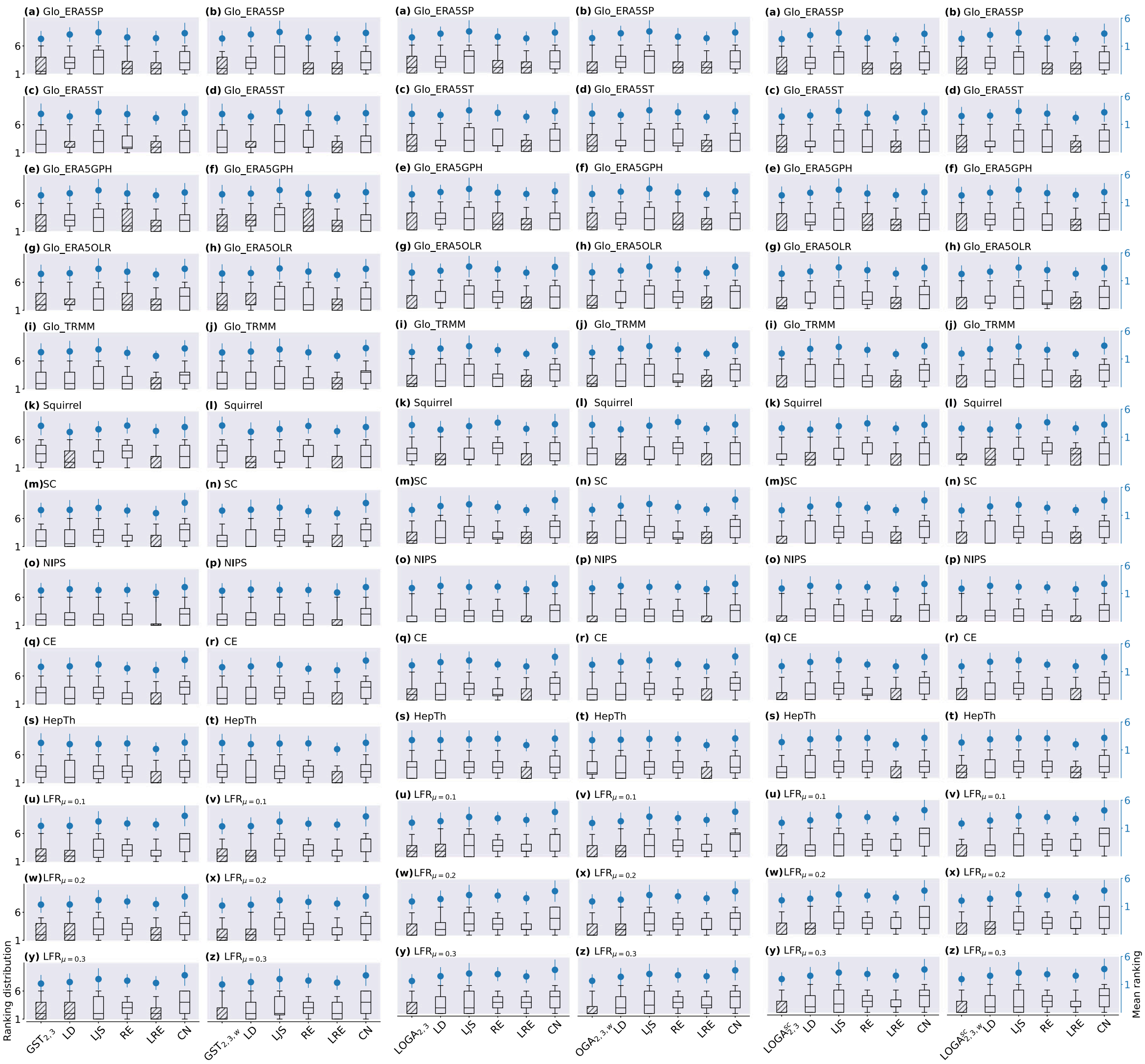
Zhen Su: Writing – review & editing, Writing – original draft, Visualization, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Jürgen Kurths:** Writing – review & editing, Supervision, Funding acquisition, Conceptualization. **Henning Meyerhenke:** Writing – review & editing, Supervision, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Z.S. was funded by the China Scholarship Council (CSC) scholarship. J.K. was supported by the Federal Ministry of Education and Research (BMBF), Germany grant No. 01LP1902J (climXtreme). H.M. was partially supported by German Research Foundation



(A) GST vs {LD, LJS, RE, LRE, CN}. (B) LOGA vs {LD, LJS, RE, LRE, CN}. (C) LOGA^{sc} vs {LD, LJS, RE, LRE, CN}.

Fig. 6. Same as Fig. 5, but for 13 networks in Table 3.

(DFG) grants ME-3619/4-1 (ALMACOM) and GR-5745/1-1 (DyANE). We acknowledge the European Regional Development Fund (ERDF), the German Federal Ministry of Education and Research and the Land Brandenburg for supporting this project by providing resources on the high performance computer system at the Potsdam Institute for Climate Impact Research.

References

[1] K. Yanagiya, K. Yamada, Y. Katsuhara, T. Takatani, Y. Tanaka, Edge sampling of graphs based on edge smoothness, in: ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, Singapore, Singapore, 2022, pp. 5932–5936, http://dx.doi.org/10.1109/ICASSP43922.2022.9747724.

[2] M. Choe, J. Yoo, G. Lee, W. Baek, U. Kang, K. Shin, Midas: representative sampling from real-world hypergraphs, in: Proceedings of the ACM Web Conference 2022, in: WWW '22, Association for Computing Machinery, New York, NY, USA, 2022, pp. 1080–1092, http://dx.doi.org/10.1145/3485447.3512157.

[3] L. Fang, C. Wu, HES: edge sampling for heterogeneous graphs, in: 2023 International Joint Conference on Neural Networks (IJCNN), IEEE, Gold Coast, Australia, 2023, pp. 1–8, http://dx.doi.org/10.1109/IJCNN54540.2023.10192005.

[4] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, L. Wang, Graph contrastive learning with adaptive augmentation, in: Proceedings of the Web Conference 2021, in: WWW '21, Association for Computing Machinery, New York, NY, USA, 2021, pp. 2069–2080, http://dx.doi.org/10.1145/3442381.3449802.

[5] M. Mathioudakis, F. Bonchi, C. Castillo, A. Gionis, A. Ukkonen, Sparsification of influence networks, in: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, in: KDD '11, Association for Computing Machinery, New York, NY, USA, 2011, pp. 529–537, http://dx.doi.org/10.1145/2020408.2020492.

[6] M. Benzi, C. Klymko, On the limiting behavior of parameter-dependent network centrality measures, SIAM J. Matrix Anal. Appl. 36 (2) (2015) 686–706, http://dx.doi.org/10.1137/130950550.

[7] D. Schoch, T.W. Valente, U. Brandes, Correlations among centrality indices and a class of uniquely ranked graphs, Social Networks 50 (2017) 46–54, http://dx.doi.org/10.1016/j.socnet.2017.03.010.

[8] Z. Su, C. Gao, J. Liu, T. Jia, Z. Wang, J. Kurths, Emergence of nonlinear crossover under epidemic dynamics in heterogeneous networks, Phys. Rev. E 102 (5) (2020) 052311, http://dx.doi.org/10.1103/PhysRevE.102.052311.

[9] Y. Ran, X.-K. Xu, T. Jia, The maximum capability of a topological feature in link prediction, PNAS Nexus 3 (3) (2024) pgae113, http://dx.doi.org/10.1093/pnasnexus/pgae113.

[10] M. Hamann, G. Lindner, H. Meyerhenke, C.L. Staudt, D. Wagner, Structure-preserving sparsification methods for social networks, Soc. Netw. Anal. Min. 6 (1) (2016) 22, http://dx.doi.org/10.1007/s13278-016-0332-2.

[11] C.M. Le, Edge sampling using local network information, J. Mach. Learn. Res. 22 (88) (2021) 1–29.

[12] J. Tětek, M. Thorup, Edge sampling and graph parameter estimation via vertex neighborhood accesses, in: Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing, ACM, Rome Italy, 2022, pp. 1116–1129, http://dx.doi.org/10.1145/3519935.3520059.

[13] Z. Su, J. Kurths, H. Meyerhenke, Network sparsification via degree- and subgraph-based edge sampling, in: 2022 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), 2022, pp. 9–16, http://dx.doi.org/10.1109/ASONAM55673.2022.10068651.

[14] P. Mahadevan, D. Krioukov, K. Fall, A. Vahdat, Systematic topology analysis and generation using degree correlations, ACM SIGCOMM Comput. Commun. Rev. 36 (4) (2006) 135–146, http://dx.doi.org/10.1145/1151659.1159930.

[15] C. Orsini, M.M. Dankulov, P. Colomer-de-Simón, A. Jamakovic, P. Mahadevan, A. Vahdat, K.E. Bassler, Z. Toroczkai, M. Boguñá, G. Caldarelli, S. Fortunato, D. Krioukov, Quantifying randomness in real networks, Nature Commun. 6 (1) (2015) 8627.

[16] C. Sun, A time variant log-linear learning approach to the SET K-COVER problem in wireless sensor networks, IEEE Trans. Cybern. 48 (4) (2018) 1316–1325, http://dx.doi.org/10.1109/TCYB.2017.2691772.

[17] C. Sun, W. Sun, X. Wang, Q. Zhou, Potential game theoretic learning for the minimal weighted vertex cover in distributed networking systems, IEEE Trans. Cybern. 49 (5) (2019) 1968–1978, http://dx.doi.org/10.1109/TCYB.2018.2817631.

[18] M. Liu, I. Kolmanovsky, H.E. Tseng, S. Huang, D. Filev, A. Girard, Potential game-based decision-making for autonomous driving, IEEE Trans. Intell. Transp. Syst. 24 (8) (2023) 8014–8027, http://dx.doi.org/10.1109/TITS.2023.3264665.

[19] P. Parchas, F. Gullo, D. Papadias, F. Bonchi, Uncertain graph processing through representative instances, ACM Trans. Database Syst. 40 (3) (2015) 20:1–20:39, http://dx.doi.org/10.1145/2818182.

[20] J. Lu, H. Wang, Uniform random sampling not recommended for large graph size estimation, Inform. Sci. 421 (2017) 136–153, http://dx.doi.org/10.1016/j.ins.2017.08.030.

[21] V. Sadhanala, Y.-X. Wang, R. Tibshirani, Graph sparsification approaches for Laplacian smoothing, in: Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, PMLR, 2016, pp. 1250–1259.

[22] P.N. McGraw, M. Menzinger, Laplacian spectra as a diagnostic tool for network structure and dynamics, Phys. Rev. E 77 (3) (2008) 031102, http://dx.doi.org/10.1103/PhysRevE.77.031102.

[23] V. Satuluri, S. Parthasarathy, Y. Ruan, Local graph sparsification for scalable clustering, in: Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data, in: SIGMOD '11, Association for Computing Machinery, New York, NY, USA, 2011, pp. 721–732, http://dx.doi.org/10.1145/1989323.1989399.

[24] J. Leskovec, C. Faloutsos, Sampling from large graphs, in: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, in: KDD '06, Association for Computing Machinery, New York, NY, USA, 2006, pp. 631–636, http://dx.doi.org/10.1145/1150402.1150479.

[25] A. Nocaj, M. Ortmann, U. Brandes, Untangling the hairballs of multi-centered, small-world online social media networks, JGAA 19 (2) (2015) 595–618, http://dx.doi.org/10.7155/jgaa.00370.

[26] E. John, I. Safro, Single- and multi-level network sparsification by algebraic distance, J. Complex Netw. 5 (3) (2017) 352–388, http://dx.doi.org/10.1093/comnet/cnw025.

[27] Y. Zeng, C. Song, T. Ge, Selective edge shedding in large graphs under resource constraints, in: 2021 IEEE 37th International Conference on Data Engineering (ICDE), 2021, pp. 2057–2062, http://dx.doi.org/10.1109/ICDE51399.2021.00200.

[28] Y. Zeng, C. Song, T. Ge, Y. Zhang, Reduction of large-scale graphs: effective edge shedding at a controllable ratio under resource constraints, Knowl.-Based Syst. 240 (2022) 108126, http://dx.doi.org/10.1016/j.knosys.2022.108126.

[29] Z. Su, Y. Liu, J. Kurths, H. Meyerhenke, Generic network sparsification via degree- and subgraph-based edge sampling, Inform. Sci. 679 (2024) 121096, http://dx.doi.org/10.1016/j.ins.2024.121096.

[30] P. Parchas, F. Gullo, D. Papadias, F. Bonchi, The pursuit of a good possible world: Extracting representative instances of uncertain graphs, in: Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data, in: SIGMOD '14, Association for Computing Machinery, New York, NY, USA, 2014, pp. 967–978, http://dx.doi.org/10.1145/2588555.2593668.

[31] D. Monderer, L.S. Shapley, Potential games, Games Econom. Behav. 14 (1) (1996) 124–143, http://dx.doi.org/10.1006/game.1996.0044.

[32] M. Newman, Networks, Oxford University Press, 2018.

[33] C.L. Staudt, A. Sazonovs, H. Meyerhenke, NetworKit: A tool suite for large-scale complex network analysis, Netw. Sci. 4 (4) (2016) 508–530, http://dx.doi.org/10.1017/nws.2016.20.

[34] J. Ash, D. Newth, Optimizing complex networks for resilience against cascading failure, Phys. A 380 (2007) 673–683, http://dx.doi.org/10.1016/j.physa.2006.12.058.

[35] O. Artime, M. Grassia, M. De Domenico, J.P. Gleeson, H.A. Makse, G. Mangioni, M. Perc, F. Radicchi, Robustness and resilience of complex networks, Nat. Rev. Phys. 6 (2) (2024) 114–131, http://dx.doi.org/10.1038/s42254-023-00676-y.

[36] A. Barrat, M. Barthélemy, R. Pastor-Satorras, A. Vespignani, The architecture of complex weighted networks, Proc. Natl. Acad. Sci. 101 (11) (2004) 3747–3752, http://dx.doi.org/10.1073/pnas.0400087101.

[37] M. Ortmann, U. Brandes, Triangle listing algorithms: back from the diversion, in: 2014 Proceedings of the Meeting on Algorithm Engineering and Experiments (ALENEX), in: Proceedings, Society for Industrial and Applied Mathematics, 2013, pp. 1–8, http://dx.doi.org/10.1137/1.9781611973198.1.

[38] N. Chiba, T. Nishizeki, Arboricity and subgraph listing algorithms, SIAM J. Comput. 14 (1) (1985) 210–223, http://dx.doi.org/10.1137/0214017.

[39] S. Gupta, N. Boers, F. Pappenberger, J. Kurths, Complex network approach for detecting tropical cyclones, Clim. Dyn. 57 (11) (2021) 3355–3364, http://dx.doi.org/10.1007/s00382-021-05871-0.

[40] N. Boers, B. Goswami, A. Rheinwalt, B. Bookhagen, B. Hoskins, J. Kurths, Complex networks reveal global pattern of extreme-rainfall teleconnections, Nature 566 (7744) (2019) 373–377, http://dx.doi.org/10.1038/s41586-018-0872-x.

[41] Z. Su, H. Meyerhenke, J. Kurths, The climatic interdependence of extreme-rainfall events around the globe, Chaos 32 (4) (2022) 043126, http://dx.doi.org/10.1063/5.0077106.

[42] A. Lancichinetti, S. Fortunato, F. Radicchi, Benchmark graphs for testing community detection algorithms, Phys. Rev. E 78 (4) (2008) 046110, http://dx.doi.org/10.1103/PhysRevE.78.046110.

[43] E. Angriman, A. van der Grinten, M. Hamann, H. Meyerhenke, M. Penschuck, Algorithms for large-scale network analysis and the NetworKit toolkit, in: H. Bast, C. Korzen, U. Meyer, M. Penschuck (Eds.), Algorithms for Big Data: DFG Priority Program 1736, in: Lecture Notes in Computer Science, Springer Nature Switzerland, Cham, 2022, pp. 3–20, http://dx.doi.org/10.1007/978-3-031-21534-6_1.

[44] C.L. Staudt, H. Meyerhenke, Engineering parallel algorithms for community detection in massive networks, IEEE Trans. Parallel Distrib. Syst. 27 (1) (2016) 171–184, http://dx.doi.org/10.1109/TPDS.2015.2390633.

[45] N.X. Vinh, J. Epps, J. Bailey, Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance, J. Mach. Learn. Res. 11 (95) (2010) 2837–2854.

[46] R. Geisberger, P. Sanders, D. Schultes, Better approximation of betweenness centrality, in: Proceedings of the Meeting on Algorithm Engineering & Expermiments, Society for Industrial and Applied Mathematics, USA, 2008, pp. 90–100, http://dx.doi.org/10.1137/1.9781611972887.9.

[47] A. Tsitsulin, M. Munkhoeva, B. Perozzi, Just slaq when you approximate: accurate spectral distances for web-scale graphs, in: Proceedings of the Web Conference 2020, in: WWW '20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 2697–2703, http://dx.doi.org/10.1145/3366423.3380026.

[48] A. Tsitsulin, D. Mottin, P. Karras, A. Bronstein, E. Müller, Netlsd: hearing the shape of a graph, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, in: KDD '18, Association for Computing Machinery, New York, NY, USA, 2018, pp. 2347–2356, http://dx.doi.org/10.1145/3219819.3219991.

[49] P.-Y. Chen, L. Wu, S. Liu, I. Rajapakse, Fast incremental von Neumann graph entropy computation: theory, algorithm, and applications, in: Proceedings of the 36th International Conference on Machine Learning, PMLR, 2019, pp. 1091–1101.