This article has been accepted for publication in IEEE Transactions on Network Science and Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2022.3216803

A Graph Entropy Measure From Urelement to Higher-Order Graphlets for Network Analysis

Ru Huang, Zijian Chen, Guangtao Zhai, *Senior Member, IEEE*, Jianhua He, *Senior Member, IEEE*, and Xiaoli Chu, *Senior Member, IEEE*

***Abstract*—Graph entropy measures have recently gained wide attention for identifying and discriminating various networks in biology, society, transportation, etc. However, existing methods cannot sufficiently explore the structural contents by merely considering the elementary invariants of a graph, ignoring the underlying patterns in higher-order features. In this paper, we propose a general *entropy-based graph representation framework* (GREET) based on four pertinent properties of graphlet topology from urelement to higher-order statistics. Specifically, we introduce an unbiased graphlet estimation strategy for obtaining both urelement and higher-order statistics. Additionally, we define a novel family of information functions based on hierarchical topological features to compute the graph entropy, then construct a graph information entropy (GIE) vector using the obtained local and global structural statistics to facilitate downstream tasks. Furthermore, there are some advantages that our GREET exhibits over other methods: (a) high accuracy with $<1\%$ relative error; (b) scalable for even larger vertex graphlets; (c) efficient calculation procedure with feasible speedup. Extensive experiments show that GREET exhibits superior performance on graph classification and clustering tasks, achieving remarkable improvements compared to several baselines. Altogether these findings pave the way for a wide range of applications of graphlet-based entropy as a complexity metric in graph analysis.**

***Index Terms*—Graph entropy, induced subgraphs, higher-order graphlets, graphlet estimation, graph characterization.**

I. INTRODUCTION

THERE are ubiquitous graph structures in various real-world complex systems, which call for trustworthy and effective graph characterization paradigms for more accurate and ultimately more useful representations. During the past decades, dozens of graph representation methods have been proposed for many networks, including communication networks [1], [2], social networks [3], [4], and biological networks [5], [6], etc. Generally, representation learning for networks is widely considered as a promising yet more challenging task, which requires the advancement of many techniques. One important issue in network characterization is to measure the complexity of these networks [7]–[9], by summarizing the underlying graph structure such as the micro-macro structure, community structure, core-periphery structure, etc.

To explore the inherent structural complexity of graphs, many entropy based methods are presented [10]–[12]. Historically, the entropy measure originates from information theory and is used to describe the chaos of a system. Built upon the Shannon entropy, the graph entropy is usually associated with features defined on a finite graph such as the probability distribution of vertices [13], [14], edge automorphism group [15], and degree sequence of the subgraph [16], etc. However, the entropy of a graph defined by those elementary properties may encounter severe difficulties in distinguishing the non-isomorphic graphs with similar structural roles and functions. For example, it is confusing to compare the compound networks, which contain different groups with homologous benzene rings [17]. In this case, the graph entropy based on shallow features merely stands for the structural similarity, which cannot capture the underlying patterns and does not consider higher-order organizations.

Graphlets or induced subgraphs [18], [19], as basic building blocks, represent the mesoscale structure of a network, which play an important role in structural measurement, function interpretation, and a wide range of network analysis applications. It has been proved that graphlets are very effective in characterizing networks [20]–[22]. For example, the graphlet kernel [23] (defined as the dot product of two vectors of normalized graphlet counts) and the euclidean distance between graphlet frequency distribution (GFD) vectors [24], are used proverbially for many network analysis tasks. Therefore, a question can be raised naturally is that "what about defining graph entropy measures by such structures?", which has not been well studied in the literature. Interestingly, as a special kind of subgraph, the main obstacle in practical~~practice~~ applications comes from the heavy computational budget when counting the graphlets. As shown in Table I, the number of graphlets increases exponentially with the number of vertices, which makes conducting downstream tasks highly inefficient. Furthermore, the existing algorithms [24]–[27] such as the exact counting methods, are severely limited to the network size. Hence, it is necessary to find an approximate method with high accuracy and scalability that can be used for calculating graph entropy efficiently.

In this paper, we present GREET[1], a network representation

This work was supported in part by the National Natural Science Foundation of China under Grant 61673178 and 61922063; in part by the Natural Science Foundation of Shanghai under Grant 20ZR1413800; in part by the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 824019 and 101022280. *(Corresponding author: Ru Huang.)*

Ru Huang and Zijian Chen are with the School of Information Science & Engineering, East China University of Science and Technology, Shanghai 200237, China (e-mail:huangrabbit@ecust.edu.cn).

Guangtao Zhai is with the Institute of Image Communication and Information Processing, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail:zhaiguangtao@sjtu.edu.cn).

Jianhua He is with the School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, U.K. (e-mail: j.he@essex.ac.uk).

Xiaoli Chu is with the Department of Electronic and Electrical Engineering, University of Sheffield, Sheffield S1 3JD, U.K. (e-mail:x.chu@sheffield.ac.uk).

[1] GREET is an anagram of the bold letters in n**E**twork charac**T**erization with **GR**aph **E**ntropy.

This article has been accepted for publication in IEEE Transactions on Network Science and Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2022.3216803

framework based on graph entropy measure. The motivation of this work is to construct a feature vector, called *graph information entropy* (GIE), which represents~~indicates~~ the structural complexity of a graph and can be used for distinguishing graphs. In particular, this paper proposes unbiased graphlet estimators for deriving both urelement and higher-order graphlet statistics. The graphlet estimators provide fast and accurate approximations compared to the existing exact counting algorithms. Concretely, we design an edge-centric neighborhood with predefined settings~~.~~, where *tree* and *cyclic* graphlets are treated as the initial evolutionary state, indicating that the subsequent search process of the higher-order graphlet is based on these basic structures and makes the exploration process more efficient. ~~*Tree* and *cyclic* graphlets are treated as the evolutionary factors, which make the exploration process of graphlets more efficient.~~ With the above designs, the proposed GREET achieves several orders of magnitude faster than the traditional iteration methods.

Inspired by the *entropy theory* in [28], we propose a novel family of information functionals based on the urelement and higher-order graphlet statistics (e.g., the frequency of k-vertex graphlets in an edge-centric neighborhood), which is then used to compute the graph entropy in the transformed space. It is noted that GREET only considers the graphlets that possess up to four vertices for performance and computation trade-offs. It can be easily inferred from Table I that the number of graphlets reaches 21 and 112 with five and six vertices, respectively, which are too rare for real-world networks and are too vague for characterizing a graph [16], [19], [20], [29]. Moreover, we justify the motivation of jointly considering urelement and higher-order features from the following aspects. First, it is obvious that only considering a node, edge, or its reachable nodes at a fixed distance with little ~~less~~ flexibility~~changeability~~ provides less structural information than higher-order closures. Second, taking both urelement and higher-order graphlets properties into account can obtain richer representation of the structural features of a graph.

In summary, this~~our~~ paper makes the following contributions:

- We propose a general network characterization framework GREET, ~~A general network characterization framework GREET is proposed for graph comparison,~~ which provides a novel graph entropy-based measurement for graph comparison. This framework is shown to be accurate, efficient, comprehensive, and scalable for various graph analysis tasks.
- Within the GREET framework, we develop unbiased estimators with fast and accurate counting strategies for estimating both urelement and higher-order statistics.~~We propose unbiased estimators for GREET framework, which provides fast and accurate counting strategies for both urelement and higher-order statistics.~~ Experimental results on various~~several~~ networks show that our proposed algorithm~~method~~ reduces the computational time by several orders of magnitude and achieves less than 1% relative error for nearly all graphlets.
- Motivated by Shannon entropy theory, we define two novel local and global information functionals based on the urelement and higher-order statistics, respectively. The resulting graph information entropy vector contains comprehensive complexity features~~both local and global topological properties~~ of the graph while capturing the underlying structural information.
- Extensive experiments are conducted over twenty real-world networks from six domains to evaluate the proposed GREET method in terms of graph classification and graph clustering tasks. The experimental results demonstrate that GREET ~~the effectiveness of GREET that~~ significantly outperforms the baselines in terms of classification accuracy and show that our proposed GIE can be used as a discriminative metric to greatly improve the graph clustering performance.

TABLE I
DISTINCT NOTATIONS~~NOTATION~~ AND PROPERTIES FOR 2-, 3- AND 4-VERTEX GRAPHLETS. AMONG THEM, Δ AND $\bar{\mathbf{d}}$ DENOTE THE MAX AND AVERAGE DEGREE, RESPECTIVELY; ρ DENOTE THE DENSITY; r IS THE ASSORTATIVITY COEFFICIENT; $Tree$ AND $Cyclic$ HOLD TRUE IF THE GRAPHLET SATISFIES THE REQUIREMENT.

Graphlet	Notation	Δ	$\bar{\mathbf{d}}$	ρ	r	$Tree$	$Cyclic$
	g_0	1	1.0	1.00	1.00	✓	✗
	g_1	2	1.33	0.67	-1.00	✓	✗
	g_2	2	2.0	1.00	1.00	✗	✓
	g_3	2	1.5	0.50	-0.50	✓	✗
	g_4	3	1.5	0.50	-1.00	✓	✗
	g_5	2	2.0	0.67	1.00	✗	✓
	g_6	3	2.0	0.67	-0.71	✓	✓
	g_7	3	2.5	0.83	-0.66	✗	✓
	g_8	3	3.0	1.00	1.00	✗	✓

The remainder of this paper is organized as follows. Section II introduces some necessary definitions and concepts that will be used in this paper and explains the proposed framework. Section III proposes a novel family of graph characterization based on the graph entropy theory. Section IV presents extensive experimental results on several graph analysis tasks over twenty networks. Section V concludes this paper with discussions and future works.

II. PRELIMINARIES

In this section, we introduce some definitions and necessary notations that will be used in this paper. Besides, we formally explain the goal of this paper is to obtain accurate representations of networks with a graph entropy-based measure.

Let $G(V, E)$ be a simple undirected graph with a finite non-empty set of vertices $V = \{v_1, v_2, \ldots, v_N\}$ and a finite set of edges $E = \{e_1, e_2, \ldots, e_M\}$. The existence of a edge between any two target vertices are determined by the neighborhoods centered on them. Hence, given a vertex $v_i \in V$, let $\Gamma^h(v_i) = \{v \in V | d(v, v_i) \leq h\}$ be the set of vertices h-*hop* adjacent to v_i in G, where $d(v, v_i)$ is the distance between node v and v_i in hops. Similarly, given two vertices $v_i, v_j \in V$, let $\Gamma^h(v_i, v_j)$ denotes the h-*hop* neighborhood around them defined as

$$\Gamma^h(v_i, v_j) = \Gamma^h(v_i) \cup \Gamma^h(v_j) \backslash \{v_i, v_j\} \tag{1}$$

This article has been accepted for publication in IEEE Transactions on Network Science and Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2022.3216803

TABLE II
FOUR PROPERTIES OF GRAPHLET TOPOLOGY IN GREET FROM THE PERSPECTIVE OF HIERARCHY AND DIMENSION

HIERARCHY	DIMENSION	
	Single Real-Value	**Distribution**
Urelement	Statistics in local closures	GFD of an node/edge
Higher-Order	Global graphlet counting	GFD of graphlets in G

For simplicity, the local neighborhood consists of the set of vertices adjacent to v_i or v_j and all edges between that set are denoted as $\mathbf{\Gamma}$. Moreover, the degree of a vertex v is denoted as $d_v = \left|\Gamma^1(v)\right|$ and $\Delta(G)$ denotes the maximum vertex degree.

Definition 1 (GRAPHLET). *A graphlet of G, $g_i = (V', E')$, is a connected induced subgraph of G if $V' \subseteq V$ and $E' \subseteq E$ with constraints that $\{e = (u, v) \,|u, v \in V', e \in E, e \notin E'\} = \emptyset$. In other words, graphlet is a subset of the vertices in graph G as well as all edges whose endpoints are both in this subset.*

In this paper, we define a graphlet with k vertices, as $g^{(k)}$ or *k-graphlet* and $g = g^{(1)} \cup \cdots \cup g^{(k)}$. Besides, we also follow the definition in [24] that a *k-graphlet* is called a *tree graphlet* if it has $k-1$ edges and the rest are named as *cyclic graphlet*. Table I reports some graphlet notations and important attributes.

Definition 2 (GRAPHLET FREQUENCY DISTRIBUTION). *Let G be a graph, the value $f_i^G = (X_i / \sum_i X_i)$ for $i = 1, \ldots, |g|$ is called* GFD, *where X_i is the frequency of graphlet g_i. The $\mathbf{f}_G = \begin{bmatrix} \cdots & f_i^G & \cdots \end{bmatrix}$ is the GFD in vector form.*

Definition 3 (SHANNON ENTROPY [30]). *Let X be a discrete random variable. The Shannon entropy of X is defined as*

$$I_X = -\sum_x p(x) \log p(x) \tag{2}$$

~~$I_X = -\sum_{x \in X} p(x) \log (p(x))$~~ , where $p(x)$ denotes the probability mass function that X is in the state x.~~where $p(x)$ is the probability distribution of X.~~

A. Graph Entropy and Information Functional

Combining the Shannon entropy [30] with graph theory [31], Dehmer *et al.* [28] proposed a new complexity measure that is graph entropy. One of the salient features for this graph entropy is that we can capture the structural content through parameterized representation.

Definition 4 (Dehmer [28]). *Given an undirected simple graph G, for a vertex $v_i \in V$, its local information functional is defined as*

$$f(v_i) = \alpha^{\beta_1 |\mathcal{S}_1(v_i, G)| + \beta_2 |\mathcal{S}_2(v_i, G)| + \cdots + \beta_\rho |\mathcal{S}_\rho(v_i, G)|} \tag{3}$$

where α is arbitrary real positive coefficient as well as β_k for $k = 1, 2, \ldots, \rho$. $|\mathcal{S}_\rho(v_i, G)|$ denotes the number of vertices in ρ-sphere, which acts as the set $\{v \in V | d(v_i, v) = \rho, \rho \geq 1\}$.

The local information functional exploit~~exploit~~ features over vertices explicitly and establish a filter-bank specialized in probability distribution bands for representation learning on intricate graph topology. Consequently, the vertex probability distribution based on information functional can be written as

$$p(v_i) = \frac{f(v_i)}{\sum_{v_j \in V} f(v_j)} \tag{4}$$

Subsequently, the graph entropy can be calculated in the form of Shannon entropy once a frequency distribution of the vertex set is obtained, which motivates us to associate the frequency of different structures with graph characterization.

B. Problem Formulation

The objective of this study is to explore a variety of graphlet statistics so as to construct a more discriminative feature vector for designing a practical graph representation learning framework. Inspired by [32], we mainly focus on two pairs of statistics with hierarchical and dimensional differences: urelement versus higher-order graphlets and single real-value versus distribution. Here, we provide an intuitive explanation for four properties of graphlet topology that are keys for learning powerful graph representations.

- Two notions of urelement are defined for local properties. The first, *Single real-value urelement statistics*, refers to the elementary structural features, e.g., the degree power of vertices, the counts of *triangle graphlet* containing a certain node or edge in the local neighborhood $\mathbf{\Gamma}_e$. The second, *urelement distribution*, measures an individual graph element (an arbitrary node or edge) through GFD.
- There are two further properties related to global structural information. *Higher-order single real-value statistics* can be defined as the total number of *arbitrary graphlets* in graph G. As the dimensional expansion, *higher-order distribution* indicates the overall graphlet frequency distribution (GFD) of G.

The proposed framework gives rise to graphlet feature exploration methods that are effective and efficient for various network applications. Recent approaches are limited to unilateral statistics [16], [20], [24], [33], whereas this paper instead proposes a comprehensive characterization framework beyond single counts that combines multidirectional graphlet statistics with information entropies.

III. THE PROPOSED METHOD

In this section, we introduce a novel family of graph entropy measures (GREET) by exploring urelement and higher-order graphlet statistics. This gives rise to the *graphlet-based entropy measure* that serves as a basis for deriving fast and accurate graph representations which are also applicable and scalable for networks in different domains. An illustration of the GREET framework is depicted in Fig. 1 for intuition.

A. The Urelement Statistics Estimating Method

Given the sampled target set of edge-centric neighborhoods, we illustrate the way to calculate the estimation of all distinct graphlets containing a specific node or edge in Algorithm 1. Specifically, the process of estimating the *urelement statistics* for an individual node or edge mainly consists of two parts,

This article has been accepted for publication in IEEE Transactions on Network Science and Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2022.3216803

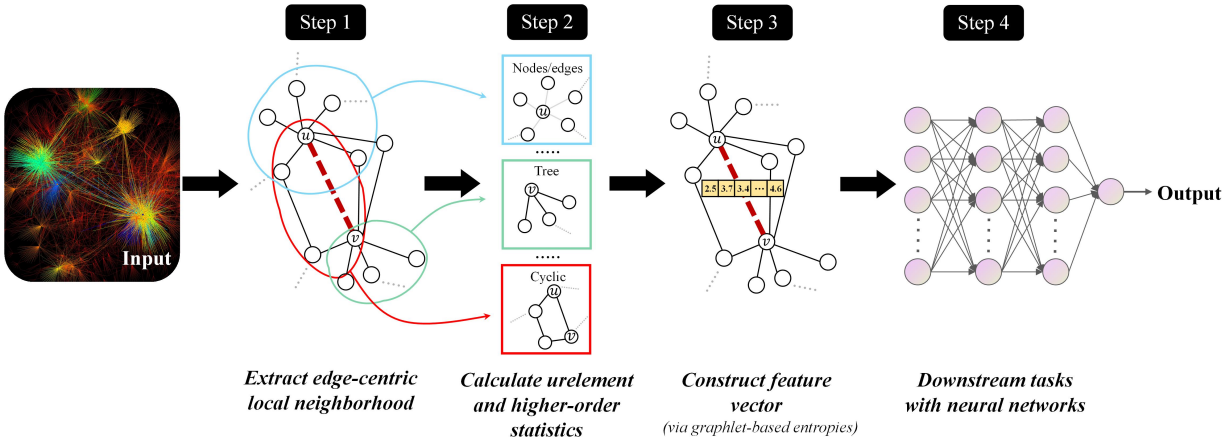


Fig. 1. The overall architecture of the proposed GREET. Given a sampled edge, which is shown in **red highlighted dash line (– –)**, the local neighborhood around it is abstracted as an enclosing subgraph. Then, the urelement and higher-order graphlet statistics can be calculated from graphlet frequencies dimensionally. Subsequently, the obtained properties are used to define information functionals and result in a set of feature vectors that can serve as the input of a neural network for downstream tasks.

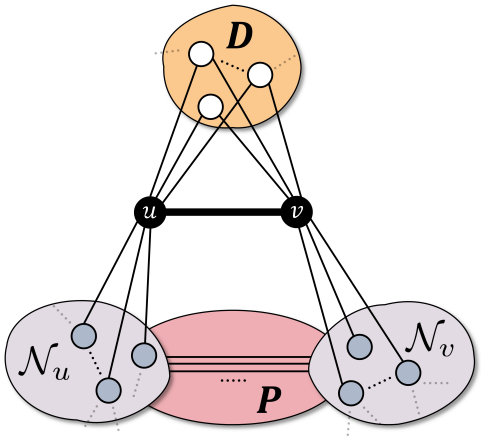


Fig. 2. Let D be the set of nodes completing a cyclic graphlet directly with a centered edge $e(u,v) \in E$. Similarly, let $\mathcal{N}_u$ and $\mathcal{N}_v$ be the set of nodes that form a 3-vertex tree graphlet with u and v, respectively. Moreover, P denotes the set of potential edges that complete a cyclic graphlet indirectly with respect to the edge $e(u,v)$. Note that, the other vertices in $\mathcal{N}_u$ are not connected with those in $\mathcal{N}_v$.

namely the *tree* and *cyclic graphlet* enumeration. Table I lists all graphlets used in this paper, referred as g_i for i ranging from 0 to 8. Among them, g_1, g_3, and g_4 are *tree graphlets*[2], while the remaining are *cyclic graphlets*. The task of *urelement statistics estimation* is to calculate an accurate yet fast approximation of graphlet distribution over a local neighborhood.

Definition 5 (URELEMENT ESTIMATION)**.** *Given an undirected graph $G = (V, E)$ and an edge-centric local neighborhood $\mathbf{\Gamma}_e$, the urelement estimation problem is to minimize function $\mathcal{L}(\mathbf{u}_i; \mathbf{y}_i)$, where $\mathbf{u}_i = [u_1\ u_2\ \cdots\ u_5\ u_6\ \cdots\ u_8]^T$ represents the approximation of the exact urelement statistics denoted by $\mathbf{y}_i$ for edge $e_i \in E$. The element u_k in $\mathbf{u}_i$ for k from 1 to 8 corresponding to the graphlet g_i presented in Table I.*

[2]Strictly speaking, g_0 is also a *tree graphlet*. However, in this paper, we exclude it due to the number of vertices. Besides, g_6 also contains *tree graphlet*.

Here, *L1-norm* is adopted as the loss function in Definition 5. Along the searching method proposed in [34], we commence by classifying basic substructures into *tree* and *cyclic*. First of all, we explain the searching of *tree graphlet*. Instead of discovering all *tree graphlets* explicitly, we relax the above problem through a general paradigm. Let us denote $\mathbf{\Gamma}_e$ as the target 1-*hop* neighborhood with respect to $e(u,v) \in E$. We construct two sets of nodes $\mathcal{N}_u = \{s \in \Gamma(u) \setminus \{v\} \,|\, s \notin \Gamma(v)\}$ and $\mathcal{N}_v = \{r \in \Gamma(v) \setminus \{u\} \,|\, r \notin \Gamma(u)\}$ within $\mathbf{\Gamma}_e$ that complete simplest *tree graphlet* with node u and v, respectively. For further intuition, please see Fig. 2. These defined structures are enable us to easily count the quantities of g_1. Furthermore, other *tree graphlet* such as g_3, g_4 or trees with more vertices can be found by the combinations of centered edge and $\mathcal{N}_{u,v}$. As for the enumeration of *cyclic graphlet*, we can utilize partial known spanning trees to generate *cyclic graphlets* (e.g., g_6, g_7 can have g_1, g_3, and g_4 as their precursor *tree graphlets*).

We now discuss the urelement estimation problem in detail. Let us set $\psi(\cdot)$ as a node label function for facilitating later procedure. Thus, nodes in local neighborhoods are labeled as θ_1, θ_2, and θ_3. Specificially, the neighbors $\Gamma(u)$ of u are marked as θ_1. Likewise, the neighbors $\Gamma(v)$ of v and the nodes in D_e are labeled as θ_2 and θ_3, respectively. First and foremost, sets $\mathcal{N}_u$, $\mathcal{N}_v$, and D_e are investigated by following iterations.

- *Start with an edge $e = (u,v) \in E$ and let $\mathcal{N}_u$, $\mathcal{N}_v$, and $D_e = \{\}$.*
- *for each node $s \in \Gamma(u)$, if $s \neq v$ then update $\mathcal{N}_u \leftarrow \mathcal{N}_u \cup \{s\}$ and set $\psi(s) = \theta_1$.*
- *for each node $s \in \Gamma(v) - u$, if $\psi(s) = \theta_1$ then update $D_e \leftarrow D_e \cup \{s\}$, $\mathcal{N}_u \leftarrow \mathcal{N}_u - \{s\}$ and set $\psi(s) = \theta_3$; else update $\mathcal{N}_v \leftarrow \mathcal{N}_v \cup \{s\}$ and set $\psi(s) = \theta_2$.*
- *Repeat the steps until all nodes in $\mathbf{\Gamma}_e$ are investigated.*

As shown in Algorithm 1, for simple 3-vertex graphlet g_1 and g_2, their estimation can be obtained directly via Lines 17–18 (as observed in [35]). Then, we investigate each $s \in \mathcal{N}_v$ and select a vertex $r \in \Gamma(s)$ with predefined probability distribution $\mathbb{P}$. Apparently, g_5, the 4-vertex *cyclic graphlet* can be checked by judging the type of vertex r and the intermediate variables

This article has been accepted for publication in IEEE Transactions on Network Science and Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2022.3216803

Algorithm 1 Estimation of Urelement Statistics

Input: An edge-centric neighborhood $\mathbf{\Gamma}_e$ and a sampling factor p

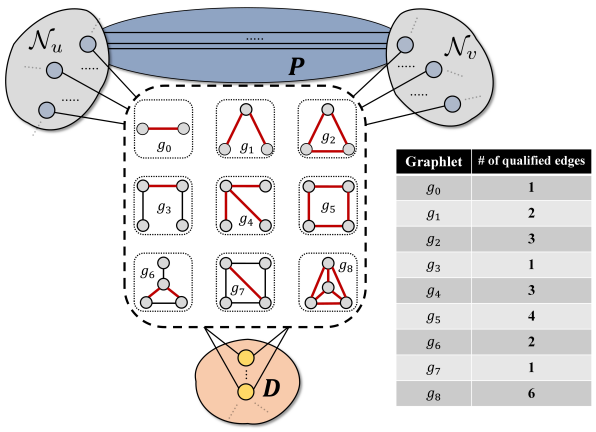
1: **for each** $s \in \mathcal{N}_v$ **do**
2: **for** $k = 1$ to $[d_s \cdot p]$ **do**
3: Sample a vertex $r \in \Gamma(s)$ with a given distribution $\mathbb{P}$
4: **if** $\psi(r) = \theta_1$ **then** update $u_5 = u_5 + (d_s/[d_s \cdot p])$ ▷ g_5
5: **if** $\psi(r) = \theta_2$ **then** update $u_6 = u_6 + (d_s/[d_s \cdot p])$ ▷ g_6
6: Reset $\psi(s) = 0$
7: **for each** $s \in \mathcal{N}_u$ **do**
8: **for** $k = 1$ to $[d_s \cdot p]$ **do**
9: Sample a vertex $r \in \Gamma(s)$ with a given distribution $\mathbb{P}$
10: **if** $\psi(r) = \theta_1$ **then** update $u_6 = u_6 + (d_s/[d_s \cdot p])$ ▷ g_6
11: Reset $\psi(s) = 0$
12: **for each** $s \in D_e$ **do**
13: **for** $k = 1$ to $[d_s \cdot p]$ **do**
14: Sample a vertex $r \in \Gamma(s)$ with a given distribution $\mathbb{P}$
15: **if** $\psi(r) = \theta_3$ **then** update $u_8 = u_8 + (d_s/[d_s \cdot p])$ ▷ g_8
16: Reset $\psi(s) = 0$
17: $u_1 = (d_u + d_v - 2) - 2|D_e|$ ▷ g_1
18: $u_2 = |D_e|$ ▷ g_2
19: $u_3 = |\mathcal{N}_u| \cdot |\mathcal{N}_v| - u_5$ ▷ g_3
20: $u_4 = \mathbf{C}^2_{|\mathcal{N}_u|} + \mathbf{C}^2_{|\mathcal{N}_v|} - u_6$ ▷ g_4
21: $u_7 = \mathbf{C}^2_{|D_e|} - u_8$ ▷ g_7
22: **return u**, where u_i is the estimate of graphlet g_i in $\mathbf{\Gamma}_e$

θ_1, as an indication that whether there exists potential edges between nodes in sets $\mathcal{N}_u$ and $\mathcal{N}_v$. Lines 5, 7–11 also show the detection of g_6 under the similar condition as g_5. In addition, the 4-vertex fully connected *cyclic graphlet* g_8 are estimated by searching area D_e, in which a node participates a triangle directly. Ultimately, the rest of graphlets are estimated in $\mathcal{O}(1)$ time using previous knowledge (Lines 19–21). Among them, the 4-vertex lines g_3 are calculated by subtracting those completing a cyclic, while graphlets g_4 and g_7 are estimated leveraging the combination number formula, where $\mathbf{C}^m_n = n!/m!\,(n-m)!$.

Complexity Analysis: The strategy we adopted enables us to estimate the urelement statistics efficiently, which utilizes the obtained knowledge in each step adequately. By dividing graphlet into two typical structures, namely the *tree graphlet* and the *cyclic graphlet*, we are able to extrapolate other more complex graphlets. Furthermore, compared to the existing methods [24], [25], [29], we specify three key sets $\mathcal{N}_u$, $\mathcal{N}_v$, and D_e that makes it possible to count k-vertex graphlets without unnecessary computation. Finally, to calculate the urelement statistics from a given edge, the computational complexity is $\mathcal{O}\left(\widetilde{\Delta}\left(|\mathcal{N}_u| + |\mathcal{N}_v| + |D_e|\right)\right)$, where $\widetilde{\Delta}$ denote the maximum degree of vertex within $\mathcal{N}_u$, $\mathcal{N}_v$, and D_e. It is evident that most of the computational budget comes from the searching process of neighbors in the above sets. To mitigate this problem, we predefine sampling factor p (Lines 3, 9, and 14), which ensures the acquisition of important network structure while leading to an acceptable time cost.

B. The Higher-Order Statistics Estimating Method

This section discusses graphlet estimation in the context of the entire graph. Note that, we retain some necessary notations and illustrations from Section III-A for better understanding. It is necessary to investigate the difference between urelement and higher-order graphlet estimation strategies. Obviously, there exist more diverse connection patterns among the whole graph



Graphlet	# of qualified edges
g_0	1
g_1	2
g_2	3
g_3	1
g_4	3
g_5	4
g_6	2
g_7	1
g_8	6

Fig. 3. **Edge connection pattern diagram:** Figure shows all possible edges that satisified the requirements for building edge-centric neighborhoods $\mathcal{N}_u$, $\mathcal{N}_v$, and D_e. Note that, all eligible edges are colored in red and highlighted. The table in the bottom-right cornor provides the summarization of qualified edges in each 2-, 3-, and 4-graphlet.

than local subgraphs, resulting in unavoidable counting errors. On account of this, we adopt the Horvitz-Thompson estimator [36] to obtain unbiased estimates. The iterative process is shown in Algorithm 2. First, for each sampled edge $e \in E_s$, we build neighborhoods $\mathcal{N}_u$, $\mathcal{N}_v$, and D_e, as it was done in Section III-A. The quantity of g_1, g_2, g_3, g_4, g_6, and g_7 are derived in $\mathcal{O}(1)$ time utilizing knowledge from the centered edge and its completed tree and cyclic structures with sets $\mathcal{N}_u$, $\mathcal{N}_v$, and D_e (refer to Lines 4–7, and 9–10 in Algorithm 2, where $\oplus$ denotes the addition assignment operator) [35]. Besides, the g_5 and g_8 are computed through Algorithm 3 and 4 using information from the $(k-1)$-vertex *tree and cyclic graphlets*. Here, we use H_i for i from 1–8 to denote the graphlet counts in graph G and sampling probability p_i is defined as $p_i = |E_s|/|E|$, where $|E_s|$ and $|E|$ are the number of sampled edges and total edges in graph G, respectively. Moreover, let the reciprocal of p be $z_i = 1/p_i$, which is then used to correct the counting errors.

Intuitively, the sampling bias can be easily associated with the edge role in each graphlet. As shown in Fig. 3, we highlight all qualified edges in red color and list a table in the bottom-right corner, which concludes the number of suitable edges per graphlet. For example, consider the fully-connected 4-cyclic g_8 graphlet, in which each edge is involved in precisely two

Algorithm 2 Estimation of Higher-Order Statistics

Input: A graph $G(V, E)$ with sampled edge set E_s

1: **for each** $e = (u, v) \in E_s$ **do**
2: Set D_e, $\mathcal{N}_u$, and $\mathcal{N}_v = \emptyset$
3: Construct local neighborhoods as in Alg. 1
4: $H_1 \oplus H_1(e) = |\mathcal{N}_u| + |\mathcal{N}_v|$
5: $H_2 \oplus H_2(e) = |D_e|$
6: $H_3 \oplus H_3(e) = |\mathcal{N}_u| \cdot |\mathcal{N}_v|$
7: $H_4 \oplus H_4(e) = \mathbf{C}^2_{|\mathcal{N}_u|} + \mathbf{C}^2_{|\mathcal{N}_v|}$
8: $H_5 \oplus H_5(e) =$ S4-CYCLIC(ψ, $\mathcal{N}_v$)
9: $H_6 \oplus H_6(e) = (|\mathcal{N}_u| + |\mathcal{N}_v|) \cdot |D_e|$
10: $H_7 \oplus H_7(e) = \mathbf{C}^2_{|D_e|}$
11: $H_8 \oplus H_8(e) =$ F4-CYCLIC(ψ, D_e)
12: **return H**, where H_i is the count for graphlet g_i

This article has been accepted for publication in IEEE Transactions on Network Science and Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2022.3216803

triangles. Therefore, the role of each edge in g_8 is exactly the same and all of them satisfy the definition of the central edge. Similarly, consider the simplest 4-cyclic g_5, where each edge is equivalent. Combined with the above views and algorithms, we can deduce that each F4-cyclic g_8 and S4-cyclic g_5 will be counted 6 and 4 times, respectively. As a result, the correction coefficients are given as 1/6 and 1/4. Formally, the correction factor of g_i account for the edge multiplicities in G is defined as

$$C\left(g_i, G\right) = \frac{1}{q_i} \tag{5}$$

where q_i is the number of edges that qualified for the definition of the local neighborhood in graphlet g_i. Afterward, the higher-order graphlet counts are estimated via the following equations:

$$X_i = z_i C_i Z_i \ , \quad Z_i = \begin{cases} H_i & i = 1, 2, 5, 6, 8 \\ H_i - H_5 & i = 3 \\ H_i - H_6 & i = 4 \\ H_i - H_8 & i = 7 \end{cases} \tag{6}$$

where X_i is the estimated counts of the graphlet g_i. Specifically, the calculation of graphlets g_3, g_4, and g_7 are different from the others. This is due to the fact that such graphlets are structurally evolved from basic *tree* and *cyclic* graphlets.

Error Analysis: For a graphlet g_i, assume $Y_i(e)$ is its ground truth in the $G(V, E)$. In the process of estimation, we construct edge-centric neighborhood over the sampled edges of the given graph G, such that an edge e accounts for the existence of g_i. Thus, the total count of g_i are obtained as $Y_i = \sum_{e \in E} Y_i\left(e\right)$. The approximation counting method of GREET estimate Y_i by sampling a set of edges $E_s \subset E$ using a certain probability τ, then

$$X_i \simeq \frac{1}{\tau} \sum_{e \in E_s} Y_i\left(e\right) \tag{7}$$

is an estimator for g_i.

Lemma 1. *X_i is an unbiased estimator of Y_i.*

Proof.

$$\mathbb{E}[X_i] = \mathbb{E}\Big[\frac{1}{\tau} \sum_{e \in E_s} Y_i\left(e\right)\Big] = \frac{1}{\tau} \sum_{e \in E_s} \mathbb{E}[Y_i\left(e\right)] \tag{8}$$

Since each of the edges are sampled randomly, $\mathbb{E}[Y_i\left(e\right)] = \mathbb{E}[\mathrm{B}_e] \cdot Y_i\left(e\right)$, where B_e denotes a Bernoulli random variable indicating whether e and its local neighborhood are selected. Plugging in Eq. 8, we obtain:

$$\mathbb{E}[X_i] = \frac{1}{\tau} \sum_{e \in E} \mathbb{E}[\mathrm{B}_e] \cdot Y_i\left(e\right) = \frac{1}{\tau} \sum_{e \in E} \tau \cdot Y_i(e) = Y_i \tag{9}$$

Hence, the lemma is proved. □

Moreover, the mean squared error (MSE) of X_i is obtained as $\mathbb{E}[(X_i - Y_i)^2]$. Since, $\mathbb{E}[X_i] = Y_i$ and we have

$$\mathrm{MSE}\left(X_i\right) = \underbrace{\mathbb{E}[(X_i - \mathbb{E}[X_i])^2}_{\text{Variance}} + \underbrace{\left(\mathbb{E}[X_i] - Y_i\right)^2}_{\text{Bias}} \tag{10}$$

where the first term on the right side of the equation is variance and the second term is the bias generated by the estimator X_i. As a result, since X_i is an unbiased estimator, $\mathrm{MSE}\left(X_i\right) = \mathbb{E}[(X_i - \mathbb{E}[X_i])^2$. In this paper, we also argue that all the edges are sampled independently.

Algorithm 3 Simplest 4-Cyclic Counting

```
procedure S4-CYCLIC(ψ, N_v)
    Set variable C_s = 0
    for each s ∈ N_v do
        for each r ∈ Γ(s) do
            if ψ(r) = θ_1 then set C_s ⊕ 1
        Reset ψ(s) = 0
    return C_s
```

Algorithm 4 Fully-Connected 4-Cyclic Counting

```
procedure F4-CYCLIC(ψ, D_e)
    Set variable C_f = 0
    for each s ∈ D_e do
        for each r ∈ Γ(s) do
            if ψ(r) = θ_3 then set C_f ⊕ 1
        Reset ψ(s) = 0
    return C_f
```

Complexity Analysis: Here, the computational complexity of counting higher-order graphlets is investigated. The maximum number of simplest cyclic and tree graphlets occurring to a sampled edge $e \in E_s$ are denoted by $D_{\max}$ and $\mathcal{N}_{\max}$, respectively, where $\mathcal{N}_{\max} = \max\left(|\mathcal{N}_u|, |\mathcal{N}_v|\right)$. For a local neighborhood $\mathbf{\Gamma}_e$, the simplest and fully-connected 4-cyclic are estimated based on the Algorithm 2 in $\mathcal{O}(\Delta\mathcal{N}_{max})$ and $\mathcal{O}(\Delta D_{max})$, respectively. Subsequently, we can derive the rest of graphlets in $\mathcal{O}(1)$ using the combinational structure of the *tree* and *cyclic* graphlets. Therefore, the time complexity in counting graphlets for $[\mathbf{\Gamma}_{e_1}, \mathbf{\Gamma}_{e_2}, \ldots, \mathbf{\Gamma}_{e_M}]$ is up to $\mathcal{O}(M\Delta(D_{\max}+\mathcal{N}_{\max}))$, where Δ is the maximum degree in G. Besides, each counting on the target edge can be conducted independently. Therefore, we can also use parallelized processing unit to further reduce the time complexity.

C. Quantifying Structural Information with Entropy

Given the hierarchical graphlet statistics, a straightforward way is to directly transform these counts into heuristic features, which turns out to be suboptimal in [37], probably due to the lack of generality. To address this, we present two novel information functionals based on a fixed size of induced subgraphs, which can be utilized to estimate the entire graph entropy. Then, such graph entropies are combined with other topological information content metrics to construct discriminative and general graph representations that can be computed efficiently.

Different from Dehmer's work [28], we cast sight on defining the probability distribution of edges for a graph. In this paper, we introduce an edge-centric neighborhood $\mathbf{\Gamma}_e$ with abundant properties, which allows us to define graph entropy based on the obtained urelement and higher-order graphlet statistics (Table II). As mentioned in [28], such graph entropies can capture the underlying structural information of G effectively.

Definition 6 (EDGE PROBABILITY DISTRIBUTION)**.** *Let $G = (V, E)$ with iterable edge labels. For an edge $e_i \in E$, we define*

$$p\left(e_i\right) := \frac{f\left(e_i\right)}{\sum_{j=1}^{|E|} f\left(e_j\right)} \tag{11}$$

This article has been accepted for publication in IEEE Transactions on Network Science and Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2022.3216803

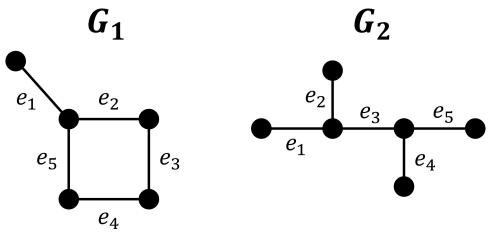


Fig. 4. Information functionals for two undirected graphs G_1 and G_2 with 6 edges and 5 edges, respectively.

where $f(\cdot)$ represents an information functional, which satisfied the equation $p(e_1) + p(e_2) + \cdots + p(e_{|E|}) = 1$.

Definition 7 (GRAPH ENTROPY)**.** *Given a undirected graph $G = (V, E)$ and let f be an arbitrary information functional. We define the entropy of G by*

$$I_f(G) = -\sum_{i=1}^{|E|} \frac{f(e_i)}{\sum_{j=1}^{|E|} f(e_j)} \log\left(\frac{f(e_i)}{\sum_{j=1}^{|E|} f(e_j)}\right) \tag{12}$$

Definition 8 (LOCAL INFORMATION FUNCTIONAL)**.** *Given a simple undirected graph G, the local information functional for an edge $e_i \in E$ is given as*

$$f^U(e_i) := e^{\alpha_1 |u_1(e_i,G)| + \alpha_2 |u_2(e_i,G)| + \cdots + \alpha_k |u_k(e_i,G)|} \tag{13}$$

where α_k is the real positive coefficient. $|u_k(e_i, G)|$ indicates the estimated urelement statistics of neighborhood $\mathbf{\Gamma}_{e_i}$ regarding G.

Definition 9 (GLOBAL INFORMATION FUNCTIONAL)**.** *For a given graph $G = (V, E)$, let $h_j(G)$ be the number of graphlet g_j. We define the global information functional by*

$$f^H(G) := h_j(G) \tag{14}$$

where j is the ordinal number of graphlet.

Apparently, according to the above definition, we can obtain $h_j(G)$ directly from the estimation of higher-order graphlets. Let $\theta(G) = \sum_{j=1}^{T} h_j(G)$ be the total number of graphlets in G, T is the number of types of graphlets. The obtained graph entropy I_{f^H} is stated as

$$I_{f^H}(G) = -\sum_{j=1}^{T} \frac{f^H(G)}{\theta(G)} \log\left(\frac{f^H(G)}{\theta(G)}\right) \tag{15}$$

Similarly, the graph entropy I_{f^U} is stated as

$$I_{f^U}(G) = -\sum_{i=1}^{|E|} \frac{f^U(e_i)}{\sum_{j=1}^{|E|} f^U(e_j)} \log\left(\frac{f^U(e_i)}{\sum_{j=1}^{|E|} f^U(e_j)}\right) \tag{16}$$

In particular, the general idea in computing the information functional $f^U(e_i)$ and $f^H(G)$ is that we build a bridge between the simple counts of graphlets and the measurement of graphs from the perspective of micro and macro. One prominent advantage of this method is that it quantifies the structural information of a graph with meaningful complexity measurement.

Numerical Analysis: We numerically investigate the entropies of the example graphs of Fig. 4 by using the proposed information functional $f^U(e_i)$ and $f^H(G)$. Based on the obtained estimation results of urelement and higher-order graphlet statistics, we are able to explore the discernibility of the resulting graph entropies. For G_1, by applying the Algorithm 1, we obtain:

$$f^U(e_2) = f^U(e_5) = e^{\alpha_1+\alpha_3+\alpha_4+\alpha_5} \tag{17}$$
$$f^U(e_3) = f^U(e_4) = e^{\alpha_3+\alpha_5} \tag{18}$$
$$f^U(e_1) = e^{2\alpha_1+\alpha_4} \tag{19}$$

Here, we set $\alpha_1 = \alpha_2 = \cdots = \alpha_k = 1$ and plug $f^U(e_i)$ into Eq. 16. Thus, the structural content of G_1 becomes to

$$\begin{aligned} I_{f^U}(G_1) = &-2\frac{e^4}{2e^4+2e^2+e^3}\log\left(\frac{e^4}{2e^4+2e^2+e^3}\right) \\ &-2\frac{e^2}{2e^4+2e^2+e^3}\log\left(\frac{e^2}{2e^4+2e^2+e^3}\right) \\ &-\frac{e^3}{2e^4+2e^2+e^3}\log\left(\frac{e^3}{2e^4+2e^2+e^3}\right). \end{aligned} \tag{20}$$

To compute the entropy of G_1 regarding f^H, we directly apply Definition 9 to the Eq. 15, and obtain:

$$\begin{aligned} I_{f^H}(G_1) &= -\sum_{j=1}^{8} \frac{h_j(G_1)}{\theta(G_1)} \log\left(\frac{h_j(G_1)}{\theta(G_1)}\right) \\ &= -\left[\frac{2}{3}\log\frac{1}{3} + \frac{1}{3}\log\frac{1}{6}\right] \end{aligned} \tag{21}$$

By applying the same steps as stated above, the calculation for the structural information of $I_{f^U}(G_2)$ and $I_{f^H}(G_2)$ is the same. Hence,

$$\begin{aligned} I_{f^U}(G_2) &= -\sum_{i=1}^{5} \frac{f^U(e_i)}{\sum_{j=1}^{5} f^U(e_j)} \log\left(\frac{f^U(e_i)}{\sum_{j=1}^{5} f^U(e_j)}\right) \\ &= -\left[4\frac{e^3}{4e^3+e^{10}} + \frac{e^{10}}{4e^3+e^{10}}\right] \end{aligned} \tag{22}$$

$$\begin{aligned} I_{f^H}(G_2) &= -\sum_{j=1}^{8} \frac{h_j(G_2)}{\theta(G_2)} \log\left(\frac{h_j(G_2)}{\theta(G_2)}\right) \\ &= -\left[\frac{1}{2}\log\frac{1}{2} + \frac{1}{3}\log\frac{1}{3} + \frac{1}{6}\log\frac{1}{6}\right] \end{aligned} \tag{23}$$

Obviously, the computation of $I_f(G)$ is able to generalize to more complicated structures. As mentioned earlier, we mainly consider 3- and 4-*graphlets* due to the computational cost. To guarantee the independence and the discrimination of features, we combine some variants of I_{f^U} and I_{f^H} with other classical graph entropy metrics to construct the graph information entropy (GIE) vector for a richer graph representation. Specifically, we add the graph entropy I_α and I_d proposed in [38], [39], which are defined by the vertex orbits and the degree sequence of all vertices in G, respectively. Motivation of this setting is straightforward. Firstly, the metrics I_α and I_d are nearly the simplest version of the graph entropy, which can be utilized to improve the discrimination ability in distinguishing networks with significant structural differences. Secondly, incorporating

This article has been accepted for publication in IEEE Transactions on Network Science and Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2022.3216803

TABLE III
PROPERTIES OF UNDIRECTED GRAPH IN REAL-WORLD DATASETS. $|V|$, $|E|$, Δ, $\bar{\mathbf{d}}$, ρ, r ARE THE NUMBER OF NODES, NUMBER OF EDGES, MAXIMUM DEGREE, AVERAGE DEGREE, DENSITY OF GRAPH, AND ASSORTATIVITY COEFFICIENT, RESPECTIVELY

Graph	$\|V\|$	$\|E\|$	Δ	$\bar{\mathbf{d}}$	ρ	r
bio-CE-LC	1,387	1,648	131	2.48	0.002	-0.17
bio-HS-HT	2,570	13,691	149	9.75	0.004	0.29
bio-dmela	7,393	25,569	190	6.23	<0.001	-0.05
bio-yeast	1,458	1,948	56	2.67	0.002	-0.21
bio-celegans	453	2,025	237	8.94	0.02	-0.23
soc-pg-tvshow	3,892	17,239	126	8.85	0.002	0.56
soc-wiki-vote	889	2,914	102	6.34	0.007	-0.03
ia-corecipient	906	11,197	442	28.38	0.04	-0.05
ia-fb-messages	1,266	6,451	112	10.19	0.008	-0.08
ia-reality	6,809	7,680	261	2.26	<0.001	-0.68
ca-GrQc	4,158	13,422	81	6.45	0.001	0.66
ca-CSphd	1,881	1,739	46	1.85	0.001	-0.20
ca-AstroPh	17,903	196,972	504	22	0.001	0.21
email-EU	32,430	54,397	623	3.36	<0.001	-0.38
email-enron-only	143	623	42	8.71	0.06	-0.02
email-univ	1,133	5,451	71	9.62	0.01	0.08
econ-wm1	258	2,744	110	21.27	0.08	0.04
econ-poli	3,915	4,180	66	2.14	0.001	-0.29
econ-mahindas	1,258	7,576	206	12.04	0.01	-0.06
econ-mbeaflw	487	49920	679	139.04	0.41	-0.27

various graph entropy can receive better generalization capability. As a result, the complete form of the feature vector is $\vec{\mathbf{v}}_{GIE} = [I_{f^U}, I_{f^H}, I_{f^U} + I_{f^H}, |I_{f^U}||I_{f^H}|, I_\alpha, I_d]$, which measures both local and global structure information of a graph and can be fed into a neural network or machine learning model for downstream tasks.

IV. EXPERIMENTS

In this section, we conduct extensive experiments to evaluate the performance of GREET on over twenty networks from six domains with different structural properties. All datasets are obtained from the Network Repository[3] [40] and an open source dataset website[4]. To make a fair comparison, our experiments are performed on a Windows (64-bit) PC with a Quad-Core Intel i5-9300HF CPU 2.4GHz processor, 16GB RAM, and NVIDIA GeForce GTX 1660Ti 6G GPU.

A. Performance on the Higher-Order Graphlet Estimation

The first class of experiments investigates the performance of GREET on higher-order graphlet estimation problems. We select twenty undirected graphs as testbeds including biological, social, interaction, collaboration, email, and economic networks. Table III reports the statistical information of each network. Here, the state-of-the-art approach PGD [35] is utilized to compute the exact graphlet counts for comparison. In our experiments, we define $\frac{|X_i - Y_i|}{Y_i}$ as the relative error of X_i, $1 \leq i \leq 8$, where X_i denotes the unbiased estimator of the genuine statistic Y_i. For better performance, the indicator is expected to be as small as possible. Note that all estimated values are calculated over 100 independent runs.

[3]https://networkrepository.com
[4]https://ls11-www.cs.tu-dortmund.de/staff/morris/graphkerneldatasets

1) Estimating the Graphlet Statistics: We first test GREET for estimating the frequency of graphlets with size $k = 3$ and $k = 4$. Table IV lists the GFD results for a variety of real-world networks at the sampling factor of 0.1. It can be observed that tree graphlets g_3 and g_4 make up the majority of graphlets in most networks, while the proportion of cyclic graphlets g_2, g_5, and g_8 remain at the bottom on most graphs. Such results enable us to better analyze the network patterns and structural information. Besides, the $L1\text{-}norm$ between exact counts and the estimated results are given to illustrate the high accuracy of the estimator among all graphs and graphlets.

2) Sampling Factor versus Counting Error: To fully evaluate our methods, we further explore the effect of the sampling probability on counting error. Fig. 5 reports the relative errors of GREET with sampling factor $p = 1\%, 10\%, 100\%$. We can observe that the counting errors are proportional to $1/p$, which is consistent with the analysis in Section IV-C. Besides, no significant differences are found between the estimated and the exact counts in all cases. For most graphs, the counting errors of the graphlets are lower than 0.01 when $p = 1\%$, except for those graphlets with relatively small counts. The overall trend of the results indicates that graphlets with large proportions exhibit less error than those with small proportions.

3) Speed-Up Comparison with Previous Work: In addition, we also study the computational time of GREET on eight typical networks from six domains that are shown in Table V. Three existing graphlet counting methods, including brute force [27], GUISE [25], and GRAFT [24], are used for comparison. However, as the graph grows larger, it is difficult to obtain accurate counting for GUISE and GRAFT. Thus, we only report the time cost until the relative error reaches below 0.01. For large graphs, traditional brute force is clearly not suitable due to its high computational costs. The hyphen "−" in Table V means that there is out of memory and the calculation can not be accomplished within 3 hours. Take the large graph "ca-AstroPh" as an example, which has nearly 200k edges and is not able to be completed in a short period of time by brute force, GUISE takes over 1 minute to compute and GRAFT returns in 18 seconds. Strikingly, our proposed GREET only takes less than 2 seconds to return the results. Benefitting from the proposed sampling strategy, for other graphs, GREET further reduces the computational time to less than 1 second. In summary, our approach is about 2 to 3 orders of magnitude faster than the traditional iteration methods, GUISE, and GRAFT, which makes GREET a better option among various estimation methods and thus leads to higher efficiency in other tasks.

B. Comparison with Existing Works on Graph Classification

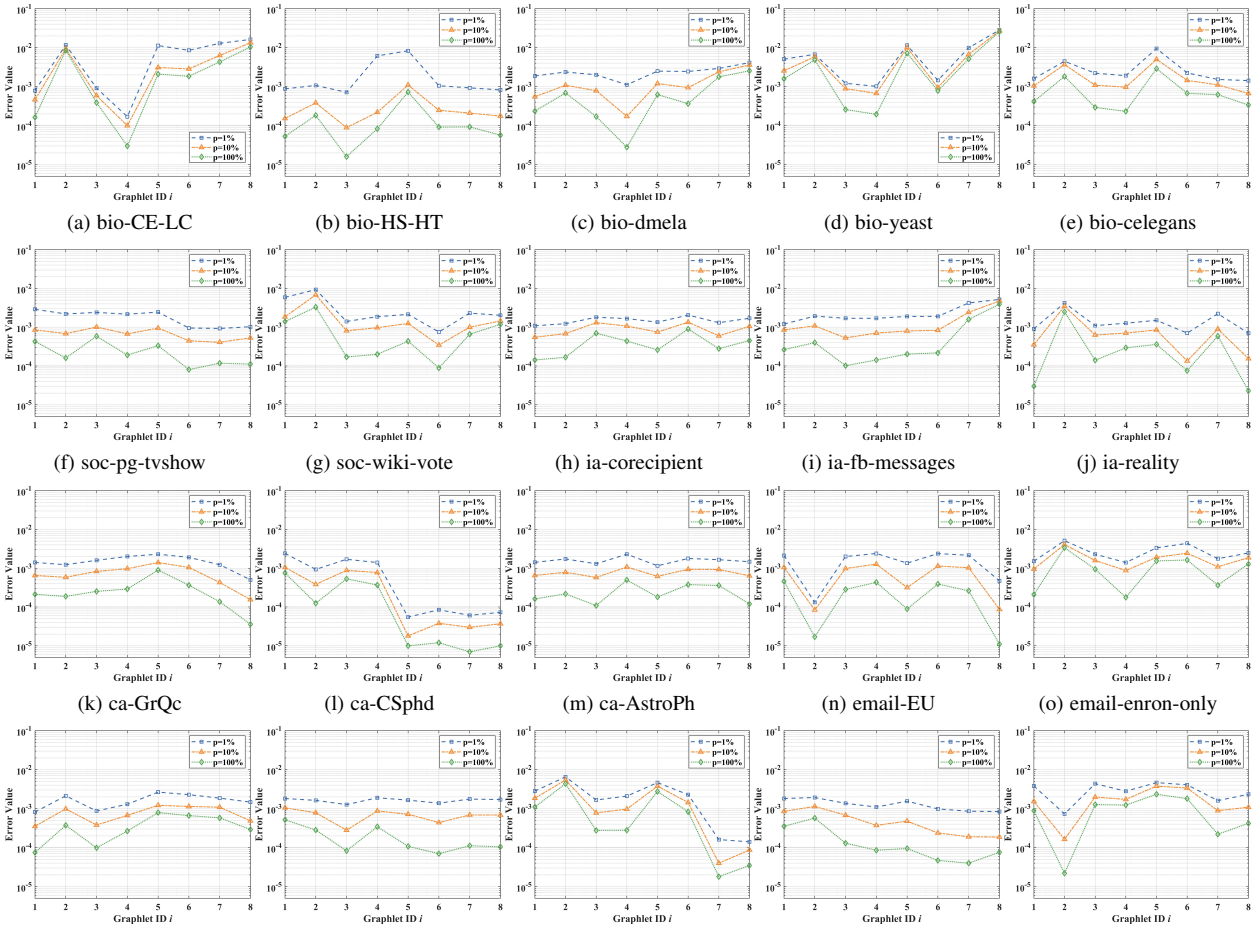
In this experiment, we evaluate the performance of GREET on graph classification tasks using eight datasets from bioinformatics and social networks. Those are PROTEINS, MUTAG, ENZYMES, COX2, IMDB-B, IMDB-MULTI, PTC-FR, and REDDIT-B datasets. Several state-of-the-art methods are used as baselines including:

- *Graph kernel-based methods:* Weisfeiler-Lehman kernels (WL) [41] serves as one of the most prominent graph kernel methods for graph representation learning, which maps the

This article has been accepted for publication in IEEE Transactions on Network Science and Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2022.3216803

TABLE IV
GFD ESTIMATES OF NINE GRAPHLETS FOR VARIOUS REAL-WORLD NETWORKS. ALL GFD ESTIMATES HAVE LESS THAN 10^{-4} ARE UNDERLINED AND THE LARGEST PROPORTION OF GRAPHLET ARE HIGHLIGHTED

Graph									L1-norm
bio-CE-LC	0.0369	0.0005	0.0257	**0.9331**	0.0010	0.0022	0.0005	0.0002	0.0362
bio-HS-HT	0.0213	0.0033	**0.4243**	0.3127	0.0130	0.1625	0.0394	0.0235	0.0026
bio-dmela	0.0266	0.0001	**0.5258**	0.4274	0.0050	0.0145	0.0006	<u>≤0.0001</u>	0.0108
bio-yeast	0.0965	0.0018	0.2636	**0.6132**	0.0012	0.0217	0.0017	0.0003	0.0546
bio-celegans	0.0179	0.0008	0.1277	**0.7486**	0.0012	0.0935	0.0095	0.0008	0.0150
soc-pg-tvshow	0.0277	0.0133	0.2454	**0.2502**	0.0055	0.2652	0.0711	0.1216	0.0056
soc-wiki-vote	0.0373	0.0018	0.3996	**0.4398**	0.0059	0.1058	0.0091	0.0007	0.0145
ia-corecipient	0.0085	0.0027	0.2423	**0.3393**	0.0073	0.3151	0.0582	0.0267	0.0074
ia-fb-messages	0.0260	0.0004	**0.5441**	0.3794	0.0087	0.0396	0.0017	<u>≤0.0001</u>	0.0120
ia-reality	0.0179	<u>≤0.0001</u>	0.0571	**0.9215**	0.0001	0.0033	0.0001	<u>≤0.0001</u>	0.0072
ca-GrQc	0.0400	0.0226	0.2616	0.1918	0.0005	**0.2969**	0.0310	0.1556	0.0061
ca-CSphd	0.1359	0.0002	0.1939	**0.6678**	0.0002	0.0019	0.0001	<u>≤0.0001</u>	0.0030
ca-AstroPh	0.0704	0.0356	**0.3631**	0.3231	0.0009	0.1831	0.0151	0.0087	0.0062
email-EU	0.0075	<u>≤0.0001</u>	0.0513	**0.9325**	0.0001	0.0084	0.0003	<u>≤0.0001</u>	0.0060
email-enron-only	0.0686	0.0128	**0.4122**	0.2430	0.0093	0.2034	0.0394	0.0112	0.0148
email-univ	0.0404	0.0027	**0.5554**	0.2742	0.0063	0.1089	0.0104	0.0017	0.0063
econ-wm1	0.0182	0.0074	0.2507	0.2089	0.0136	**0.3342**	0.1253	0.0418	0.0055
econ-poli	0.1035	0.0008	0.1677	**0.7135**	0.0013	0.0128	0.0004	<u>≤0.0001</u>	0.0144
econ-mahindas	0.0255	0.0003	0.2978	**0.5275**	0.1191	0.0255	0.0043	<u>≤0.0001</u>	0.0041
econ-mbeaflw	0.0105	0.0013	**0.3427**	0.3332	0.1047	0.1428	0.0581	0.0067	0.0145



(a) bio-CE-LC (b) bio-HS-HT (c) bio-dmela (d) bio-yeast (e) bio-celegans

(f) soc-pg-tvshow (g) soc-wiki-vote (h) ia-corecipient (i) ia-fb-messages (j) ia-reality

(k) ca-GrQc (l) ca-CSphd (m) ca-AstroPh (n) email-EU (o) email-enron-only

(p) email-univ (q) econ-wm1 (r) econ-poli (s) econ-mahindas (t) econ-mbeaflw

Fig. 5. Relative error of estimates for graphlet g_i, $1 \leq i \leq 8$, given by algorithms in Section IV-C with sampling factor $p = 1\%, 10\%, 100\%$.

This article has been accepted for publication in IEEE Transactions on Network Science and Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2022.3216803

TABLE V
COMPUTATIONAL TIME OF GREET IN COMPARISON WITH EXISTING GRAPHLET COUNTING METHODS. THE BEST TIME FOR EACH PROBLEM INSTANCE IS IN BOLD

Graph	Brute Force	GUISE	GRAFT	GREET
bio-HS-HT	317.41 s	8.34 s	1.35 s	**0.21 s**
bio-dmela	60.98 s	7.66 s	1.12 s	**0.08 s**
soc-pg-tvshow	203.75 s	6.24 s	0.98 s	**0.08 s**
ia-fb-messages	34.33 s	2.21 s	0.32 s	**<0.01 s**
ca-GrQc	54.08 s	4.71 s	0.55 s	**0.03 s**
ca-AstroPh	–	83.43 s	18.33 s	**1.71 s**
email-EU	–	21.44 s	2.67 s	**0.35 s**
econ-mbeaflw	–	27.68 s	3.01 s	**0.36 s**

original graph to graph sequences with topological and label information. Shortest Path kernel [42] tackles the NP-hard problem that computes all paths and longest paths in a graph and offers a computable solution in polynomial time. Graphlet kernel [23] compare graphs by simply counting graphlets.

- *Graph neural network-based methods:* ~~Graph convolutional networks (GCN) is a variant of convolutional neural networks that operate directly on graphs.~~ DGCNN [43] applies a localized graph convolution model and designs a novel SortPooling layer to achieve ordered training. Graph-U-Nets [44] proposes a novel pooling strategy that employs a projection vector to measure the rank of each node. StructPool [45] models graph pooling as a node clustering problem and leverages conditional random fields (CRF) to capture the relationships among the assignments of different nodes.
- *Graph entropy-based methods:* The classical topological information content $I_{vd} = -\sum_{i=1}^{n} d_i \log d_i$, proposed by Bonchev *et al.* [46], is calculated based on the degree of vertex. Dehmer [28] introduced a graph entropy measure based on the cardinalities of vertex spheres. Aziz *et al.* [16] defined an information functional based on the degree statistics of the graphlet, which are used to estimate the entropy of networks in a transformed space.

For this experiment, the graph entropy-based feature vector introduced in Section III-C is utilized as the input to train the classifier. Considering its simplicity and strong discriminative ability, an easily constructed three-layer fully-connected neural network is adopted as the classifier. The width of hidden layers are set to 32, 32, 16. Dropout with a rate of 0.5 is used to prevent overfitting. Besides, we adopt rectified linear unit (ReLU) and softmax function for the active function in all hidden layers and output layer, respectively. To compare the graph classification accuracies of GREET and the alternate approaches, we perform 10-fold cross-validation with 9 folds for training and the rest for testing the model, as described in [16], [23], [41], [43], [44]. We also employ the Adam optimizer [47] with a learning rate of 0.001 and a batch size of 64 for training. An early stopping mechanism is utilized for better efficiency. Specifically, for the graph entropy-based method proposed in [28], [46], we use the Euclidean distance of two metrics I_{vd} and $I_f(G)$ between two graphs for classification. For other baselines, we use the original implementations and parameter settings released by the authors

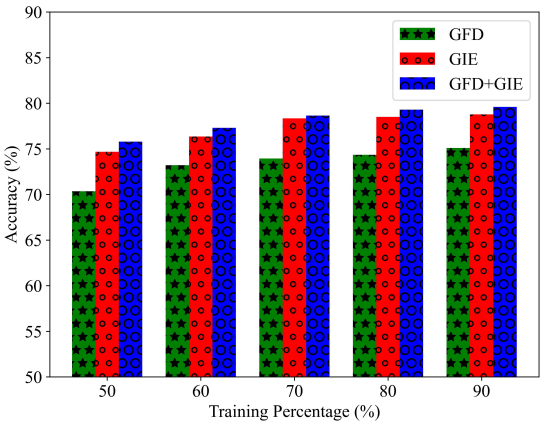


Fig. 6. Accuracy comparison on PROTEINS dataset using different training ratios. The results of simply considering GFD and GIE are indicated by green and red histograms, respectively. The concatenation of GFD and GIE is marked with blue histogram.

to ensure fair comparisons. All the experiments are conducted five times independently. Table VI summarizes the comparison results.

It is observed that our proposed GREET remains on the top for most datasets. Concretely, our method outperforms the previous best approach by margins of 1.94, 3.06, 3.04, 0.99, 2.33, and 4.38 percent on MUTAG, ENZYMES, COX2, IMDB-MULTI, PTC-FR, and REDDIT-B, respectively~~1.04, 1.94, 3.06, 3.04, 2.28, 2.93, and 5.02 percent on PROTEINS, MUTAG, ENZYMES, COX2, IMDB-MULTI, PTC-FR, and REDDIT-B, respectively~~. The competitive performances, particularly on REDDIT-B, which has a larger number of nodes, demonstrate the effectiveness of our method in handling large graphs. It is not surprising since most graph kernel approaches are designed for capturing the shallow~~explicit~~ structural information, while our GREET takes both urelement statistics and higher-order graphlet features into consideration. From an overall perspective, our proposed method also performs well on other small datasets, demonstrating its generalizability. Besides, during the implementation of GNN-based methods, a severe over-fitting issue in training deep graph neural networks has been discovered to yield inferior results. It is worth mentioning that our proposed GREET gains insight into a novel measurement for graph representation, which introduces the graphlet-based entropy to address the problem of the prohibitively high training burden and thus overcome the obstacles for the subsequent tasks. Furthermore, in terms of standard deviation, the fluctuation of GREET is less than most baselines, especially compared to those GNN-based models, which also indicates the stability of our method. Moreover, we find that compared with other datasets, the experiments on ENZYMES have achieved unsatisfactory performance, i.e., the accuracy results on ENZYMES are lower than other datasets. We reckon that it is related to the design of the neural networks, which is not conducive to handling multi-classification tasks. Here, we leave it to future works.

Fig. 6 gives a further comparison of the graph classification performances on dataset PROTEINS with three different features concerned. The first uses only GFD as the input feature,

This article has been accepted for publication in IEEE Transactions on Network Science and Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2022.3216803

TABLE VI
COMPARISONS BETWEEN GREET AND OTHER APPROACHES IN TERMS OF GRAPH CLASSIFICATION ACCURACY (%) ON EIGHT DATASETS INCLUDING BIOINFORMATICS AND SOCIAL DOMAINS. THE BEST AND THE SUBOPTIMAL RESULTS ALONG EACH COLUMN ARE BOLDFACED AND UNDERLINED, RESPECTIVELY

	PROTEINS	MUTAG	ENZYMES	COX2	IMDB-B	IMDB-MULTI	PTC-FR	REDDIT-B
#graphs	1,113	188	600	467	1,000	1,500	351	2,000
#avg.nodes	39.1	17.9	32.6	41.2	19.8	13.0	26.7	429.6
#classes	2	2	6	2	2	3	2	2
WL [41]	74.32±3.13	87.62±5.35	51.30±1.67	76.41±6.07	72.89±3.93	50.87±3.95	67.64±1.74	81.03±3.17
Shortest-Path [42]	73.14±3.05	86.05±2.14	41.53±1.97	76.58±0.71	71.53±3.63	48.23±2.57	66.39±1.73	77.44±3.15
Graphlet [23]	72.58±2.46	85.06±2.76	32.55±2.78	78.16±1.03	72.63±5.49	47.30±3.56	68.57±1.04	80.27±4.51
DGCNN [43]	75.35±2.48	85.83±2.34	65.51±5.32	81.13±5.67	71.91±5.86	46.26±2.84	65.09±5.66	82.56±2.38
Graph-U-Nets [44]	77.54±2.35	87.21±5.57	64.98±4.88	78.17±5.81	**75.48±3.31**	51.18±3.88	66.11±6.55	84.37±1.57
StructPool [45]	**79.13±2.67**	87.51±3.46	63.85±2.91	81.53±4.66	74.08±3.35	52.47±2.58	69.17±3.35	85.97±2.52
BONCHEV. *et al.* [46]	67.35±0.17	82.56±0.97	47.31±0.88	73.45±1.17	68.77±2.02	48.35±1.88	64.73±2.12	80.65±2.34
DEHMER [28]	70.21±0.56	83.18±0.78	48.67±0.59	75.36±1.93	70.31±2.26	50.16±1.91	66.84±2.03	82.24±2.44
AZIZ. *et al.* [16]	77.22±2.38	86.58±2.36	53.77±4.34	78.59±0.83	72.89±4.45	50.41±2.53	68.43±1.83	85.31±2.42
GREET (ours)	78.58±1.71	**89.56±3.42**	**67.57±3.60**	**84.57±2.63**	74.33±3.51	**53.46±2.98**	**71.50±3.42**	**90.35±2.23**

TABLE VII
QUANTITATIVE RESULTS OF AGGLOMERATIVE HIERARCHICAL GRAPH CLUSTERING USING GFD

CLUSTERS	REAL GROUPS					
	Bio.	**Soc.**	**Inter.**	**Collab.**	**Email**	**Econ.**
Bio.	③	①	–	①	–	①
Soc.	–	①	–	①	–	①
Inter.	–	–	③	–	①	–
Collab.	②	–	–	①	–	①
Email	–	–	–	–	②	–
Econ.	–	–	–	–	–	①

[1] The circled numbers indicate the amount of graphs assigned to each cluster.
[2] $Purity = \frac{3+1+3+1+2+1}{20} = 0.55$

TABLE VIII
QUANTITATIVE RESULTS OF AGGLOMERATIVE HIERARCHICAL GRAPH CLUSTERING USING GIE

CLUSTERS	REAL GROUPS					
	Bio.	**Soc.**	**Inter.**	**Collab.**	**Email**	**Econ.**
Bio.	⑤	–	–	–	①	–
Soc.	–	①	–	–	–	–
Inter.	–	①	③	–	①	–
Collab.	–	–	–	③	–	–
Email	–	–	–	–	①	–
Econ.	–	–	–	–	–	④

[1] $Purity = \frac{5+1+3+3+1+4}{20} = 0.85$

the second merely utilizes GIE as the input, and the third takes both GFD and GIE into consideration. We can observe that the concatenate feature vectors always achieve better performance than only GFD or GIE. In addition, we test the sensitivity of these variations with different training ratios. From Fig. 6, we can observe that the classification accuracy improves by 5.43, 4.13, 4.74, 4.96, and 4.51 percent with 50%, 60%, 70%, 80%, and 90% training ratio, respectively, when GIE is considered as complement term. These experiments further prove that our proposed graph entropy features can greatly promote the performance of graph classification tasks.

C. GIE as a Property for Graph-Clustering

In this section, we also conduct experiments to demonstrate the usability of GIE for clustering graphs into their functional domains. To achieve this, we compare the performance of only GIE or GFD considered methods on twenty datasets using the mainstream agglomerative clustering technique. The Euclidean distance is utilized as the distance metric. Here, we consider a total of twenty graphs from six domains (Table III). Six graphs from the gene functional associations of biological networks, two graphs from the user social information networks, three graphs from the interaction networks, three graphs from the collaboration networks, three graphs from the email networks, and four graphs built from the economic networks.

To assign original graphs into six clusters, we first calculate the GFD and GIE vectors of all graphs using the proposed estimation method. Obviously, graphs with the same category have GFD and GIE features that are likely to be comparable. In this experiment, $Purity$ is chosen as the evaluation criteria, which is a primary validation measure to determine the cluster quality. It is defined as $\sum_{i=1}^{K} \frac{m_i}{m} \max(p_{ij})$, where K and m are the number of clusters and the total number of members involved in the whole clustering, respectively. p_{ij} denotes the probability that network j is assigned to cluster i.

In Table VII and Table VIII, we report the quantitive results of agglomerative hierarchical graph clustering using GFD and GIE, respectively. It can be observed that the $purity$ increases from 0.55 to 0.85 when we take the GIE as the input. From a numerical point of view, the estimated GFD patterns of graphs vary even if they belong to the same category. As the commonality of GFD between two networks decreases, the clustering results become unreliable, and consequently cannot be applied to cluster a graph into its functional domain. The advantage of our proposed GIE lies in the exploration of both urelement and high-order properties that provide multiple perspectives for us to obtain the underlying structural information. To sum up, this experiment demonstrates that GIE can also be used in unsupervised graph clustering tasks with acceptable performance.

V. Conclusion

This paper proposed GREET, a general entropy-based framework with efficient graphlet estimation algorithms for graph characterization. Firstly, an enclosing edge-centric neighborhood is constructed for exploring basic 3-, 4-vertex tree and cyclic graphlets, which leads to the high efficiency for further graphlet counting. Secondly, we propose two novel information functionals with both urelement and higher-order features considered, which are shown to be effective in understanding the structural information of networks. Then the resulting graph information entropy vector is the combination of its local and global graph entropy, which has proven its superiority in generating comprehensive network representation. Moreover, GREET also has many interchangeable parts, which enable its generalization to other applications. For example, the procedure of estimating graphlets can be replaced by alternative approaches, thus improving its flexibility and scalability. Experimental results on over twenty datasets demonstrate that our proposed method is able to calculate graphlets with high efficiency and low error, and the experiments on the graph classification task show that GREET outperforms current approaches with higher accuracy. In particular, the existing graph representation methods can also benefit from the proposed GIE in other domains.

The findings of this work give rise to many important future research directions. For instance, it would be valuable to extend the definition of local and global information functionals for directed or attribute networks. Besides, it is of great significance to measure the similarity between graph entropy and structural information so as to approximate entropies with less entropy gap. Moreover, we also intend to investigate the upper and lower bounds for the entropy measure in future work, which is still in its infancy.

References

[1] C.-H. Zeng and K.-C. Chen, "Social network analysis facilitates cognition in large wireless networks: clustering coefficient aided geographical routing," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 3, pp. 618–634, 2018.

[2] K. Venugopalachary and V. K. Chakka, "A small-scale wireless distributed cooperative secure communication network design using graph fir filters," *IEEE Sensors Letters*, vol. 5, no. 4, pp. 1–4, 2021.

[3] A. Manada and H. Morita, "Graph theoretical analysis on distributed line graphs for peer-to-peer networks," *IEEE Transactions on Network Science and Engineering*, vol. 6, no. 3, pp. 326–335, 2018.

[4] F. Xia, S. Yu, C. Liu, J. Li, and I. Lee, "Chief: Clustering with higher-order motifs in big networks," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 3, pp. 990–1005, 2022.

[5] X. Mei, X. Cai, L. Yang, and N. Wang, "Relation-aware heterogeneous graph transformer based drug repurposing," *Expert Systems with Applications*, vol. 190, p. 116165, 2022.

[6] Y. Liu, M. K. Ng, and S. Wu, "Multi-domain networks association for biological data using block signed graph clustering," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 17, no. 2, pp. 435–448, 2018.

[7] K. R. Žalik and B. Žalik, "Memetic algorithm using node entropy and partition entropy for community detection in networks," *Information Sciences*, vol. 445, pp. 38–49, 2018.

[8] K. Anand, G. Bianconi, and S. Severini, "Shannon and von neumann entropy of random networks with heterogeneous expected degree," *Physical Review E*, vol. 83, no. 3, p. 036109, 2011.

[9] A. Li and Y. Pan, "Structural information and dynamical complexity of networks," *IEEE Transactions on Information Theory*, vol. 62, no. 6, pp. 3290–3339, 2016.

[10] L. Han, F. Escolano, E. R. Hancock, and R. C. Wilson, "Graph characterizations from von neumann entropy," *Pattern Recognition Letters*, vol. 33, no. 15, pp. 1958–1967, 2012.

[11] X. Liu, L. Fu, X. Wang, and C. Zhou, "On the similarity between von neumann graph entropy and structural information: Interpretation, computation, and applications," *IEEE Transactions on Information Theory*, 2022.

[12] G. Luo, J. Li, H. Peng, C. Yang, L. Sun, P. S. Yu, and L. He, "Graph entropy guided node embedding dimension selection for graph neural networks," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, 2021, pp. 2767–2774.

[13] M. Dehmer, "A novel method for measuring the structural information content of networks," *Cybernetics and Systems: An International Journal*, vol. 39, no. 8, pp. 825–842, 2008.

[14] C. Ye, R. C. Wilson, C. H. Comin, L. d. F. Costa, and E. R. Hancock, "Approximate von neumann entropy for directed graphs," *Physical Review E*, vol. 89, no. 5, p. 052804, 2014.

[15] E. Trucco, "A note on the information content of graphs," *The bulletin of mathematical biophysics*, vol. 18, no. 2, pp. 129–135, 1956.

[16] F. Aziz, M. S. Akbar, M. Jawad, A. H. Malik, M. I. Uddin, and G. V. Gkoutos, "Graph characterisation using graphlet-based entropies," *Pattern Recognition Letters*, vol. 147, pp. 100–107, 2021.

[17] Q. Xuan, J. Wang, M. Zhao, J. Yuan, C. Fu, Z. Ruan, and G. Chen, "Subgraph networks with application to structural feature space expansion," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 6, pp. 2776–2789, 2019.

[18] S.-Y. Liu, J. Xiao, and X.-K. Xu, "Link prediction in signed social networks: From status theory to motif families," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 3, pp. 1724–1735, 2019.

[19] R. A. Rossi, A. Rao, S. Kim, E. Koh, N. K. Ahmed, and G. Wu, "From closing triangles to higher-order motif closures for better unsupervised online link prediction," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 4085–4093.

[20] B. Shi, J. Zhong, H. Qiu, Q. Bao, K. Liu, and J. Liu, "Hybrid embedding via cross-layer random walks on multiplex networks," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 1815–1827, 2021.

[21] K. Tu, J. Li, D. Towsley, D. Braines, and L. D. Turner, "gl2vec: Learning feature representation using graphlets for directed networks," in *Proceedings of the 2019 IEEE/ACM international conference on advances in social networks analysis and mining*, 2019, pp. 216–221.

[22] K. Paramonov, D. Shemetov, and J. Sharpnack, "Estimating graphlet statistics via lifting," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 587–595.

[23] N. Shervashidze, S. Vishwanathan, T. Petri, K. Mehlhorn, and K. Borgwardt, "Efficient graphlet kernels for large graph comparison," in *Artificial intelligence and statistics*. PMLR, 2009, pp. 488–495.

[24] M. Rahman, M. A. Bhuiyan, and M. Al Hasan, "Graft: An efficient graphlet counting method for large graph analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 10, pp. 2466–2478, 2014.

[25] M. Rahman, M. A. Bhuiyan, M. Rahman, and M. A. Hasan, "Guise: a uniform sampler for constructing frequency histogram of graphlets," *Knowledge and information systems*, vol. 38, no. 3, pp. 511–536, 2014.

[26] T. Hočevar and J. Demšar, "A combinatorial approach to graphlet counting," *Bioinformatics*, vol. 30, no. 4, pp. 559–565, 2014.

[27] O. Kuchaiev, A. Stevanović, W. Hayes, and N. Pržulj, "Graphcrunch 2: software tool for network modeling, alignment and clustering," *BMC bioinformatics*, vol. 12, no. 1, pp. 1–13, 2011.

[28] M. Dehmer, "Information-theoretic concepts for the analysis of complex networks," *Applied Artificial Intelligence*, vol. 22, no. 7-8, pp. 684–706, 2008.

[29] P. Wang, J. Zhao, X. Zhang, Z. Li, J. Cheng, J. C. Lui, D. Towsley, J. Tao, and X. Guan, "Moss-5: A fast method of approximating counts of 5-node graphlets in large graphs," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 1, pp. 73–86, 2017.

[30] C. E. Shannon, "A mathematical theory of communication," *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

[31] J. A. Bondy, U. S. R. Murty *et al.*, *Graph theory with applications*. Macmillan London, 1976, vol. 290.

[32] U. Brandes, *Network analysis: methodological foundations*. Springer Science & Business Media, 2005, vol. 3418.

[33] M. Jha, C. Seshadhri, and A. Pinar, "Path sampling: A fast and provable method for estimating 4-vertex subgraph counts," in *Proceedings of the 24th international conference on world wide web*, 2015, pp. 495–505.

[34] R. A. Rossi, R. Zhou, and N. K. Ahmed, "Estimation of graphlet counts in massive networks," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 1, pp. 44–57, 2018.

[35] N. K. Ahmed, J. Neville, R. A. Rossi, and N. Duffield, "Efficient graphlet counting for large networks," in *2015 IEEE International Conference on Data Mining*. IEEE, 2015, pp. 1–10.

[36] D. G. Horvitz and D. J. Thompson, "A generalization of sampling without replacement from a finite universe," *Journal of the American statistical Association*, vol. 47, no. 260, pp. 663–685, 1952.

[37] S. Maharaj, Z. Ohiba, and W. Hayes, "Comparing different graphlet measures for evaluating network model fits to biogrid ppi networks," in *International Conference on Algorithms for Computational Biology*. Springer, 2019, pp. 52–67.

[38] A. Mowshowitz and M. Dehmer, "Entropy and the complexity of graphs revisited," *Entropy*, vol. 14, no. 3, pp. 559–570, 2012.

[39] S. Cao, M. Dehmer, and Y. Shi, "Extremality of degree-based graph entropies," *Information Sciences*, vol. 278, pp. 22–33, 2014.

[40] R. A. Rossi and N. K. Ahmed, "An interactive data repository with visual analytics," *ACM SIGKDD Explorations Newsletter*, vol. 17, no. 2, pp. 37–41, 2016.

[41] N. Shervashidze, P. Schweitzer, E. J. Van Leeuwen, K. Mehlhorn, and K. M. Borgwardt, "Weisfeiler-lehman graph kernels." *Journal of Machine Learning Research*, vol. 12, no. 9, 2011.

[42] K. M. Borgwardt and H.-P. Kriegel, "Shortest-path kernels on graphs," in *Fifth IEEE international conference on data mining (ICDM'05)*. IEEE, 2005, pp. 8–pp.

[43] M. Zhang, Z. Cui, M. Neumann, and Y. Chen, "An end-to-end deep learning architecture for graph classification," in *Thirty-second AAAI conference on artificial intelligence*, 2018.

[44] H. Gao and S. Ji, "Graph u-nets," in *international conference on machine learning*. PMLR, 2019, pp. 2083–2092.

[45] H. Yuan and S. Ji, "Structpool: Structured graph pooling via conditional random fields," in *Proceedings of the 8th International Conference on Learning Representations*, 2020.

[46] D. Bonchev and G. A. Buck, "Quantitative measures of network complexity," in *Complexity in chemistry, biology, and ecology*. Springer, 2005, pp. 191–235.

[47] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, 2015.

Ru Huang received his B.S. degree from Nanjing University, Nanjing, China, in 1999 and his Ph.D. degree in circuit and system from Shanghai Jiao Tong University, Shanghai, China, in 2008. He was a visiting scholar at the University of Wisconsin-Madison, WI, USA, from March 31, 2015 to March 30, 2016. He is currently an associate professor of electronics and communication engineering at East China University of Science and Technology, Shanghai, China. His current research interests include wireless sensor networks, complex networks, and deep learning.

Zijian Chen received the B.S. degree from Wenzhou University, Wenzhou, China, in 2020. He is currently working toward the M.S. degree with the School of Information Science and Engineering, East China University of Science and Technology, Shanghai, China. His research interests include complex network modeling, graph data mining, and machine learning.

Guangtao Zhai (Senior Member, IEEE) received the B.E. and M.E. degrees from Shandong University, Shandong, China, in 2001 and 2004, respectively, and the Ph.D. degree from Shanghai Jiao Tong University, Shanghai, China, in 2009. From 2008 to 2009, he was a Visiting Student with the Department of Electrical and Computer Engineering, McMaster University, Hamilton, ON, Canada, where he was a Post-Doctoral Fellow from 2010 to 2012. From 2012 to 2013, he was a Humboldt Research Fellow with the Institute of Multimedia Communication and Signal Processing, Friedrich Alexander University of Erlangen–Nuremberg, Germany. He is currently a Research Professor with the Institute of Image Communication and Information Processing, Shanghai Jiao Tong University. His research interests include multimedia signal processing and perceptual signal processing. He received the Award of National Excellent Ph.D. Thesis from the Ministry of Education of China in 2012.

Jianhua He (Senior Member, IEEE) received the Ph.D. degree from Nanyang Technological University, Singapore, in 2002. He is currently a Reader with the University of Essex, Colchester, U.K. He has authored or coauthored more than 150 research papers in international journals and conferences in his main research field, which includes wireless communications and networks, connected vehicles, autonomous driving, Internet of Things, mobile edge computing, data analytics, AI, and machine learning. He is the Workshop Chair of MobiArch'20 and ICAV'21, a Steering Committee Member of MobiArch'21, and a Member of Editorial Board for several international journals. He is the Coordinator of EU Horizon2020 projects COSAFE and VESAFE on cooperative connected autonomous vehicles.

Xiaoli Chu (Senior Member, IEEE) received the B.Eng. degree in electronic and information engineering from Xi'an Jiaotong University in 2001 and the Ph.D. degree in electrical and electronic engineering from The Hong Kong University of Science and Technology in 2005. From 2005 to 2012, she was with the Centre for Telecommunications Research, Kings College London. She is currently a Professor with the Department of Electronic and Electrical Engineering, The University of Sheffield, U.K. She has coauthored over 180 peer-reviewed journals and conference papers. She co-authored/co-edited the books *Fog-Enabled Intelligent IoT Systems* (Springer 2020), *Ultra-Dense Networks for 5G and Beyond* (Wiley 2019), *Heterogeneous Cellular Networks: Theory*, *Simulation and Deployment* (Cambridge University Press 2013), and *4G Femtocells: Resource Allocation and Interference Management* (Springer 2013). She was a co-recipient of the IEEE Communications Society 2017 Young Author Best Paper Award. She received the IEEE COMMUNICATIONS LETTERS Exemplary Editor Award in 2018. She is a Senior Editor for the IEEE WIRELESS COMMUNICATIONS LETTERS and an Editor for the IEEE COMMUNICATIONS LETTERS.