

Same Stats, Different Graphs: Exploring the Space of Graphs in Terms of Graph Properties

Hang Chen , Utkarsh Soni, Yafeng Lu , Vahan Huroyan , Ross Maciejewski , and Stephen Kobourov 

Abstract—Data analysts commonly utilize statistics to summarize large datasets. While it is often sufficient to explore only the summary statistics of a dataset (e.g., min/mean/max), Anscombe’s Quartet demonstrates how such statistics can be misleading. We consider a similar problem in the context of graph mining. To study the relationships between different graph properties, we examine low-order non-isomorphic graphs and provide a simple visual analytics system to explore correlations across multiple graph properties. However, for larger graphs, studying the entire space quickly becomes intractable. We use different random graph generation methods to further look into the distribution of graph properties for higher order graphs and investigate the impact of various sampling methodologies. We also describe a method for generating many graphs that are identical over a number of graph properties and statistics yet are clearly different and identifiably distinct.

Index Terms—Graph mining, graph properties, graph generators

1 INTRODUCTION

STATISTICS are often used to summarize a large dataset. In a way, one hopes to find the “most important” statistics that capture one’s data. For example, when comparing two countries, we often specify the population size, GDP, employment rate, etc. The idea is that if two countries have a “similar” statistical profile, they are similar (e.g., France and Germany have a more similar demographic profile than France and USA). However, Anscombe’s quartet [1] convincingly illustrates that datasets with the same values over a limited number of statistical properties can be fundamentally different—a great argument for the need to visualize the underlying data; see Fig. 1.

Similarly, in the graph analytics community, a variety of properties and statistics are being used to summarize graphs, such as graph density, average path length, global clustering coefficient, etc. However, summarizing a graph with a fixed set of graph properties leads to the problem illustrated by Anscombe. It is easy to construct several graphs that have the same properties (e.g., number of vertices, number of edges, number of triangles, girth, clustering coefficient) while the underlying graphs are clearly different and identifiably distinct; see Fig. 2. From a graph theoretical point of view, these graphs are very different: they differ in connectivity, planarity, symmetry, and other properties.

- H. Chen, V. Huroyan, and S. Kobourov are with the Department of Computer Science, The University of Arizona, Tucson, AZ 85721 USA.
E-mail: hangchen@email.arizona.edu, vahanhuroyan@math.arizona.edu, kobourov@cs.arizona.edu.
- U. Soni, Y. Lu, and R. Maciejewski are with School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Phoenix, AZ 85004 USA. E-mail: {usoni1, lyafeng, rmacieje}@asu.edu.

Manuscript received 5 Dec. 2018; revised 27 Sept. 2019; accepted 1 Oct. 2019.
Date of publication 10 Oct. 2019; date of current version 28 Jan. 2021.

(Corresponding author: Hang Chen.)

Recommended for acceptance by S. C North.

Digital Object Identifier no. 10.1109/TVCG.2019.2946558

Recently, Matejka and Fitzmaurice [2] proposed a dataset generation method that can modify a given 2-dimensional point set (like the ones in Anscombe’s quartet) while preserving its summary statistics but changing its visualization (what they call “graph”). Given the graphs in Fig. 2, we consider whether it is also possible to modify a given graph and preserve a given set of graph properties while changing other graph properties. Note that the problem is much easier for 2D point sets and statistics, such as mean, deviation and correlation, than for graphs where many graph properties are structurally correlated (e.g., diameter and average path length). With this in mind, we first consider how can we fix a few graph properties (such as the number of nodes, number of edges, number of triangles) and vary another property (such as the clustering coefficient or connectivity). We find that there is a spectrum of possibilities. Sometimes the “unrestricted” properties can vary dramatically, sometimes not, and the outcome depends on two issues: (1) the inherent correlation between some properties (e.g., density and number of triangles), and (2) the bias in graph generators.

We begin by studying the correlation between graph properties across the set of all non-isomorphic graphs with up to 10 vertices. Recall that two vertex-labeled graphs are isomorphic if just by relabeling the vertices of one of the graphs we can obtain the other. Thus, two non-isomorphic graphs must be structurally different. The statistical properties derived for all graphs for a fixed number of vertices provide further information about certain “restrictions.” In other words, the range of one property may be restricted if another property is fixed. However, we cannot explore the entire space of graph properties and correlations. As the number of vertices grows, the number of different non-isomorphic graphs grows super-exponentially. For $|V| = 1, 2 \dots 9$ the numbers are 1, 2, 4, 11, 34, 156, 1044, 12346, 274668, but already for $|V| = 16$ we have 6×10^{22} non-isomorphic graphs.

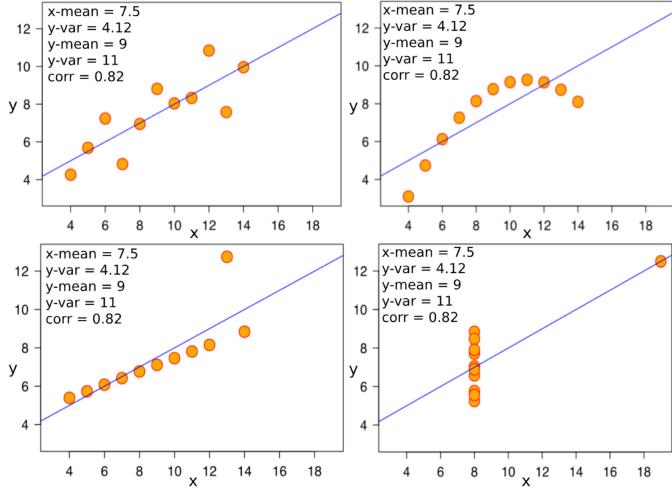


Fig. 1. Anscombe's quartet: All four datasets have the same mean and standard deviation in x and y and (x, y) -correlation.

To go beyond ten vertices, we use graph generators based on models, such as Erdős-Rényi and Watts-Strogatz. However, different graph generators have different biases and these can impact the results. We study the extent to which sampling using random generators can represent the whole graph set for an arbitrary number of vertices with respect to their coverage of the graph properties. One way to evaluate the performance of random generators is to compare the generated graphs to the total set of non-isomorphic graphs, which is available for $|V| \leq 10$ vertices. If we randomly generate a small set of graphs (also for $|V| \leq 10$ vertices) using a given graph generator, we can explore how well the sample and generator cover the space of graph properties. In this way, we can begin exploring the problem "same stats, different graphs" for larger graphs.

We have put together a visual analytics exploration tool for the space of all low-order (≤ 10) non-isomorphic graphs and sampled higher order graphs. We include a generator for "same stats, different graphs," i.e., multiple graphs that are identical over a number of graph properties, yet are clearly different. Data and tools are available at <http://differentgraphs.cs.arizona.edu>. This work illustrates the challenges associated with analyzing graphs based solely on sampling summary properties. Overcoming such challenges is critical for network analysis, graph mining, etc., where the analysis of graphs is often done at the graph property level, which can miss critical topological information.

1.1 Definitions and Conventions

The object of study in this paper are simple, undirected, unweighted graphs, $G = (V, E)$ with $|V|$ nodes and $|E|$ edges. Specifically, we study the set of non-isomorphic graphs, considering each of them as a high-dimensional point. Recall that two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are isomorphic if they are identical up to relabeling their vertices; that is, if there exists a bijection $f : V_1 \rightarrow V_2$ such that for any $u_1, v_1 \in V_1$ and $\{u_1, v_1\} \in E_1$ if and only if $\{f(u_1), f(v_1)\} \in E_2$.

Throughout the paper we use the term *graph properties* to refer to the properties of a graph (e.g., diameter, density). We also use the term *statistics* when describing a graph via a collection of graph properties (e.g., the values

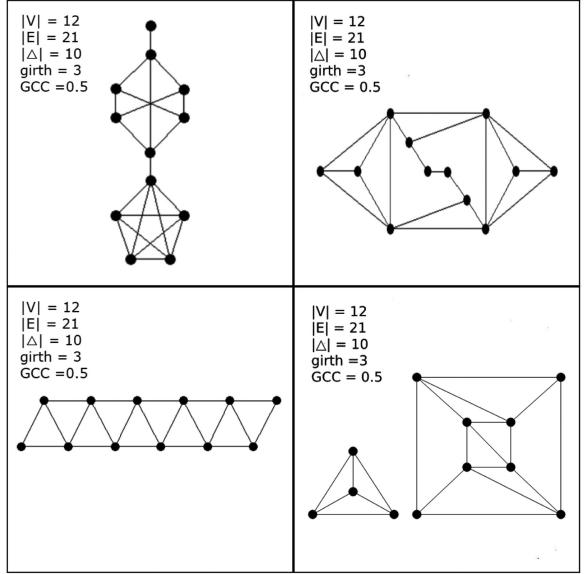


Fig. 2. These four graphs share the same 5 common properties: $|V| = 12$, $|E| = 21$, number of triangles $|\Delta| = 10$, girth = 3 and global clustering coefficient $GCC = 0.5$. However, structurally the graphs are very different: some are planar, others are not, some show regular patterns and are symmetric, others are not, and finally, one of the graphs is disconnected, another is 1-connected and the rest are 2-connected.

for diameter, density). The 10 properties under consideration are discussed in Section 2.1. By *ground truth* dataset we refer to the dataset in \mathbb{R}^{10} that represents the set of 10 properties of all non-isomorphic graphs with fixed number of vertices $|V|$.

Note that the set of all non-isomorphic graphs is not in a bijection with the ground truth dataset in 10 dimensional space. A pair of non-isomorphic graphs does indeed correspond to a pair of distinct points in \mathbb{R}^{10} . On the other hand, a pair of non-isomorphic graphs that share the same 10 graph properties get mapped to the same point in \mathbb{R}^{10} , which is an instance of the titular "same stats, different graphs."

1.2 Structure of This Paper

This paper is organized as follows: Section 2 summarizes related work; Section 3 presents preliminary experiments and findings about low-order graphs; Section 4 discusses methods of generating different graphs with same or similar properties; Section 5 discusses different measures for estimating graph generators from the point of view of coverage and representation of the underlying space of non-isomorphic graphs; Section 6 compares the coverage and representation of the graph generators used in the paper. We conclude with a brief discussion and directions for future work in Section 7.

2 RELATED WORK

There is a great deal of related work, but here we focus on graph properties that are studied in the graph analysis and graph mining literature, a review of the major random graph generators, and work on exploration and visualization of graph properties.

TABLE 1
The Set of Graph Properties Considered in this Paper

Name	Formula	Reference
Average Clustering Coefficient	$ACC(G) = \frac{1}{n} \sum_{i=1}^n c(u_i), u_i \in V, n = V $ $c(u_i) = \frac{2T(u_i)}{\binom{\deg(u_i)}{2}}$, where $T(u_i)$ is the number of triangles through node u_i	[3], [4], [5], [6], [7]
Global Clustering Coefficient	$GCC(G) = \frac{3 \times \text{triangles} }{ \text{connected triples} }$	[5], [6]
Square Clustering	$SCC(G) = \frac{1}{n} \sum_{i=1}^n c_4(u_i), u_i \in V, n = V $ where $c_4(u_i)$ is the quotient between the number of squares which u_i participated and the total number of possible squares	[8]
Average Path Length	$APL = \text{mean}\left\{ \frac{\sum_{v \in V} d(u, v), u \neq v}{n-1} \right\}$	[3], [4], [6], [7]
Degree Assortativity	$r = \frac{\sum_{xy} x y (e_{xy} - a_x b_y)}{\sigma_a \sigma_b}$	[7], [9]
Diameter	$\text{diam}(G) = \max\{\text{dist}(v, w), v, w \in V\}$	[3], [5], [7], [10]
Density	$\text{den} = \frac{2 E }{ V (V -1)}$	
Ratio of Triangles	$Rt = \frac{ \text{triangles} }{\binom{ V }{3}}$	
Node Connectivity	Cv: the minimum number of nodes to remove to disconnect the graph	[11]
Edge Connectivity	Ce: the minimum number of edges to remove to disconnect the graph	[11]

2.1 Graph Properties

Graph mining is applied in different domains from bioinformatics and chemistry, to software engineering and social science. Essential to graph mining is the efficient calculation of various graph properties (e.g., diameter, density) and summary statistics (e.g., averages, modes) that can provide useful insight about the structural properties of a graph. For consistency, we refer to the graph properties and summary statistics as *graph properties*. We reviewed recent graph mining systems and identified some of the most frequently extracted graph properties. We list those, along with their definitions, in Table 1. These properties range from basic, e.g., vertex count and edge count, to complex, e.g., clustering coefficients [3], [4], [5], [6], [7] and average path length [3], [4], [6], [7]. Many of them can be used to derive further graph properties. For example, graph density can be determined directly as the ratio of the number of edges $|E|$ to the maximum number of edges possible $|V| \times (|V| - 1)/2$, and real-world graphs are often found to have a low graph density [12]. Node connectivity and edge connectivity may be used to describe the resilience of a graphs [13], [14], and graph diameter [15] captures the maximum among all pairs of shortest paths [16], [17].

Other graph properties measure how tightly nodes are grouped in a graph. For example, clustering coefficients have been used to describe many real-world graphs and can be measured locally and globally. Nodes in a highly connected clique tend to have a high local clustering coefficient, and a graph with clear clustering patterns will have a high global clustering coefficient [18], [19], [20], [21]. Studies have shown that the global clustering coefficient has been found to typically be larger in real-world graphs than in Erdős-Rényi graphs with the same number of vertices and edges [6], [21], [22], and a small-world graph should have a relatively large average clustering coefficient [23], [24], [25]. Small-world graphs also have an average path length (APL) that is logarithmic in the number of vertices, while real-world graphs have small (often constant) APL [21], [22], [23], [24], [25], [26].

Degree distribution is one frequently used property describing the graph degree statistics. Many real-world graphs, including communication, citation, biological and

social graphs, have been found to follow a power-law shaped degree distribution [6], [21], [27]. Other real world graphs have been found to follow an exponential degree distribution [28], [29], [30]. Degree assortativity is of particular interest in the study of social graphs and is calculated based on the Pearson correlation between the vertex degrees of connected pairs [31]. A random graph generated by Erdős-Rényi model has an expected assortative coefficient of 0. Newman [31] extensively studied assortativity in real-world graphs and found that social networks often have positive assortativity, i.e., vertices with a similar degree preferentially connect together, whereas technological and biological graphs tend to have negative assortativity implying that vertices with a smaller degree tend to connect to high degree vertices. Assortativity has been shown to affect clustering [32], resilience [31], and epidemic-spread [33].

Note that there are many other graph properties, but many of them are local, defined on the level of individual vertices, or individual edges. Examples include degree centrality of a vertex, betweenness centrality of a vertex or an edge, etc. Since we are interested in global properties, properties of the graph as a whole, we cannot directly use such local properties. We represent graphs by the 10 properties in Table 1, i.e., we represent each graph as a single data point in \mathbb{R}^{10} .

2.2 Graph Generators

Graph properties have been used to describe various classes of graphs (e.g., geometric, small-world, scale-free) and a variety of algorithms have been developed to automatically generate graphs that mimic these various properties. Charabati et al. [3] divide graph models and generators into the following broad categories:

- 1) Random Graph Models: The graphs are generated by a random process.
- 2) Preferential Attachment Models: In these models, the “rich get richer,” as the graph grows, leading to power law effects.
- 3) Geographical Models: These models consider the effects of geography (i.e., the positions of the nodes)

TABLE 2

This Table Demonstrates the Relationship between the Graph Nodes $|V| = n$, the Size of the Sample k and the Probability of Having Two Isomorphic Graphs in the Sample Calculated by the Formula (1) Defined in Section 2.2

n	5	6	7	8	9	10	11	12	13	14	15	16
k	2	2	4	14	60	440	4,240	55,500	985,000	23,800,000	787,000,000	35,600,000,000
probability	0.88	0.98	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99

on the topology of the graph. This is relevant for modeling router or power grid graphs.

The Erdős-Rényi (ER) graph model is a simple graph generation model [6] that creates graphs either by choosing a graph randomly with equal probability from a set of all possible graphs of size $|V|$ with $|E|$ edges [34] or by creating each possible edge of a graph with $|V|$ vertices with a given probability p [35]. The latter process gives a binomial degree distribution that can be approximated with a Poisson distribution. Note that fixing the number of nodes and using $p = 1/2$ results in a uniform sampling from the space of isomorphic graphs. Even though this graph generator does not sample uniformly from the space of non-isomorphic graphs, the probability of having two isomorphic graphs if we generate k graphs with $|V| = n$ according to ER graph generator with $p = 1/2$ is greater than or equal to

$$1 - \binom{k}{2} * \frac{n!}{2^{n(n-1)/2}}. \quad (1)$$

Table 2 shows the number of graphs for fixed number of vertices $|V|$ that one needs to generate to have a high probability of avoiding two isomorphic graphs in the sample.

The Watts and Strogatz [25] (WS) model addresses a limitation of the ER model. Specifically, the WS model can be used to generate graphs that exhibit small-world properties and that have higher clustering coefficients. However, the WS model can generate disconnected graphs. We utilize the variation suggested by Newman and Watts [36] to ensure connectivity as some graph properties are not well defined for disconnected graphs.

It is also possible to create graphs where the degrees follow other common probability distributions, e.g., exponential [37] or Gaussian [38]. Graphs with any given degree sequence can be generated using the configuration model [21]. Models have also been proposed for generating synthetic scale-free graphs with a scaling exponent [39].

The model proposed by Gilbert [40], the geometric model (GE), places nodes according to a Poisson point process in some metric space (e.g., the unit square in 2D), and adds edges between pairs of nodes that are within a pre-specified distance threshold. Barabasi and Albert (BA) [41] described another popular model for generating undirected graphs. It is a graph growth model in which each added vertex has a fixed number of edges $|E|$, and the probability of each edge connecting to an existing vertex v is proportional to the degree of v . Dorogovtsev et al. [42] and Albert and Barabasi [43] also developed a variation of the BA model with a tunable scaling exponent. We use the NetworkX version of the BA model in this paper, which is a direct implementation of the original BA model [41].

2.3 Generating Non-Isomorphic Graphs

The graph generators in the previous subsection generate graphs by sampling the space of all isomorphic graphs. However, there is also work on generating graphs from the set of all non-isomorphic graphs with fixed number of vertices. McKay [44] proposes an isomorphic-free generation method, which is known as canonical deletion. Such generators rely on an algorithm (implemented in the Nauty program) to efficiently test whether two graphs are isomorphic [45]. The algorithm has been revised to improve performance [46] (in terms of running time); for example, testing whether two graphs on 100 vertices are isomorphic takes 1 second on average. Goldberg [47] proposes several algorithms to generate non-isomorphic graphs uniformly at random. Even though the algorithm provides good theoretical guarantees on the running time and space, it is practical only for low-order graphs. There are other methods to generate graphs from the set of all non-isomorphic graphs uniformly at random [48], [49], but they are also not practical beyond low-order graphs.

2.4 Exploring Graph Properties

Bach et al. [50] introduce an interactive system to create random graphs that match user-specified properties based on a genetic algorithm. The properties considered are $|V|$, $|E|$, average vertex degree, number of components, diameter, ACC, density, and the number of clusters (as defined by Newman and Girvan [51]). The goal is to generate graphs that get as close as possible to a set of target properties; however, there are no guarantees that the target values can be obtained. Somewhat differently, we are interested in creating graphs that match several target properties exactly, but differ drastically in other parameters.

Kennedy et al. [52] provide an interactive graph analysis system called Graph Landscape, which allow researchers to explore graphs, graph sets, and benchmark collections regarding their properties. Unlike our paper, the system aims to enable the analysis of differences and similarities between different sets of graphs and to assess their value for experimental evaluations.

Also related is work on graph anonymization, where the goal is to generate one or more graphs with same set of fixed properties as those in a given source graph [53], [54], [55]. As the given graph could contain sensitive data, the generated graphs can be used instead in order to preserve anonymity. There are various kinds of graphs anonymization algorithms, each of which serve different purposes. Some examples of anonymization algorithms include K-neighborhood anonymity, edge randomization and cluster based generalization; see survey by Wu et al. [53]. Although related, this work is different from ours as only certain parts of the graph need to be modified and only certain graph properties need to be

maintained, e.g., preserving the spectral information of the underlying graph as in Ying et al. [56].

3 PRELIMINARY EXPERIMENTS AND FINDINGS

Unlike the traditional setting of Anscombes quartet, in the graph setting some properties are correlated with each other. Thus, fixing one property (e.g., high edge density) may allow us to vary a property such as assortativity, but not other properties such as diameter. This lead us to study such correlations, which in turn lead us to graph generators (as the ground truth data is too large for $|V| > 10$), which in turn lead us to evaluating the generators, which in turn lead us to the qualities of the generators, which lead us to the notions of coverage and representativeness.

In this section we present some preliminary findings about the relationships between the graph properties described in Section 2.1. Note that our analysis focuses on the 10 dimensional space, where each dimension corresponds to one of 10 graph properties. As discussed in Section 2.1, there are many other graph properties. However, many properties are local (vertex-based or edge-based). Our choice of 10 global properties is the result of analyzing the graph mining literature and selecting the most commonly studied graph properties that can also be efficiently computed. Similarly, as discussed in Section 2.2, while there are many different graph generators, we focus on 5, ensuring that we have one from each of the three major types. In Section 3.1 we consider the properties of graphs with 100 vertices and next, in Section 3.2 we analyze properties of graphs with up to 10 vertices.

3.1 Graph Properties in Higher-Order Graphs

In a recent study of the ability to perceive different graph properties, such as edge density and clustering coefficient in different types of graph layouts (e.g., force-directed, circular), we generated a large number of graphs with 100 vertices. Specifically, we generated graphs that vary in a controlled way in edge density and graphs that vary in a controlled way in the average clustering coefficient [57]. A post-hoc analysis of this data (<http://vader.lab.asu.edu/GraphAnalytics/>) reveals some interesting patterns among the properties listed in Table 1.

The edge density dataset has 4,950 graphs and for each graph, we compute the 10 properties described in Table 1. We then compute Pearson correlation coefficients and observe high positive (purple) correlations and negative (brown) correlations for many property pairs, Fig 3. For example, the average clustering coefficient is highly correlated with the global clustering coefficient and the number of triangles.

These graphs were created for a very specific purpose and cover only a limited space of all graphs with $|V| = 100$. The type of generators we used, and the way we used them (some properties were controlled), could bias the results and influence the correlations. The fact that these correlations exist when some properties are fixed indicates that we can keep certain graph properties fixed while manipulating others. This motivated us to conduct the following experiments:

- 1) Generate all non-isomorphic lower order graphs ($|V| \leq 10$) and analyze the relationships between

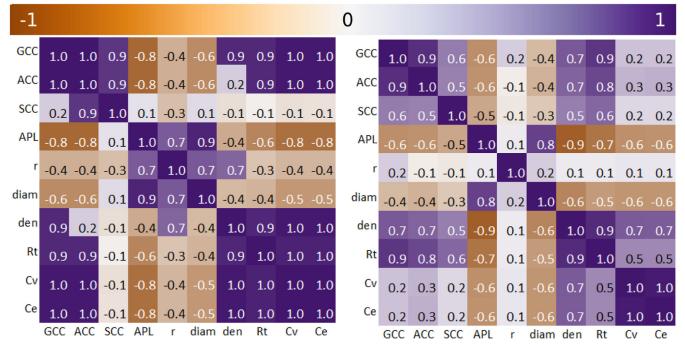


Fig. 3. Graph properties correlation matrix plots for the edge density dataset (left) and the ground truth set of all non-isomorphic graphs on $|V| = 9$ vertices (right).

graph properties. We consider this type of data as ground truth due to its completeness.

- 2) Use different graph generators and compare how well they represent the space of non-isomorphic graphs and how well they cover the range of possible values in the ground truth data.

An analysis of the set of 274,668 non-isomorphic graphs on $|V| = 9$ vertices shows that the correlations are quite different than those in graphs from our edge density experiment; see Fig. 3.

3.2 Graph Properties in Lower-Order Graphs

We start the experiment by looking at the pairwise relationships of graph properties of low-order graphs, where all non-isomorphic graphs can be enumerated. If two properties, say s_1 and s_2 , are highly correlated, then fixing s_1 is likely to restrict the range of possible values for s_2 . On the other hand, if s_1 and s_2 are independent, fixing s_1 might not impact the range of values for s_2 , yielding the same stats (s_1) for different graphs (s_2). We first study the correlations between the properties under consideration.

We focus on the analysis of graphs with $|V| = 5, 6, \dots, 10$ (unless otherwise stated) and compute all properties for all non-isomorphic graphs in this range. One reason we do not consider $|V| < 5$ is that many of the properties require at least 3 vertices (e.g., GCC, ACC). Another reason is that the number of different graphs on 1, 2, 3 and 4 vertices is very small: 1, 2, 4 and 11.

We next consider the pairwise correlations between the different graph properties and how this changes as the graph order increases, Fig 4. To compare the coverage of graph properties with different $|V|$, we scale the values of graph properties into the same range. By definition, clustering coefficients (ACC, GCC, SCC) are in the $[0, 1]$ range and degree assortativity is in the $[-1, 1]$ range. We keep their values and ranges without scaling. Edge density, number of triangles, diameter and connectivity (C_v and C_e), are normalized into $[0, 1]$ (dividing by the corresponding maximum value). APL, is also normalized into $[0, 1]$, subject to some complications: we compute the exact average path length to divide by in our ground truth datasets, but not when we use the generators, where we use the maximal path length encountered instead (which may not be the same as the maximum).

It is easy to see that the coverage of values expands with increasing $|V|$. Fig. 5 shows this pattern for three pairs of properties. This indicates that we are more likely to find larger

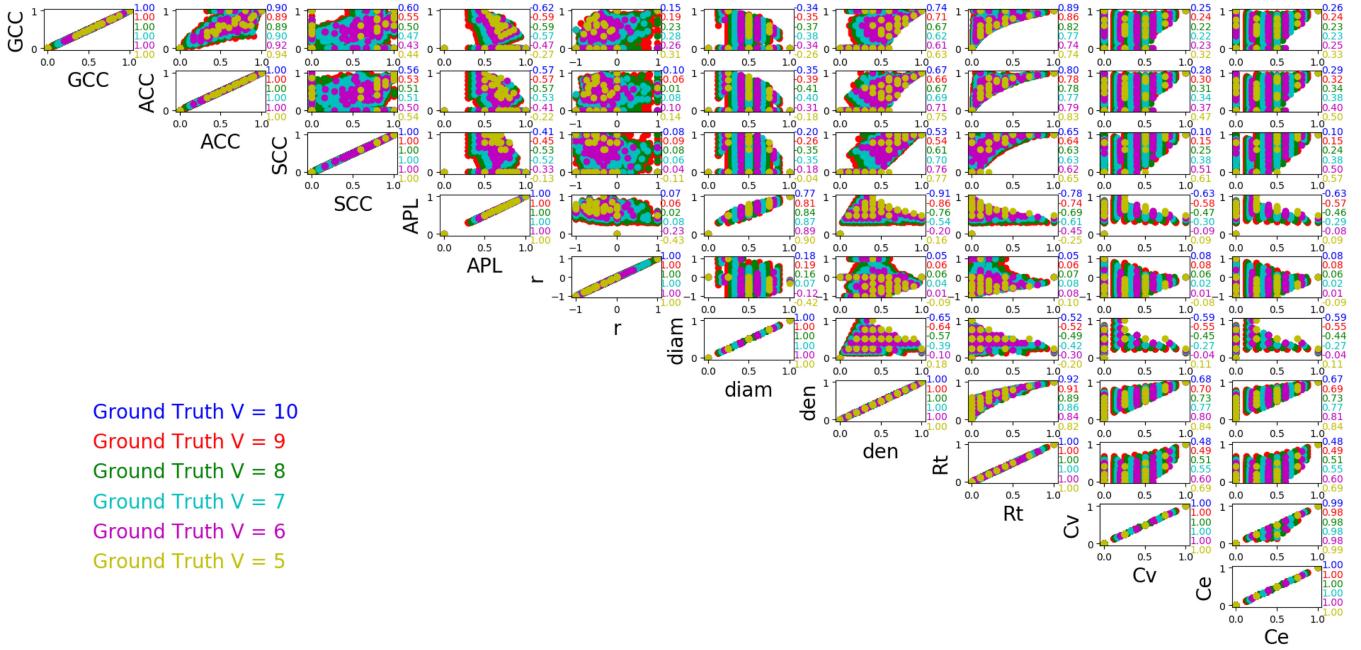


Fig. 4. Correlations between graph properties in the ground truth for $|V| = 5, 6, 7, 8, 9$. Note that for $|V| = 9$ there are already 274,668 points. Points are plotted to overlap, with the largest sets plotted first (i.e., $|V| = 9, \dots, |V| = 5$) to enable us to identify the range of properties that can be covered with a given number of vertices.

ranges of different graph properties for graphs with more vertices given the same set of fixed properties. With this in mind, we consider graphs with more than 10 vertices, but this time relying on random graph generators. Fig. 6 shows how correlation values between all pairs of graph properties change when the number of vertices increases. The blue trend lines for the ground truth data show the correlation values calculated using the set of all possible graphs for a given number of nodes. The

orange trend lines show the correlation values calculated from graphs generated with the ER model. Specifically, the ER-model data is created as follows: for each value of $|V| = 5, 6, \dots, 15$ we generate 100,000 graphs with p selected uniformly at random in the $[0, 1]$ range.

For most of the cells in the matrix, Fig. 6, the correlation values seem to converge as $|V|$ becomes larger than 8 (both in the ground truth and the ER-model generated graph sets). The pattern of the change in correlation values appears to be the same for both sets. Analyzing the trend lines of the ER-model, we observe four patterns of change in the correlation values: convergence to a constant value, monotonic decrease, monotonic increase, and non-monotonic change. Fig. 6 highlights these patterns using box outlines of different colors. There are exceptions that do not fit these patterns, e.g., (S_c, r) and in two cases, (r, C_v) and (r, C_e) , the trend lines show different patterns.

4 FINDING SAME STATS, DIFFERENT GRAPHS

In this section we consider examples of the “same stats, different graphs” phenomenon. In order to study and find “same stats, different graphs,” we run several small experiments:

- The first experiment indicates that very sparse or very dense graphs have few isomorphic copies, while graphs with edge density close to 0.5 have many such copies.
- The second experiment shows that graphs with the same edge densities do not necessarily have the same number of isomorphic copies.
- The third experiment shows that even the best graph generator (in terms of representativeness), ER, does not represent the space of all non-isomorphic graphs well.

We start with the case when the two graphs are indeed isomorphic. Structurally the path graph 1-2-3-4 is

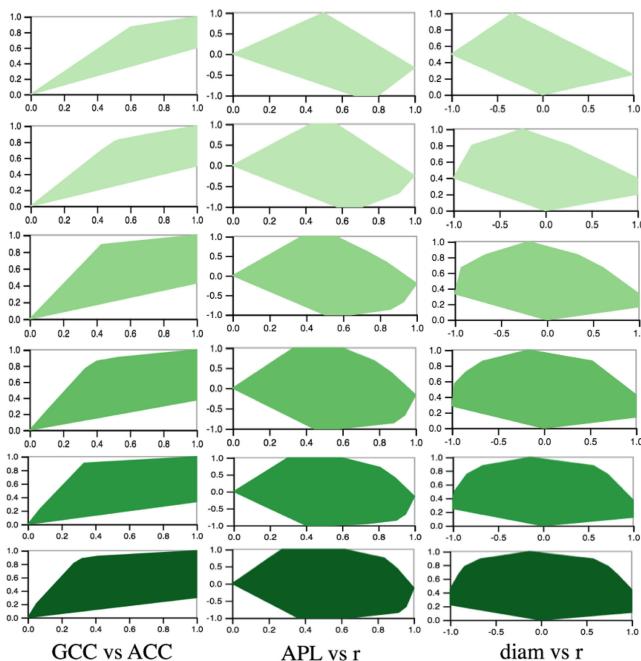


Fig. 5. The convex hull of graph coverage across several graph properties. Each row represents all graphs for a fixed number of vertices ($|V| = 5 \dots |V| = 10$). Columns are pairs of graph properties. The color is uniform for each image and corresponds to the average number of graphs in the image (the more of them, the darker the color).

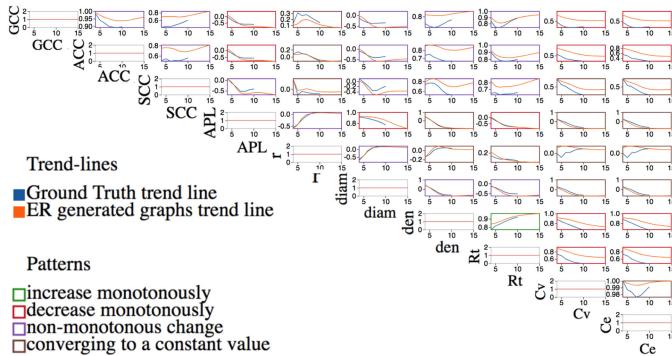


Fig. 6. Trends in the correlations with increasing $|V|$: the x -axis shows the number of vertices and the y -axis shows the correlation value for the pair of graph properties.

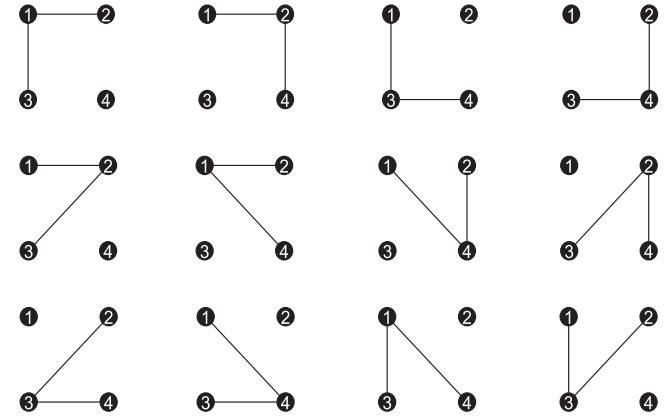


Fig. 8. All isomorphic graphs $|V| = 4, |E| = 2$ whose edges form a path.

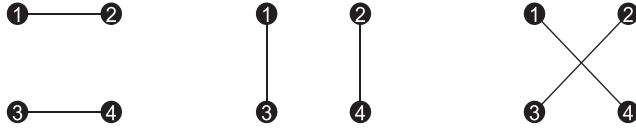


Fig. 7. All isomorphic graphs $|V| = 4, |E| = 2$ whose edges are disjoint.

indistinguishable from the path graph 1-3-2-4, however, for an isomorphic graph generator, these two graphs are different. Intuitively, very sparse or very dense graphs have few isomorphic copies, while graphs with edge density 0.5 have many such copies. Consider for example graphs with 4 vertices. There is only one graph for with 4 vertices and 0 edges (the empty graph) and only one graph with 4 vertices and 6 edges (the complete graph). Relabeling those graphs does not create different isomorphic copies. On the other hand, there are three structurally different graphs with 4 vertices and with 3 edges; their degree sequences are $\{2, 2, 1, 1\}$, $\{3, 1, 1, 1\}$, $\{2, 2, 2, 0\}$. Since their degree sequences are different, changing labels leads to different isomorphic copies.

It is then natural to ask whether two graphs with the same edge densities have the same number of isomorphic copies. Unfortunately, this is not the case, as illustrated in Figs. 7 and 8 by two small graphs with $|V| = 4$ and $|E| = 2$. Structurally, there are only two different types of graphs with 4 vertices and 2 edges. In the first type the two edges form a path; see Fig. 8. In the second type the two edges are disjoint; see Fig. 7. Importantly, there are only 3 graphs for which the edges are disjoint, while there are 12 graphs for which the edges form a path. If a pair of graphs is isomorphic then they have the same properties, including the 10 properties that we are measuring. However, a pair of graphs can be structurally different (non-isomorphic) while still having the same properties. To find such low-order graphs we can explore the entire set of non-isomorphic graphs (as we have explicit representations for all of them). For larger graphs, we use the graph generators together with some filters.

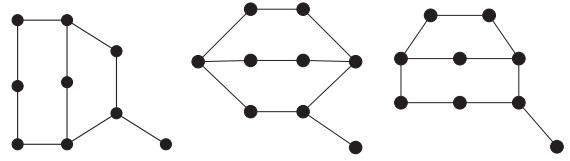


Fig. 9. An example of three different graphs that have exactly the same 10 properties: 9 vertices, 10 edges, $GCC = ACC = SCC = Rt = 0$, $APL = 2.11$, $r = -0.47$, $diam = 4$, $Cv = Ce = 1$.

Examining the ground truth data for $|V| = 7$, we find a pair of graphs that have exactly the same 10 properties. For $|V| = 8$, we find 8 pairs of graphs that have exactly the same 10 properties. For $|V| > 8$ we have found many more graphs with exactly the same 10 properties, such as the triple of graphs shown in Fig. 9.

In Sections 6.1 and 6.3 we analyze the 5 graph generators in terms of *representation* and *coverage*, and we explore their performance as generators of non-isomorphic graphs. We use the ER generator with probability of adding any edge $p = 1/2$ and $|V| = 8$, and generate 12,346 graphs (equal to the number of non-isomorphic graphs on $|V| = 8$). We compute the number of repetitions (over the 10 properties). Results are given in Table 3.

Comparing to the total set of non-isomorphic graphs for $|V| = 8$, the set generated by ER has more graphs with the same properties. There are 1,955 pairs that have exactly the same 10 graph properties in the sample, while in the total set of non-isomorphic graphs there are only 7 such pairs. Fig. 10 shows the two most common graphs that appear in the ER model samples.

4.1 A Tool for Finding Similar Graphs

We provide an analytic tool (<http://findgraph.cs.arizona.edu/>) for exploring the “same stats, different graphs” phenomenon, that is, looking for graphs with several fixed properties and one that varies. In this tool, we separate our

TABLE 3

Illustration of the Number of Repetitions in the Set of 12,346 Graphs Generated by the ER Model with $|V| = 8$ and $p = 1/2$

# of sets with graphs that have the same 10 properties	2	7	37	109	348	833	1955	3713
# of graphs in the set	8	7	6	5	4	3	2	1

The size of the sample is equal to the number of non-isomorphic graphs with $|V| = 8$. The table indicates the number of sets of graphs having the same exact 10 properties (although they may not be isomorphic). For example, there are 2 sets with 8 graphs that each have the exact same 10 properties. To put this in perspective, in the ground truth there are only 7 pairs of “same 10 stats, but different graphs,” while in this sample we have 1,955 such pairs.

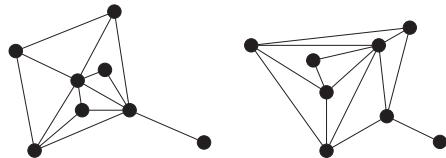


Fig. 10. Illustration of the graphs that appeared most frequently in the sample generated by the ER model with $p = 1/2$, $V = 8$. Each of the two graphs appeared 8 times.

task into two parts and integrate them in one interface. For graphs with $|V| \leq 10$ we examine all possible non-isomorphic graphs. For graphs with $|V| > 10$ we use graph generators and a filter.

For graphs with $|V| \leq 10$, since the ground truth is known, we know whether graphs with fixed several properties exist or no. In order to find such graphs, we create one interactive parallel coordinate plot (PCP) for each set of non-isomorphic graphs with fixed $|V|$. We provide an online version of PCP (<http://differentgraphs.cs.arizona.edu/pcp/index.html>). In order to make the waiting time affordable, we use k-means clustering instead of all non-isomorphic graphs with fixing $|V|$.

4.2 Finding Similar Low-Order Graphs

For $|V| \leq 10$, we examine our dataset of all possible non-isomorphic graphs, looking for graphs with several fixed properties and one that varies. We use a spring layout to visualize this series of same stats, different graphs.

For the first experiment, we fix $|V| = 9$, $APL \in (1.42, 1.47)$, $den \in (0.52, 0.57)$, $GCC \in (0.5, 0.6)$, $R_t \in (0.15, 0.25)$. Since all our properties are normalized to $[0, 1]$ and assortativity is in $[-1, 1]$, we consider 10 possible ranges for assortativity, each of size 0.2 and we find graphs for seven of the ten possible ranges;

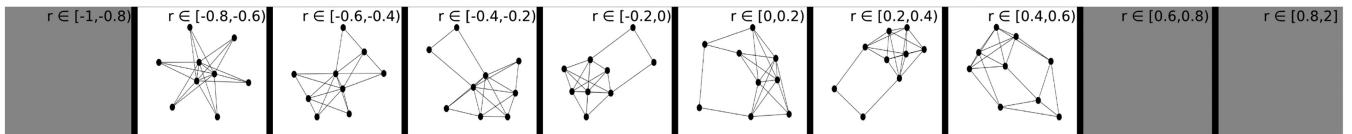


Fig. 11. Examples of graphs with fixed number of vertices $|V| = 9$ and 4 other fixed properties ($APL \in (1.42, 1.47)$, $den \in (0.52, 0.57)$, $GCC \in (0.5, 0.6)$, $R_t \in (0.15, 0.25)$), but with varying assortativity values (between -1 and 1 by jumps of 0.2). The empty slots represent that there is no graph with the specified set of properties.

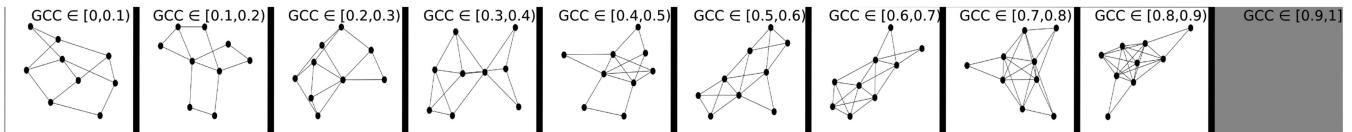


Fig. 12. Examples of graphs with fixed number of vertices $|V| = 9$ and 4 other fixed properties ($diam = 3$, $Cv = 2$, $Ce = 2$, and $r \in (-0.29, -0.22)$), but with varying global clustering coefficient (GCC in the range between 0 and 1 by jumps of 0.1). The empty slots represent that there is no graph with the specified set of properties.

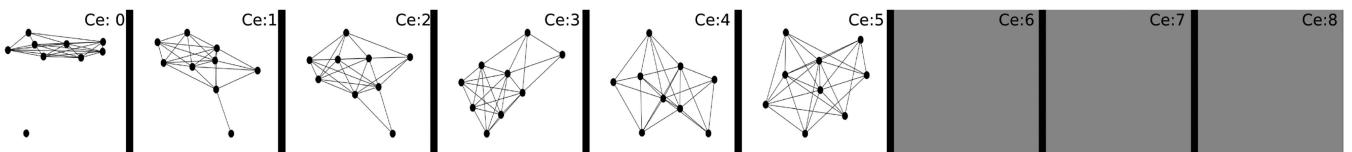


Fig. 13. Examples of graphs with fixed number of vertices $|V| = 9$ and 4 other fixed properties ($SCC \in (0.75, 0.85)$, $ACC \in (0.75, 0.8)$, $r \in (-0.3, -0.2)$, $R_t \in (0.35, 0.45)$), but with varying edge connectivity (in the range between 0 and 8 by jumps of 1). The empty slots represent that there is no graph with the specified set of properties.

see Fig. 11. This figure also illustrates the output of our “same stats, different graphs” generator: fix several graph properties and generate graphs that vary in another properties.

Similarly, for the second experiment, we fix $|V| = 9$, $diam = 3$, $Cv = 2$, $Ce = 2$, and $r \in (-0.29, -0.22)$ and look for graphs with different GCC values. GCC is in the range 0 to 1 and we find graphs for nine of the ten possible ranges by jumps of 0.1 ; Fig. 12.

As a final example, we fix $|V| = 9$, $SCC \in (0.75, 0.85)$, $ACC \in (0.75, 0.8)$, $r \in (-0.3, -0.2)$, $R_t \in (0.35, 0.45)$ and find graphs with varying connectivity. Specifically, we look for different values of edge connectivity, C_e , which is in the range 0 to $|V| - 1$; Fig. 13.

Note that the graphs in Figs. 11, 12, and 13 are different in structure even though they possess similar values for many properties. For example, in Fig. 12, there are more and more triangles as we go from left to right.

4.3 Finding Similar Graphs Using Generators

This approach relies on generating many graphs and filtering graphs based on several fixed properties. For the two most important properties of a graph, $|V|$ and $|E|$, we generate all graphs with a fixed $|V|$ and choose $|E|$ as follows:

- 1) UN: select $|E|$ uniformly from its range. This is equivalent to forcing the edge density in the generated set to follow a uniform distribution;
- 2) ER: select $|E|$ from the binomial distribution, that is, each edge appears with fifty percent probability.

Using both edge selection strategies for all four generators, we compare the statistics distribution to the ground truth for $|V| = 9$. Fig. 14 illustrates how different properties are distributed for the UN and ER generators. It shows that although the ER generates a distribution that is more similar

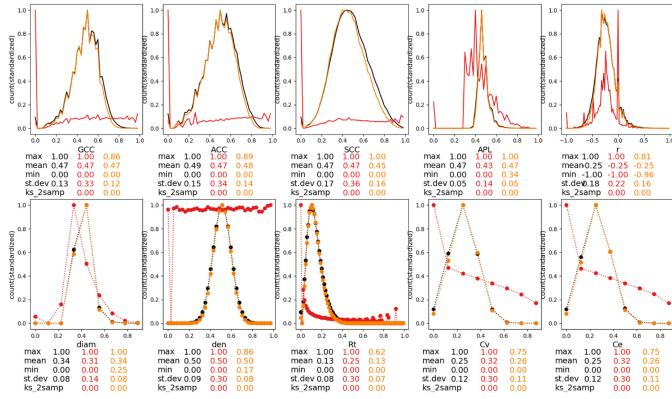


Fig. 14. Distribution of the 10 properties, including min/mean/max and standard deviation. Ground truth is colored black, ER with $p = 1/2$ in orange, UN in red.

to the ground truth, it does not cover the range of values (larger min and smaller max) than the UN generator. The WS and BA models also do not provide good coverage of the various graph properties. We next discuss the notions of representation and coverage of the ground truth.

5 GENERATOR EVALUATION

For low order graphs (in this setting graphs with $|V| \leq 10$), we can explore statistical coverage and representation of a generated sample by comparing it with the set of all graphs with fixed number of vertices. However, it is difficult to generate all non-isomorphic graphs with more than 10 vertices due to the super-exponential increase in the number of graphs (e.g., for $|V| = 16$ there are 6×10^{22} different graphs). Nevertheless, these higher order graphs are common in many domains. We turn to graph generators in order to further explore the issue of “same stats, different graphs” for larger graphs.

In this section, we discuss two approaches for measuring the quality of the statistical approximation of the set of properties for the sampled graphs when compared against the ground truth. Note that we are going to generate graphs with a fixed number of vertices (e.g., $|V| = 9$) and compare the sample to the set of all different graphs with the same number of vertices. The comparison is done with respect to the 10 properties defined in Section 2.1. The first measure, which we refer to as *representation*, evaluates the extent to which the set of sampled graphs represents the properties of the set of all graphs with fixed number of vertices. The second measure, which we refer to as *coverage*, evaluates the extent to which the sampled set of graphs covers a similar range of values as the set of all graphs with a fixed number of vertices. Both settings refer to the 10 dimensional space defined in Section 2.1.

We analyze graphs sampled from the following five models: ER with probability 1/2 (ER), ER with p selected uniformly at random from the [0,1] range (UN), geometric (GE), Watts and Strogatz (WS), and Barabasi and Albert (BA). We use the implementations of the generators (ER, WS, BA, GE) from NetworkX [58].

5.1 Representation

Our goal here is to explore whether a small sample of graphs with a fixed number of vertices can represent the set of all

non-isomorphic graphs with the same number of vertices and how this representation changes as the sample size becomes larger (i.e., going from 1 to 100 percent). For this purpose, we review and analyze the following four methods: Pearson correlations, Kolmogorov-Smirnov (KS) test, Kullback-Leibler (KL) divergence, and Wasserstein distance, which is also known as the earth mover (EM) distance.

One possible way to measure how representative a graph generator is to generate graphs with it, compute the graph properties described in Table 1, calculate relative correlations between the graph properties, and compare the results with those in the ground truth. Since we consider 10 properties we have 45 such comparisons, Fig. 17, which makes it difficult to compare the different generators.

The KS test [59] is a nonparametric test used to compare a sample with a reference distribution (one-sample case), that is, to quantify how well the sample represents the given distribution. The KS test can also be used to compare two samples (two-sample case) and quantify whether both samples represent the same distribution. The null distribution of this statistic is calculated under the null hypothesis that the samples are drawn from the same distribution (in the two-sample case). In our setting, we need to compare sampled data with the ground truth. To do so, we propose to uniformly sample 10 percent of the ground truth and use the KS test with the generated sample and the uniformly sampled dataset. We repeat this procedure 10 times and average the results. However, similar to pairwise correlations, the KS test results in 10 different numbers, one for each property.

Unlike relative correlations and the KS test, KL divergence and EM distance would give us a single value associated with the generated dataset. However, since we have a discrete dataset and the underlying distribution is unknown, we use the formulas for a multinomial normal distribution. For this, one needs to calculate the mean and the covariance of the generated data and the ground truth, then use the formulas for KL divergence and EM distance

$$D_{KL}(N_0|N_1) = \frac{1}{2} \left(\text{tr}(\Sigma_1^{-1}\Sigma_0) + (\mu_0 - \mu_1)^T \Sigma_1 (\mu_0 - \mu_1) - k + \ln \left(\frac{\det(\Sigma_1)}{\det(\Sigma_0)} \right) \right),$$

$$D_{EM}(N_0|N_1) = \|\mu_0 - \mu_1\|_2^2 + \text{tr} \left(\Sigma_0 + \Sigma_1 - 2(\Sigma_1^{1/2}\Sigma_0\Sigma_1^{1/2})^{1/2} \right).$$

5.2 Coverage

In this section we introduce four different measures: diameter, volume of the bounding box, split bounding box, and volume of the robust ellipse to measure the coverage of the sampled dataset. The ultimate goal is to see whether these coverage measures are consistent for different generators and under different sample sizes. Next, we define the four measures mentioned above and discuss their advantages and limitations for our setting. For a discrete dataset $S \subset \mathbb{R}^d$, the *diameter* measures the largest possible distance between all the pairs of points in S . The precise mathematical definition of the diameter is

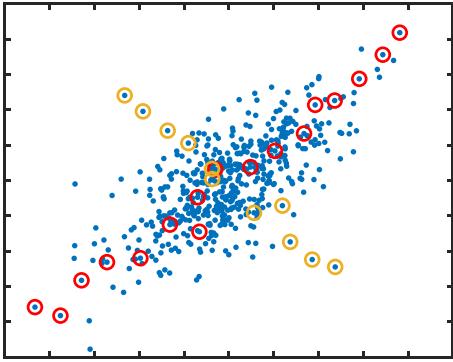


Fig. 15. Example of a dataset in 2D where the diameter of the sample (points circled in red) is the same as the diameter for the ground truth (all points in blue), but the sample does not cover the dataset well (see points with yellow circle).

$$\text{diam}(S) = \sup_{x,y \in S} \|x - y\|.$$

If the diameter of the sampled dataset is smaller than the diameter of the ground truth, then the sampled dataset does not cover the complete range of graph properties. However, if the diameter of the sampled dataset is the same, or in the same range as the diameter of the ground truth, it does not necessarily imply that the sampled dataset covers the range of the graph properties of all graphs; see Fig. 15.

The complexity of calculating the diameter of a discrete set S is $O(|S|^2)$, this is the complexity of finding the distances between all possible points of $|S|$. For large datasets it would be too time consuming to compute the diameter. To overcome this issue, we uniformly subsample points from the dataset and find the diameter of the subsampled dataset. To make sure that the results are accurate, we calculate the diameter for 10 such subsamples and report their average.

We also propose to use the *volume of the bounding box*. For a set $S = \{x_1, \dots, x_n\} \subset \mathbb{R}^d$ the bounding box of S is defined as

$$\begin{aligned} BB(S) = \{(a_i, b_i), i = 1, 2, \dots, d \mid &a_i = \min\{x_1(i), \dots, x_n(i)\}, \\ &b_i = \max\{x_1(i), \dots, x_n(i)\}, \text{ for } i = 1, \dots, d\}. \end{aligned}$$

The volume of the bounding box for a set $S \subset \mathbb{R}^d$ is the volume of the d -dimensional hyperrectangle $BB(S)$ which is $\prod_{i=1}^d (b_i - a_i)$.

Although the bounding box captures information for each dimension and maintains extreme point information for each dimension, similar to the diameter, it can suffer if the dataset is not concentrated around a hyperrectangle. The bounding box volume is highly influenced by outliers, especially for small sample sizes. If the ratio of the bounding boxes volumes is small it implies that the sample does not cover the ground truth; however, it is hard to make a conclusion if the ratio is around 1; Fig. 16.

We also consider the *split bounding box* measure as a generalization of the standard bounding box, where we create multiple bounding boxes that span the data. We divide the range of each data dimension into multiple parts of equal size resulting in multiple hypercubes. Next, for each hypercube, we compute the bounding box of the sample and the ground truth data (restricted to this hypercube), calculate

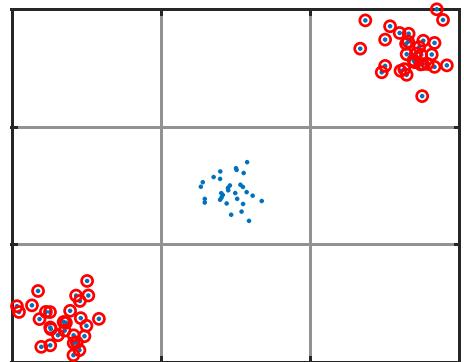


Fig. 16. Example of a dataset in 2D where the volume of the bounding box (points circled in red) is the same as the volume of the bounding box of the ground truth (all points in blue), but the sample does not cover the dataset well (see the blue points in the middle). This example demonstrates that split bounding box would perform better in this case.

the ratio of their volumes, and average them across all hypercubes; see Fig. 16. Dividing the range of each measure into k equal parts results in k^{10} hypercubes for our dataset. As a result, when the number of dimensions is high, this measure requires a large sample size.

A possible way to overcome the limitations associated with the bounding box and split bounding box is to consider the *convex hull*. We can compute the convex hull of the sample and the ground truth dataset and calculate the ratio of their volumes. However, the $O(n^{\lfloor d/2 \rfloor})$ convex hull computation is computationally expensive in high dimensions. We show the convex hull results based on 8 of the 10 properties (excluding Ce and Cv) computed using the QHull algorithm [60]; see Table 4.

A computationally efficient alternative to the convex hull is the *robust ellipse* measure. We compute the singular values of the dataset (similar to principal component analysis) and multiply them to obtain the *robust ellipse* measure. As in the other approaches, we compute the robust ellipse measures for the generated sample and the ground truth dataset, calculate their volumes, and consider their ratio. This measure, unlike the diameter, bounding box and convex hull measures, should be more robust to outliers. One limitation of this measure is that it depends on the density of the dataset. Note that for the representation measures, small values mean that the sample represents the ground truth well. For the coverage measures, the bigger the value, the better the coverage.

6 COMPARISON BETWEEN GRAPH GENERATORS

In this section, we use the measures of representation and coverage defined in Section 5 to compare between the five generators discussed in Section 2.2. In Section 6.1 we present results for the representation measure, in Section 6.3 we present results for the coverage measure, and in Section 6.5 we discuss the limitations of graph generators.

TABLE 4
The Ratio of the Volumes of Convex Hulls for Sampled (1 percent) and the Ground Truth in 8D (Excluding Ce and Cv), $|V| = 9$

	UN	GE	ER	WS	BA
Ratio	11.96%	14.78%	0.81%	1.25%	0.11%

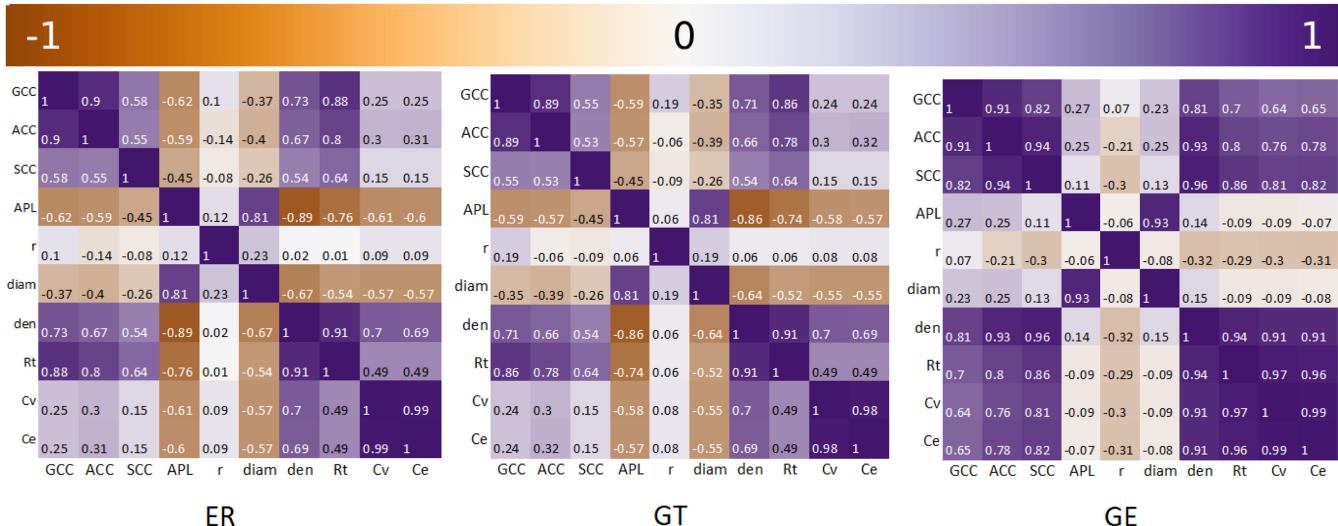


Fig. 17. Relative correlations between the 10 properties for graphs with $|V| = 9$. The left table shows the correlations for the dataset sampled by the ER $p = 1/2$ graph generator of size 1 percent of the ground truth dataset. The middle table shows correlations for the ground truth. The right table presents the relative correlations for the dataset sampled by the GE graph generator of size 1 percent of the ground truth dataset.

6.1 Representation of the Graph Generators

We start our analysis with Pearson correlations. For each of the five generators, we generate a sample with a size of 1 percent of the ground truth dataset. We present the relative correlations for the sampled and ground truth datasets by calculating all pairwise 2D statistics of all non-isomorphic graphs with $|V| = 9$. While we computed these tables for all five generators, we only show the results for the best (ER) and worst (GE) performing generators, Fig. 17. When comparing the ground truth and ER values, we see that they are nearly identical for all entries in the matrices and the largest difference is 0.09. When comparing the ground truth and GE, however, the differences are obvious and as large as 1.00 (e.g., in the correlation between APL and density). For the most representative model, ER, all pairwise correlations are similar for the ground truth and the sampled datasets. However, the graphs generated by ER do not cover the entire spectrum of possible values for each property.

To effectively visualize the data, we use *violin plots* [61] (as implemented in the matplotlib library). Violin plots show more information than Box plots, as the kernel density visualization makes it possible to see more details about the data

distribution (e.g., clusters). To make it easy to distinguish the results for the different generators, we use persistent colors across the visualizations and a color scheme from colorbrewer (Colorbrewer 2.0: <http://colorbrewer2.org/>).

Going beyond correlations, we also compute the KL divergence and the EM distance defined in Section 5.1 for samples generated by the five generators for $|V| = 7, 8, 9$. (We do not report results for $|V| = 5, 6$ as the sample sizes are too small and they do not contain enough information to calculate the KL divergence). In Fig. 18, we report the results of the 10 different samples for each generator with sample size equal to the size of the ground truth dataset. As discussed in Section 5.1, numbers close to 0 mean that the sample generated by the generator represents the ground truth well. The best generator for $|V| = 7, 8, 9$ is ER under both KL divergence and EM distance measures. Overall, GE, UN and WS are consistently outperformed by ER and BA.

6.2 Consistency of the Representation Measure

An important question is whether the measures that we use (Section 5) behave consistently. Consistency refers to being able to obtain similar results for the representation measure when the size of the ground truth data is changing, which

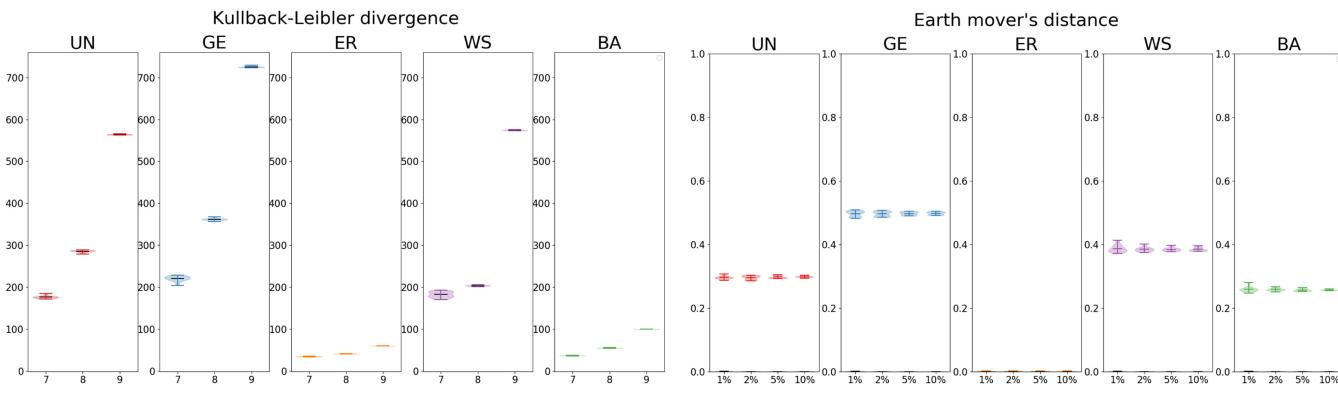


Fig. 18. Representation comparison of the five graph generators (UN, GE, ER, WE and BA) over ten samples, with sample sizes equal to the size of the ground truth dataset for $|V| = 7$, $|V| = 8$, and $|V| = 9$. KL divergence is on the left and EM distance is on the right; see Section 5.1.

TABLE 5
The Ranking (Best to Worst from Left to Right) of Graph Generators based on the Representation Measures KL and EM

V	generators					
	ER	BA	UN	WS	GE	
	ER	BA	WS	UN	GE	
7						
8						
9						

It is notable that both the KL and EM measures give the same ranking of the 5 generators. We are particularly interested in the best generators, and for $|V| = 7$, $|V| = 8$, $|V| = 9$ both KL and EM rank ER as best and BA as second best.

happens (1) when we consider graphs with different number of vertices and (2) when we use different sample sizes. We summarize the results from our consistency experiments, Fig. 18, in Table 5. According to our experiments, KL and EM measures for representation are consistent for $|V| = 7$, $|V| = 8$ and $|V| = 9$.

Another question is, how big does the sample need to be to represent the underlying ground truth dataset well. To answer this question, we run experiments for $|V| = 9$. For each of the five generators, we generate a sample with size 1, 2, 5, and 10 percent of the ground truth dataset. The results are reported in Fig. 19. We also show the results with a sample generated from the ground truth dataset by taking a uniform sample from it, which can be considered as a benchmark. In Fig. 19, we see that ER performs as well as the uniform sample from the ground truth dataset. We also note that as the sample gets larger, the values for both KL divergence and EM distance get smaller (they are decreasing). This behavior is expected as the more graphs contained in the sample, the more representative they can be. Also worth noting is that the larger the sample size is, the less variation within the ten samples, as shown by the progressively smaller violin plots.

6.3 Coverage of the Graph Generators

We start our analysis of the coverage measures by visualizing the 2D plots between the ten graph properties; see Fig. 20. We only show the results for the UN and WS graph generators which achieve the best and worst coverage results, respectively. The ground truth is represented in black, the UN data in red (left) and the WS data in purple (right). Note that under each colored dot (red or purple)

there exists a black dot, as the generators (red or purple) sample from the ground truth (black). Further, the fewer black points that are visible in the plot, the better the generator covers the ground truth. It is easy to see that UN performs much better, whereas WS misses large ranges of possible values, most noticeable in the density and assortativity columns (den and r).

Next, we use the four coverage measures discussed in Section 5.2. We run experiments for $|V| = 5, 6, 7, 8$, and 9 with sample size equal to the ground truth dataset size, compute the four measures for ten different samples and report the results using violin plots; see Fig. 21. Unlike representation, where the ER-model was the best model, the ER-model performs poorly in coverage.

Fig. 21 shows that for $|V| = 5, 6, 7, 8, 9$ the samples generated by the WS and BA models achieve poor coverage results, that is, the generated samples do not cover the range of graph properties for the ground truth well. The samples generated by the UN and GE models achieve the best coverage (high values for bounding box, split-bounding box and diameter). This implies that the UN and GE cover the ground truth data best. Three of the measures, namely diameter, bounding box and robust ellipse give consistent results. The results for split bounding box do not always agree with the other three measures, which can be explained by taking into consideration two observations: (1) for ten dimensional data such as ours, the number of boxes jumps from 1 to 1,024, and (2) for small values of $|V|$ there are not enough data points to calculate an accurate split bounding box measure.

Note that UN and GE have worse results in terms of representation but the best results in terms of coverage. This is not particularly surprising as good representation and good coverage are correlated with different properties of the graph generators. For example, the UN and GE generators are more likely to create unusual/extreme graphs (fully connected, very sparse, etc.), whereas ER generates the most likely/typical graphs. We also note that if we use the strategy described in Section 5.2 to approximate the diameter, in some cases the diameter of the ground truth dataset might be smaller than the diameter of the sampled dataset. We observe such behaviour in Figs. 21 and 22.

6.4 Consistency of the Coverage Measure

As with the representation measure, we also consider the issue of consistency of the different coverage measures

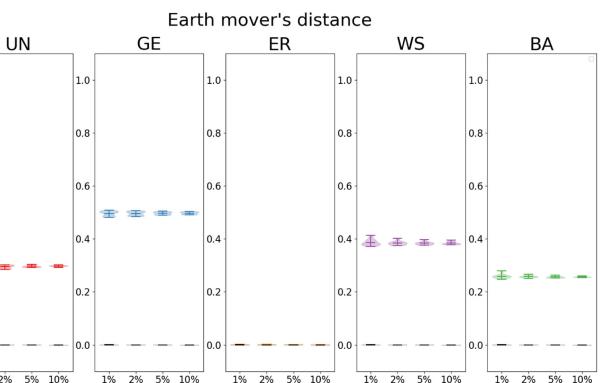
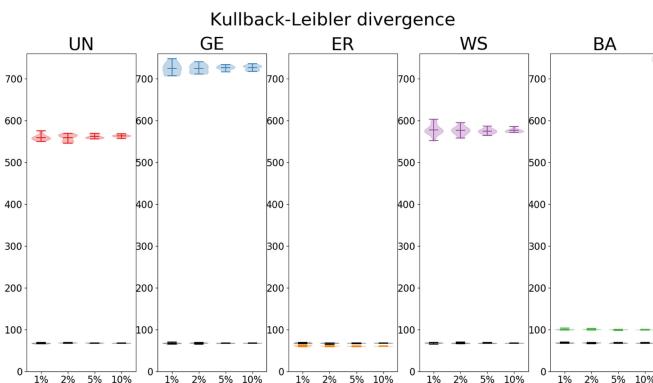


Fig. 19. Comparison of stability with different sample sizes, 1, 2, 5 and 10 percent (with respect to the ground truth size) for graphs with $|V| = 9$ for two different representation measures and five different graph generators. The uniform sample from the ground truth data is shown in black.

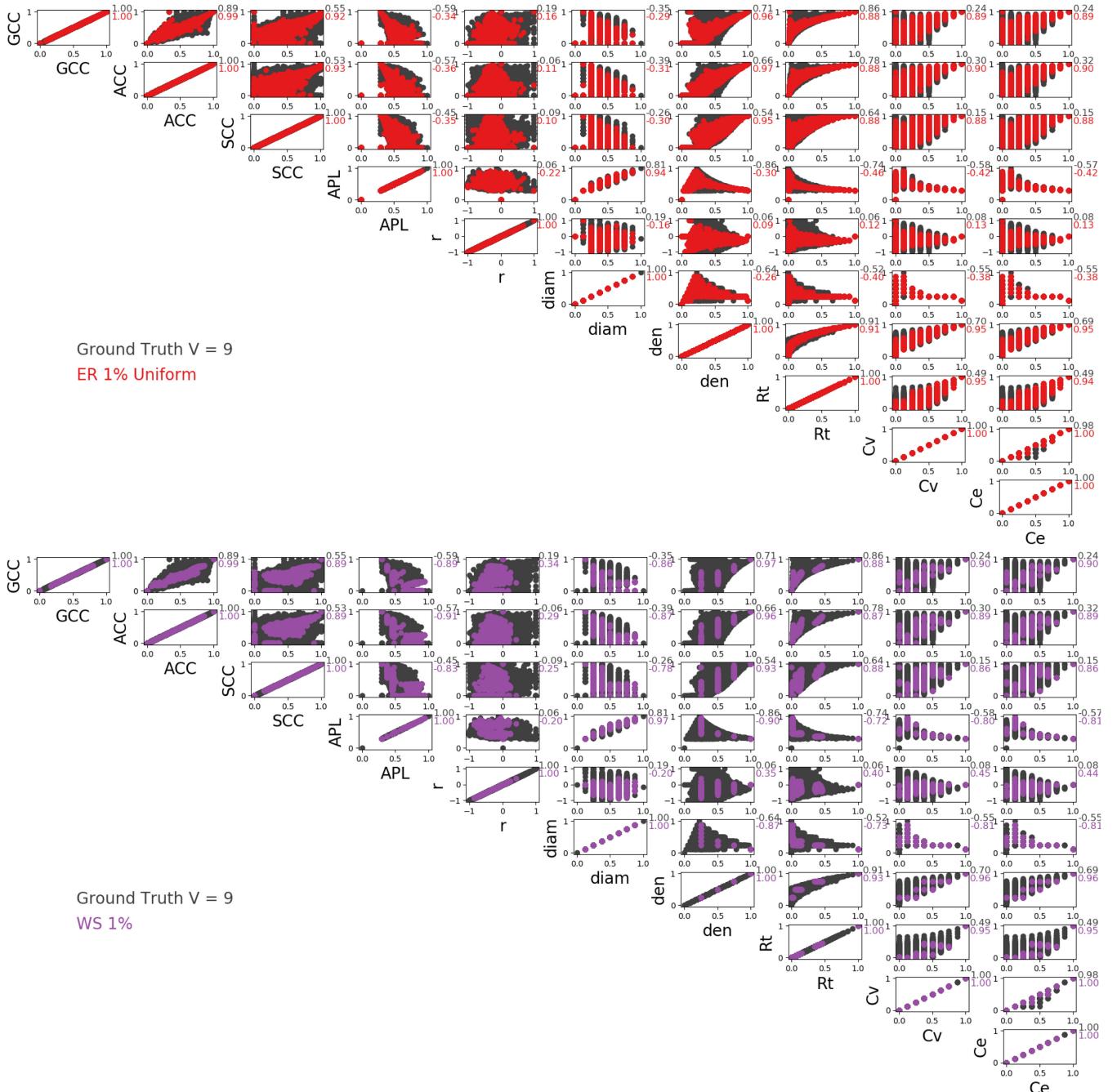


Fig. 20. Illustration of coverage for different generators based on 2D projections of the ground truth and generated data for $|V| = 9$. The ground truth is colored in black. The left figure shows a 1 percent sample from the UN generator (red). The right figure shows a 1 percent sample from the WS generator (purple). The corresponding correlation for the sample and the ground truth are shown next to each plot. Note that under each colored dot (red or purple) there exists a black dot, as the generators (red or purple) sample from the ground truth (black). Further, the fewer black points that are visible in the plot, the better the generator covers the ground truth.

when we vary the sample size or the size of the ground truth data (by changing the number of vertices). Unlike the measures for representation (KL divergence and EM distance), some of the measures for coverage are not stable for small values of $|V|$. In Fig. 21, we observe that for the bounding box and the split-bounding box measures, for $|V| = 5, 6, 7$ there is a high variation among the 10 samples.

We also observe some variation in the diameter, but only for the ER generator; see Figs. 21 and 22. However, this is an expected behaviour, since the UN and GE graph generators are able to capture the extreme cases. Thus, even for small

samples, the ratio of diameters of the sample and the ground truth is close to 1. However, for the ER graph generator, for smaller samples the generator does not always capture the extreme cases.

6.5 Limitations of BA and WS

None of the graph generators explicitly optimize representation or coverage, but some are better than others. As shown in Figs. 18, 19, and 21 the BA and WS generators perform poorly across all representation and coverage measures. The underlying generation methods and the

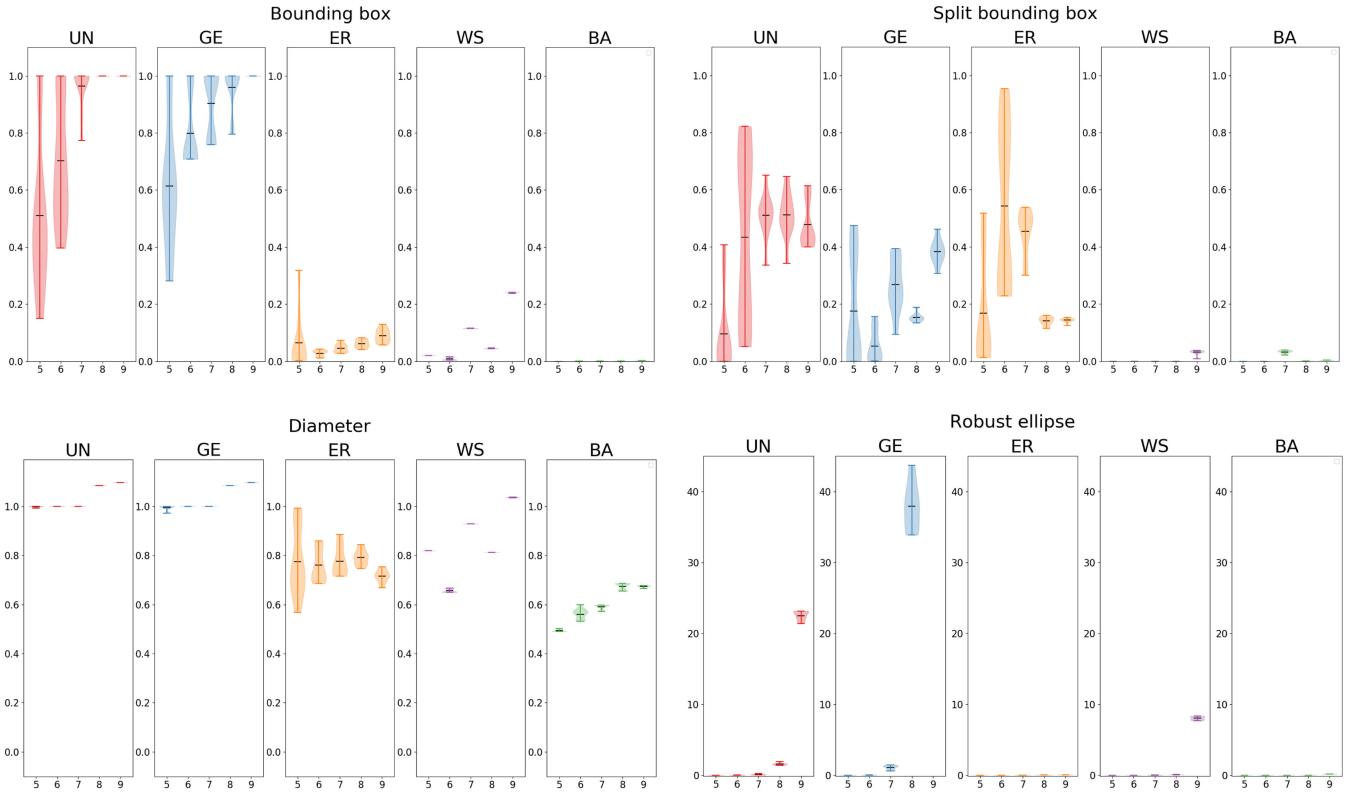


Fig. 21. Coverage comparison of the five graph generators (UN, GE, ER, WE and BA) over ten samples with sample sizes equal to the size of the ground truth dataset for $|V| = 5, 6, 7, 8, 9$. Each subfigure shows results for different coverage measures; see Section 5.2.

specific parameter settings used might explain why. The WS small-world graph generator requires 3 parameters: the number of vertices n , a number k that specifies how

many neighbors each node should be connected to, and a probability p for adding these edges. For our experiments we used $n = 9$, k chosen uniformly at random in the range

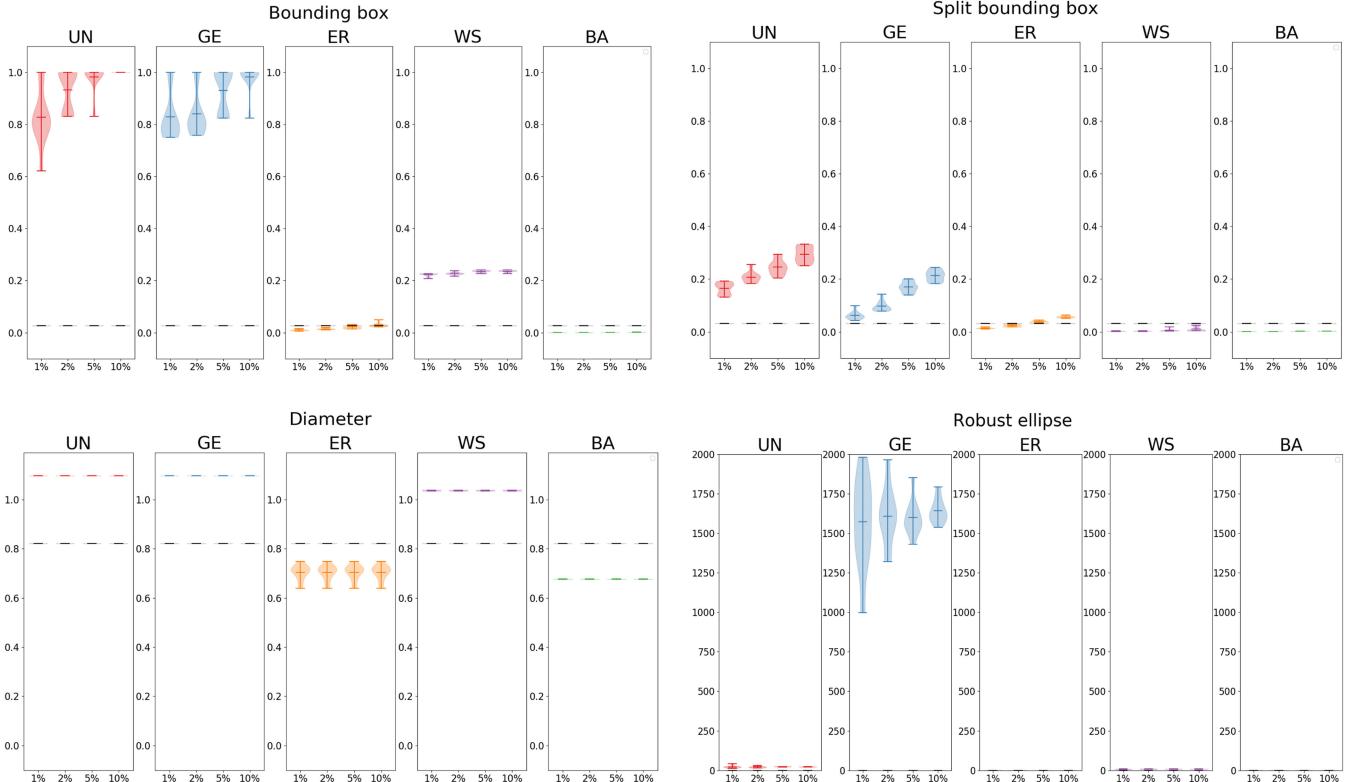


Fig. 22. Comparison of stability when using different sample sizes, 1, 2, 5 and 10 percent (with respect to the ground truth size) for graphs with $|V| = 9$ for four different coverage measures and five different graph generators. The uniform sample from the ground truth data is shown in black in all plots.

2 to $|V| - 1$, and p chosen uniformly at random in the range $[0, 1]$. Since the WS generator begins with a k -connected ring, and only switches an edge from one node to another, the result is graphs with $k * n$ edges, which limits the number of non-isomorphic graphs.

Similarly, the BA generator requires 2 parameters: the number of vertices n and the number of edges m to attach from a new node to existing nodes. We use $n = 9$ and randomly chose an integer value from 1 to $|V| - 1$ for m . Thus, the possible number of edges can only be $(n - m) * m$, which in the case of $|V| = 9$ leads to only four possible values of $|E|$ (8, 14, 18, 20). This restricts the range of different graphs that can be generated.

7 CONCLUSIONS AND FUTURE WORK

We discussed an exploration of the space of graphs, treating each graph as a high dimensional object, where the number of dimensions is determined by the number of global properties associated with each graph. For low-order graphs ($|V| \leq 10$) and for a small number of fixed properties (10 in this paper) we can directly explore the space. This allows us to find many instances of the “same stats, different graphs” phenomenon. One natural question is to determine the “true” set of dimensions for the space of non-isomorphic graphs, where we consider a set of graph properties that suffice to always distinguish two non-isomorphic graphs.

For larger graphs we can use random graph generators. Since most practical random graph generators sample the space of isomorphic graphs, they are not equally well-suited for the task of generating non-isomorphic graphs. We considered two particular measures, coverage and representation, and find that UN does a good job in coverage and ER is best for representation. We provide online exploration tool, source-code and data at <http://differentgraphs.cs.arizona.edu>

To “see” the difference between two non-isomorphic graphs, it often suffices to look at the drawings of the graphs. However, as graphs get larger, some graph drawing algorithms may not allow us to distinguish differences in properties between two graphs purely from their drawings. We recently studied how the perception properties, such as density and ACC, is affected by different graph drawing algorithms [57]. The results confirm the intuition that some drawing algorithms are more appropriate than others in aiding viewers to perceive differences between underlying graph properties. Further work in this direction might help ensure that differences between graphs are captured in the different drawings. Other interesting open problems that arise from this work include: designing efficient generators for graphs with the same stats or graphs that share some stats and vary in others, finding theoretical guarantees and bounds on how likely it is to find such graphs for specific generators, computing bounds on how many graphs have the same subset of stats.

ACKNOWLEDGMENTS

This is a journal version of a paper that appeared in the proceedings of the 26th Symposium on Graph Drawing and

Network Visualization (GD’18). This work was supported in part by NSF grants CCF-1740858 and DMS-1839274.

REFERENCES

- [1] F. J. Anscombe, “Graphs in statistical analysis,” *The Amer. Statistician*, vol. 27, no. 1, pp. 17–21, 1973. [Online]. Available: <http://www.jstor.org/stable/2682899>
- [2] J. Matejka and G. Fitzmaurice, “Same stats, different graphs: Generating datasets with varied appearance and identical statistics through simulated annealing,” in *Proc. CHI Conf. Human Factors Comput. Syst.*, 2017, pp. 1290–1294.
- [3] D. Chakrabarti and C. Faloutsos, “Graph patterns and the R-mat generator,” in *Mining Graph Data*, Hoboken, NJ, USA: Wiley, 2007, pp. 65–95.
- [4] G. Li, M. Semerci, B. Yener, and M. J. Zaki, “Graph classification via topological and label attributes,” in *Proc. 9th Int. Workshop Mining Learn. Graphs*, 2011, vol. 2, pp. 1–9.
- [5] S. Kairam, D. MacLean, M. Savva, and J. Heer, “GraphPrism: Compact visualization of network structure,” in *Proc. Int. Work. Conf. Adv. Visual Interfaces*, 2012, pp. 498–505.
- [6] D. Chakrabarti and C. Faloutsos, “Graph mining: Laws, generators, and algorithms,” *ACM Comput. Surv.*, vol. 38, no. 1, 2006, Art. no. 2.
- [7] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee, “Measurement and analysis of online social networks,” in *Proc. 7th ACM SIGCOMM Conf. Internet Meas.*, 2007, pp. 29–42.
- [8] P. G. Lind, M. C. Gonzalez, and H. J. Herrmann, “Cycles and clustering in bipartite networks,” *Phys. Rev. E*, vol. 72, no. 5, 2005, Art. no. 056127.
- [9] M. E. Newman, “Mixing patterns in networks,” *Phys. Rev. E*, vol. 67, no. 2, 2003, Art. no. 026126.
- [10] M. McGlohon, L. Akoglu, and C. Faloutsos, “Statistical properties of social networks,” in *Social Network Data Analytics*, Berlin, Germany: Springer, 2011, pp. 17–42.
- [11] S. Even and R. E. Tarjan, “Network flow and testing graph connectivity,” *SIAM J. Comput.*, vol. 4, no. 4, pp. 507–518, 1975.
- [12] G. Melancon, “Just how dense are dense graphs in the real world?: A methodological note,” in *Proc. AVI Workshop BEyond Time Errors: Novel Eval. Methods Inf. Vis.*, 2006, pp. 1–7.
- [13] D. Cartwright and F. Harary, “Structural balance: A generalization of heider’s theory,” *Psychological Rev.*, vol. 63, no. 5, 1956, Art. no. 277.
- [14] D. Loguinov, A. Kumar, V. Rai, and S. Ganesh, “Graph-theoretic analysis of structured peer-to-peer systems: Routing distances and fault resilience,” in *Proc. Conf. Appl. Technol. Architectures Protocols Comput. Commun.*, 2003, pp. 395–406.
- [15] R. A. Hanneman and M. Riddle, “Introduction to social network methods,” Riverside, CA, USA: Univ. California Riverside Press, 2005.
- [16] R. Albert, H. Jeong, and A.-L. Barabási, “Internet: Diameter of the world-wide web,” *Nature*, vol. 401, no. 6749, 1999, Art. no. 130.
- [17] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener, “Graph structure in the web,” *Comput. Netw.*, vol. 33, no. 1/6, pp. 309–320, 2000.
- [18] S. L. Feld, “The focused organization of social ties,” *Amer. J. Sociology*, vol. 86, no. 5, pp. 1015–1035, 1981.
- [19] O. Frank and F. Harary, “Cluster inference by using transitivity indices in empirical graphs,” *J. Amer. Statistical Assoc.*, vol. 77, no. 380, pp. 835–840, 1982.
- [20] M. Karlberg, “Testing transitivity in graphs,” *Social Netw.*, vol. 19, no. 4, pp. 325–343, 1997.
- [21] M. E. Newman, “The structure and function of complex networks,” *SIAM Rev.*, vol. 45, no. 2, pp. 167–256, 2003.
- [22] B. Uzzi and J. Spiro, “Collaboration and creativity: The small world problem,” *Amer. J. Sociology*, vol. 111, no. 2, pp. 447–504, 2005.
- [23] G. F. Davis, M. Yoo, and W. E. Baker, “The small world of the american corporate elite, 1982–2001,” *Strategic Org.*, vol. 1, no. 3, pp. 301–326, 2003.
- [24] H. Ebel, L.-I. Mielsch, and S. Bornholdt, “Scale-free topology of e-mail networks,” *Phys. Rev. E*, vol. 66, no. 3, 2002, Art. no. 035103.
- [25] D. J. Watts and S. H. Strogatz, “Collective dynamics of small-world networks,” *Nature*, vol. 393, no. 6684, 1998, Art. no. 440.

- [26] V. Van Noort, B. Snel, and M. A. Huynen, "The yeast coexpression network has a small-world, scale-free architecture and can be explained by a simple model," *Eur. Mol. Biol. Org. Rep.*, vol. 5, no. 3, pp. 280–284, 2004.
- [27] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, "Complex networks: Structure and dynamics," *Phys. Rep.*, vol. 424, no. 4/5, pp. 175–308, 2006.
- [28] R. Guimera, L. Danon, A. Diaz-Guilera, F. Giralt, and A. Arenas, "Self-similar community structure in a network of human interactions," *Phys. Rev. E*, vol. 68, no. 6, 2003, Art. no. 065103.
- [29] P. Sen, S. Dasgupta, A. Chatterjee, P. Sreeram, G. Mukherjee, and S. Manna, "Small-world properties of the indian railway network," *Phys. Rev. E*, vol. 67, no. 3, 2003, Art. no. 036106.
- [30] D. Wei-Bing, G. Long, L. Wei, and C. Xu, "Worldwide marine transportation network: Efficiency and container throughput," *Chin. Phys. Lett.*, vol. 26, no. 11, 2009, Art. no. 118901.
- [31] M. E. Newman, "Assortative mixing in networks," *Phys. Rev. Lett.*, vol. 89, no. 20, 2002, Art. no. 208701.
- [32] S. Maslov, K. Sneppen, and A. Zaliznyak, "Detection of topological patterns in complex networks: Correlation profile of the Internet," *Physica A: Statistical Mechanics Appl.*, vol. 333, pp. 529–540, 2004.
- [33] M. Boguná and R. Pastor-Satorras, "Epidemic spreading in correlated complex networks," *Phys. Rev. E*, vol. 66, no. 4, 2002, Art. no. 047104.
- [34] E. N. Gilbert, "Random graphs," *The Ann. Math. Statist.*, vol. 30, no. 4, pp. 1141–1144, 1959.
- [35] P. Erdős and A. Rényi, "On random graphs," *Publicationes Mathematicae*, vol. 6, pp. 290–297, 1959.
- [36] M. E. Newman and D. J. Watts, "Scaling and percolation in the small-world network model," *Phys. Rev. E*, vol. 60, no. 6, 1999, Art. no. 7332.
- [37] W. Deng, W. Li, X. Cai, and Q. A. Wang, "The exponential degree distribution in complex networks: Non-equilibrium network theory, numerical simulation and empirical data," *Physica A: Statistical Mechanics Appl.*, vol. 390, no. 8, pp. 1481–1485, 2011.
- [38] M. A. Javarone, "Gaussian networks generated by random walks," *J. Statistical Phys.*, vol. 159, no. 1, pp. 108–119, 2015.
- [39] D. D. S. Price, "A general theory of bibliometric and other cumulative advantage processes," *J. Assoc. Inf. Sci. Technol.*, vol. 27, no. 5, pp. 292–306, 1976.
- [40] E. N. Gilbert, "Random plane networks," *J. Soc. Ind. Appl. Math.*, vol. 9, no. 4, pp. 533–543, 1961.
- [41] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Sci.*, vol. 286, no. 5439, pp. 509–512, 1999.
- [42] S. N. Dorogovtsev, J. F. F. Mendes, and A. N. Samukhin, "Structure of growing networks with preferential linking," *Phys. Rev. Lett.*, vol. 85, no. 21, 2000, Art. no. 4633.
- [43] R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks," *Rev. Modern Phys.*, vol. 74, no. 1, 2002, Art. no. 47.
- [44] B. D. McKay, "Isomorph-free exhaustive generation," *J. Algorithms*, vol. 26, no. 2, pp. 306–324, 1998.
- [45] B. McKay, "Practical graph isomorphism", *Congr. Numer.*, vol. 30, pp. 45–87, 1981.
- [46] B. D. McKay and A. Piperno, "Practical graph isomorphism, II," *J. Symbolic Comput.*, vol. 60, pp. 94–112, 2014.
- [47] L. A. Goldberg, "Efficient algorithms for listing unlabeled graphs," *J. Algorithms*, vol. 13, no. 1, pp. 128–143, 1992.
- [48] G. Tinhofer, "Generating graphs uniformly at random," in *Computational Graph Theory*. Berlin, Germany: Springer, 1990, pp. 235–255.
- [49] J. D. Dixon and H. S. Wilf, "The random selection of unlabeled graphs," *J. Algorithms*, vol. 4, no. 3, pp. 205–213, 1983.
- [50] B. Bach, A. Spritzer, E. Lutton, and J.-D. Fekete, "Interactive random graph generation with evolutionary algorithms," in *Proc. Int. Symp. Graph Drawing*, 2012, pp. 541–552.
- [51] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proc. Nat. Acad. Sci. United States America*, vol. 99, no. 12, pp. 7821–7826, 2002.
- [52] A. Kennedy, K. Klein, A. Nguyen, and F. Y. Wang, "The graph landscape: Using visual analytics for graph set analysis," *J. Vis.*, vol. 20, no. 3, pp. 417–432, 2017.
- [53] X. Wu, X. Ying, K. Liu, and L. Chen, "A survey of privacy-preservation of graphs and social networks," in *Managing and Mining Graph Data*. Berlin, Germany: Springer, 2010, pp. 421–453.
- [54] C. Sun, S. Y. Philip, X. Kong, and Y. Fu, "Privacy preserving social network publication against mutual friend attacks," in *Proc. IEEE 13th Int. Conf. Data Mining Workshops*, 2013, pp. 883–890.
- [55] C. C. Aggarwal, Y. Li, and S. Y. Philip, "On the hardness of graph anonymization," in *Proc. IEEE 11th Int. Conf. Data Mining*, 2011, pp. 1002–1007.
- [56] X. Ying and X. Wu, "Randomizing social networks: A spectrum preserving approach," in *Proc. SIAM Int. Conf. Data Mining*, 2008, pp. 739–750.
- [57] U. Soni, Y. Lu, B. Hansen, H. Purchase, S. Kobourov, and R. Maciejewski, "The perception of graph properties in graph layouts," in *Proc. 20th IEEE Eurographics Conf. Vis.*, 2018, pp. 169–181.
- [58] A. Hagberg, P. Swart, and D. S. Chult, "Exploring network structure, dynamics, and function using networkX," Rep. No.: LA-UR-08-05495, Los Alamos National Lab.(LANL), Los Alamos, NM, 2008.
- [59] F. J. Massey Jr., "The kolmogorov-smirnov test for goodness of fit," *J. Amer. Statistical Assoc.*, vol. 46, no. 253, pp. 68–78, 1951.
- [60] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Trans. Math. Softw.*, vol. 22, no. 4, pp. 469–483, 1996.
- [61] J. Hintze and R. Nelson, "Violin plots: A box plot-density trace synergism," *The Amer. Statistician*, vol. 52, no. 2, pp. 181–184, May 1998.



Hang Chen received the BS degree in mathematics and computer science from the University of Arizona. He is working toward the graduate degree in the Department of Computer Science, University of Arizona. His research interests include graph theory, information visualisation, and machine learning.



Utkarsh Soni received the master's degree in computer science from Arizona State University. He is working toward the PhD degree in the School of Computing, Informatics, and Decision Systems Engineering, Arizona State University. His master's thesis was on Graph Perception. His current research interests are human-aware AI and human preference modeling.



Yafeng Lu received the PhD degree from Arizona State University. She is a software engineer at Bloomberg L.P. Before joining Bloomberg, she was a postdoctoral research associate with the School of Computing, Informatics, and Decision Systems Engineering, Arizona State University. Her research interests include predictive visual analytics, social media and textual data analysis.



Vahan Huroyan received the PhD degree in mathematics from the University of Minnesota. He is a postdoctoral research associate with the Department of Mathematics, University of Arizona and is affiliated with the US National Science Foundation Tripos research group. His research interests include machine learning, mathematical data analysis, computer vision, and non-convex optimization.



Ross Maciejewski received the PhD degree from Purdue University. He is an associate professor with the School of Computing, Informatics, and Decision Systems Engineering, Arizona State University. His research interests include visual analytics, geographic visualization, and data science.



Stephen Kobourov received the BS degree in mathematics and computer science from Dartmouth College and the MS and PhD degrees from Johns Hopkins University. He is a professor with the Department of Computer Science, University of Arizona. His research interests include information visualisation, graph theory, and geometric algorithms.

▷ For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/csdn.