scientific data

OPEN

DATA DESCRIPTOR

An Open Graph Dataset Organized by Scales

Ying Zhao[1], Xianzhe Zou[1], Xiao Wang[1], Zhanpeng Yang[1], Xuan Wang[1], Xin Zhao[1], Ning Zhang[2], Xin Huang[2] & Fangfang Zhou[1] ✉

Graph-related technologies, including social networks, transportation systems, and bioinformatics, are continually evolving in various application domains. The advancement of these technologies often relies on high-quality graph datasets for validating performance, such as scalability and time/space complexity. However, existing datasets are typically categorized by domains or types, lacking an explicit organization by scales and a wide range of scale levels. This situation may hinder comprehensive performance validations. This paper introduces an open graph dataset organized by scales named OGDOS. The dataset encompasses 470 preset scale levels, covering node counts from 100 to 200,000 and edge-to-node ratios from 1 to 10. The dataset combines scale-aligned real-world graphs and synthetic graphs, offering a versatile resource for evaluating various graph-related technologies. This paper also presents the OGDOS's construction process, provides a technical validation, and discusses its limitations.

Background & Summary

Graph is a fundamental data model that encodes entities and their relationships as nodes and edges, respectively, thereby facilitating the analysis and understanding of complex relational data. Graphs have played critical roles across a vast array of application domains, including social networks, transportation systems, bioinformatics, and scholarly citation networks[1–4].

In recent years, graph-related algorithms[5–7], such as community detection algorithms, graph sampling algorithms, and centrality analysis algorithms, have continuously advanced. Graph visualization methods[8,9], such as force-directed layout algorithms and graph rendering techniques, have been continuously improved. Well-established graph libraries and computing systems[10–12], such as NetworkX, Pregel, and G6.js, provide increasingly powerful and flexible tools for researchers and developers.

The abovementioned technological advancements are closely tied to the availability of high-quality graph datasets. The introduction of new algorithms requires running on graph datasets to validate scalability, time/space complexity, and stability[13–15]. Improvements in graph visualization methods require testing on graph datasets to assess the quality of graph layouts and the performance of graph rendering[16,17]. Newly released graph libraries and computing systems need to analyze graph datasets to ensure the efficient integration of innovative features[18–20].

Currently, many open graph datasets exist to support the advancement of graph technologies. We list the well-known open graph datasets in Table 1. These datasets have provided different explicit categorizations to support various research purposes. Common categorization perspectives include application domains (e.g., social networks, web networks, and transportation networks), data sources (e.g., Twitter, citation databases, and the human brain), and graph types (e.g., directed or undirected graphs, dynamic or static graphs, and planar or nonplanar graphs). Most of the datasets allow for categorizations by application domains or data sources to facilitate domain-specific research. A small number of datasets allow categorization by graph types, supporting the analysis of distinct graph structures. Additionally, there are domain-specific datasets designed for targeted applications, such as Transportation Networks[21] for traffic assignment, Graph Layout Benchmark Datasets[22] for layout algorithms, and the Open Graph Benchmark[23] for graph machine learning, which facilitate research and development in their respective fields.

However, existing open graph datasets rarely categorize graphs explicitly according to scale levels. This situation is generally reflected in two aspects. First, existing datasets do provide the basic scale information of each graph, such as the number of nodes and edges, but they commonly do not organize graphs into distinct

[1]School of Computer Science and Engineering, Central South University, Changsha, China. [2]Qi An Xin Technology Group Inc., Layer Platform, Beijing, China. ✉e-mail: zff@csu.edu.cn

Name	URL	Classification Perspectives	Target Applications
Network Repository*[39]	https://networkrepository.com/	Data sources, Application domains	\
UCI Network Data repository*[40]	https://networkdata.ics.uci.edu/	Data sources, Application domains	\
SNAP*[41]	https://snap.stanford.edu/data/	Data sources, Application domains	Graph algorithmic benchmarks
Network corpus*[42]	https://github.com/micrograviтas/network-corpus	Data sources, Application domains	\
SocioPatterns*[43]	http://www.sociopatterns.org/datasets/	Data sources, Application domains	\
HIN-Datasets-for-Recommendation-and-Network-Embedding[44]	https://github.com/librahu/HIN-Datasets-for-Recommendation-and-Network-Embedding	Data sources, Application domains	Network embedding and recommendation
TransportationNetworks[21]	https://github.com/bstabler/TransportationNetworks	Data sources, Application domains	Traffic assignment
FirstCourse NetworkScience*[45]	https://github.com/CambridgeUniversityPress/FirstCourseNetworkScience	Data sources, Application domains	\
Graph Layout Benchmark Datasets*[22]	https://visdunneright.github.io/gd_benchmark_sets/	Data sources, Application domains, Graph types	Layout algorithms benchmark
House of Graphs[46]	https://houseofgraphs.org	Graph types	Graph theroies analysis
Open Graph Benchmark[23]	https://ogb.stanford.edu/	Applications domains	Machine learning on graphs

Table 1. The list of well-know graph datasets, detailing with the categorization perspectives and the target applications they provided. Datasets marked with "*" indicate that some graphs have been processed and integrated into OGDOS.

scale categories. Second, existing datasets may not fully cover a wide range of scale levels. That is, some scale levels may be absent in a graph dataset. A scale-oriented dataset is essential for research and development in graph-related technologies. The main reason is that the relationship between the scalability of a graph-related technique and the scale of the graph is complex[24,25]. Evaluation experiments that are conducted on a graph dataset covering wide-range graph scales are necessary to comprehensively assess the relationship between the scalability and graph scales.

To address these limitations, we present an approach involving an open graph dataset organized by a scale named OGDOS. Our OGDOS categorizes graphs into 47 node-scale levels (ranging from 100 to 200,000 nodes) and 10 edge-to-node ratio levels (ranging from 1 to 10), resulting in 470 unique graph scale levels. OGDOS carefully selects reliable graph data from both real-world and synthesized sources, offering comprehensive coverage of these graph scale levels. Additionally, OGDOS covers various application domains, such as social networks, transportation, and biology. It also covers various graph types, including scale-free networks and small-world networks.

Methods

The construction process of OGDOS follows a tabular-filling approach. First, we predefined the node and edge scale levels of OGDOS to explicitly cover a wide range of graph scales. Second, we prioritized selecting real-world graphs from common graph datasets to cover the predefined scale levels in OGDOS, ensuring that OGDOS includes high-quality real-world graphs and covers multiple graph application domains. Finally, we generated diverse types of synthetic graphs with rich topological structures to cover the remaining scale levels. The source types of the graphs in OGDOS are provided in Table 2. The detailed methods for these three steps are provided in the following sections.

Presetting Graph Scale Levels.

We preset the graph scale levels in OGDOS on two dimensions: node scale levels and edge scale levels. Users generally consider node scales, rather than edge scales, as the primary factor when evaluating the size of a graph. Therefore, we used the node count to define the node scale levels as the main factor and used the edge-to-node ratio (i.e., the number of edges divided by the number of nodes) to define the edge scale levels as the secondary factor.

The node count in OGDOS ranges from 100 to 200,000, covering a wide variety of graph application scenarios. Pilot experiments on common graph-related applications (e.g., graph visualization and centrality analysis) indicate that without costly acceleration technologies, processing graphs with more than 200,000 nodes on mainstream desktop computers is prohibitively time-consuming. As a result, the maximum node count was capped at 200,000. Within this range, we preset 47 node scale levels distributed across four progressively increasing intervals. In particular, for nodes ranging from 100 to 1,000, we predefined 10 levels with an interval of 100 nodes. For number of nodes ranging from 1,000 to 10,000, we predefined 9 levels with an interval of 1,000 nodes. For number of nodes ranging from 10,000 to 100,000, we selected 18 levels with an interval of 5,000 nodes. For number of nodes ranging from 100,000 to 200,000, we selected 10 levels with an interval of 10,000 nodes. This approach provides fine-grained scale levels for small- to medium-size graphs while offering coarser-grained scale levels for large-size graphs, thus meeting the evaluation needs of a wide range of graph applications.

The edge scale levels were preset from 1 to 10 edge-to-node ratios with a fixed interval. These edge scale levels cover a variety of graph types. For example, sparse graphs (e.g., transportation, biological, and logistics networks) typically exhibit edge-to-node ratios of approximately 1[26], whereas small-world graphs[27] (e.g., brain networks, power grids, and social networks) tend to have edge-to-node ratios within 10. Scale-free graphs (e.g.,

Source Type		Edge-to-node Ratios									
		1	2	3	4	5	6	7	8	9	10
Node Scales	100	Real-O	Real-O	Real-O	Real-T	Real-T	Real-T	Syn-S	Real-T	Syn-S	Syn-S
	200	Real-O	Real-O	Real-O	Real-O	Real-O	Real-T	Real-O	Real-T	Syn-S	Syn-S
	300	Real-O	Real-O	Real-T	Real-T	Real-T	Real-O	Syn-S	Syn-S	Syn-S	Syn-S
	400	Real-O	Real-T	Real-T	Real-T	Syn-F	Syn-S	Real-T	Syn-S	Syn-S	Syn-S
	500	Real-T	Real-T	Real-T	Real-T	Real-T	Syn-F	Real-T	Syn-S	Syn-S	Real-O
	600	Real-T	Real-T	Real-T	Real-T	Syn-F	Real-T	Syn-S	Real-T	Syn-S	Real-T
	700	Real-O	Real-T	Real-T	Syn-F	Real-O	Syn-F	Real-T	Syn-S	Syn-S	Syn-S
	800	Real-T	Real-T	Real-T	Real-T	Real-T	Real-O	Syn-S	Real-T	Syn-S	Syn-S
	900	Real-T	Real-T	Real-T	Real-T	Real-T	Syn-F	Syn-F	Real-O	Syn-S	Syn-S
	1k	Real-O	Real-T	Real-T	Real-T	Real-T	Real-O	Real-T	Syn-F	Syn-S	Real-T
	2k	Real-O	Real-T	Real-O	Real-T	Real-T	Real-T	Real-T	Real-O	Syn-S	Real-T
	3k	Real-T	Real-T	Real-T	Real-T	Syn-F	Real-T	Syn-F	Syn-F	Syn-S	Syn-S
	4k	Real-O	Real-O	Real-T	Real-T	Real-O	Real-T	Syn-F	Syn-F	Real-O	Syn-S
	5k	Real-T	Real-T	Real-O	Real-T	Syn-F	Real-O	Real-T	Syn-F	Syn-S	Syn-S
	6k	Real-T	Real-T	Real-T	Syn-F	Syn-F	Real-T	Real-T	Real-T	Syn-S	Syn-S
	7k	Real-T	Real-O	Real-T	Real-T	Real-T	Real-T	Real-T	Real-T	Syn-S	Syn-S
	8k	Real-T	Real-O	Real-T	Real-T	Real-O	Syn-F	Syn-F	Syn-F	Syn-S	Syn-S
	9k	Real-T	Real-T	Real-T	Real-T	Syn-F	Real-T	Syn-F	Syn-F	Syn-F	Syn-S
	10k	Real-T	Real-O	Real-O	Real-T	Real-T	Real-T	Real-T	Real-T	Syn-F	Real-T
	15k	Real-T	Real-T	Real-O	Real-T	Real-T	Real-T	Real-T	Syn-F	Syn-F	Real-T
	20k	Real-T	Real-T	Real-O	Real-T	Real-O	Syn-F	Syn-F	Real-T	Syn-F	Syn-S
	25k	Real-T	Real-T	Real-T	Syn-F	Syn-F	Syn-F	Real-T	Syn-F	Syn-F	Syn-F
	30k	Real-T	Real-T	Real-T	Real-O	Syn-F	Syn-F	Real-O	Syn-F	Syn-F	Syn-F
	35k	Real-T	Real-T	Real-T	Real-O	Real-T	Syn-F	Syn-F	Real-T	Syn-F	Syn-F
	40k	Real-T	Real-O	Real-O	Real-T	Syn-F	Real-T	Real-T	Real-O	Syn-F	Syn-F
	45k	Syn-F	Real-T	Syn-F	Real-T	Real-T	Syn-F	Syn-F	Syn-F	Syn-F	Syn-F
	50k	Syn-F	Real-T	Real-T	Real-T	Real-T	Real-T	Syn-F	Syn-F	Syn-F	Syn-F
	55k	Syn-F	Real-T	Syn-F	Real-T	Real-T	Syn-F	Syn-F	Syn-F	Real-O	Syn-F
	60k	Real-T	Real-T	Syn-F	Real-T	Real-T	Real-T	Syn-F	Syn-F	Syn-F	Syn-F
	65k	Syn-F	Real-O	Real-T	Syn-F	Syn-F	Syn-F	Real-T	Syn-F	Syn-F	Syn-F
	70k	Real-T	Real-T	Real-T	Syn-F	Syn-F	Syn-F	Syn-F	Syn-F	Syn-F	Syn-F
	75k	Syn-F	Real-T	Real-T	Syn-F	Real-T	Real-T	Syn-F	Syn-F	Syn-F	Syn-F
	80k	Real-T	Real-T	Syn-F	Syn-F	Syn-F	Real-T	Syn-F	Syn-F	Syn-F	Syn-F
	85k	Syn-F	Real-T	Real-T	Real-T	Syn-F	Syn-F	Syn-F	Syn-F	Syn-F	Syn-F
	90k	Real-T	Syn-F	Syn-F	Real-T	Syn-F	Syn-F	Syn-F	Syn-F	Syn-F	Syn-F
	95k	Real-T	Syn-F	Syn-F	Syn-F	Syn-F	Syn-F	Syn-F	Syn-F	Syn-F	Syn-F
	100k	Syn-F	Real-T	Real-O	Syn-F	Syn-F	Syn-F	Syn-F	Syn-F	Syn-F	Syn-F
	110k	Real-T	Real-T	Syn-F	Syn-F	Syn-F	Syn-F	Syn-F	Syn-F	Syn-F	Syn-F
	120k	Real-T	Syn-F	Real-T	Syn-F	Syn-F	Syn-F	Syn-F	Syn-F	Syn-F	Syn-F
	130k	Real-T	Syn-F	Real-T	Real-T	Syn-F	Syn-F	Syn-F	Syn-F	Syn-F	Syn-F
	140k	Real-T	Syn-F	Real-O	Real-T	Syn-F	Syn-F	Real-T	Syn-F	Syn-F	Syn-F
	150k	Real-T	Real-O	Syn-F	Real-T	Real-T	Syn-F	Syn-F	Syn-F	Syn-F	Syn-F
	160k	Real-T	Real-T	Syn-F	Syn-F	Syn-F	Syn-F	Syn-F	Syn-F	Syn-F	Syn-F
	170k	Real-T	Real-T	Syn-F	Syn-F	Syn-F	Syn-F	Syn-F	Syn-F	Syn-F	Syn-F
	180k	Syn-F	Real-T	Syn-F	Real-T	Syn-F	Syn-F	Syn-F	Syn-F	Syn-F	Syn-F
	190k	Syn-F	Syn-F	Real-O	Real-T	Real-T	Syn-F	Syn-F	Syn-F	Syn-F	Syn-F
	200k	Syn-F	Syn-F	Real-T	Real-T	Real-O	Syn-F	Syn-F	Syn-F	Syn-F	Syn-F

Table 2. The source types of graphs in OGDOS. Each cell corresponds to a specific combination of a preset node count (rows) and edge-to-node ratio (columns). The Real-O represents an original real-world graph, and the Real-T represents a real-world graph after tuning. Syn-S represents a synthesized small-world graph, Syn-F represents a synthesized scale-free graph.

social networks, internet topology, and scientific collaboration networks) typically have edge-to-node ratios between 1 and 5, with some reaching between 6 and 10, although very few exceed 10[28].

Each combination of a node scale and an edge scale corresponds to a scale level in OGDOS, resulting in a total of 470 scale levels (47 node scale levels × 10 edge-to-node ratios). All graphs in OGDOS were standardized as simple graphs, with edges and nodes being attribute-free and content-agnostic, ensuring uniformity across graphs from different sources in OGDOS.

Selecting and Tuning Real-world Graphs. We selected real-world graphs from the common datasets in Table 1 to cover the corresponding scale levels in OGDOS. However, graphs that perfectly match the preset scale levels are rare. In practice, many graphs exhibit slight deviations from our preset scale levels. These graphs were utilized by fine-tuning their scale with a consideration to preserve their overall structural properties. The process for selecting and tuning real-world graphs is as follows:

- **Step** 1: Finding the target scale levels. We traverse the predefined list of node scale levels and calculate the numerical difference between each level and the node count in the current graph. The node scale level with the smallest difference is selected as the target node scale. Similarly, the target edge-to-node ratio is selected by comparing the current graph's edge-to-node ratio with the predefined ratios. The combination of the target node scale and edge-to-node ratio is identified as the target scale level.
- **Step** 2: Determining how to adjust. The quality of the selected real-world graphs is retained only by retaining graphs where both the node count and the edge-to-node ratio are between 95% and 110% of the target scale level. On the basis of the current graph's node count and edge-to-node ratio, we categorize these graphs into two scenarios. (1) If the node count of the current graph is between 95% and 105% of the target node scale and its edge-to-node ratio is also between 95% and 105% of the target ratio, then the scheme is retained. In this case, the graph is considered sufficiently close to the target scale levels, and it is selected without modification. (2) For graphs that fall outside this threshold, we adjust the node and edge counts of the current graph. If the node count exceeds 105% of the target node scale, then the number of nodes is initially reduced via Step 3, and the edge-to-node ratio is further evaluated. If the edge-to-node ratio exceeds 105% of the target ratio, then the number of edges is reduced via Step 4. Adjusted graphs are selected to represent the corresponding scale levels.
- **Step** 3: Reducing the number of nodes. The nodes in the current graph are ranked in ascending order by degree, and excess nodes and their connected edges are removed until the node count falls within 105% of the target node scale to preserve nodes that significantly affect the overall graph structure. We also skip the removal of a node when its deletion increases the number of connected components in the graph[29].
- **Step** 4: Reducing the number of edges. An edge is considered to have minimal impact on the overall structure of a graph if it satisfies the following two conditions: (1) its removal does not increase the number of connected components in the graph[30], and (2) its removal does not reduce the number of closed triplets, thereby preserving the global clustering coefficient (GCC) of the graph. The GCC is defined as the ratio of closed triplets (triangles) to all possible triplets (two edges with a shared node) in a graph. A higher GCC indicates denser local clustering of nodes, which is often associated with richer topological structures[31,32]. Edges meeting both conditions are prioritized for removal until the edge-to-node ratio is reduced to within 105% of the target ratio.

Moreover, all complex graphs are first converted to simple graphs before selection and tuning and are then marked as fine-tuned graphs in Table 2. As a result, out of 224 real-world graphs, 49 were selected without modification, whereas 175 were selected after converting or fine-tuning.

Generating Synthetic Graphs. For scale levels with edge-to-node ratios ranging from 1 to 10 and not covered by real-world graphs, we opted to generate small-world or scale-free networks to cover corresponding scale levels for two reasons. First, many real-world networks exhibit small-world and scale-free properties. For example, in real-world networks such as social networks, the internet, and genetic networks, nodes may not be directly connected but can be reached through relatively short path lengths, which are characteristic of small-world networks[33,34]. In financial, propagation, and internet architecture networks, a small number of nodes play a crucial role in connecting various parts of the network, displaying scale-free behavior[33,35,36]. Second, these two types of networks offer richer graph structures than randomly generated networks do, which are essential for supporting graph applications that require more complex topological features.

We generated small-world networks for scale levels that satisfied the condition $n \geq k \geq ln(n)$, where n represents the number of nodes and k represents the edge-to-node ratio. Graphs meeting this condition are considered to have many sparsely connected nodes and rich structural topologies, which supports certain graph-theoretic applications[27]. We used the watts_strogatz_graph function from NetworkX to generate a small-word graph[10], with the rewiring probability p set to the commonly used value of 0.025. The graphs generated by the Watts-Strogatz model in this function exhibit key properties similar to those of real-world networks, such as homophily (the tendency of similar nodes to connect) and weak ties (connections that bridge distant clusters)[32].

For the remaining scale levels, we used the barabasi_albert_graph function from NetworkX to generate graphs[10]. The Barabási-Albert model is a fundamental model for constructing normal scale-free networks[36]. However, the graphs generated by this method may not have rich topological structures to support common graph applications. Similar to Step 4 in last section, we used the GCC as an indicator of structural richness to evaluate the quality of the generated scale-free graphs. For each combination of a node scale and an edge scale, we generated 20 graphs and selected the graph with the highest GCC to represent the graph scale level[31,32].

Thus, among the 246 synthetic graphs, 40 are small-world graphs, and 206 are scale-free graphs.

Distance Metrics	Kolmogorov-Smirnov Distance (KSD)				Skew Divergence Distance (SDD)				L2-normalization Distance (L2ND)			
	Fine-tune*	RE	RNE	TIES	Fine-tune*	RE	RNE	TIES	Fine-tune*	RE	RNE	TIES
Degree Distribution	0.138 ± 0.014	0.480 ± 0.036	0.142 ± 0.026	0.153 ± 0.014	0.008 ± 0.002	0.023 ± 0.003	0.008 ± 0.002	0.005 ± 0.001	0.052 ± 0.01	0.153 ± 0.014	0.048 ± 0.01	0.092 ± 0.006
Clustering Coefficient	0.072 ± 0.019	0.332 ± 0.039	0.077 ± 0.019	0.077 ± 0.011	0.005 ± 0.002	0.019 ± 0.003	0.006 ± 0.002	0.003 ± 0.001	0.036 ± 0.009	0.127 ± 0.015	0.034 ± 0.009	0.082 ± 0.007
PageRank Distribution	0.137 ± 0.019	0.415 ± 0.024	0.154 ± 0.024	0.420 ± 0.026	0.011 ± 0.002	0.011 ± 0.002	0.018 ± 0.002	0.019 ± 0.003	0.065 ± 0.012	0.161 ± 0.009	0.064 ± 0.012	0.163 ± 0.011

Table 3. Results of technique validation experiment. Mean KSD, SDD, and L2ND values with standard error (95% confidence intervals) for degree, clustering coefficient, and PageRank distributions measured across all 77 fine-tuned scale levels. For our fine-tuning approach (indicated by "*"), structural metrics were computed between the original and fine-tuned graphs for each level. For the reference methods, the original simple graph was sampled five times per scale level (with the edge count adjusted to match that of the fine-tuned version), and the metrics were computed in the same way. Values close to zero indicate high similarity between the distributions.

Data Records

OGDOS collection. The proposed OGDOS[37] are available in the following figshare repository: https://doi.org/10.6084/m9.figshare.27992339. The source types of the graphs in OGDOS are provided in Table 2. All real-world graphs in OGDOS are sourced from well-known graph datasets listed in Table 1 and have undergone careful selection and refinement. The process of selecting and refining these graphs is documented in the '.py' files in the Code folder.

The data accessibility is facilitated by storing each graph in OGDOS as a ".json" file in a specific directory path, which reflects its corresponding scale levels. For example, a ".json" file located at the "…\OGDOS \ 100 \ 1 \" folder represents a graph with a node scale of 100 and an edge-to-node ratio of 1.

Technical Validation

In constructing OGDOS, we applied a fine-tuning approach on a subset of real-world graphs to adjust the numbers of nodes and edges for matching predefined scale levels. This process selectively removes nodes and edges with the goal of preserving the original structural properties. To verify the preservation, we conducted a validation experiment.

Metric Selection. We assessed the similarity between the original and fine-tuned simple graphs by comparing three common structural distributions (the degree distribution (DD), clustering coefficient (CC), and PageRank (PR) distribution), using three distance metrics[6]. Specifically, Kolmogorov-Smirnov Distance (KSD) is defined as the maximum absolute difference between the cumulative distribution functions of the distributions, providing a global measure of distributional change. Skew Divergence Distance (SDD) quantifies differences between the histogram representations of the distributions, effectively capturing the difference in their overall distribution shapes. L2-normalization Distance (L2ND) is calculated as the normalized L2 norm of the differences between the histograms, offering a scale-independent measure of the magnitude of changes. For the three distance metrics, values close to zero indicate high similarity between the structural distributions of the original and fine-tuned graphs.

Reference Methods. Our fine-tuning approach can be regarded as an edge sampling method combined with node removal. Therefore, we selected three reference edge sampling methods, namely, Random Edge Sampling (RE), Random Node-Edge Sampling (RNE), and Totally Induced Edge Sampling (TIES). RE is a basic edge sampling method that serves as a foundational baseline. RNE and TIES represent sophisticated techniques designed to respectively balance local node properties and global structural features[38].

Experimental Design. We evaluated our fine-tuning approach across all 77 fine-tuned graph scale levels. For our fine-tuning approach, structural metrics were computed between the original and fine-tuned simple graphs for each level, and results were averaged. For the reference methods, the original simple graph was sampled five times per scale level (with the edge count adjusted to match that of the fine-tuned version), and the metrics were computed and averaged in the same way.

Results and Analysis. Table 3 summarizes the results. Our fine-tuning approach consistently produces KSD values below 0.16, SDD values below 0.02, and L2ND values below 0.08 across all the three distributions, indicating that the fine-tuned graphs closely resemble the original graphs in terms of the structural properties.

In further comparison with the reference methods, our fine-tuning approach demonstrates superior performance for all KSD metrics across the three distributions, likely because our approach effectively preserves the nodes and edges that significantly impact the graph's structural properties. For all SDD metrics, our approach outperforms RE, is comparable to RNE, and slightly trails TIES in the degree and clustering coefficient distributions, likely due to TIES's global optimization strategy, whereas our method emphasizes local node degree and connectivity. For all L2ND metrics, our approach surpasses both RE and TIES and performs on par with RNE, underscoring its effectiveness in minimizing magnitude differences.

Usage Notes

OGDOS is a collection of multiscale simple graphs; thus, it is a versatile resource for evaluating various graph-related technologies. The dataset supports fundamental graph computing methods (e.g., graph partitioning, graph traversal and random walks), graph visualization methods (e.g., graph simplification, graph layout and graph rendering) and graph analysis methods (e.g., hotspot analysis, propagation modeling, and epidemic prediction).

OGDOS allows the evaluation of these technologies along two key dimensions: node scale and edge scale. Regarding node scale, the dataset encompasses graphs with node scales ranging from 100 to 200,000. By selecting graphs with a uniform edge-to-node ratio and applying the corresponding techniques across different node scales, researchers can obtain approximate performance metrics (e.g., processing time, visualization frame rates, and memory usage) across multiple node scales. This strategy enables node-centric evaluations of how the performance of various methods changes with node scales. Similarly, by selecting graphs with a uniform node scale levels and applying the corresponding techniques across different edge-to-node ratios, OGDOS enables edge-centric evaluations of how the performance of various methods changes with edge scales.

OGDOS also exhibits several limitations that should be considered:

Simple, attribute-free storage format. Graphs in OGDOS are stored in a simple, attribute-free format, which conceals attribute differences across various graph sources (e.g. real-world or synthesized sources). While this design facilitates a broad range of fundamental performance evaluations, it restricts the OGDOS's applicability for tasks requiring specific node or edge attributes or complex graph structures, such as protein interaction networks, dynamic graph learning, or graph-based recommendation systems. Notably, some graphs in OGDOS originate from real-world datasets and include detailed attributes; users seeking such information are encouraged to consult the original sources (indicated by "*" in Table 1).

Scale-based organization. OGDOS is uniquely organized by node scale and edge-to-node ratio, which is sufficient for general performance evaluations. However, many public datasets also classify graphs by domains, types, or additional structural features. Incorporating these dimensions through further subdivision of each scale level could enhance the dataset's utility. Future work may consider integrating these additional classification criteria.

Variable node scale intervals. OGDOS employs fine-grained intervals for small graphs (e.g., 100-1,000 nodes at intervals of 100) and coarser intervals for large graphs (e.g., 100k-200k nodes at intervals of 10k), to capture the non-linear performance changes of many graph-related techniques. While this design can effectively highlight overall performance trends, it provides only approximate estimates for applications that require evaluation at specific scales (e.g., exactly 1,050 or 10,051,500 nodes). In such cases, users may need to interpolate or extrapolate the available performance metrics.

Code availability

The code for processing real-world graphs is publicly available in the Supplyment Information. The repository includes instructions on how to convert the format of source files, how to judge and fine-tune complex graphs, how to find the target scale levels, and how to filter and refine real-world graphs. All synthetic graphs in OGDOS are generated using the barabasi_albert_graph or watts_strogatz_graph functions from Python NetworkX library (version 2.8.8), which offers flexible functions and parameters (such as target node scale and edge-to-node ratio) to generate the desired graph datasets.

Received: 16 December 2024; Accepted: 28 April 2025;
Published online: 28 May 2025

References

1. Sahoo, S. R. & Gupta, B. Multiple features based approach for automatic fake news detection on social networks using deep learning. *Appl. Soft Comput.* **100**, 106983, https://doi.org/10.1016/j.asoc.2020.106983 (2021).
2. Gu, Y., Fu, X., Liu, Z., Xu, X. & Chen, A. Performance of transportation network under perturbations: Reliability, vulnerability, and resilience. *Transp. Res. Part E: Logist. Transp. Rev.* **133**, 101809, https://doi.org/10.1016/j.tre.2019.11.003 (2020).
3. Szklarczyk, D. *et al.* The string database in 2021: customizable protein-protein networks, and functional characterization of user-uploaded gene/measurement sets. *Nucleic Acids Res.* **49**, D605–D612, https://doi.org/10.1093/nar/gkaa1074 (2020).
4. Goyal, K. & Kumar, S. Financial literacy: A systematic review and bibliometric analysis. *Int. J. Consumer Stud.* **45**, 80–105, https://doi.org/10.1111/ijcs.12605 (2021).
5. Tang, L. & Liu, H.*Community detection and mining in social media*. Synthesis Lectures on Data Mining and Knowledge Discovery (Springer International Publishing, 2022).
6. Zhao, Y. *et al.* Preserving minority structures in graph sampling. *IEEE Transactions on Vis. Comput. Graph.* **27**, 1698–1708, https://doi.org/10.1109/TVCG.2020.3030428 (2021).
7. Saxena, A. & Iyengar, S. Centrality measures in complex networks: A survey. Preprint at https://arxiv.org/abs/2011.07190 (2020).
8. Kobourov, S. G. Spring embedders and force directed graph drawing algorithms. Preprint at https://arxiv.org/abs/1201.3011 (2012).
9. Kwon, O.-H., Muelder, C., Lee, K. & Ma, K.-L. A study of layout, rendering, and interaction methods for immersive graph visualization. *IEEE Transactions on Vis. Comput. Graph.* **22**, 1802–1815, https://doi.org/10.1109/TVCG.2016.2520921 (2016).
10. Hagberg, A., Swart, P. J. & Schult, D. A. Exploring network structure, dynamics, and function using networkx. *osti* https://www.osti.gov/biblio/960616 (2008).
11. Malewicz, G. *et al.* Pregel: a system for large-scale graph processing. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, 135–146, https://doi.org/10.1145/1807167.1807184 (2010).

12. Wang, Y. *et al*. G6: A web-based library for graph visualization. *Vis. Informatics.* **5**, 49–55, https://doi.org/10.1016/j.visinf.2021.12.003 (2021).
13. Xian, X., Semenov, A., Hu, Y., Wang, A. & Jin, Y. Adaptive sampling and quick anomaly detection in large networks. *IEEE Transactions on Autom. Sci. Eng.* **20**, 2253–2267, https://doi.org/10.1109/TASE.2022.3214193 (2023).
14. Çatalyürek, U. *et al*. More recent advances in (hyper)graph partitioning. *ACM Comput.Surv.* **55**, 38, https://doi.org/10.1145/3571808 (2023).
15. Ying, Z., Xin, Z., Siming, C., Zhuo, Z. & Xin, H. An Indoor Crowd Movement Trajectory Benchmark Dataset. *IEEE Transactions on Reliability.* **70**, 1368–1380, https://doi.org/10.1109/TR.2021.3109122 (2021).
16. Balci, H. & Dogrusoz, U. fcose: A fast compound graph layout algorithm with constraint support. *IEEE Transactionson Vis. Comput. Graph.* **28**, 4582–4593, https://doi.org/10.1109/TVCG.2021.3095303 (2022).
17. Horak, T., Kister, U. & Dachselt, R. Comparing rendering performance of common web technologies for large graphs. In *Poster Program of the 2018 IEEE VIS Conference* (2018).
18. Michail, D., Kinable, J., Naveh, B. & Sichi, J. V. Jgrapht—a java library for graph data structures and algorithms. *ACM Trans. Math. Softw.* **46**, 29, https://doi.org/10.1145/3381449 (2020).
19. Lu, Y., Cheng, J., Yan, D. & Wu, H. Large-scale distributed graph computing systems: an experimental evaluation. *Proc. VLDB Endow.* **8**, 281–292, https://doi.org/10.14778/2735508.2735517 (2014).
20. Ying, Z. *et al*. A Benchmark for Visual Analysis of Insider Threat Detection. *SCIENCE CHINA Information Sciences.* **65**, 1–4, https://doi.org/10.1007/s11432-019-2776-4 (2022).
21. Ben Stabler, H. B.-G. & Sall, E. Transportationnetworks. *TransportationNetworks* https://github.com/bstabler/TransportationNetworks (2023).
22. Di Bartolomeo, S., Puerta, E., Wilson, C., Crnovrsanin, T. & Dunne, C. A collection of benchmark datasets for evaluating graph layout algorithms. *Graph Layout Benchmark Datasets* https://visdunneright.github.io/gd_benchmark_sets (2023).
23. Hu, W. *et al*. Open graph benchmark: Datasets for machine learning on graphs. *Adv. Neural Inf. Process Syst.* **33**, 22118–22133, https://proceedings.neurips.cc/paper/2020/hash/fb60d411a5c5b72b2e7d3527cfc84fd0-Abstract.html (2020).
24. Heidari, S., Simmhan, Y., Calheiros, R. N. & Buyya, R. Scalable graph processing frameworks: A taxonomy and open challenges. *ACM Comput. Surv.* **51**, 53, https://doi.org/10.1145/3199523 (2018).
25. Yildirim, H., Chaoji, V. & Zaki, M. J. Grail: scalable reachability index for large graphs. *Proc. VLDB Endow.* **3**, 276–284, https://doi.org/10.14778/1920841.1920879 (2010).
26. Singh, A. & Humphries, M. D. Finding communities in sparse networks. *Sci. reports.* **5**, 8828, https://doi.org/10.1038/srep08828 (2015).
27. Watts, D. J. & Strogatz, S. H. Collective dynamics of 'small-world'networks. *nature.* **393**, 440–442, https://doi.org/10.1038/30918 (1998).
28. Melancon, G. Just how dense are dense graphs in the real world? a methodological note. In *Proceedings of the 2006 AVI Workshop on BEyond Time and Errors: Novel Evaluation Methods for Information Visualization*, 1–7, https://doi.org/10.1145/1168149.1168167 (2006).
29. Pavez, E., Egilmez, H. E. & Ortega, A. Learning graphs with monotone topology properties and multiple connected components. *IEEE Transactions on Signal Processing* **66**, 2399–2413, https://doi.org/10.1109/TSP.2018.2813337 (2018).
30. Chang, L. *et al*. Efficiently computing k-edge connected components via graph decomposition. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, 205–216, https://doi.org/10.1145/2463676.2465323 (2013).
31. Opsahl, T. Triadic closure in two-mode networks: Redefining the global and local clustering coefficients. *Soc. Networks.* **35**, 159–167, https://doi.org/10.1016/j.socnet.2011.07.001 (2013).
32. Luce, R. D. & Perry, A. D. A method of matrix analysis of group structure. *Psychometrika.* **14**, 95–116, https://doi.org/10.1007/BF02289146 (1949).
33. Wang, X. F. & Chen, G. Complex networks: small-world, scale-free and beyond. *IEEE Circuits Syst. Mag.* **3**, 6–20, https://doi.org/10.1109/MCAS.2003.1228503 (2003).
34. Takes, F. W. & Kosters, W. A. Determining the diameter of small world networks. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, 1191–1196, https://doi.org/10.1145/2063576.2063748 (Association for Computing Machinery, 2011).
35. Barabási, A.-L. Scale-free networks: A decade and beyond. *Science.* **325**, 412–413, https://doi.org/10.1126/science.1173299 (2009).
36. Holme, P. & Kim, B. J. Growing scale-free networks with tunable clustering. *Phys. Rev. E* **65**, 026107, https://doi.org/10.1103/PhysRevE.65.026107 (2002).
37. Zhao, Y. *et al*. OGDOS: An open graph dataset organized by scales. Figshare https://doi.org/10.6084/m9.figshare.27992339 (2025).
38. Leskovec, J. & Faloutsos, C. Sampling from large graphs. *Proceedings of the ACM SIGKDD international conference on Knowledge discovery and data mining.* **6**, 631–636, https://doi.org/10.1145/1150402.1150479 (2006).
39. Rossi, R. & Ahmed, N. The network data repository with interactive graph analytics and visualization. *Proc. AAAI Conf. on Artif. Intell.* **29**, 4292–4293, https://doi.org/10.1609/aaai.v29i1.9277 (2015).
40. DuBois, C. L. UCI network data repository. *networkdata* http://networkdata.ics.uci.edu (2008).
41. Leskovec, J. & Sosič, R. Snap: A general-purpose network analysis and graph-mining library. *ACM Trans. Intell. Syst. Technol.* **8**, https://doi.org/10.1145/2898361 (2016).
42. Microgravitas & Pgdr. networkcorpus. *networkcorpus* https://github.com/microgravitas/network-corpus/tree/master#network-corpus (2019).
43. Barrat, A., Cattuto, J. P. C. & den Broeck, W. V. Sociopatterns. *SocioPatterns* http://www.sociopatterns.org/datasets (2008).
44. Librahu.Hin-datasets-for-recommendation-and-network-embedding. *HIN-Datasets-for-Recommendation-and-Network-Embedding* https://github.com/librahu/HIN-Datasets-for-Recommendation-and-Network-Embedding (2019).
45. Filippo Menczer, S. F. & Davis, C. A. Firstcoursenetworkscience. *FirstCourseNetworkScience* https://github.com/CambridgeUniversityPress/FirstCourseNetworkScience (2023).
46. Coolsaet, K., D'hondt, S. & Goedgebeur, J. House of Graphs 2.0: A database of interesting graphs and more. *Discrete Applied Mathematics.* **325**, 97-107, https://doi.org/10.1016/j.dam.2022.10.013 (2023).

Acknowledgements

This work has been supported by the National Natural Science Foundation of China (No. 62272480 and 62472444).

Author contributions

Data collection, Data processing, Implementation and Writing: Ying Zhao, Xianzhe Zou; Data collection, Data processing, Reviewing the manuscript, Data analysis: Xiao Wang, Zhanpeng Yang; Data collection, Data processing: Xuan Wang, Xin Zhao, Ning Zhang, Xin Huang; Reviewing the manuscript, Data analysis: Fangfang Zhou.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41597-025-05077-7.

Correspondence and requests for materials should be addressed to F.Z.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.