GraphLand: Evaluating Graph Machine Learning Models on Diverse Industrial Data

Gleb Bazhenov*
HSE University, Yandex Research
gv-bazhenov@yandex-team.ru

Oleg Platonov*
HSE Universoty, Yandex Research
olegplatonov@yandex-team.ru

Liudmila Prokhorenkova
Yandex Research
ostroumova-la@yandex-team.ru

Abstract

Although data that can be naturally represented as graphs is widespread in real-world applications across diverse industries, popular graph ML benchmarks for node property prediction only cover a surprisingly narrow set of data domains, and graph neural networks (GNNs) are often evaluated on just a few academic citation networks. This issue is particularly pressing in light of the recent growing interest in designing graph foundation models. These models are supposed to be able to transfer to diverse graph datasets from different domains, and yet the proposed graph foundation models are often evaluated on a very limited set of datasets from narrow applications. To alleviate this issue, we introduce GraphLand: a benchmark of 14 diverse graph datasets for node property prediction from a range of different industrial applications. GraphLand allows evaluating graph ML models on a wide range of graphs with diverse sizes, structural characteristics, and feature sets, all in a unified setting. Further, GraphLand allows investigating such previously underexplored research questions as how realistic temporal distributional shifts under transductive and inductive settings influence graph ML model performance. We evaluate a range of GNNs on GraphLand datasets and show that they significantly outperform graph-agnostic models in realistic settings. We also evaluate currently available general-purpose graph foundation models and find that they fail to produce competitive results on our proposed datasets.

1 Introduction

Recently, there has been a significant push for data-centric approaches in machine learning. In particular, high-quality, realistic, reliable, and diverse benchmarks are paramount for proper evaluation of the performance of machine learning methods. In the field of graph machine learning (GML), there has recently been a lot of criticism of existing popular benchmark datasets concerning such aspects as lacking practical relevance [1], low structural diversity that leaves most of the possible graph structure space not represented [2, 3], low application domain diversity [1], graph structure not being beneficial for the considered tasks [1, 4–6], potential bugs in the data collection processes leading to incorrect labels [7] and duplicated graph nodes [8]. While there have recently been efforts to create more realistic graph benchmarks, they focus on more specific domains (e.g., 3D molecular data) that require specialized models. At the same time, benchmarks for the standard and most widespread GML setting of node property prediction in a single large graph have received considerably less attention, and evaluation of the performance of classic graph neural networks and recent graph foundation models

*Equal contribution.

Bazhenov et al., GraphLand: Evaluating Graph Machine Learning Models on Diverse Industrial Data (Extended Abstract). Presented at the Fourth Learning on Graphs Conference (LoG 2025), Hybrid Event, December 10–12, 2025.

is still often limited to a few academic citation networks despite this setting and models developed for it having vast real-world applications in diverse industries.

We believe the historical focus of GNN evaluation on academic citation networks, which represent only a single (and a rather narrow) application domain, is primarily a consequence of the availability of open data of this type, rather than its relevance to real-world applications. At the same time, some of the most classical and simultaneously practically important examples of real-world graphs — social networks, web graphs, and road networks — are surprisingly rarely used for GNN evaluation, perhaps due to the lack of easily accessible high-quality datasets.

In our work, we aim to alleviate the issue of a lack of realistic GNN benchmarks for node property prediction by introducing GraphLand: a collection of graphs and associated machine learning tasks collected from a variety of industrial applications that represent real-world GML usage. GraphLand significantly extends the set of available datasets for GML model evaluation, providing in a unified format 14 graph datasets, many of which represent applications or structural properties that have not been covered by standard GML benchmarks before. The datasets in GraphLand have been collected both from open data that has been underutilized or not utilized at all in the field of GML, and from newly released data from services of a large technological company for which the use of GML has internally proven its usefulness. A key feature of GraphLand is its diversity, with graphs spanning a wide range of domains, sizes, and structural properties, and having rich node features with different types, meanings, and distributions.

For datasets in GraphLand, we provide several data splits, including a realistic temporal one, which allows for investigating practically important questions previously underexplored in GML literature: how temporal distributional shifts affect the performance of GML models in both transductive and inductive settings.

We run extensive experiments on GraphLand datasets with a range of GNNs and several openly available GFMs. We find that GNNs can achieve great results in industrial applications with attention-based GNNs often performing better than more classic ones. However, their performance can be strongly affected by temporal distributional shifts and dynamically evolving graph structure, which highlights the importance of developing models more resilient to such changes. Further, we find that currently available GFMs perform poorly on our datasets and fail to achieve results competitive with more classic methods. For a more detailed discussion of experimental results and additional comparison with GBDT models supplied with simple neighborhood feature aggregation, see the extended version of this paper [9].

We hope that GraphLand will allow more diverse and realistic evaluation of GML models, as well as encourage research into currently underexplored directions such as designing GML models that are more resilient to temporal distributional shifts and dynamically evolving graphs, and designing GFMs that are truly generalizable to graph data from different domains with different node feature sets.

2 Text-attributed graphs and generalization of graph foundation models

Most of the datasets frequently used for node property prediction only have textual descriptions as node attributes. However, graphs representing real-world networks often have rich and diverse node attributes that go beyond just texts and encompass a variety of numerical and categorical features with different meanings and distributions. There has recently been a lot of interest in developing general-purpose GFMs that are expected to generalize to graphs from different domains [10, 11]. A key challenge for such GFMs is being able to adapt to graphs with different node feature sets which is required for a model truly generalizable to different domains. Yet, the GFMs that have been proposed in the current literature typically overlook this challenge and are often only evaluated on text-attributed graphs [12–14]. Textual attributes can be easily projected to a common latent feature space by applying pretrained text encoders based on Large Language Models, thus allowing a single GFM to work with different text-attributed graphs. The prevalence of such text-attributed graphs in GML benchmarks has led to most of current GFM research overlooking the problem of generalization to different node feature sets, since it is not required to solve tasks from standard benchmarks. However, this problem is very important for real-world industrial applications of GML in which graphs often come attributed with a mixture of numerical and categorical node features. GFMs must therefore be able to work with such features to effectively solve practical tasks. The problem of devising a single foundation model that can work with arbitrary numerical and categorical features

Table 1: Characteristics of the proposed GraphLand datasets.

	node classification							node regression						
	hm-categories	pokec-regions	web-topics	tolokers-2	city-reviews	artnet-exp	web-fraud	hm-prices	avazu-ctr	city-roads-M	city-roads-L	twitch-views	artnet-views	web-traffic
# nodes	46.5K	1.6M	2.9M	11.8K	148.8K	50.4K	2.9M	46.5K	76.3K	57.1K	142.3K	168.1K	50.4K	2.9M
# edges	10.7M	22.3M	12.4M	519.0K	1.2M	280.3K	12.4M	10.7M	11.0M	107.1K	231.6K	6.8M	280.3K	12.4M
avg degree	460.92	27.32	8.56	88.28	15.66	11.12	8.56	460.92	288.04	3.75	3.26	80.87	11.12	8.56
median degree	45	13	2	30	4	2	2	45	71	4	3	32	2	2
avg distance	2.45	4.68	3.08	2.79	4.91	4.42	3.08	2.45	3.55	126.75	194.05	2.88	4.42	3.08
diameter	13	14	36	11	19	13	36	13	14	383	553	8	13	36
global clustering	0.27	0.05	0.00	0.23	0.26	0.03	0.00	0.27	0.24	0.00	0.00	0.02	0.03	0.00
avg local clustering	0.70	0.11	0.33	0.53	0.41	0.08	0.33	0.70	0.85	0.00	0.00	0.16	0.08	0.33
degree assortativity	−0.35	0.00	−0.14	−0.08	0.01	0.03	−0.14	−0.35	−0.30	0.70	0.74	−0.09	0.03	−0.14
# classes	21	183	28	2	2	2	2	N/A	N/A	N/A	N/A	N/A	N/A	N/A
unbiased homophily	0.38	0.98	0.55	0.10	0.69	0.28	0.32	N/A	N/A	N/A	N/A	N/A	N/A	N/A
target assortativity	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.12	0.18	0.74	0.72	−0.41	0.19	−0.21
# node features	35	56	263	16	37	75	266	41	260	26	26	4	50	267

has received significant attention from the ML for tabular data community, where such features are standard, and first successful attempts to develop such a model have recently emerged [15–21]. However, these ideas have not yet spread to the GML community (likely because current standard GML benchmarks do not require working with non-textual node features) despite the significant benefits of designing a successful GFM that can handle arbitrary node feature sets for practical applications.

3 GraphLand: a collection of diverse industrial graph datasets

GraphLand is a collection of 14 graph datasets with node property prediction tasks (either classification or regression). Some of these datasets are newly released for this benchmark, while others are collected from open data sources that are underutilized or not utilized at all in current GML benchmarking. In selecting datasets for GraphLand, we aim to fulfill the following desiderata: datasets should come from diverse fields representing impactful industrial applications of GML, graphs should exhibit a range of different sizes and structural characteristics, nodes should have rich features, graph structure should be beneficial for the considered tasks.

Here we briefly describe our datasets, while detailed information is provided in Appendix A.1. First, we describe our newly released datasets. `web-fraud`, `web-topics`, and `web-traffic` represent a part of the Internet (web-graph). `artnet-views` and `artnet-exp` represent a social network of art creators. `city-roads-M` and `city-roads-L` represent road networks of two major cities. `city-reviews` represents a network of users of a service for reviewing places and organizations. Further, we describe datasets obtained from open sources that were previously underutilized. `hm-categories` and `hm-prices` represent co-purchasing networks. `avazu-ctr` represents a network of devices from which advertisements can be viewed. `pokec-regions` represents a large social networks. `twitch-views` represents a network of content streamers. `tolokers-2` represents a network of crowdsourcing platform workers. Overall, our datasets cover a diverse range of industrial applications. In Table 1, we provide some characteristics of the datasets. As the table shows, our datasets also exhibit very diverse graph structural properties. A more detailed discussion of the characteristics from Table 1 is provided in Appendix A.2.

For each dataset, we prepare several different train/val/test data splits. Random-low (`RL`) and random-high (`RH`) are random splits with low and high labeled rates (10%/10%/80% and 50%/25%/25% train/val/test proportions, respectively). Temporal-high (`TH`) is a temporal split with high label rate. Additionally, temporal-high/inductive (`THI`) setting allows evaluation models under an inductive settings by presenting different snapshots of dynamically evolving networks for train, val, and test parts of `TH` split. More details on our data splits and experimental settings are provided in Appendix B.

4 Experiments

In Table 2 we present our experimental results for the `RL` split. It can be seen that classic GNNs, such as GCN [22] and GraphSAGE [23], demonstrate strong results significantly outperforming

Table 2: Experimental results under the RL (random low) data split in the transductive setting. The best result and those statistically indistinguishable from it are highlighted in orange. TLE stands for time limit exceeded (24 hours); MLE stands for memory limit exceeded (80 GB VRAM); RTE stands for runtime error in the official code of GFMs.

(a) Results for classification datasets.

	multiclass classification (accuracy is reported)			binary classification (average precision is reported)			
	hm-categories	pokec-regions	web-topics	tolokers-2	city-reviews	artnet-exp	web-fraud
best const. pred.	19.46 ± 0.00	3.77 ± 0.00	28.36 ± 0.00	21.82 ± 0.00	12.09 ± 0.00	10.00 ± 0.00	0.66 ± 0.00
ResMLP	37.72 ± 0.18	4.88 ± 0.01	42.41 ± 0.02	41.16 ± 1.13	71.32 ± 0.11	35.07 ± 2.34	8.77 ± 0.18
GCN	61.70 ± 0.35	34.96 ± 0.38	46.45 ± 0.10	51.32 ± 0.96	77.15 ± 0.28	43.09 ± 0.38	10.02 ± 0.18
GraphSAGE	56.75 ± 0.53	37.88 ± 0.41	47.41 ± 0.13	53.73 ± 0.53	77.82 ± 0.13	42.65 ± 0.59	12.11 ± 0.23
GAT	67.96 ± 0.33	46.17 ± 0.32	48.25 ± 0.05	53.78 ± 1.34	77.67 ± 0.13	46.62 ± 0.32	13.32 ± 0.29
GT	69.23 ± 0.50	46.47 ± 0.16	48.00 ± 0.05	54.50 ± 1.20	76.97 ± 0.21	45.16 ± 0.46	12.74 ± 0.42
OpenGraph (ICL)	9.49 ± 0.93	1.73 ± 0.31	RTE	40.49 ± 0.31	58.44 ± 1.08	15.65 ± 1.23	RTE
AnyGraph (ICL)	15.47 ± 2.36	24.65 ± 1.51	6.67 ± 3.88	31.33 ± 2.89	64.37 ± 1.29	13.14 ± 1.15	0.68 ± 0.03
TS-GNN (ICL)	20.09 ± 1.29	MLE	MLE	38.54 ± 0.94	43.46 ± 5.17	20.44 ± 1.05	MLE
GCOPE (FT)	19.51 ± 0.07	TLE	TLE	28.67 ± 1.42	67.38 ± 1.23	16.10 ± 2.79	TLE

(b) Results for regression datasets (R^2 is reported).

	hm-prices	avazu-ctr	city-roads-M	city-roads-L	twitch-views	artnet-views	web-traffic
best const. pred.	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
ResMLP	62.66 ± 0.37	24.54 ± 0.36	54.77 ± 0.15	46.47 ± 0.29	13.35 ± 0.02	29.71 ± 0.60	72.42 ± 0.05
GCN	69.76 ± 0.38	30.47 ± 0.27	59.05 ± 0.16	53.26 ± 0.14	75.55 ± 0.05	55.99 ± 0.26	82.07 ± 0.14
GraphSAGE	70.54 ± 0.21	31.84 ± 0.24	57.51 ± 0.53	52.43 ± 0.25	66.87 ± 0.11	49.79 ± 0.51	83.50 ± 0.11
GAT	73.17 ± 0.50	33.20 ± 0.20	59.11 ± 0.20	53.43 ± 0.20	72.93 ± 0.17	53.36 ± 0.78	84.68 ± 0.06
GT	71.87 ± 0.65	30.87 ± 0.47	58.05 ± 0.58	53.38 ± 0.12	72.19 ± 0.14	54.23 ± 0.22	84.49 ± 0.07

graph-agnostic model ResMLP and demonstrating the potential of GNNs in industrial applications. Attention-augmented GNNs (GAT [24] and neighborhood-attention Graph Transformer (GT) [25]) also produce strong results and often outperform simpler GNNs, showing the importance of being able to assign weights to messages from graph neighbors depending on their content in realistic industrial datasets. Further, we evaluate currently openly available GFMs. Despite a lot of works proposing different GFMs, we find only four open models that can perform node classification in graphs with arbitrary node features (OpenGraph [26], AnyGraph [27], TS-GNN [28], GCOPE [29]) and none that can perform node regression. In our experiments, these models produce poor results on all datasets (following the setups in the original papers, the former three GFMs are evaluated in in-context learning (ICL) regime, while the latter GFM is finetuned (FT) for the downstream task). This shows that currently available GFMs are still far from being able to compete with classic GNNs on realistic datasets with rich node attributes.

We present the results for the RH, TH, and THI settings combined in a single table in Table 3 (individual tables for each setting are provided in Appendix D). First, it can be seen that the observations from the RL setting about benefits of attention-based GNNs and weak performance of GFMs also apply to these settings. Further, these three settings come with the same data split proportions, allowing us to compare model performance under the random split, the realistic temporal split, and the inductive setting. We observe that temporal data splits are significantly more challenging for all models than random ones (with the exception of the avazu-ctr dataset). This is important as in real-world applications temporal distributional shifts are common, and not considering them can provide overly optimistic performance estimates. Further, the considered models perform significantly worse in the inductive setting than in the transductive setting. These observations highlight the importance of developing GML methods that are more resilient to temporal distributional shifts and dynamic changes in the graph structure for industrial applications. In the absence of such methods, frequent retraining of GNNs on new data is recommended to achieve the best results. Note that GFMs that only utilize in-context learning to adapt to new graphs do not suffer from the transductive/inductive mismatch and thus represent a promising direction, but their performance is currently very weak compared with GNNs on all datasets.

Overall, our main findings are as follows:

- GNNs can provide substantial benefits in industrial applications where graph information is available, with attention-based GNNs showing particularly strong results.

Table 3: Experimental results under the RH (random high), TH (temporal high), and THI (temporal high / inductive) settings. The best result and those statistically indistinguishable from it are highlighted in red for RH, violet for TH, and blue for THI.

(a) Results for classification datasets.

		multiclass classification (accuracy is reported)			binary classification (average precision is reported)		
		hm-categories	pokec-regions	web-topics	tolokers-2	artnet-exp	web-fraud
best const. pred.	RH	19.46 ± 0.00	3.77 ± 0.00	28.36 ± 0.00	21.82 ± 0.00	10.00 ± 0.00	0.66 ± 0.00
	TH	19.57 ± 0.00	2.98 ± 0.00	23.28 ± 0.00	8.61 ± 0.00	7.84 ± 0.00	0.15 ± 0.00
	THI	19.57 ± 0.00	2.98 ± 0.00	23.28 ± 0.00	8.61 ± 0.00	7.84 ± 0.00	0.15 ± 0.00
ResMLP	RH	43.12 ± 0.25	5.09 ± 0.01	44.55 ± 0.08	45.96 ± 0.46	43.55 ± 0.23	13.52 ± 0.21
	TH	32.44 ± 0.54	4.18 ± 0.01	35.49 ± 0.03	21.72 ± 6.69	37.48 ± 0.51	2.83 ± 0.26
	THI	32.44 ± 0.54	4.18 ± 0.01	35.49 ± 0.03	21.72 ± 6.69	37.48 ± 0.51	2.83 ± 0.26
GCN	RH	73.38 ± 0.42	35.08 ± 0.62	48.88 ± 0.09	60.49 ± 0.86	49.80 ± 0.35	15.58 ± 0.20
	TH	56.91 ± 0.55	11.88 ± 0.31	38.20 ± 0.19	46.72 ± 1.19	40.64 ± 0.40	4.85 ± 1.42
	THI	47.05 ± 1.69	6.88 ± 0.51	37.76 ± 0.06	32.43 ± 8.03	41.28 ± 0.28	3.44 ± 0.33
GraphSAGE	RH	73.34 ± 0.68	40.76 ± 0.21	50.05 ± 0.03	58.42 ± 0.92	48.49 ± 0.37	20.47 ± 0.17
	TH	59.62 ± 0.51	16.60 ± 0.28	39.00 ± 0.09	17.05 ± 7.65	40.50 ± 0.84	16.01 ± 2.22
	THI	48.11 ± 2.08	8.04 ± 0.26	38.04 ± 0.18	30.86 ± 9.48	40.53 ± 0.40	13.88 ± 1.32
GAT	RH	79.19 ± 0.21	46.72 ± 0.69	50.54 ± 0.04	63.76 ± 1.30	50.62 ± 0.35	20.43 ± 0.21
	TH	61.28 ± 0.97	20.43 ± 0.55	39.24 ± 0.23	38.59 ± 6.19	41.85 ± 0.63	16.50 ± 1.14
	THI	59.34 ± 1.09	13.38 ± 0.35	38.77 ± 0.35	24.53 ± 9.55	41.64 ± 0.32	11.98 ± 1.54
GT	RH	79.28 ± 0.31	50.06 ± 0.53	50.58 ± 0.04	60.32 ± 1.21	49.32 ± 1.00	19.73 ± 0.34
	TH	63.31 ± 0.45	25.09 ± 0.58	39.19 ± 0.15	34.15 ± 4.81	40.10 ± 0.60	11.97 ± 1.13
	THI	59.54 ± 1.59	17.22 ± 0.42	38.78 ± 0.08	22.89 ± 10.4	40.26 ± 0.82	7.84 ± 2.35
OpenGraph (ICL)	RH	11.69 ± 0.84	2.56 ± 0.42	RTE	44.62 ± 1.35	23.72 ± 1.86	RTE
	TH	5.76 ± 1.03	0.80 ± 0.45	RTE	9.12 ± 1.74	16.19 ± 1.36	RTE
	THI	5.76 ± 1.03	0.80 ± 0.45	RTE	9.12 ± 1.74	16.19 ± 1.36	RTE
AnyGraph (ICL)	RH	15.65 ± 2.82	27.67 ± 2.48	6.30 ± 2.82	30.21 ± 3.32	15.80 ± 1.90	0.67 ± 0.02
	TH	9.47 ± 1.13	9.20 ± 0.67	11.14 ± 5.16	13.52 ± 4.74	11.80 ± 1.01	0.16 ± 0.01
	THI	9.47 ± 1.13	9.20 ± 0.67	11.14 ± 5.16	13.52 ± 4.74	11.80 ± 1.01	0.16 ± 0.01
TS-GNN (ICL)	RH	15.48 ± 1.93	MLE	MLE	31.83 ± 2.55	13.38 ± 2.59	MLE
	TH	18.91 ± 0.32	MLE	MLE	12.59 ± 1.97	9.46 ± 1.25	MLE
	THI	18.91 ± 0.32	MLE	MLE	12.59 ± 1.97	9.46 ± 1.25	MLE
GCOPE (FT)	RH	19.99 ± 0.12	TLE	TLE	31.79 ± 1.95	23.86 ± 2.33	TLE
	TH	19.14 ± 0.58	TLE	TLE	8.46 ± 1.15	20.83 ± 1.18	TLE
	THI	15.69 ± 3.44	TLE	TLE	10.73 ± 1.48	19.10 ± 0.96	TLE

(b) Results for regression datasets (R^2 is reported).

		hm-prices	avazu-ctr	twitch-views	artnet-views
best const. pred.	RH	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
	TH	-2.85 ± 0.00	0.00 ± 0.00	-22.31 ± 0.00	-9.32 ± 0.00
	THI	-2.85 ± 0.00	0.00 ± 0.00	-22.31 ± 0.00	-9.32 ± 0.00
ResMLP	RH	70.11 ± 0.48	28.03 ± 0.22	13.36 ± 0.01	36.10 ± 0.17
	TH	61.64 ± 0.79	20.35 ± 1.50	11.91 ± 8.00	42.15 ± 0.52
	THI	61.64 ± 0.79	20.35 ± 1.50	11.91 ± 8.00	42.15 ± 0.52
GCN	RH	79.76 ± 0.76	34.96 ± 0.11	77.12 ± 0.11	61.02 ± 0.13
	TH	65.20 ± 0.84	37.49 ± 0.26	68.17 ± 0.24	54.44 ± 0.43
	THI	64.31 ± 0.82	34.78 ± 0.48	63.58 ± 0.54	53.73 ± 0.47
GraphSAGE	RH	79.89 ± 0.46	35.20 ± 0.20	72.02 ± 0.16	56.65 ± 0.50
	TH	67.93 ± 1.24	38.38 ± 0.39	61.46 ± 0.68	51.87 ± 0.48
	THI	65.80 ± 0.56	36.79 ± 0.55	56.60 ± 0.71	53.37 ± 0.30
GAT	RH	81.68 ± 0.41	35.74 ± 0.19	76.06 ± 0.30	59.01 ± 0.52
	TH	70.83 ± 0.99	39.21 ± 0.17	66.32 ± 0.59	52.30 ± 0.34
	THI	69.74 ± 1.50	37.18 ± 1.02	61.41 ± 1.30	52.44 ± 0.59
GT	RH	80.90 ± 0.42	34.38 ± 0.31	75.57 ± 0.15	58.97 ± 0.25
	TH	69.70 ± 0.84	38.27 ± 0.27	65.83 ± 0.24	51.67 ± 0.54
	THI	67.33 ± 2.05	36.49 ± 0.83	60.67 ± 1.02	52.26 ± 0.48

- GML methods can be very strongly affected by temporal distributional shifts and dynamically evolving graphs, so the design of GML methods capable of better handling such settings is an important research direction.
- The currently available general-purpose graph foundation models achieve very weak results on our datasets. Thus, the problem of creating truly general GFMs is far from being solved yet. We hope our benchmark can serve as a reliable testbed for future research in this direction, and first such successful use cases have already appeared [30, 31].

We hope GraphLand will encourage the evaluation of GML methods under more realistic and diverse settings, the development of GML methods that are more resilient to temporal distributional shifts and dynamically changing graph structure, and the development of better performing GFMs that can handle different node feature sets that go beyond just textual descriptions.

Acknowledgments

We thank Islam Umarov for providing the data for `city-reviews` dataset; Daniil Fedulov and Daniil Bondarenko for collecting the data for `artnet-views` and `artnet-exp` datasets; Dmitry Bikulov and Alexander Ulyanov for sharing the data for `web-traffic`, `web-fraud`, and `web-topics` datasets; Alexandr Ruchkin and Aleksei Istomin for collecting the data for `city-roads-M` and `city-roads-L` datasets; Fedor Velikonivtsev for helping with data processing for `city-roads-M` and `city-roads-L` datasets. We also thank Dmitry Eremeev for useful discussions.

References

[1] Maya Bechler-Speicher, Ben Finkelshtein, Fabrizio Frasca, Luis Müller, Jan Tönshoff, Antoine Siraudin, Viktor Zaverkin, Michael M. Bronstein, Mathias Niepert, Bryan Perozzi, Mikhail Galkin, and Christopher Morris. Position: Graph learning will lose relevance due to poor benchmarks. In *International Conference on Machine Learning*, 2025. 1

[2] John Palowitch, Anton Tsitsulin, Brandon Mayer, and Bryan Perozzi. GraphWorld: Fake graphs bring real insights for GNNs. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3691–3701, 2022. 1

[3] Seiji Maekawa, Koki Noda, Yuya Sasaki, and Makoto Onizuka. Beyond real-world benchmark datasets: An empirical study of node classification with GNNs. *Advances in Neural Information Processing Systems*, 35:5562–5574, 2022. 1

[4] Federico Errica, Marco Podda, Davide Bacciu, and Alessio Micheli. A fair comparison of graph neural networks for graph classification. In *International Conference on Learning Representations (ICLR)*, 2020. 1

[5] Zhengdao Li, Yong Cao, Kefan Shuai, Yiming Miao, and Kai Hwang. Rethinking the effectiveness of graph classification datasets in benchmarks for assessing GNNs. *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, 2024.

[6] Corinna Coupette, Jeremy Wayland, Emily Simons, and Bastian Rieck. No metric to rule them all: Toward principled evaluations of graph-learning datasets. In *International Conference on Machine Learning*, 2025. 1

[7] Yuwen Li, Miao Xiong, and Bryan Hooi. GraphCleaner: Detecting mislabelled samples in popular graph learning benchmarks. In *International Conference on Machine Learning*, pages 20195–20209, 2023. 1

[8] Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. A critical look at the evaluation of GNNs under heterophily: Are we really making progress? In *International Conference on Learning Representations (ICLR)*, 2023. 1, 10, 12, 13

[9] Gleb Bazhenov, Oleg Platonov, and Liudmila Prokhorenkova. GraphLand: Evaluating graph machine learning models on diverse industrial data. *Advances in Neural Information Processing Systems*, 2025. 2

[10] Yuxiang Wang, Wenqi Fan, Suhang Wang, and Yao Ma. Towards graph foundation models: A transferability perspective. *arXiv preprint arXiv:2503.09363*, 2025. 2

[11] Haitao Mao, Zhikai Chen, Wenzhuo Tang, Jianan Zhao, Yao Ma, Tong Zhao, Neil Shah, Mikhail Galkin, and Jiliang Tang. Position: Graph foundation models are already here. In *International Conference on Machine Learning*, 2024. 2

[12] Hao Liu, Jiarui Feng, Lecheng Kong, Ningyue Liang, Dacheng Tao, Yixin Chen, and Muhan Zhang. One for all: Towards training one graph model for all classification tasks. In *International Conference on Learning Representations (ICLR)*, 2024. 2

[13] Yuhan Li, Peisong Wang, Zhixun Li, Jeffrey Xu Yu, and Jia Li. ZeroG: Investigating cross-dataset zero-shot transferability in graphs. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1725–1735, 2024.

[14] Yufei He, Yuan Sui, Xiaoxin He, and Bryan Hooi. UniGraph: Learning a unified cross-domain foundation model for text-attributed graphs. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pages 448–459, 2025. 2

[15] Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A transformer that solves small tabular classification problems in a second. In *International Conference on Learning Representations (ICLR)*, 2023. 3

[16] Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025.

[17] Han-Jia Ye, Qile Zhou, Huai-Hong Yin, De-Chuan Zhan, and Wei-Lun Chao. Rethinking pre-training in tabular data: A neighborhood embedding perspective. *arXiv preprint arXiv:2311.00055*, 2023.

[18] Han-Jia Ye, Si-Yang Liu, and Wei-Lun Chao. A closer look at TabPFN v2: Understanding its strengths and extending its capabilities. *arXiv preprint arXiv:2502.17361*, 2025.

[19] Andreas C Mueller, Carlo A Curino, and Raghu Ramakrishnan. MotherNet: Fast training and inference via hyper-network transformers. In *International Conference on Learning Representations (ICLR)*, 2025.

[20] Junwei Ma, Valentin Thomas, Rasa Hosseinzadeh, Hamidreza Kamkari, Alex Labach, Jesse C Cresswell, Keyvan Golestan, Guangwei Yu, Maksims Volkovs, and Anthony L Caterini. TabDPT: Scaling tabular foundation models on real data. *arXiv preprint arXiv:2410.18164*, 2024.

[21] Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICL: A tabular foundation model for in-context learning on large data. In *International Conference on Machine Learning*, 2025. 3

[22] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017. 3, 13

[23] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in Neural Information Processing Systems*, 30, 2017. 3, 13

[24] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations (ICLR)*, 2018. 4, 13

[25] Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjin Wang, and Yu Sun. Masked label prediction: Unified message passing model for semi-supervised classification. *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, 2021. 4, 13

[26] Lianghao Xia, Ben Kao, and Chao Huang. OpenGraph: Towards open graph foundation models. *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024. 4, 13

[27] Lianghao Xia and Chao Huang. AnyGraph: Graph foundation model in the wild. *arXiv preprint arXiv:2408.10700*, 2024. 4, 13

[28] Ben Finkelshtein, İsmail İlkan Ceylan, Michael Bronstein, and Ron Levie. Equivariance everywhere all at once: A recipe for graph foundation models. *arXiv preprint arXiv:2506.14291*, 2025. 4

[29] Haihong Zhao, Aochuan Chen, Xiangguo Sun, Hong Cheng, and Jia Li. All in one and one for all: A simple yet effective method towards cross-domain graph pretraining. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4443–4454, 2024. 4, 14

[30] Dmitry Eremeev, Gleb Bazhenov, Oleg Platonov, Artem Babenko, and Liudmila Prokhorenkova. Turning tabular foundation models into graph foundation models. *arXiv preprint arXiv:2508.20906*, 2025. 5

[31] Dmitry Eremeev, Oleg Platonov, Gleb Bazhenov, Artem Babenko, and Liudmila Prokhorenkova. GraphPFN: A prior-data fitted graph foundation model. *arXiv preprint arXiv:2509.21489*, 2025. 5

[32] Steve Wang and Will Cukierski. Click-through rate prediction, 2014. URL `https://kaggle.com/competitions/avazu-ctr-prediction`. 10

[33] Sergei Ivanov and Liudmila Prokhorenkova. Boost then convolve: Gradient boosting meets graph neural networks. In *International Conference on Learning Representations (ICLR)*, 2021. 10

[34] Carlos García Ling et al. H&M personalized fashion recommendations, 2022. URL https://kaggle.com/competitions/h-and-m-personalized-fashion-recommendations. 10

[35] Lubos Takac and Michal Zabovsky. Data analysis in public social networks. In *International Scientific Conference and International Workshop Present Day Trends of Innovations*, 2012. 10

[36] Derek Lim, Felix Hohne, Xiuyu Li, Sijia Linda Huang, Vaishnavi Gupta, Omkar Bhalerao, and Ser Nam Lim. Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods. *Advances in Neural Information Processing Systems*, 34:20887–20902, 2021. 10, 11

[37] Benedek Rozemberczki and Rik Sarkar. Twitch gamers: a dataset for evaluating proximity preserving and structural role-based node embeddings, 2021. 10

[38] Daniil Likhobaba, Nikita Pavlichenko, and Dmitry Ustalov. Toloker graph: Interaction of crowd annotators, 2023. URL https://github.com/Toloka/TolokerGraph. 10

[39] Stefano Boccaletti, Ginestra Bianconi, Regino Criado, Charo I Del Genio, Jesús Gómez-Gardenes, Miguel Romance, Irene Sendina-Nadal, Zhen Wang, and Massimiliano Zanin. The structure and dynamics of multilayer networks. *Physics Reports*, 544(1):1–122, 2014. 11

[40] Oleg Platonov, Denis Kuznedelev, Artem Babenko, and Liudmila Prokhorenkova. Characterizing graph datasets for node classification: Homophily-heterophily dichotomy and beyond. *Advances in Neural Information Processing Systems*, 36:523–548, 2023. 11

[41] Mikhail Mironov and Liudmila Prokhorenkova. Revisiting graph homophily measures. *Learning on Graphs Conference*, 2024. 11, 12

[42] Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-GCN: Geometric graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2020. 12

[43] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *Advances in Neural Information Processing Systems*, 33:22118–22133, 2020. 12

[44] Shurui Gui, Xiner Li, Limei Wang, and Shuiwang Ji. GOOD: A graph out-of-distribution benchmark. *Advances in Neural Information Processing Systems*, 35:2059–2073, 2022. 12

[45] Gleb Bazhenov, Denis Kuznedelev, Andrey Malinin, Artem Babenko, and Liudmila Prokhorenkova. Evaluating robustness and uncertainty of graph models under structural distributional shifts. *Advances in Neural Information Processing Systems*, 36:75567–75594, 2023. 12

[46] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 13

[47] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 13

[48] Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. *Advances in Neural Information Processing Systems*, 34: 18932–18943, 2021. 13

[49] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International Conference on Machine Learning*, pages 1263–1272, 2017. 13

[50] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs). *arXiv preprint arXiv:1606.08415*, 2016. 13

[51] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *Advances in Neural Information Processing Systems*, 33:5812–5823, 2020. 14

[52] Jun Xia, Lirong Wu, Jintao Chen, Bozhen Hu, and Stan Z Li. SimGRACE: A simple framework for graph contrastive learning without data augmentation. In *Proceedings of the ACM Web Conference*, pages 1070–1079, 2022. 14

[53] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. 14

[54] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. 14

[55] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 2019. 14

[56] Minjie Wang, Da Zheng, Zihao Ye, Quan Gan, Mufei Li, Xiang Song, Jinjing Zhou, Chao Ma, Lingfan Yu, Yu Gai, Tianjun Xiao, Tong He, George Karypis, Jinyang Li, and Zheng Zhang. Deep graph library: A graph-centric, highly-performant package for graph neural networks. *arXiv preprint arXiv:1909.01315*, 2019. 14

A GraphLand benchmark details

A.1 Dataset descriptions

In this section, we provide a more detailed description of the GraphLand datasets. Note that none of the proposed datasets contain any personal information. For the datasets based on data newly collected for our benchmark, the public release of the data was approved by a legal team. GraphLand datasets are available at Zenodo and Kaggle. Our source code and instructions on how to reproduce our experimental results are available in our GitHub repository.

`web-fraud`, **`web-topics`**, and **`web-traffic`** These three datasets are web-graphs — they represent a segment of the Internet. The nodes are websites, and a directed edge connects two nodes if at least one user followed a link from one website to the other in a selected period of time. We prepared three datasets with the same graph but different tasks: in `web-fraud`, the task is to predict which websites are fraudulent (strongly imbalanced binary classification); in `web-topics`, the task is to predict the topic that a website belongs to (multiclass classification); and in `web-traffic`, the task is to predict how many users visited a website in a specific period of time (regression). With almost 3 million nodes, this is one of the largest publicly available attributed graphs that is not a citation network. Nodes in this graph have more than two hundred features, examples of which include the number of videos on the website (numerical feature), the website's zone, and whether the website is on a free hosting (categorical features). Data for these datasets was obtained from the Yandex search engine.

`artnet-views` and **`artnet-exp`** These two datasets represent a social network of art creators. The nodes are users, and an undirected edge connects two nodes if the users are friends. We prepared two datasets with the same graph but different tasks: in `artnet-views`, the task is to predict how many views a user receives in a specific period of time (regression); and in `artnet-exp`, the task is to predict which users create explicit art content (binary classification). Examples of node features include user interests (categorical features). Data for these datasets was obtained from the generative AI art creation platform Shedevrum.

`city-roads-M` and **`city-roads-L`** These datasets are obtained from the logs of a navigation service and represent the road networks of two major cities, with the second one being several times larger than the first. The nodes are segments of roads, and a directed edge connects two nodes if the segments are incident to each other and moving from one segment to the other is permitted by traffic rules. The task is to predict the average travel speed on a road segment at a specific timestamp (regression). The features include various information about the road segment such as binary indicators of whether there is a bike dismount sign, whether the road segment ends with a crosswalk or a toll post, whether the road segment is in poor condition, whether it is restricted for trucks, and whether it has a mass transit lane (categorical features). Numerical features include the length of the road segment and the geographic coordinates of the road endpoints. Data for these datasets was obtained from the Yandex Maps service.

`city-reviews` This dataset is obtained from the logs of a review service in which users can leave reviews and ratings for places and organizations in two major cities. The nodes are users, and an undirected edge connects two nodes if the users often leave reviews for the same organizations. The

task is fraud detection — to predict which users leave fraudulent reviews (binary classification). The node features are based on user interactions with the service and their examples include the share of negative reviews among all reviews left by a user (numerical feature) and the browser that is used to access the service by the user (categorical feature). Data for this dataset was obtained from the Yandex Maps service.

avazu-ctr This dataset is based on open data that has been introduced at the Kaggle competition organized by Avazu [32]. The data contains information about interactions between devices used to access the Internet, websites, and advertisements. In our graph, the nodes are devices, and an edge connects two nodes if the devices often visit the same websites. The graph is undirected. A smaller version of a similar dataset has been used by Ivanov and Prokhorenkova [33]; however, it contained only a small subset of devices, while for our dataset we collected data for all the available devices, making our graph more than 50 times larger. The task is to predict the advertisement click-through rate (CTR) observed on devices (regression). Nodes in this graph have more than two hundred numerical features; however, most of them were anonymized in the original data source.

hm-categories and **hm-prices** These datasets are based on open data that has been introduced at the Kaggle competition organized by H&M [34]. The graph represents a co-purchasing network. The nodes are products, and an edge connects two nodes if the products are often bought by the same customers. The graph is undirected. We prepared two datasets with the same graph but different tasks: in `hm-categories`, the task is to predict the product category (multiclass classification), and in `hm-prices`, the task is to predict the product price (regression). The node features in this dataset include product metadata such as product color (categorical feature), as well as information obtained from product purchasing statistics such as what proportion of product purchases occurs on different weekdays (numerical features).

pokec-regions This dataset is based on the data from Takac and Zabovsky [35]. It represents the online social network Pokec. The nodes are users, and a directed edge connects two nodes if one user has marked the other one as a friend. While this graph is quite popular in network analysis as an example of a classic social network, it is relatively rarely used for machine learning, with the exception of Lim et al. [36] who use the same graph as we do but with different task, node features, and data split. In our dataset, the task is to predict which region a user is from (extreme multiclass classification with 183 classes). The node features in our dataset are based on user profile information, examples of them include the profile completion proportion (numerical feature) and binary indicators of whether different profile fields are filled (categorical features).

twitch-views This dataset is based on the data from Rozemberczki and Sarkar [37]. It represents the live-streaming network Twitch. The nodes are users, and an undirected edge connects two nodes if both users follow each other. The task is to predict how many views a user receives in a specific period of time (regression). The node features are based on user profile information and examples of them include user language and affiliate status (categorical features).

tolokers-2 This is a new version of the dataset `tolokers` from Platonov et al. [8], Likhobaba et al. [38] with a significantly extended set of node features. It is based on the data from the Toloka crowdsourcing platform and the graph represents a network of tolokers (workers). The nodes are tolokers, and an edge connects two nodes if these tolokers have worked on the same task. The graph is undirected. The task is fraud detection — to predict which tolokers have been banned in one of the projects (binary classification). The new node features include various performance statistics of workers, such as the number of approved assignments and the number of skipped assignments (numerical features), as well as workers' profile information, such as their education level (categorical feature).

Some graphs in our benchmark are undirected, and some are directed (see individual dataset descriptions above). All our undirected graphs consist of a single connected component, and all our directed graphs consist of a single weakly connected component.

For all datasets, we provide random stratified `RL` and `RH` data splits. Further, we provide temporal `TH` data split (with the possibility of using the inductive learning setting `THI`) for all datasets with the exception of `city-roads-M` and `city-roads-L` datasets (since well-established road network graphs typically do not evolve over time significantly), as well as `city-reviews` and `web-traffic` datasets (since for them some of the necessary temporal information was not available).

A.2 Dataset properties

A key characteristic of our benchmark is its diversity. As described above, our graphs come from different domains and have different prediction tasks. Their edges are also constructed in different ways (based on user interactions, activity similarity, physical connections, etc.). However, the proposed datasets also differ in many other ways. Some properties of our graphs are presented in Table 1 (see below for the details on how the provided characteristics are defined). First, note that the sizes of our datasets range from 11K to 3M nodes. The smaller graphs can be suitable for compute-intensive models, while the larger graphs can provide a moderate scaling challenge. The average and median degrees of our graphs also vary significantly and our benchmark has both sparse and relatively dense graphs, including graphs with the average degree in the order of hundreds which is larger than the average degrees of most datasets used in current GML research (such graphs may highlight the importance of attention-based GNNs with their soft edge selection mechanisms). The average distance between two nodes in our graphs varies from 2.45 for `hm-categories` and `hm-prices` to 194 for `city-roads-L`; and graph diameter (maximum distance) varies from 8 for `twitch-views` to 553 for `city-roads-L`. Further, we report the values of clustering coefficients, which show how typical closed node triplets are for the graph. In the literature, there are two definitions of clustering coefficients [39]: the global clustering coefficient and the average local clustering coefficient. We have both graphs where the clustering coefficients are high and graphs where they are almost zero, as well as graphs where global and local clustering coefficients significantly disagree (which is possible for graphs with imbalanced degree distributions). The degree assortativity coefficient is defined as the Pearson correlation coefficient of degrees among pairs of linked nodes. For most of our graphs, the degree assortativity is either negative or close to zero, which means that nodes do not tend to connect to other nodes with similar degrees, while `city-roads-M` and `city-roads-L` datasets are the exceptions — for them the degree assortativity is positive and large.

Further, let us discuss the graph-label relationships in our datasets. To measure the similarity of labels of connected nodes for regression datasets, we use target assortativity — the Pearson correlation coefficient of target values between pairs of connected nodes. For instance, for the `city-roads-M` and `city-roads-L` datasets, the target assortativity is positive and quite large, which shows that nodes tend to connect to other nodes with similar target values (which is expected for the task of speed prediction in road networks), while for the `twitch-views` and `web-traffic` datasets, the target assortativity is negative. For classification datasets, the similarity of neighbors' labels is usually called *homophily*: in homophilous datasets, nodes tend to connect to nodes of the same class. How to properly measure homophily has recently attracted some research. It has been noted by Lim et al. [36] and Platonov et al. [40] that homophily measures typically used in the literature — such as the proportion of edges connecting nodes of the same class — are not appropriate for comparing homophily levels between graphs with different numbers of classes and their size balance. Platonov et al. [40] proposed a set of properties that a homophily measure appropriate for use in such comparisons should satisfy and Mironov and Prokhorenkova [41] constructed the first known homophily measure that satisfies all these properties — *unbiased homophily*. Thus, in our work, we use unbiased homophily to measure the homophily levels of our datasets. Unbiased homophily (with $\alpha = 0$, see Mironov and Prokhorenkova [41] for more details) takes values in $[-1,\ 1]$ with 1 indicating perfect homophily, -1 indicating perfect heterophily, and 0 indicating no preference between homophilous and heterophilous edges (such graphs are typically referred to as heterophilous in the literature, although a more appropriate term would be non-homophilous). Note that the values of unbiased homophily should not be compared to values of other homophily measures used in the literature; the unbiased homophily levels for some popular graph node classification datasets are provided by Mironov and Prokhorenkova [41]. Unbiased homophily indicates that among our datasets `pokec-regions` and `city-reviews` are homophilous, while the remaining datasets are non-homophilous. Thus, our benchmark significantly expands the set of available non-homophilous graph datasets.

Finally, our datasets have diverse sets of node features consisting of numerical and categorical features with different meanings and distributions. All our datasets except `twitch-views` and `tolokers-2` have at least several dozen node features, while some have several hundred node features.

Overall, our datasets are diverse in domain, scale, structural properties, graph-label relations, and node attributes. Coming from real-world GML applications, they may serve as a valuable tool for the research and development of GML methods for the industry.

Computing dataset characteristics. Further, we describe the characteristics that are used in Table 1. Note that, while some graphs in our benchmark are directed, we transformed all the graphs to be undirected before computing all the considered graph characteristics, since some of the characteristics are not defined for directed graphs.

Average degree and *median degree* are the average and median numbers of neighbors a node has, respectively. Since all our graphs are connected (when treated as undirected graphs), for any two nodes there is a path between them. *Average distance* is the average length of the shortest paths between all pairs of nodes, while *diameter* is the maximum length of the shortest paths between all pairs of nodes. For our largest graphs — the ones used for the `pokec-regions`, `web-traffic`, `web-fraud`, and `web-topics` datasets — we approximate the average distance with the average over distances for 100K randomly sampled node pairs. *Global clustering* coefficient is computed as three times the number of triangles divided by the number of pairs of adjacent edges (i.e., it is the fraction of closed triplets of nodes among all connected triplets). *Average local clustering* coefficient first computes the local clustering of each node, which is the fraction of connected pairs of its neighbors, and then averages the obtained values among all nodes. *Degree assortativity* is the Pearson correlation coefficient between the degrees of connected nodes. Further, *target assortativity* for regression datasets is the Pearson correlation coefficient between target values of connected nodes. For computing *unbiased homophily*, we follow Mironov and Prokhorenkova [41] and use the simplest version of this measure with the α parameter set to 0.

B Data splits and experimental settings

In GML literature, there are two most popular settings for node property prediction regarding the relative sizes of train, validation, and test sets: one with a high label rate and one with a low label rate. In the high label rate setting, the train set encompasses 50% of all graph nodes or more. This setting is common in heterophilous benchmarks, e.g., it is used by datasets from Pei et al. [42] and Platonov et al. [8], and it is also used by most datasets from OGB [43]. In the low label rate setting, much smaller train set sizes are used (typically, no more than 10% of all graph nodes). This setting is commonly used with the classic `cora`, `citeseer`, `pubmed` citation networks, and it is also used by the `ogbn-products` dataset from OGB. Both of these settings can appear in real-world GML usage scenarios, depending on the resources available for data labeling. It is important to provide predetermined splits to ensure experiments in different works are run in the same setting and their results are comparable, but it is also important to accommodate different needs of different research projects. Thus, for datasets in our benchmark we provide fixed splits for both settings. We refer to these splits as the `RL` (random low) and `RH` (random high) splits. Specifically, the `RL` split randomly divides nodes into train/validation/test sets with $10\%/10\%/80\%$ proportions, while the `RH` split randomly divides nodes into train/validation/test sets with $50\%/25\%/25\%$ proportions.

The `RL` and `RH` data splits are random, as is common in current GML benchmarks. However, in real-world applications, data splits are often *temporal*, i.e., the labeled objects are the ones that appeared in the network earlier, while the ones that appeared later are not labeled and belong to the test set. Despite their prevalence in applications, temporal splits are very rarely used in current GML node property prediction benchmarks. To the best of our knowledge, the only such datasets with temporal splits available are citation networks from OGB, which represent only a single application. At the same time, temporal data splits may significantly affect the prediction problem and the model performance, as they often result in *distributional shifts* between train, validation, and test data. While some types of distributional shifts have been previously explored in the GML literature, e.g., shifts in node features [44] and shifts in graph structure [45], realistic temporal distributional shifts often combine shifts in several aspects of data simultaneously (e.g., shifts in the distributions of node features, labels, and graph structural characteristics), and the effect of such realistic shifts on GML model performance is currently under-explored. To close this gap, we provide a temporal split for most datasets in our benchmark. We refer to this split as the `TH` (temporal high) split; it divides nodes into train/validation/test sets with $50\%/25\%/25\%$ proportions, i.e., exactly the same proportions as in the `RH` split, which allows comparing model results between the `RH` and `TH` splits to see how the complexity of the task changes when temporal distributional shifts are introduced.

Further, many real-world networks are not static, but evolve over time. Thus, in many applications, not only are there no labels available for nodes that appear in the network later, but the nodes themselves (with their attributes and incident edges) are not available at training time. This setting is

known in GML as the *inductive* setting. In contrast to the *transductive* setting in which the whole graph is available at training time (including the nodes for which predictions should be made), in the inductive setting validation and test nodes are not available at training time. Despite temporally evolving graphs being common in practical applications, most standard node property prediction datasets only provide the transductive setting. While it is well-known that GNNs, in contrast to some other GML methods like shallow node embeddings, can work not only in the transductive but also in the inductive setting [23], it is not well-explored how the lack of complete graph information at training time in the inductive setting affects GNN performance. Moreover, when the inductive setting is used for GNN evaluation in the current literature, the validation and train nodes are typically chosen randomly, which is not realistic, since in real-world applications the inductive setting is almost always induced by the temporal evolution of the graph. To fill this gap, for all datasets in our benchmark for which temporal information is available, we additionally provide the inductive experimental setting. We refer to this setting as `THI` (temporal high / inductive); it has the exact same data split as the transductive `TH` setting, but provides three snapshots of the graph: one for training, one for validation, and one for testing. This allows investigating how model performance changes between the transductive and the inductive setting. To the best of our knowledge, our work is the first to compare GNN performance between random and temporal data splits in the transductive setting under the same split ratios, and between transductive and inductive settings under the same temporal data split. This comparison allows us to investigate how much these differences can affect GNN performance.

C Experimental setup details

C.1 Models

A graph-agnostic baseline. As a simple baseline, we use ResMLP — an MLP with skip-connections [46] and layer normalization [47]. This model does not have any information about the graph structure and operates on nodes as independent samples — we call such models *graph-agnostic*. It has been shown that such MLP-like models with skip-connections can serve as very strong baselines for industrial data with mixed numerical and categorical features [48].

Graph neural networks. We consider several representative GNN architectures. First, we use GCN [22] and GraphSAGE [23] as simple classical GNN models. For GraphSAGE, we use the version with the mean aggregation function, and we do not use the neighbor sampling technique proposed in the original paper, instead training the model on the full graph, like all other GNNs in our experiments. Further, we use two GNNs with attention-based neighborhood aggregation functions: GAT [24] and Graph Transformer (GT) [25]. Note that GT is a *local* graph transformer, i.e., each node only attends to its neighbors in the graph (in contrast to *global* graph transformers, in which each node attends to all other nodes in the graph, and which are thus not instances of the standard message-passing neural networks (MPNNs) framework of Gilmer et al. [49]). Following Platonov et al. [8], we equip all the considered GNNs with skip-connections and layer normalization, which we found important for their strong performance on our datasets. We also add a two-layer MLP with the GELU activation function [50] after every neighborhood aggregation block in GNNs. Our graph models are implemented in the same codebase as our ResMLP — we simply swap each residual block of ResMLP with a residual neighborhood aggregation block of the selected GNN architecture. Therefore, comparing the performance of ResMLP and GNNs allows us to see if graph information is helpful for the task. Indeed, in our experiments, GNNs significantly outperform graph-agnostic ResMLP on all our datasets, confirming the usefulness of the provided graph structure for the considered tasks.

Graph foundation models. Most currently available GFMs do not support node property prediction tasks in graphs with arbitrary node features. Of those that do, we were able to find only two models with open weights: OpenGraph [26] and AnyGraph [27]. Both OpenGraph and AnyGraph exploit the Transformer architecture and are pretrained with a link prediction objective on a mixture of different graph datasets. These methods differ in what data they can operate on. Specifically, OpenGraph only uses relational information and constructs node representations based on SVD factors of the adjacency matrix, while AnyGraph also uses the available node feature information and combines SVD factors for both the feature matrix and the adjacency matrix. Both these models were designed to be adapted to new node classification datasets without fine-tuning, using an in-context learning

(ICL) setting instead. Specifically, they can perform link prediction in arbitrary graphs, and they cast any node classification task as a link prediction task where links to virtual nodes representing classes are predicted for unlabeled nodes. Further, we were able to reproduce the pretraining for one more GFM — GCOPE [29] (weights for this model are not publicly available, but training code is). GCOPE also applies a projection to node features (e.g., based on SVD or attention mechanism), but exploits additional virtual nodes as graph coordinators to simultaneously process different graph datasets at the pretraining stage. The authors use GraphCL [51] or SimGRACE [52] as the pretraining objective. In contrast to OpenGraph and AnyGraph, GCOPE uses fine-tuning for adaptation to new node classification datasets. However, neither of the three models supports node regression. Further, we found that these GFMs cannot scale to large datasets.

C.2 Experimental setup and hyperparameter selection details

Some of the graphs in our benchmark are directed. For our experiments, we converted directed graphs to undirected ones (by replacing each directed edge with an undirected one and then removing duplicated edges). We leave the investigation of different ways to consider edge directions to further research.

We train all models 10 times with different random seeds to compute the mean and standard deviation of model performance, except for our largest datasets `pokec-regions`, `web-traffic`, `web-fraud`, `web-topics`, for which we train all models 5 times. We train all our GNNs in a full-batch setting, i.e., we do not use any subgraph sampling techniques and train the models on the full graph. Our ResMLP baseline is implemented in the same codebase as our GNNs and thus is also trained in the full-batch setting. All the experiments were run on an NVIDIA Tesla A100 80GB GPU.

Hyperparameter choice is extremely important for the performance of GNNs. Thus, we conducted hyperparameter search on the validation set for all models. Specifically, we found that the learning rate and dropout probability [53] are the most important hyperparameters for our GNN implementations on our datasets. Thus, we ran grid search selecting the learning rate from $\{3 \times 10^{-5}, 1 \times 10^{-4}, 3 \times 10^{-4}, 1 \times 10^{-3}, 3 \times 10^{-3}\}$ and dropout probability from $\{0, 0.1, 0.2\}$ (note that the highest learning rate of 3×10^{-3} often resulted in `NaN` issues; however, we still included it in our hyperparameter search, as in our preliminary experiments we found it to be beneficial for some of our dataset/model combinations). In our preliminary experiments we found that the performance of our GNNs is quite stable for a wide variety of reasonable architecture hyperparameter values (we found the use of skip-connections and layer normalization to be important for this stability). Hence, for our final experiments, we kept these hyperparameters fixed. We set these values as follows: the number of graph neighborhood aggregation blocks to 3 and the hidden dimension to 512. The only exceptions to this hidden dimension size were made for our largest datasets: to avoid GPU out-of-memory issues, we decreased the hidden dimension to 400 for `pokec-regions` and to 200 for `web-traffic`, `web-fraud`, and `web-topics`. For GNNs with attention-based graph neighborhood aggregation (GAT and GT), the number of attention heads was set to 4. We used the Adam optimizer [54] in all our GNN experiments. We trained each model for 1000 steps and then selected the best step based on the performance on the validation set.

When applying deep learning models to data with numerical features, the preprocessing of these features is critically important. In our experiments, we considered two possible numerical feature transformation techniques: standard scaling and quantile transformation to standard normal distribution. We included them in the hyperparameter search for ResMLP and GNNs. For categorical features, we used one-hot encoding for all models. For regression datasets, neural models might perform better if the target variable is transformed. Therefore, in our experiments on regression datasets with ResMLP and GNNs, we considered the options of using the original targets or preprocessing targets with standard scaling, including these two options in the hyperparameter search.

Our GNNs are implemented using PyTorch [55] and DGL [56].

D Additional experimental results

We provide experimental results for the `RH` setting in Table 4, for the `TH` setting in Table 5, and for the `THI` setting in Table 6. In all tables with results, for each dataset, we highlight with color the best result as well as those results for which the mean differs from the best one by no more than the sum of the two results' standard deviations.

Table 4: Experimental results under the RH (random high) data split in the transductive setting. The best result and those statistically indistinguishable from it are highlighted in red. TLE stands for time limit exceeded (24 hours); MLE stands for memory limit exceeded (80 GB VRAM); RTE stands for runtime error in the official code of GFMs.

(a) Results for classification datasets.

	multiclass classification (accuracy is reported)			binary classification (average precision is reported)			
	hm-categories	pokec-regions	web-topics	tolokers-2	city-reviews	artnet-exp	web-fraud
best const. pred.	19.46 ± 0.00	3.77 ± 0.00	28.36 ± 0.00	21.82 ± 0.00	12.09 ± 0.00	10.00 ± 0.00	0.66 ± 0.00
ResMLP	43.12 ± 0.25	5.09 ± 0.01	44.55 ± 0.08	45.96 ± 0.46	75.21 ± 0.08	43.55 ± 0.23	13.52 ± 0.21
GCN	73.38 ± 0.42	35.08 ± 0.62	48.88 ± 0.09	60.49 ± 0.86	81.05 ± 0.10	49.80 ± 0.35	15.58 ± 0.20
GraphSAGE	73.34 ± 0.68	40.76 ± 0.21	50.05 ± 0.03	58.42 ± 0.92	80.75 ± 0.06	48.49 ± 0.37	20.47 ± 0.17
GAT	79.19 ± 0.21	46.72 ± 0.69	50.54 ± 0.04	63.76 ± 1.30	81.10 ± 0.11	50.62 ± 0.35	20.43 ± 0.21
GT	79.28 ± 0.31	50.06 ± 0.53	50.58 ± 0.04	60.32 ± 1.21	80.50 ± 0.14	49.32 ± 1.00	19.73 ± 0.34
OpenGraph (ICL)	11.69 ± 0.84	2.56 ± 0.42	RTE	44.62 ± 1.35	62.96 ± 0.84	23.72 ± 1.86	RTE
AnyGraph (ICL)	15.65 ± 2.82	27.67 ± 2.48	6.30 ± 2.82	30.21 ± 3.32	65.04 ± 1.41	15.80 ± 1.90	0.67 ± 0.02
TS-GNN (ICL)	15.48 ± 1.93	MLE	MLE	31.83 ± 2.55	11.84 ± 1.98	13.38 ± 2.59	MLE
GCOPE (FT)	19.99 ± 0.12	TLE	TLE	31.79 ± 1.95	69.74 ± 0.36	23.86 ± 2.33	TLE

(b) Results for regression datasets (R^2 is reported).

	hm-prices	avazu-ctr	city-roads-M	city-roads-L	twitch-views	artnet-views	web-traffic
best const. pred.	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
ResMLP	70.11 ± 0.48	28.03 ± 0.22	62.43 ± 0.32	53.09 ± 0.17	13.36 ± 0.01	36.10 ± 0.17	73.88 ± 0.05
GCN	79.76 ± 0.76	34.96 ± 0.11	69.95 ± 0.11	64.65 ± 0.27	77.12 ± 0.11	61.02 ± 0.13	83.49 ± 0.14
GraphSAGE	79.89 ± 0.46	35.20 ± 0.20	70.20 ± 0.59	65.77 ± 0.43	72.02 ± 0.16	56.65 ± 0.50	85.19 ± 0.11
GAT	81.68 ± 0.41	35.74 ± 0.19	70.53 ± 0.40	66.03 ± 0.24	76.06 ± 0.30	59.01 ± 0.52	85.70 ± 0.08
GT	80.90 ± 0.42	34.38 ± 0.31	67.45 ± 0.82	64.02 ± 0.59	75.57 ± 0.15	58.97 ± 0.25	85.54 ± 0.23

E Limitations and broader impact

The aim of our benchmark is to introduce a diverse set of graph datasets for node property prediction that covers a wide range of domains and graph structural properties, including those not encountered in commonly used datasets for GML model evaluation. However, data that can be naturally represented as graphs is so widespread across different domains that no benchmark can cover them all. Thus, our collection of 14 datasets still only covers a small part of the wide range of situations where modeling data as a graph can be useful. But we hope that it will encourage the GML research community to use more diverse sets of datasets and focus on practically relevant applications where graph-structured data appears.

Our benchmark includes datasets with realistic tasks such as fraud detection and user engagement prediction. Poorly performing machine learning models used for these tasks in real-world services can negatively affect the users of these services. For example, type I errors of fraud detection systems, i.e., wrongly predicting that an innocent person is fraudulent, have an undesirable negative impact. Thus, particular care should be taken to minimize the probability of such errors. We believe that the release of high-quality and properly anonymized datasets for these tasks such as the ones in our benchmark will encourage the community to develop better models, since the community will be able to use these datasets as a realistic and reliable testbed to investigate which methods lead to reductions in undesirable model errors.

Table 5: Experimental results under the TH (temporal high) data split in the transductive setting. The best result and those statistically indistinguishable from it are highlighted in violet. TLE stands for time limit exceeded (24 hours); MLE stands for memory limit exceeded (80 GB VRAM); RTE stands for runtime error in the official code of GFMs.

(a) Results for classification datasets.

	multiclass classification (accuracy is reported)			binary classification (average precision is reported)		
	hm-categories	pokec-regions	web-topics	tolokers-2	artnet-exp	web-fraud
best const. pred.	19.57 ± 0.00	2.98 ± 0.00	23.28 ± 0.00	8.61 ± 0.00	7.84 ± 0.00	0.15 ± 0.00
ResMLP	32.44 ± 0.54	4.18 ± 0.01	35.49 ± 0.03	21.72 ± 6.69	37.48 ± 0.51	2.83 ± 0.26
GCN	56.91 ± 0.55	11.88 ± 0.31	38.20 ± 0.19	46.72 ± 1.19	40.64 ± 0.40	4.85 ± 1.42
GraphSAGE	59.62 ± 0.51	16.60 ± 0.28	39.00 ± 0.09	17.05 ± 7.65	40.50 ± 0.84	16.01 ± 2.22
GAT	61.28 ± 0.97	20.43 ± 0.55	39.24 ± 0.23	38.59 ± 6.19	41.85 ± 0.63	16.50 ± 1.14
GT	63.31 ± 0.45	25.09 ± 0.58	39.19 ± 0.15	34.15 ± 4.81	40.10 ± 0.60	11.97 ± 1.13
OpenGraph (ICL)	5.76 ± 1.03	0.80 ± 0.45	RTE	9.12 ± 1.74	16.19 ± 1.36	RTE
AnyGraph (ICL)	9.47 ± 1.13	9.20 ± 0.67	11.14 ± 5.16	13.52 ± 4.74	11.80 ± 1.01	0.16 ± 0.01
TS-GNN (ICL)	18.91 ± 0.32	MLE	MLE	12.59 ± 1.97	9.46 ± 1.25	MLE
GCOPE (FT)	19.14 ± 0.58	TLE	TLE	8.46 ± 1.15	20.83 ± 1.18	TLE

(b) Results for regression datasets (R^2 is reported).

	hm-prices	avazu-ctr	twitch-views	artnet-views
best const. pred.	-2.85 ± 0.00	0.00 ± 0.00	-22.31 ± 0.00	-9.32 ± 0.00
ResMLP	61.64 ± 0.79	20.35 ± 1.50	11.91 ± 8.00	42.15 ± 0.52
GCN	65.20 ± 0.84	37.49 ± 0.26	68.17 ± 0.24	54.44 ± 0.43
GraphSAGE	67.93 ± 1.24	38.38 ± 0.39	61.46 ± 0.68	51.87 ± 0.48
GAT	70.83 ± 0.99	39.21 ± 0.17	66.32 ± 0.59	52.30 ± 0.34
GT	69.70 ± 0.84	38.27 ± 0.27	65.83 ± 0.24	51.67 ± 0.54

Table 6: Experimental results under the THI (temporal high / inductive) setting. The best result and those statistically indistinguishable from it are highlighted in blue. TLE stands for time limit exceeded (24 hours); MLE stands for memory limit exceeded (80 GB VRAM); RTE stands for runtime error in the official code of GFMs.

(a) Results for classification datasets.

	multiclass classification (accuracy is reported)			binary classification (average precision is reported)		
	hm-categories	pokec-regions	web-topics	tolokers-2	artnet-exp	web-fraud
best const. pred.	19.57 ± 0.00	2.98 ± 0.00	23.28 ± 0.00	8.61 ± 0.00	7.84 ± 0.00	0.15 ± 0.00
ResMLP	32.44 ± 0.54	4.18 ± 0.01	35.49 ± 0.03	21.72 ± 6.69	37.48 ± 0.51	2.83 ± 0.26
GCN	47.05 ± 1.69	6.88 ± 0.51	37.76 ± 0.06	32.43 ± 8.03	41.28 ± 0.28	3.44 ± 0.33
GraphSAGE	48.11 ± 2.08	8.04 ± 0.26	38.04 ± 0.18	30.86 ± 9.48	40.53 ± 0.40	13.88 ± 1.32
GAT	59.34 ± 1.09	13.38 ± 0.35	38.77 ± 0.35	24.53 ± 9.55	41.64 ± 0.32	11.98 ± 1.54
GT	59.54 ± 1.59	17.22 ± 0.42	38.78 ± 0.08	22.89 ± 10.40	40.26 ± 0.82	7.84 ± 2.35
OpenGraph (ICL)	5.76 ± 1.03	0.80 ± 0.45	RTE	9.12 ± 1.74	16.19 ± 1.36	RTE
AnyGraph (ICL)	9.47 ± 1.13	9.20 ± 0.67	11.14 ± 5.16	13.52 ± 4.74	11.80 ± 1.01	0.16 ± 0.01
TS-GNN (ICL)	18.91 ± 0.32	MLE	MLE	12.59 ± 1.97	9.46 ± 1.25	MLE
GCOPE (FT)	15.69 ± 3.44	TLE	TLE	10.73 ± 1.48	19.10 ± 0.96	TLE

(b) Results for regression datasets (R^2 is reported).

	hm-prices	avazu-ctr	twitch-views	artnet-views
best const. pred.	-2.85 ± 0.00	0.00 ± 0.00	-22.31 ± 0.00	-9.32 ± 0.00
ResMLP	61.64 ± 0.79	20.35 ± 1.50	11.91 ± 8.00	42.15 ± 0.52
GCN	64.31 ± 0.82	34.78 ± 0.48	63.58 ± 0.54	53.73 ± 0.47
GraphSAGE	65.80 ± 0.56	36.79 ± 0.55	56.60 ± 0.71	53.37 ± 0.30
GAT	69.74 ± 1.50	37.18 ± 1.02	61.41 ± 1.30	52.44 ± 0.59
GT	67.33 ± 2.05	36.49 ± 0.83	60.67 ± 1.02	52.26 ± 0.48