DOI: 10.15514/ISPRAS-2026-38(5)-15

Оценка качества графовых моделей машинного обучения на разнообразных промышленных данных

Г.В. Баженов, ORCID: 0009-0009-7967-9573 <gvbazhenov@hse.ru>

НИУ Высшая школа экономики,
101978, Россия, г. Москва, ул. Мясницкая, д. 20.

Аннотация. Хотя данные, которые можно естественным образом представить в виде графов, широко распространены в реальных приложениях самых разных отраслей, популярные наборы задач машинного обучения на графах (МО на графах) для предсказания свойств вершин охватывают удивительно узкий набор предметных областей, а графовые нейронные сети (GNN) нередко оцениваются лишь на нескольких академических сетях цитирований. Эта проблема особенно насущна в свете растущего интереса к разработке графовых фундаментальных моделей (GFM). Предполагается, что такие модели способны переносить знания на разнообразные графовые наборы данных из различных областей, однако предлагаемые GFM нередко оцениваются лишь на весьма ограниченном наборе данных из узкого круга приложений. Для восполнения этого пробела мы представляем GraphLand – 14 разнообразных графовых наборов данных для предсказания свойств вершин из широкого спектра различных промышленных приложений. GraphLand позволяет оценивать модели МО на графах на широком спектре графов с разнообразными размерами, структурными характеристиками и наборами признаков в едином формате. Помимо этого, GraphLand позволяет исследовать ранее слабо изученные научные вопросы, в частности как реалистичные временны́е сдвиги распределения в трансдуктивном и индуктивном режимах влияют на качество моделей МО на графах. Симулируя реалистичные промышленные условия, мы используем GraphLand для сравнения GNN с моделями градиентного бустинга над решающими деревьями (GBDT), популярными в промышленных приложениях, и показываем, что GBDT, снабжённые дополнительными признаками на основе графа, могут в ряде случаев быть весьма сильными базовыми моделями. Также мы оцениваем современные универсальные GFM и обнаруживаем, что они не достигают конкурентных результатов на предложенных наборах данных. Исходный код и наборы данных доступны в нашем репозитории на GitHub. Кроме того, наборы данных интегрированы в PyTorch Geometric.

Ключевые слова: машинное обучение на графах; наборы задач; предсказание свойств вершин; графовая нейронная сеть; графовая фундаментальная модель; промышленные наборы данных.

Для цитирования: Баженов Г.В. Оценка качества графовых моделей машинного обучения на разнообразных промышленных данных. Труды ИСП РАН, 2026, том 38, вып. 5, с. 257-286. DOI: 10.15514/ISPRAS-2026-38(5)-15

Evaluating Graph Machine Learning Models on Diverse Industrial Data

G.V. Bazhenov, ORCID: 0009-0009-7967-9573 <gvbazhenov@hse.ru>

National Research University, Higher School of Economics,
20, Myasnitskaya st., Moscow, 101978, Russia.

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. To mimic realistic industrial settings, we use GraphLand to compare GNNs with gradient-boosted decision trees (GBDT) models that are popular in industrial applications and show that GBDTs provided with additional graph-based input features can sometimes be very strong baselines. Further, we evaluate current general-purpose graph foundation models and find that they fail to produce competitive results on our proposed datasets. Our source code and datasets are available in our GitHub repository, and our datasets have also been integrated into PyTorch Geometric.

Keywords: graph machine learning; benchmark; node property prediction; graph neural network; graph foundation model; industrial datasets.

For citation: Bazhenov G.V. Evaluating Graph Machine Learning Models on Diverse Industrial Data. Trudy ISP RAN/Proc. ISP RAS, 2026, vol. 38, issue 5, pp. 257-286. DOI: 10.15514/ISPRAS-2026-38(5)-15

1. Введение

В машинном обучении (МО) всё большую важность обретают данные: для корректной оценки качества методов необходимы высококачественные, реалистичные, надёжные и разнообразные наборы задач для тестирования (benchmarks). В области машинного обучения на графах (МО на графах) устоявшиеся наборы задач подвергаются критике по ряду причин: отсутствие практической релевантности [1], низкое структурное разнообразие, не охватывающее большинство возможных типов графовой структуры [2-3], малое разнообразие прикладных областей [1], бесполезность графовой структуры для рассматриваемых задач [1,4-6], а также потенциальные ошибки при сборе данных, приводящие к некорректным меткам [7] и дублированию вершин [8]. Хотя новые наборы графовых задач и появляются, они, как правило, ориентированы на специфические предметные области (например, трёхмерные молекулярные данные) и требуют специализированных моделей. При этом стандартной и наиболее распространённой постановке задачи МО на графах – предсказанию свойств вершин в едином крупном графе – уделялось значительно меньше внимания. Оценка качества как классических графовых нейронных сетей (graph neural network, GNN), так и современных графовых фундаментальных моделей (graph foundation model, GFM) по-прежнему нередко ограничивается несколькими академическими сетями цитирований, хотя и сама постановка, и разработанные для неё модели находят широкое применение в различных отраслях.

Исторически сложившийся акцент при оценке GNN на академических сетях цитирований, представляющих лишь одну, достаточно узкую прикладную область, обусловлен, по нашему

мнению, скорее доступностью открытых данных такого типа, нежели их реальной значимостью для практических приложений. В то же время наиболее классические и практически значимые примеры реальных графов – социальные сети, веб-графы и дорожные сети – редко используются для оценки GNN, что, по-видимому, связано с отсутствием легкодоступных и качественных наборов данных.

Особый интерес представляет разработка GFM – моделей, которые после крупномасштабного предобучения способны работать с разнообразными графовыми данными без или с применением минимального дообучения [9-10]. Корректная оценка таких моделей требует реалистичных данных, однако предлагаемые графовые фундаментальные модели зачастую оцениваются только на графах с текстовыми признаками (в основном, на сетях цитирований), игнорируя проблему переноса на графы с другими типами признаков. В то же время именно такая переносимость и является ключевым свойством по-настоящему универсальной модели GFM, поскольку графы из разных прикладных областей имеют совершенно разные наборы признаков на вершинах.

Из-за нехватки разнообразных и реалистичных промышленных наборов данных некоторые исследователи предлагают переходить к оценке моделей МО на графах на синтетических данных [2,11]. Мы, однако, убеждены в необходимости оценивать модели на реальных данных, чтобы получать несмещённые оценки их качества в реалистичных сценариях и демонстрировать потенциал МО на графах для промышленных приложений. Таким образом, потребность в открытых, разнообразных и реалистичных графовых наборах данных остаётся крайне высокой.

Чтобы восполнить недостаток реалистичных задач для графовых моделей, мы представляем GraphLand – собрание наборов графовых данных с соответствующими задачами МО из разнообразных промышленных приложений, отражающих реальные сценарии использования МО на графах. GraphLand существенно расширяет спектр доступных данных для оценки моделей, предоставляя в едином формате 14 наборов. Многие из них представляют приложения или структурные свойства, ранее не охваченные стандартными наборами задач. Наборы данных для GraphLand собраны как из открытых источников, которые ранее редко или вовсе не использовались в исследованиях МО на графах, так и из впервые публикуемых данных сервисов крупной технологической компании, где МО на графах уже доказало свою практическую эффективность. Ключевая особенность собрания GraphLand – его разнообразие: графы охватывают широкий спектр предметных областей, размеров и структурных свойств, а также обладают богатыми признаками вершин с различными типами, семантикой и распределениями.

Для каждого набора данных в GraphLand мы предлагаем несколько разбиений, в том числе реалистичное временно́е. Это позволяет исследовать важные для практики, но слабо изученные в литературе вопросы: например, как временны́е сдвиги распределения влияют на качество моделей в трансдуктивном и индуктивном режимах.

Мы провели обширные эксперименты на наборах данных GraphLand, используя как различные GNN и современные GFM, так и классические модели градиентного бустинга над деревьями решений (gradient boosting on decision trees, GBDT) [12]. Последние популярны в промышленных приложениях; мы адаптировали их к графовым данным, добавив признаки, извлечённые из структуры графа. Наши эксперименты показывают, что GNN способны достигать высокого качества в промышленных приложениях, причём модели с механизмом внимания нередко превосходят классические архитектуры. Однако их качество значительно падает под влиянием временны́х сдвигов и динамического изменения структуры графа, что подчёркивает необходимость разработки более устойчивых моделей. Современные же GFM показали на наших данных слабые результаты, уступив более классическим методам.

Мы надеемся, что собрание GraphLand позволит проводить более разностороннюю и реалистичную оценку качества моделей МО на графах, а также что он простимулирует

исследования в малоизученных областях: разработка моделей, устойчивых к временны́м сдвигам и динамическим изменениям графа, а также создание GFM, способных к действительному обобщению на графовые данные из различных предметных областей с разными признаками на вершинах.

2. Вклад

1. **GraphLand** – собрание из 14 разнообразных промышленных графовых наборов данных для предсказания свойств вершин (классификация и регрессия) в едином формате с 4 типами разбиений данных: RL (случайное, низкая доля размеченных вершин, random labelling, low), RH (случайное, высокая доля, random labelling, high), TH (временно́е, высокая доля, temporal split, high) и THI (временно́е, высокая доля, индуктивный режим, temporal split, high, inductive). Наборы данных охватывают широкий спектр промышленных приложений, структурных свойств и типов признаков, ранее слабо представленных в стандартных наборах задач МО на графах.
2. Эмпирическое исследование влияния временны́х сдвигов распределения и индуктивного режима предсказания на качество моделей GNN, GBDT и GFM на реалистичных промышленных данных – исследовательский вопрос, ранее слабо изученный в литературе по МО на графах.
3. Практическая демонстрация того, что модели GBDT, снабжённые дополнительными графовыми признаками посредством агрегации признаков соседей (neighborhood feature aggregation, NFA), могут служить сильными базовыми моделями в реалистичных промышленных условиях, особенно в задачах регрессии. Современные же модели из класса GFM не достигают конкурентных результатов на предложенных наборах данных, что свидетельствует о нерешённости задачи создания по-настоящему универсальных моделей GFM.

3. Ограничения популярных тестовых наборов МО на графах

Наиболее популярными наборами данных в современной литературе по МО на графах, безусловно, являются три академические сети цитирований: cora, citeseer и pubmed [13-17]. Широкое распространение этих наборов данных, по всей видимости, обусловлено их использованием в основополагающей работе по современным GNN [18]. Однако эти наборы данных охватывают лишь одно и довольно узкое приложение – предсказание тематики статей в сетях цитирований. Другой популярный набор данных для МО на графах был представлен в работе [19]; он включает академические сети соавторства coauthor-cs и coauthor-physics, а также сети совместных покупок в электронной коммерции amazon-computers и amazon-photo. Однако все перечисленные наборы данных в совокупности охватывают лишь три приложения, тогда как методы МО на графах применимы к значительно более широкому кругу задач. Позднее в рамках набора задач Open Graph Benchmark (OGB) [20] были представлены более крупномасштабные графовые наборы данных. Однако из пяти наборов данных для предсказания свойств вершин три являются академическими сетями цитирований (ogbn-arxiv, ogbn-mag, ogbn-papers100M), а ещё один – сетью совместных покупок в электронной коммерции (ogbn-products). Таким образом, OGB существенно не расширяет спектр реальных приложений, доступных для оценки моделей МО на графах. Помимо этого, все вышеупомянутые наборы данных предоставляют в качестве признаков вершин только текстовые описания. Между тем сети совместных покупок – весьма практически важное приложение МО на графах – в реалистичных условиях снабжены богатыми метаданными о товарах, которые можно представить в виде числовых и категориальных признаков. Популярные наборы задач МО на графах не содержат наборов данных с такими метаданными. Все вышеупомянутые наборы данных также являются *гомофильными*, то есть рёбра в них, как правило, соединяют вершины одного класса. [21] показали, что даже очень

простые модели могут давать высокие результаты в гомофильных сетях. Поэтому важно, чтобы стандартные наборы задач МО на графах включали и широкий выбор *негомофильных* графов, то есть графов, в которых рёбра не имеют тенденции соединять вершины одного класса. Долгое время единственным популярным источником негомофильных графовых наборов данных была работа [22]. Однако недавно авторы работы [8] показали, что эти наборы данных имеют многочисленные проблемы, включая дублирующиеся вершины, малый размер (что приводит к зашумлённым оценкам метрик) и недостаточное представление классов (например, в наборе данных texas один из классов состоит всего из одной вершины). Хотя в работе [8] авторы ввели несколько новых негомофильных графовых наборов данных, они предназначались для повторной оценки качества различных моделей в условиях отсутствия гомофилии, а не для представления реалистичных приложений МО на графах. Поэтому некоторые из этих наборов данных являются синтетическими (minesweeper), полусинтетическими (roman-empire) или имеют ограниченные признаки вершин, хотя исходный источник данных потенциально предоставляет более полную информацию о вершинах (tolokers, questions).

В целом, широко используемые на сегодняшний день графовые наборы данных для предсказания свойств вершин не позволяют оценивать GNN и другие модели МО на графах в широком спектре практически значимых промышленных приложений.

3.1 Графы с текстовыми признаками на вершинах и обобщаемость графовых фундаментальных моделей

Большинство часто используемых наборов данных для предсказания свойств вершин содержат в качестве признаков вершин только текстовые описания. Однако графы, представляющие реальные сети, зачастую обладают богатыми и разнообразными признаками вершин, выходящими за рамки текста и включающими числовые и категориальные признаки с различными смыслами и распределениями. Большой интерес вызывает разработка универсальных моделей GFM, от которых ожидается умение обобщаться на графы из разных предметных областей [9-10]. Ключевой задачей для таких моделей является способность адаптироваться к графам с различными наборами признаков вершин – а это необходимо для модели, действительно обобщаемой на разные области. Тем не менее предлагаемые в современной литературе по моделям GFM, как правило, игнорируют эту задачу и нередко оцениваются только на графах с текстовыми признаками [23-25]. Текстовые признаки легко проецируются в общее скрытое пространство с помощью предобученных текстовых энкодеров на основе больших языковых моделей (large language model, LLM), что позволяет единственной GFM работать с различными графами, снабжёнными текстовыми признаками. Преобладание таких графов в наборах задач МО на графах привело к тому, что большинство современных исследований по GFM игнорируют проблему обобщения на различные наборы признаков вершин, поскольку для решения задач из стандартных наборов данных этого не требуется. Однако данная проблема чрезвычайно важна для реальных промышленных приложений МО на графах, в которых графы нередко снабжены смесью числовых и категориальных признаков вершин. Следовательно, модели GFM должны уметь работать с такими признаками для эффективного решения практических задач. Проблема создания единой фундаментальной модели, способной работать с произвольными числовыми и категориальными признаками, привлекла значительное внимание в сообществе МО на табличных данных, где такие признаки являются стандартными; первые успешные попытки разработать подобную модель уже появились [26-32]. Однако эти идеи пока не распространились в сообщество МО на графах (вероятно, потому что стандартные наборы задач МО на графах не требуют работы с нетекстовыми признаками вершин), несмотря на значительные преимущества успешной модели GFM, способной работать с произвольными наборами признаков вершин, для практических приложений.

4. Graphland: разнообразные промышленные графовые наборы данных

GraphLand – собрание из 14 графовых наборов данных с задачами предсказания свойств вершин (классификация или регрессия). Часть из них впервые опубликована специально для данного набора, другие собраны из открытых источников данных, которые редко или вовсе не используются в современных наборах задач МО на графах. При выборе данных для GraphLand мы стремимся выполнить следующие требования:

- Наборы данных должны охватывать широкий спектр практически значимых промышленных приложений, включая такие архетипические примеры реальных сетей, как социальные сети, веб-графы и дорожные сети;
- Наборы данных должны иметь структуру графа, полезную для рассматриваемой задачи;
- Графы из разных наборов данных должны обладать разнообразными структурными свойствами;
- Наборы данных должны иметь богатые признаки вершин, включающие числовые и/или категориальные признаки;
- Если граф динамически развивается во времени, то в идеале он должен содержать необходимую временну́ю информацию для построения реалистичного временно́го разбиения данных и содержательной индуктивной постановки;
- Наборы данных должны охватывать диапазон размеров графов, позволяя проводить оценку при различных доступных вычислительных ресурсах, но должны содержать не менее 10,000 вершин во избежание особенно зашумлённых оценок метрик.

В данном разделе мы кратко описываем наши наборы данных и соответствующие задачи предсказания. Более подробные описания приведены в Приложении А, а обсуждение структурных свойств и других характеристик наших наборов данных – в Приложении В.

Сначала опишем наборы данных, основанные на данных, впервые опубликованных специально для нашего тестового набора.

Наборы данных web-fraud, web-topics и web-traffic

Эти три набора данных представляют часть интернета (веб-граф). Вершины – веб-сайты, направленное ребро соединяет две вершины, если хотя бы один пользователь перешёл по ссылке с одного сайта на другой в выбранный период времени. Мы подготовили три набора данных с одним и тем же графом, но разными задачами: в web-fraud задача – предсказать, какие сайты являются вредоносными (сильно несбалансированная бинарная классификация); в web-topics – предсказать тематику, к которой принадлежит сайт (многоклассовая классификация); в web-traffic – предсказать, сколько пользователей посетило сайт за определённый период (регрессия). Почти с 3 миллионами вершин это один из крупнейших публично доступных графов, несущих признаковое описание на вершинах и не являющихся сетью цитирований.

Наборы данных artnet-views и artnet-exp

Эти два набора данных представляют социальную сеть авторов художественного контента. Вершины – пользователи, ребро соединяет две вершины, если пользователи являются друзьями. Мы подготовили два набора данных с одним и тем же графом, но разными задачами: в artnet-views задача – предсказать, сколько просмотров получит пользователь за определённый период (регрессия); в artnet-exp – предсказать, какие пользователи склонны создавать вызывающий контент (бинарная классификация).

Наборы данных city-roads-M и city-roads-L

Эти наборы данных представляют дорожные сети двух крупных городов, причём второй город в несколько раз больше первого. Вершины – участки дорог, направленное ребро

соединяет две вершины, если участки смежны и переход с одного участка на другой разрешён правилами дорожного движения. Задача – предсказать среднюю скорость движения на дорожном участке в определённый момент времени (регрессия).

Набор данных city-reviews

Этот набор данных представляет сервис отзывов о местах и организациях в двух крупных городах. Вершины – пользователи, оставляющие оценки и комментарии о различных местах; ребро соединяет двух пользователей, если они часто оставляют отзывы об одних и тех же организациях. Задача – определение вредоносного поведения (fraud detection): необходимо предсказать, какие пользователи оставляют ложные отзывы (бинарная классификация).

Мы также обнаружили, что помимо академических сетей цитирований существует немало источников открытых сетевых данных, которые, однако, редко или вовсе не используются для оценки моделей МО на графах. Ниже описаны наборы данных, полученные из этих открытых источников. Для этих наборов данных мы определили или расширили признаки и метки вершин, а также построили структуру графа там, где она явно не была представлена.

Набор данных avazu-ctr

Этот набор данных основан на данных о взаимодействии пользователей с рекламой, предоставленных компанией Avazu [33]. Вершины – устройства, используемые для доступа в интернет; ребро соединяет два устройства, если они часто посещают одни и те же сайты. Задача – предсказать отношение числа кликов к числу показов (click-through rate, CTR) рекламы для устройств (регрессия).

Наборы данных hm-categories и hm-prices

Эти два набора данных основаны на сети совместных покупок товаров компании H&M [34]. Вершины – товары, ребро соединяет два товара, если их часто покупают одни и те же покупатели. Мы подготовили два набора данных с одним и тем же графом, но разными задачами: в hm-categories задача – предсказать категорию товара (многоклассовая классификация); в hm-prices – предсказать цену товара (регрессия).

Набор данных pokec-regions

Этот набор данных основан на данных из работы [35] и представляет онлайн-социальную сеть Pokec. Вершины – пользователи, направленное ребро соединяет две вершины, если один пользователь отметил другого как друга. Задача – предсказать регион проживания пользователя (сильно многоклассовая классификация: 183 класса).

Набор данных twitch-views

Этот набор данных основан на данных из работы [36] и представляет стриминговую сеть Twitch. Вершины – пользователи, ребро соединяет две вершины, если пользователи взаимно подписаны друг на друга. Задача – предсказать, сколько просмотров получит пользователь за определённый период (регрессия).

Набор данных tolokers-2

Это новая версия набора данных tolokers из работ [8,37] с существенно расширенным набором признаков вершин. Данный набор данных представляет сеть исполнителей на платформе массового привлечения работников. Вершины – исполнители, ребро соединяет две вершины, если они принимали участие в работе над одними и теми же проектами. Задача – определение вредоносного поведения (fraud detection): необходимо предсказать, какие исполнители были заблокированы в одном из проектов (бинарная классификация).

В целом наши наборы данных охватывают широкий спектр промышленных приложений. В табл. 1 и 2 приведены некоторые характеристики наборов данных: как видно, они также обладают весьма разнообразными структурными свойствами. Более подробное обсуждение этих характеристик приведено в Приложении В.

5. Разбиения данных и условия проведения экспериментов

В задачах предсказания свойств вершин в МО на графах принято выделять две основные постановки в зависимости от объёма доступных размеченных данных. Первая, с высокой долей разметки, предполагает, что обучающая выборка охватывает 50% и более всех вершин графа. Эта постановка характерна для гетерофильных наборов данных, представленных среди прочих в работах [22] и [8], а также для большинства наборов из OGB [20]. Вторая, с низкой долей разметки, использует значительно меньшие обучающие выборки (обычно не более 10% вершин) и традиционно применяется с классическими сетями цитирований cora, citeseer, pubmed и набором данных ogbn-products из OGB. Обе ситуации встречаются на практике в зависимости от ресурсов, выделенных на разметку данных. Для сопоставимости результатов между разными научными работами важно использовать предопределённые разбиения, учитывая при этом различные исследовательские потребности. Поэтому для всех данных в нашем наборе задач мы предоставляем фиксированные разбиения для обеих постановок: RL (случайное, низкая доля) и RH (случайное, высокая доля). В RL вершины случайным образом делятся на обучающую, валидационную и тестовую выборки в пропорции 10%/10%/80%; в RH – в пропорции 50%/25%/25%.

Табл. 1. Характеристики предлагаемых наборов данных GraphLand с задачами классификации.
Table 1. Characteristics of the proposed GraphLand classification datasets.

Метрика	hm-cat.	pokec	web-top.	tolok.-2	city-rev.	artnet-exp	web-fraud
Вершин	46.5K	1.6M	2.9M	11.8K	148.8K	50.4K	2.9M
Рёбер	10.7M	22.3M	12.4M	519.0K	1.2M	280.3K	12.4M
Ср. степень	460.92	27.32	8.56	88.28	15.66	11.12	8.56
Мед. степень	45	13	2	30	4	2	2
Ср. расстояние	2.45	4.68	3.08	2.79	4.91	4.42	3.08
Диаметр	13	14	36	11	19	13	36
Гл. кластер.	0.27	0.05	0.00	0.23	0.26	0.03	0.00
Ср. лок. кл.	0.70	0.11	0.33	0.53	0.41	0.08	0.33
Ассорт. степ.	−0.35	0.00	−0.14	−0.08	0.01	0.03	−0.14
Классов	21	183	28	2	2	2	2
Несм. гомоф.	0.38	0.98	0.55	0.10	0.69	0.28	0.32
Признаков	35	56	263	16	37	75	266

Табл. 2. Характеристики предлагаемых наборов данных GraphLand с задачами регрессии.
Table 2. Characteristics of the proposed GraphLand regression datasets.

Метрика	hm-prices	avazu-ctr	city-M	city-L	twitch	artnet-v.	web-tr.
Вершин	46.5K	76.3K	57.1K	142.3K	168.1K	50.4K	2.9M
Рёбер	10.7M	11.0M	107.1K	231.6K	6.8M	280.3K	12.4M
Ср. степень	460.92	288.04	3.75	3.26	80.87	11.12	8.56
Мед. степень	45	71	4	3	32	2	2
Ср. расстояние	2.45	3.55	126.75	194.05	2.88	4.42	3.08
Диаметр	13	14	383	553	8	13	36
Гл. кластер.	0.27	0.24	0.00	0.00	0.02	0.03	0.00
Ср. лок. кл.	0.70	0.85	0.00	0.00	0.16	0.08	0.33
Ассорт. степ.	−0.35	−0.30	0.70	0.74	−0.09	0.03	−0.14
Цел. ассорт.	0.12	0.18	0.74	0.72	−0.41	0.19	−0.21
Признаков	41	260	26	26	4	50	267

В отличие от случайных разбиений RL и RH, характерных для большинства современных наборов задач, в реальных приложениях данные часто разделяют *по времени*: объектами для обучения служат те, что появились в сети раньше, а появившиеся позже попадают в тест. Несмотря на практическую значимость, временны́е разбиения крайне редко встречаются в наборах данных для предсказания свойств вершин. Насколько нам известно, они доступны только для сетей цитирований из OGB, которые представляют лишь одну прикладную область. В то же время, временны́е разбиения могут заметно усложнить задачу предсказания и повлиять на качество модели, поскольку часто порождают *сдвиги распределения* между обучающей, валидационной и тестовой выборками. Хотя отдельные виды сдвигов уже исследовались в литературе – например, в признаках вершин [38] или в структуре графа [39], – реалистичные временны́е сдвиги обычно затрагивают сразу несколько аспектов данных (признаки, метки, структуру графа). Влияние таких комплексных сдвигов на модели МО на графах изучено слабо. Поэтому для большинства задач в нашем наборе данных GraphLand мы предлагаем разбиение по времени TH (временно́е, высокая доля). Оно делит вершины в пропорции 50%/25%/25% – так же, как и RH. Это позволяет напрямую сравнить результаты на разбиениях RH и TH и оценить, насколько временны́е сдвиги усложняют задачу.

Кроме того, многие реальные сети не статичны, а меняются со временем. В таких приложениях в момент обучения часто недоступны не только метки для новых вершин, но и сами вершины с их признаками и рёбрами. Этот сценарий в машинном обучении на графах известен как *индуктивный*. В отличие от *трансдуктивного* подхода, где при обучении доступен весь граф (включая тестовые вершины), в индуктивном подходе валидационные и тестовые вершины и связанные с ними рёбра на этапе обучения отсутствуют. И хотя известно, что GNN (в отличие от, например, неглубоких эмбеддингов вершин) могут работать в обеих постановках [40], влияние индуктивного подхода, то есть отсутствия полной информации о графе, на качество GNN изучено недостаточно. Более того, в работах, где индуктивная постановка всё же используется, валидационные и тестовые вершины обычно выбирают случайно, что нереалистично: на практике индуктивный сценарий почти всегда обусловлен именно развитием графа во времени. В связи с этим для всех наборов данных в нашем собрании задач с доступной временно́й информацией мы предоставляем индуктивную постановку THI (временно́е разбиение, высокая доля, индуктивное). Она использует то же разделение данных, что и TH, но предоставляет три отдельных снимка графа: для обучения, валидации и тестирования. Это позволяет напрямую сопоставить качество моделей в трансдуктивном и индуктивном сценариях. Насколько нам известно, наша работа – первая, в которой проводится такое прямое сравнение GNN на случайных и временны́х разбиениях (в трансдуктивной постановке), а также на трансдуктивном и индуктивном разбиениях (при одинаковом временном разделении данных). Такое сравнение покажет, насколько существенно эти различия в постановке задачи влияют на итоговое качество GNN.

6. Эксперименты

В наших экспериментах мы используем ряд моделей, описанных в данном разделе. Для всех моделей мы проводим поиск гиперпараметров. Подробности, а также описание других аспектов нашей экспериментальной постановки и более детальные описания моделей приведены в Приложении C.

6.1 Модели

6.1.1 Базовые модели, не использующие структуру графа

Задачи, рассматриваемые в нашей работе, такие как определение вредоносного поведения и предсказание CTR, в промышленных условиях нередко решаются без учёта структуры графа. Поэтому в качестве базовых моделей мы используем несколько моделей, не имеющих

доступа к структуре графа. Их сравнение с моделями, учитывающими граф, позволяет выявить, насколько велика польза от использования структуры графа для рассматриваемой задачи. Первой из них является простая базовая модель ResMLP – многослойный перцептрон (multilayer perceptron, MLP) со связями прямого доступа (skip-connections) [41] и нормализацией слоёв (layer normalization) [42]. Эта модель рассматривает все вершины как независимые образцы и не использует структуру графа. Было показано, что модели такого типа могут служить очень сильными базовыми моделями для промышленных данных со смешанными числовыми и категориальными признаками [43]. Далее мы рассматриваем модели GBDT [12]. Эти модели широко используются в промышленных приложениях и нередко считаются более подходящими, чем нейронные сети, для работы с числовыми признаками [43-44] и задачами регрессии [45], что делает их весьма востребованными базовыми моделями для нашего набора задач. Мы используем три наиболее популярные реализации GBDT: XGBoost [46], LightGBM [47] и CatBoost [48].

Далее, мы стремимся усилить исходно не использующие граф модели в нашем наборе задач, предоставляя им некоторую информацию о графе посредством аугментации признаков. В частности, для каждой вершины графа мы агрегируем признаки всех её одношаговых соседей в графе, вычисляем среднее, максимальное и минимальное значения каждого признака и добавляем эти статистики к исходным признакам вершины. Это имитирует один шаг пространственной свёртки GNN на графе и позволяет изначально не графовым моделям использовать часть той информации о графе, которую используют GNN. Мы называем эту стратегию аугментации признаков *агрегацией признаков соседей* (neighborhood feature aggregation, NFA) и обозначаем модели, использующие её, суффиксом -NFA. Более подробное описание NFA приведено в Приложении C.2.

6.1.2 Графовые нейронные сети

В качестве основных моделей мы используем четыре репрезентативные архитектуры GNN: две классические – графовая свёрточная сеть GCN [18] и GraphSAGE [40], – и две с механизмом внимания – графовая сеть с механизмом внимания GAT [49] и локальный графовый трансформер с вниманием между соседними вершинами GT [50]. Заметим, что GT является локальным графовым трансформером с вниманием только над соседями вершины, что отличает его от глобальных графовых трансформеров с попарным вниманием. Для всех GNN мы применяем модификации из работы [8], добавляющие к моделям связи прямого доступа, нормализацию слоёв и блоки MLP. Мы обнаружили, что эти модификации нередко существенно улучшают качество на наших наборах данных.

Недавно высказывалось мнение, что в некоторых наборах данных МО на графах структура графа фактически не помогает решать рассматриваемые задачи [1, 4-6]. Поэтому при введении новых графовых наборов данных важно экспериментально подтвердить, что структура графа действительно полезна. Для этого можно сравнить качество моделей, учитывающих структуру графа, с моделями, которым граф недоступен. Чтобы исключить влияние других факторов на сравнение, два сравниваемых варианта модели должны быть идентичны во всём, кроме доступа к структуре графа. Наш набор моделей обеспечивает два таких сравнения. Первое – модели без графа можно сравнивать с их NFA-аугментированными версиями, имеющими доступ к информации о признаках соседей. Второе – модель ResMLP можно сравнивать с GNN: наши модели GNN имеют точно такой же базовый блок, что и наша базовая модель ResMLP, за исключением добавленных в GNN модулей пространственной свёртки на графе, что делает сравнение корректным.

6.1.3 Графовые фундаментальные модели

Значительный интерес вызывает разработка GFM – моделей, которые могут применяться к разнообразным графовым наборам данных без дообучения или с минимальным дообучением.

Однако мы обнаружили, что современные GFM преимущественно ориентированы на графы с текстовыми признаками и игнорируют задачу переноса единой модели на наборы данных с различными наборами признаков вершин. Это не позволяет таким моделям быть по-настоящему универсальными, поскольку графы в реальных приложениях нередко имеют множество различных числовых и/или категориальных признаков вершин, важных для решения соответствующих задач. Нам удалось найти лишь несколько GFM, поддерживающих предсказание свойств вершин в графах с произвольными признаками вершин: SAMGPT [51], GCOPE [52], OpenGraph [53], GraphFM [54], AnyGraph [55] и TS-GNN [56]. Из этих моделей только OpenGraph и AnyGraph имеют публично доступные веса. Поэтому мы оцениваем именно эти две модели. Мы высоко оцениваем инициативу исследователей, открыто публикующих свои GFM, и призываем других поступать так же. Для обеих моделей мы используем рекомендованную авторами постановку обучения из контекста (ICL) для классификации вершин, в которой задача формулируется как предсказание связей с виртуальными вершинами класса. Помимо этого, нам удалось воспроизвести предобучение моделей TS-GNN и GCOPE, и мы также оцениваем эти две модели (в частности, для TS-GNN мы используем вариант TS-Mean). Мы используем постановку ICL для TS-GNN и постановку дообучения (FT) для GCOPE, как рекомендовано авторами (заметим, что TS-GNN решает несколько задач наименьших квадратов для подгонки весов перед предсказанием, что можно считать формой обучения, однако, следуя авторам данной модели, мы по-прежнему относим это к ICL). Однако ни одна из рассматриваемых моделей не поддерживает регрессию на вершинах, поэтому мы можем оценивать эти модели только на наших наборах данных классификации (хотя TS-GNN и GCOPE теоретически могут поддерживать регрессию на вершинах, их официальные реализации не предоставляют такой поддержки). При этом наши эксперименты показывают, что рассматриваемые GFM не масштабируются на крупные наборы данных.

6.2 Низкая доля размеченных вершин, случайное разбиение, трансдуктивная постановка

Результаты экспериментов при разбиении с низкой долей размеченных вершин (RL) приведены в табл. 3 и табл. 4.

Прежде всего, структура графа в наших наборах данных весьма полезна для рассматриваемых задач: все GNN всегда значительно превосходят ResMLP, а NFA-аугментированные модели всегда значительно превосходят аналогичные модели без NFA. Предоставление исходно не графовым моделям информации о структуре графа с помощью NFA значительно улучшает их качество и даже позволяет LightGBM-NFA достичь наилучших результатов на четырёх наборах данных, включая три регрессионных. При этом GNN превосходят ResMLP-NFA на этих наборах данных, что свидетельствует о том, что успех LightGBM-NFA обусловлен лучшей способностью GBDT работать с числовыми признаками и целевыми переменными регрессии [43-45]. Это подчёркивает, что GBDT с аугментированными признаками являются сильными базовыми моделями в реалистичных промышленных условиях. При этом GNN всё же превосходят их на большинстве наборов данных, достигая хорошего качества в промышленных приложениях.

Вместе с тем, хотя среди GNN нет единственной лучшей модели, GNN с механизмом внимания (GAT и GT) превосходят классические GNN (GCN и GraphSAGE) на большинстве наборов данных, что показывает важность возможности взвешивать сообщения от соседей в зависимости от их содержимого в реалистичных промышленных наборах данных.

Что касается GFM, все рассмотренные модели показывают очень слабые результаты на всех наших наборах данных. Это свидетельствует о том, что современные GFM ещё далеки от способности конкурировать с классическими GNN на реалистичных наборах данных с богатыми признаками вершин.

Табл. 3. Результаты для наборов данных классификации. Для наборов данных многоклассовой классификации указана точность (Accuracy), для бинарной классификации – Average Precision. Лучший результат и статистически неотличимые от него выделены жирным. TLE – превышение лимита времени (24 часа); MLE – превышение лимита памяти (80 ГБ); RTE – ошибка времени выполнения в официальном коде GFM.
Table 3. Results for classification datasets. Accuracy is reported for multiclass classification datasets and Average Precision is reported for binary classification datasets. The best result and those statistically indistinguishable from it are marked bold. 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.

Модель / метод	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
XGBoost	40.04 ± 0.09	4.93 ± 0.01	TLE	45.76 ± 1.00	74.70 ± 0.13	41.92 ± 0.82	11.54 ± 0.04
LightGBM	39.73 ± 0.08	4.89 ± 0.00	TLE	44.60 ± 0.12	74.51 ± 0.04	41.21 ± 0.12	TLE
CatBoost	40.72 ± 0.40	TLE	TLE	46.10 ± 0.35	74.77 ± 0.10	42.50 ± 0.12	TLE
ResMLP-NFA	48.72 ± 0.38	8.05 ± 0.03	MLE	48.14 ± 1.40	76.02 ± 0.14	38.25 ± 0.56	MLE
LightGBM-NFA	56.55 ± 0.15	9.53 ± 0.01	TLE	**56.16 ± 0.28**	**78.33 ± 0.04**	45.40 ± 0.13	TLE
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

Табл. 4. Результаты для наборов данных регрессии. Для всех наборов данных указана метрика R^2.
Table 4. Results for regression datasets. R^2 is reported for all datasets.

Модель / метод	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
XGBoost	65.68 ± 0.16	26.72 ± 0.02	59.14 ± 0.11	53.75 ± 0.07	13.39 ± 0.00	32.74 ± 0.04	TLE
LightGBM	65.44 ± 0.09	25.83 ± 0.04	57.76 ± 0.10	52.65 ± 0.08	13.38 ± 0.01	32.47 ± 0.04	TLE
CatBoost	66.85 ± 0.28	26.10 ± 0.04	57.53 ± 0.18	51.43 ± 0.17	13.20 ± 0.03	32.89 ± 0.05	TLE
ResMLP-NFA	67.19 ± 0.30	31.11 ± 0.30	57.82 ± 0.14	50.85 ± 0.18	51.43 ± 0.60	51.03 ± 0.41	MLE
LightGBM-NFA	70.46 ± 0.09	31.72 ± 0.06	**61.00 ± 0.05**	**55.26 ± 0.04**	60.20 ± 0.01	**56.55 ± 0.04**	TLE
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

6.3 Высокая доля размеченных вершин, случайное и временно́е разбиения, трансдуктивная и индуктивная постановки

Сводные результаты для разбиений с высокой долей размеченных вершин – случайного (RH) и временно́го (TH), а также для индуктивной постановки (THI) приведены в табл. 5 и 6. Полные результаты представлены в Приложении D.

*Табл. 5. Результаты экспериментов при разбиениях RH (случайное, высокая доля), TH (временно́е, высокая доля) и в постановке THI (временно́е, высокая доля / индуктивное) – наборы данных классификации. Для наборов данных многоклассовой классификации указана точность, для бинарной – Average Precision. Лучший результат и статистически неотличимые от него выделены **красным** для RH, **фиолетовым** для TH и **синим** для THI.*
*Table 5. 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 – classification datasets. Accuracy is reported for multiclass classification datasets and Average Precision is reported for binary classification datasets.*

Модель / метод	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
best const. pred. (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
best const. pred. (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
LightGBM (RH)	47.28 ± 0.14	5.10 ± 0.01	TLE	49.92 ± 0.19	44.98 ± 0.80	TLE
LightGBM (TH)	32.08 ± 0.25	4.13 ± 0.01	TLE	15.85 ± 3.89	37.34 ± 0.18	TLE
LightGBM (THI)	32.08 ± 0.25	4.13 ± 0.01	TLE	15.85 ± 3.89	37.34 ± 0.18	TLE
LightGBM-NFA (RH)	67.09 ± 0.15	12.15 ± 0.02	TLE	**62.19 ± 0.35**	49.26 ± 0.18	TLE
LightGBM-NFA (TH)	52.45 ± 0.18	5.54 ± 0.01	TLE	31.81 ± 7.51	39.89 ± 0.37	TLE
LightGBM-NFA (THI)	47.46 ± 0.28	4.58 ± 0.02	TLE	**45.45 ± 0.82**	39.91 ± 0.25	TLE
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
GCN (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
GCN (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**
GraphSAGE (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**
GraphSAGE (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**
GAT (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**
GAT (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
GT (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
GT (THI)	**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) (RH)	11.69 ± 0.84	2.56 ± 0.42	RTE	44.62 ± 1.35	23.72 ± 1.86	RTE
OpenGraph (ICL) (TH)	5.76 ± 1.03	0.80 ± 0.45	RTE	9.12 ± 1.74	16.19 ± 1.36	RTE
OpenGraph (ICL) (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
AnyGraph (ICL) (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
AnyGraph (ICL) (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
TS-GNN (ICL) (TH)	18.91 ± 0.32	MLE	MLE	12.59 ± 1.97	9.46 ± 1.25	MLE
TS-GNN (ICL) (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
GCOPE (FT) (TH)	19.14 ± 0.58	TLE	TLE	8.46 ± 1.15	20.83 ± 1.18	TLE
GCOPE (FT) (THI)	15.69 ± 3.44	TLE	TLE	10.73 ± 1.48	19.10 ± 0.96	TLE

Табл. 6. Результаты экспериментов при разбиениях RH, TH и в постановке THI – наборы данных регрессии. Для всех наборов данных указана метрика R^2. Цветовое кодирование – как в табл. 5.
Table 6. Experimental results under the RH, TH and THI settings — regression datasets. R^2 is reported for all datasets. Color codes are the same as in table 5.

Модель / метод	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
best const. pred. (TH)	-2.85 ± 0.00	0.00 ± 0.00	-22.31 ± 0.00	-9.32 ± 0.00
best const. pred. (THI)	-2.85 ± 0.00	0.00 ± 0.00	-22.31 ± 0.00	-9.32 ± 0.00
LightGBM (RH)	73.99 ± 0.11	29.60 ± 0.02	13.37 ± 0.00	36.38 ± 0.08
LightGBM (TH)	63.56 ± 0.39	26.23 ± 0.04	-9.60 ± 0.03	42.87 ± 0.08
LightGBM (THI)	63.56 ± 0.39	26.23 ± 0.04	-9.60 ± 0.03	42.87 ± 0.08
LightGBM-NFA (RH)	79.78 ± 0.09	35.35 ± 0.04	62.14 ± 0.01	60.30 ± 0.07
LightGBM-NFA (TH)	**71.36 ± 0.41**	38.69 ± 0.03	43.60 ± 0.03	52.42 ± 0.06
LightGBM-NFA (THI)	**68.88 ± 0.11**	33.04 ± 0.17	24.81 ± 0.11	51.95 ± 0.05
GCN (RH)	79.76 ± 0.76	34.96 ± 0.11	**77.12 ± 0.11**	**61.02 ± 0.13**
GCN (TH)	65.20 ± 0.84	37.49 ± 0.26	**68.17 ± 0.24**	**54.44 ± 0.43**
GCN (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
GraphSAGE (TH)	67.93 ± 1.24	38.38 ± 0.39	61.46 ± 0.68	51.87 ± 0.48
GraphSAGE (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
GAT (TH)	**70.83 ± 0.99**	**39.21 ± 0.17**	66.32 ± 0.59	52.30 ± 0.34
GAT (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
GT (TH)	69.70 ± 0.84	38.27 ± 0.27	65.83 ± 0.24	51.67 ± 0.54
GT (THI)	**67.33 ± 2.05**	**36.49 ± 0.83**	60.67 ± 1.02	52.26 ± 0.48

Наблюдения из предыдущего подраздела о полезности структуры графа, преимуществах GNN с механизмом внимания и слабом качестве GFM справедливы и для постановок с высокой долей размеченных вершин. Далее мы наблюдаем, что временно́е разбиение данных значительно сложнее для всех моделей, чем случайное (за исключением набора данных avazu-ctr). Это важно, поскольку в реальных приложениях временны́е сдвиги распределения широко распространены, и их игнорирование может давать чрезмерно оптимистичные оценки качества.

При этом рассмотренные модели в индуктивной постановке показывают значительно более слабые результаты, чем в трансдуктивной. Эти наблюдения подчёркивают важность разработки методов МО на графах, более устойчивых к временны́м сдвигам распределения и динамическим изменениям структуры графа, для промышленных приложений. В отсутствие таких методов для достижения наилучших результатов рекомендуется часто переобучать GNN на новых данных. GFM, использующие только обучение из контекста для адаптации к новым графам, не страдают от несоответствия между трансдуктивной и индуктивной постановками и потому представляют перспективное направление, однако их качество пока очень слабое по сравнению с GNN на всех наборах данных.

В целом основные результаты таковы:

- GNN способны давать существенные преимущества в промышленных приложениях, где доступна информация о графе, причём GNN с механизмом внимания достигают особенно высоких результатов. Однако модели GBDT, популярные в промышленных приложениях, могут служить сильными базовыми моделями при наличии

информации о графе в качестве дополнительных входных признаков, особенно в задачах регрессии.

- Методы МО на графах могут быть сильно подвержены влиянию временны́х сдвигов распределения и динамически развивающихся графов, поэтому разработка методов МО на графах, способных лучше справляться с такими условиями, является важным направлением исследований.
- Современные универсальные GFM достигают очень слабых результатов на наших наборах данных. Таким образом, проблема создания по-настоящему универсальных GFM далека от решения. Мы надеемся, что наш набор задач сможет служить надёжной испытательной базой для будущих исследований в этом направлении. Первые успешные применения уже появились: после публикации GraphLand в открытом доступе первые GFM, достигающие высоких результатов на наших наборах данных, были представлены в работах [57-58].

7. Заключение

Мы представляем GraphLand – собрание разнообразных наборов графовых данных, отражающих реалистичные промышленные приложения машинного обучения на графах. GraphLand не только предоставляет расширенную платформу для испытаний GNN и GFM, но и позволяет исследовать вопросы, ранее слабо изученные в литературе: влияние временны́х сдвигов распределения и индуктивной постановки задачи на качество графовых моделей. Мы надеемся, что наш набор данных будет стимулировать оценку моделей в более реалистичных и разнообразных условиях, разработку методов, более устойчивых к временны́м сдвигам и динамически меняющимся графам, а также создание более совершенных GFM, способных работать с различными типами признаков на вершинах, а не только с текстовыми описаниями.

7.1 Реализация

Исходный код, включая конвейер предобработки данных и экспериментальную базу, доступен в репозитории на GitHub [74]. Наборы данных GraphLand также интегрированы [75] в библиотеку PyTorch Geometric [59]; это наиболее удобный способ работы с ними.

7.2 Ограничения

Цель нашей работы – предложить разнообразное собрание наборов графовых данных для задачи предсказания свойств вершин. Оно охватывает широкий спектр предметных областей и структурных свойств, в том числе и таких, которые не встречаются в популярных наборах данных для оценки графовых моделей. Однако данные, представимые в виде графов, настолько распространены, что ни одно собрание не может охватить все возможные сценарии их применения. Наше собрание из 14 наборов данных покрывает лишь малую часть ситуаций, где графовое представление данных может быть полезным. Тем не менее мы надеемся, что наша работа побудит сообщество исследователей в области МО на графах использовать более разнообразные данные и сосредоточиться на практически значимых приложениях с графовой структурой данных.

Список литературы / References

[1]. Bechler-Speicher M. et al., “Position: Graph learning will lose relevance due to poor benchmarks,” in International conference on machine learning, 2025.
[2]. Palowitch J., Tsitsulin A., Mayer B., Perozzi B. GraphWorld: Fake graphs bring real insights for GNNs. In Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining, 2022, pp. 3691-3701.

[3]. Maekawa S., Noda K., Sasaki Y., Onizuka M. Beyond real-world benchmark datasets: An empirical study of node classification with GNNs. Advances in Neural Information Processing Systems, 2022, vol. 35, pp. 5562-5574.
[4]. Errica F., Podda M., Bacciu D., Micheli A. A fair comparison of graph neural networks for graph classification. In International conference on learning representations, 2020.
[5]. Li Z., Cao Y., Shuai K., Miao Y., Hwang K. 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]. Coupette C., Wayland J., Simons E., Rieck B. No metric to rule them all: Toward principled evaluations of graph-learning datasets. In International conference on machine learning, 2025.
[7]. Li Y., Xiong M., Hooi B. GraphCleaner: Detecting mislabelled samples in popular graph learning benchmarks. In International conference on machine learning, 2023, pp. 20195-20209.
[8]. Platonov O., Kuznedelev D., Diskin M., Babenko A., Prokhorenkova L. A critical look at the evaluation of GNNs under heterophily: Are we really making progress? In International conference on learning representations, 2023.
[9]. Wang Y., Fan W., Wang S., Ma Y. Towards graph foundation models: A transferability perspective. 2025. DOI: 10.48550/arXiv.2503.09363.
[10]. Mao H. et al. Position: Graph foundation models are already here. In International conference on machine learning, 2024.
[11]. Yoon M., Wu Y., Palowitch J., Perozzi B., Salakhutdinov R. Graph generative model for benchmarking graph neural networks. In International conference on machine learning, 2023.
[12]. Friedman J. H. Greedy function approximation: A gradient boosting machine. Annals of Statistics, 2001, pp. 1189-1232.
[13]. Giles C.L., Bollacker K.D., Lawrence S. CiteSeer: An automatic citation indexing system. In Proceedings of the third ACM conference on digital libraries, 1998, pp. 89-98.
[14]. McCallum A. K., Nigam K., Rennie J., Seymore K. Automating the construction of internet portals with machine learning. Information Retrieval, 2000, vol. 3, no. 2, pp. 127-163.
[15]. Sen P., Namata G., Bilgic M., Getoor L., Galligher B., Eliassi-Rad T. Collective classification in network data. AI Magazine, 2008, vol. 29, no. 3, pp. 93-93.
[16]. Namata G., London B., Getoor L., Huang B. Query-driven active surveying for collective classification. In Proceedings of the workshop on mining and learning with graphs (MLG-2012), 2012.
[17]. Yang Z., Cohen W., Salakhudinov R. Revisiting semi-supervised learning with graph embeddings. In International conference on machine learning, 2016, pp. 40-48.
[18]. Kipf T.N., Welling M. Semi-supervised classification with Graph Convolutional Networks. In International conference on learning representations, 2017.
[19]. Shchur O., Mumme M., Bojchevski A., Günnemann S. Pitfalls of graph neural network evaluation, 2018. DOI: 10.48550/arXiv.1811.05868.
[20]. Hu W. et al. Open graph benchmark: Datasets for machine learning on graphs. Advances in Neural Information Processing Systems, 2020, vol. 33, pp. 22118-22133.
[21]. Huang Q., He H., Singh A., Lim S.-N., Benson A.R. Combining label propagation and simple models out-performs Graph Neural Networks. In International conference on learning representations, 2021.
[22]. Pei H., Wei B., Chang K.C.-C., Lei Y., Yang B. Geom-GCN: Geometric graph convolutional networks. In International conference on learning representations, 2020.
[23]. Liu H. et al. One for all: Towards training one graph model for all classification tasks. In International conference on learning representations, 2024.
[24]. Li Y., Wang P., Li Z., Yu J.X., and J. Li. ZeroG: Investigating cross-dataset zero-shot transferability in graphs. In Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining, 2024, pp. 1725-1735.
[25]. He Y., Sui Y., He X., Hooi B. 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, 2025, pp. 448-459.
[26]. Hollmann N., Müller S., Eggensperger K., Hutter F. TabPFN: A transformer that solves small tabular classification problems in a second. In International conference on learning representations, 2023.
[27]. Hollmann N. et al. Accurate predictions on small data with a tabular foundation model. Nature, 2025, vol. 637, no. 8045, pp. 319-326.
[28]. Ye H.-J., Zhou Q., Yin H.-H., Zhan D.-C., Chao W.-L. Rethinking pre-training in tabular data: A neighborhood embedding perspective, 2023. DOI: 10.48550/arXiv.2311.00055.

[29]. Ye H.-J., Liu S.-Y., Chao W.-L. A closer look at TabPFN v2: Understanding its strengths and extending its capabilities, 2025. DOI: 10.48550/arXiv.2502.17361
[30]. Mueller A.C., Curino C.A., Ramakrishnan R. MotherNet: Fast training and inference via hyper-network transformers. In International conference on learning representations, 2025.
[31]. Ma J. et al. TabDPT: Scaling tabular foundation models on real data, 2024. DOI: 10.48550/arXiv.2410.18164.
[32]. Qu J., Holzmüller D., Varoquaux G., Le Morvan M. TabICL: A tabular foundation model for in-context learning on large data. In International conference on machine learning, 2025.
[33]. Wang S., Cukierski W. Click-through rate prediction. Kaggle, 2014. Available at: https://kaggle.com/competitions/avazu-ctr-prediction, accessed 29.08.2026.
[34]. García Ling C. et al. H&M personalized fashion recommendations. Kaggle, 2022. Available at: https://kaggle.com/competitions/h-and-m-personalized-fashion-recommendations, accessed 29.08.2026.
[35]. Takac L., Zabovsky M. Data analysis in public social networks. In International scientific conference and international workshop present day trends of innovations, 2012.
[36]. Rozemberczki B., Sarkar R. Twitch gamers: A dataset for evaluating proximity preserving and structural role-based node embeddings, 2021. DOI: 10.48550/arXiv.2101.03091
[37]. Likhobaba D., Pavlichenko N., Ustalov D. Toloker graph: Interaction of crowd annotators. Zenodo, 2023. DOI: 10.5281/zenodo.7620795.
[38]. Gui S., Li X., Wang L., Ji S. GOOD: A graph out-of-distribution benchmark. Advances in Neural Information Processing Systems, 2022, vol. 35, pp. 2059-2073.
[39]. Bazhenov G., Kuznedelev D., Malinin A., Babenko A., Prokhorenkova L. Evaluating robustness and uncertainty of graph models under structural distributional shifts. Advances in Neural Information Processing Systems, 2023, vol. 36, pp. 75567-75594.
[40]. Hamilton W., Ying Z., Leskovec J. Inductive representation learning on large graphs. Advances in Neural Information Processing Systems, 2017, vol. 30.
[41]. He K., Zhang X., Ren S., Sun J. Deep residual learning for image recognition. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2016, pp. 770-778.
[42]. Ba J.L., Kiros J.R., Hinton G.E. Layer normalization, 2016. DOI: 10.48550/arXiv.1607.06450.
[43]. Gorishniy Y., Rubachev I., Khrulkov V., Babenko A. Revisiting deep learning models for tabular data. Advances in Neural Information Processing Systems, 2021, vol. 34, pp. 18932-18943.
[44]. Gorishniy Y., Rubachev I., Babenko A. On embeddings for numerical features in tabular deep learning. Advances in Neural Information Processing Systems, 2022, vol. 35, pp. 24991-25004.
[45]. S. Zhang, L. Yang, M. B. Mi, X. Zheng, and A. Yao, "Improving deep regression with ordinal entropy," in International conference on learning representations, 2023.
[46]. Chen T., Guestrin C. XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, 2016, pp. 785-794.
[47]. Ke G. et al. LightGBM: A highly efficient gradient boosting decision tree. Advances in Neural Information Processing Systems, 2017, vol. 30.
[48]. Prokhorenkova L., Gusev G., Vorobev A., Dorogush A.V., Gulin A. CatBoost: Unbiased boosting with categorical features. Advances in Neural Information Processing Systems, 2018, vol. 31.
[49]. Veličković P., Cucurull G., Casanova A., Romero A., Lio P., Bengio Y. Graph Attention Networks. In International conference on learning representations, 2018.
[50]. Shi Y., Huang Z., Feng S., Zhong H., Wang W., Sun Y. Masked label prediction: Unified message passing model for semi-supervised classification. Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, 2021.
[51]. Yu X., Gong Z., Zhou C., Fang Y., Zhang H. SAMGPT: Text-free graph foundation model for multi-domain pre-training and cross-domain adaptation. In Proceedings of the ACM on web conference 2025, 2025, pp. 1142-1153.
[52]. Zhao H., Chen A., Sun X., Cheng H., Li J. 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, 2024, pp. 4443-4454.
[53]. L. Xia, B. Kao, and C. Huang, "OpenGraph: Towards open graph foundation models," Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2024.
[54]. Lachi D., Azabou M., Arora V., Dyer E. GraphFM: A scalable framework for multi-graph pretraining. 2024. DOI: 10.48550/arXiv.2407.11907.
[55]. Xia L., Huang C. AnyGraph: Graph foundation model in the wild, 2024. DOI: 10.48550/arXiv.2408.10700.

[56]. Finkelshtein B., Ceylan İ. İ., Bronstein M., Levie R. Equivariance everywhere all at once: A recipe for graph foundation models, 2025. DOI: 10.48550/arXiv.2506.14291.
[57]. Eremeev D., Bazhenov G., Platonov O., Babenko A., Prokhorenkova L. Turning tabular foundation models into graph foundation models, 2025. DOI: 10.48550/arXiv.2508.20906.
[58]. Eremeev D., Platonov O., Bazhenov G., Babenko A., Prokhorenkova L. GraphPFN: A prior-data fitted graph foundation model, 2025. DOI: 10.48550/arXiv.2509.21489.
[59]. Fey M., Lenssen J. E. Fast graph representation learning with PyTorch Geometric. Representation Learning on Graphs and Manifolds Workshop at The Seventh International Conference on Learning Representations, 2019.
[60]. Ivanov S., Prokhorenkova L. Boost then convolve: Gradient boosting meets graph neural networks. In International conference on learning representations, 2021.
[61]. Lim D. et al. Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods. Advances in Neural Information Processing Systems, 2021, vol. 34, pp. 20887-20902.
[62]. Boccaletti S. et al. The structure and dynamics of multilayer networks. Physics Reports, 2014, vol. 544, no. 1, pp. 1-122.
[63]. Platonov O., Kuznedelev D., Babenko A., Prokhorenkova L. Characterizing graph datasets for node classification: Homophily–Heterophily dichotomy and beyond. In Advances in Neural Information Processing Systems, 2023, pp. 523-548.
[64]. Mironov M., Prokhorenkova L. Revisiting graph homophily measures. Learning on Graphs Conference, 2024.
[65]. Gilmer J., Schoenholz S.S., Riley P.F., Vinyals O., Dahl G.E. Neural message passing for quantum chemistry. In International conference on machine learning, 2017, pp. 1263-1272.
[66]. Hendrycks D., Gimpel K. Gaussian error linear units (GELUs). 2016. DOI: 10.48550/arXiv.1606.08415.
[67]. You Y., Chen T., Sui Y., Chen T., Wang Z., Shen Y. Graph contrastive learning with augmentations. Advances in Neural Information Processing Systems, 2020, vol. 33, pp. 5812-5823.
[68]. Xia J., Wu L., Chen J., Hu B., Li S.Z. SimGRACE: A simple framework for graph contrastive learning without data augmentation. In Proceedings of the ACM web conference, 2022, pp. 1070-1079.
[69]. Akiba T., Sano S., Yanase T., Ohta T., Koyama M., Optuna: A next-generation hyperparameter optimization framework. In Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery and data mining, 2019, pp. 2623-2631.
[70]. Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R. Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 2014, vol. 15, no. 56, pp. 1929-1958.
[71]. Kingma D.P., Ba J. Adam: A method for stochastic optimization. In International conference on learning representations, 2015.
[72]. Paszke A. et al. PyTorch: An imperative style, high-performance deep learning library. Advances in Neural Information Processing Systems, 2019, vol. 32.
[73]. Wang M. et al. Deep Graph Library: A graph-centric, highly-performant package for graph neural networks. Representation Learning on Graphs and Manifolds Workshop at The Seventh International Conference on Learning Representations, 2019.
[74]. https://github.com/yandex-research/graphland
[75]. https://pytorch-geometric.readthedocs.io/en/latest/generated/torch_geometric.datasets.GraphLandDataset.html
[76]. https://zenodo.org/records/16895532
[77]. https://www.kaggle.com/datasets/bazhenovgleb/graphland

Информация об авторах / Information about authors

Глеб Владимирович БАЖЕНОВ – аспирант факультета компьютерных наук Национального исследовательского университета «Высшая школа экономики». Область научных интересов: машинное обучение на табличных данных, графах и реляционных базах данных.

Gleb Vladimirovich BAZHENOV – postgraduate student at the Faculty of Computer Science, HSE University. Research interests: tabular machine learning, graph machine learning, relational machine learning.

Приложение А. Описание наборов данных

В данном разделе приведено более подробное описание наборов данных GraphLand. Ни один из предложенных наборов данных не содержит персональных данных. Для наборов, впервые собранных в нашей работе, публикация данных была одобрена юридической командой. Наборы данных GraphLand в исходном формате доступны на Zenodo [76] и Kaggle [77]. Также они доступны в репозитории на GitHub [74] и интегрированы в PyTorch Geometric [75].

Наборы данных web-fraud, web-topics и web-traffic

Эти три набора данных являются веб-графами: они представляют сегмент сети Интернет. Вершины – веб-сайты; направленное ребро соединяет две вершины, если хотя бы один пользователь перешёл по ссылке с одного сайта на другой в выбранный период. Подготовлены три набора данных с одинаковым графом, но разными задачами: в web-fraud – предсказать, какие сайты являются вредоносными (значительно несбалансированная бинарная классификация); в web-topics – предсказать тематику сайта (многоклассовая классификация); в web-traffic – предсказать число посетителей сайта за определённый период (регрессия). Граф содержит почти 3 млн вершин – это один из наибольших публично доступных атрибутированных графов, не являющихся сетями цитирований. Вершины имеют более двухсот признаков: среди них – количество видео на сайте (числовой признак), зона доменного имени сайта и принадлежность к бесплатному хостингу (категориальные признаки). Данные получены из поисковой системы Яндекса.

Наборы данных artnet-views и artnet-exp

Эти два набора данных представляют социальную сеть авторов изображений. Вершины – пользователи; ненаправленное ребро соединяет двух пользователей, являющихся друзьями. Подготовлены два набора данных с одинаковым графом: в artnet-views – предсказать, сколько просмотров получает пользователь за определённый период (регрессия); в artnet-exp – предсказать, какие пользователи склонны создавать вызывающий контент (бинарная классификация). Среди признаков вершин – интересы пользователей (категориальные признаки). Данные получены с платформы генеративного искусства для создания изображений Shedevrum.

Наборы данных city-roads-M и city-roads-L

Эти наборы данных получены из журналов навигационного сервиса и представляют дорожные сети двух крупных городов, причём второй набор данных в несколько раз крупнее. Вершины – сегменты дорог; направленное ребро соединяет две вершины, если сегменты смежны и правила дорожного движения разрешают передвижение с одного сегмента на другой. Задача – предсказать среднюю скорость движения на участке дороги в конкретный момент времени (регрессия). Признаки включают различную информацию о сегменте дороги: бинарные индикаторы наличия знака объезда на велосипеде, окончания сегмента пешеходным переходом или платным пунктом, наличия неудовлетворительного покрытия дорожного полотна, ограничения для грузовых автомобилей, наличия полосы для общественного транспорта (категориальные признаки). Числовые признаки – длина сегмента дороги и географические координаты концевых точек дороги. Данные получены из сервиса Яндекс.Карты.

Набор данных city-reviews

Этот набор данных получен из журналов сервиса отзывов, в котором пользователи могут оставлять отзывы и оценки для мест и организаций в двух крупных городах. Вершины – пользователи; ненаправленное ребро соединяет двух пользователей, часто оставляющих отзывы на одни и те же заведения. Задача – определение вредоносного поведения: необходимо выявить, какие пользователи оставляют ложные отзывы (бинарная классификация). Признаки вершин базируются на взаимодействии пользователей с сервисом; среди них – доля негативных отзывов среди всех оставленных пользователем отзывов

(числовой признак) и браузер для доступа к сервису (категориальный признак). Данные получены из сервиса Яндекс.Карты.

Набор данных avazu-ctr

Этот набор данных основан на открытых данных, представленных на соревновании Kaggle, организованном Avazu [33]. Данные содержат информацию о взаимодействиях устройств, используемых для доступа в Интернет, с веб-сайтами и рекламой. В нашем графе вершины – устройства, ребро соединяет два устройства, если они часто посещают одни и те же веб-сайты; граф ненаправленный. Меньшая версия аналогичного набора данных использовалась в работе [60]; однако в ней хранилось лишь небольшое подмножество устройств, тогда как наш набор данных включает все доступные устройства – в результате возникает граф, более чем в 50 раз превышающий по размеру. Задача – предсказать отношение числа кликов к числу показов рекламы (CTR), наблюдаемый на устройствах (регрессия). Вершины графа имеют более двухсот числовых признаков, однако большинство из них анонимизированы в исходном источнике данных.

Наборы данных hm-categories и hm-prices

Эти наборы данных основаны на открытых данных соревнования Kaggle, организованного H&M [34]. Граф представляет сеть совместных покупок: вершины – товары, ребро соединяет два товара, если одни и те же покупатели часто приобретают их одновременно; граф ненаправленный. Подготовлены два набора данных с одинаковым графом: в hm-categories – предсказать категорию товара (многоклассовая классификация); в hm-prices – предсказать цену товара (регрессия). Признаки включают метаданные о товаре – цвет товара (категориальный признак) и статистика покупок, например доля покупок товара в разные дни недели (числовые признаки).

Набор данных pokec-regions

Этот набор данных основан на данных из работы [35]. Он представляет онлайн-социальную сеть Pokec. Вершины – пользователи; направленное ребро соединяет двух пользователей, если один добавил другого в друзья. Хотя этот граф достаточно популярен в анализе сетей как пример классической социальной сети, для задач машинного обучения он используется относительно редко, за исключением работы [61], использующей тот же граф, но с иной задачей, другими признаками вершин и другим разбиением. В нашем наборе данных задача – предсказать, из какого региона происходит пользователь (многоклассовая классификация со 183 классами). Признаки вершин базируются на информации из профиля пользователя – среди них доля заполненности профиля (числовой признак) и бинарные индикаторы заполнения различных полей профиля (категориальные признаки).

Набор данных twitch-views

Этот набор данных основан на данных из работы [36]. Он представляет сеть потоковых трансляций Twitch. Вершины – пользователи; ненаправленное ребро соединяет двух пользователей, если они оба подписаны друг на друга. Задача – предсказать, сколько просмотров получает пользователь за определённый период (регрессия). Признаки вершин базируются на информации из профиля; среди них – язык пользователя и аффилиация (категориальные признаки).

Набор данных tolokers-2

Это новая версия набора данных tolokers из работ [8, 37] со значительно расширенным набором признаков вершин. Он основан на данных краудсорсинговой платформы Toloka; граф представляет сеть исполнителей на платформе массового привлечения работников. Вершины – исполнители, ребро соединяет две вершины, если они работали над одними и теми же проектами. Задача – определение вредоносного поведения: необходимо предсказать, какие работники были заблокированы в одном из проектов (бинарная классификация). Новые признаки вершин включают различную статистику работы исполнителей – число

подтверждённых заданий и число пропущенных заданий (числовые признаки), а также информацию из профиля – уровень образования (категориальный признак).

Часть графов в нашем наборе задач ненаправленные, часть – направленные (описания выше). Все ненаправленные графы состоят из одной связной компоненты, все направленные графы – из одной слабо связной компоненты.

Для всех наборов данных предоставляются случайное стратифицированное разбиение RL и RH. Помимо этого, предоставляется временно́е разбиение TH (с возможностью индуктивного режима THI) для всех наборов данных, за исключением city-roads-M и city-roads-L (дорожные сети в основном не меняются со временем), а также city-reviews и web-traffic (для них необходимая временна́я информация оказалась недоступной).

Приложение B. Свойство наборов данных

Ключевой характеристикой нашего набора задач является разнообразие. Как описано выше, наши графы приходят из разных областей и имеют разные задачи. Рёбра в них также построены по-разному (на основе взаимодействий пользователей, сходства активности, физических связей и так далее). Некоторые свойства наших графов представлены в табл. 1 и 2 (см. ниже определения используемых характеристик). Размеры наборов данных варьируются от 11K до 3M вершин. Меньшие графы подходят для вычислительно затратных моделей, тогда как большие дают умеренный вызов для масштабируемости моделей. Средние и медианные степени вершин существенно различаются – тестовый набор включает как разреженные, так и относительно плотные графы, включая графы со средней степенью порядка сотен – больше, чем на большинстве существующих наборов данных для МО на графах (такие графы могут подчеркивать важность механизмов мягкого выбора рёбер в GNN с механизмом внимания). Среднее расстояние между вершинами варьируется от 2,45 (для hm-categories и hm-prices) до 194 (для city-roads-L); диаметр графа (максимальное расстояние) – от 8 (для twitch-views) до 553 (для city-roads-L). Далее приводятся значения коэффициентов кластеризации, показывающих, насколько типичны замкнутые тройки вершин для данного графа. В литературе существуют два определения [62]: глобальный коэффициент кластеризации и средний локальный коэффициент кластеризации. Представлены наборы данных как с высокими, так и с почти нулевыми коэффициентами кластеризации, а также графы, где глобальный и локальный коэффициенты существенно расходятся (что возможно при несбалансированном распределении степеней вершин). Коэффициент ассортативности по степени – корреляция Пирсона степеней пар связанных вершин – для большинства графов отрицателен или близок к нулю (вершины не стремятся соединяться с вершинами схожей степени); исключения – city-roads-M и city-roads-L, для которых ассортативность по степени положительна и велика.

Обсудим также связь меток и структуры графа. Для регрессионных наборов данных сходство меток связанных вершин измеряется целевой ассортативностью – корреляцией Пирсона целевых значений пар связанных вершин. Например, для city-roads-M и city-roads-L целевая ассортативность положительна и достаточно велика (что ожидаемо для задачи предсказания скорости на дорожной сети), тогда как для twitch-views и web-traffic она отрицательна. Для задач классификации сходство меток соседних вершин принято называть *гомофилией*: в гомофильных наборах данных вершины стремятся соединяться с вершинами того же класса. Вопрос о том, как правильно измерять гомофилию, привлёк определённое внимание исследователей. Было отмечено [61] и [63], что традиционно используемые в литературе метрики гомофилии – например, доля рёбер, соединяющих вершины одного класса, – не подходят для сравнения уровней гомофилии между графами с разным числом классов и разным балансом размеров. [63] предложили набор свойств, которым должна удовлетворять корректная метрика гомофилии; [64] построили первую известную меру гомофилии, удовлетворяющую всем этим свойствам, – *несмещённую гомофилию*. Поэтому в данной

работе используется несмещённая гомофилия. Несмещённая гомофилия (при $\alpha = 0$, подробнее см. [64]) принимает значения в $[-1, 1]$: значение 1 соответствует идеальной гомофилии, -1 – идеальной гетерофилии, 0 – отсутствию предпочтений (такие графы обычно называют гетерофильными, хотя точнее называть их негомофильными). Значения несмещённой гомофилии не следует сравнивать со значениями других метрик гомофилии; уровни несмещённой гомофилии для ряда популярных наборов данных приведены в [64]. Несмещённая гомофилия показывает, что среди наших наборов данных pokec-regions и city-reviews являются гомофильными, тогда как остальные – негомофильными. Таким образом, наш набор задач значительно расширяет набор доступных негомофильных графовых наборов данных.

Наконец, наши наборы данных имеют разнообразные наборы признаков вершин, состоящих из числовых и категориальных признаков с различными значениями и распределениями. Все наборы данных, кроме twitch-views и tolokers-2, содержат не менее нескольких десятков признаков вершин, а некоторые – несколько сотен.

В целом наши наборы данных разнообразны по области применения, масштабу, структурным свойствам, связям граф-метка и признаковым описаниям вершин. Исходя из реальных приложений МО на графах, они могут служить ценным инструментом для исследования и разработки методов МО на графах в промышленности.

B.1 Вычисление характеристик наборов данных

Далее описаны характеристики, используемые в табл. 1. Заметим, что хотя часть графов в нашем наборе данных ориентированная, для вычисления всех приведённых характеристик все графы предварительно преобразовывались в неориентированные, поскольку некоторые из характеристик не определены для ориентированных графов.

Средняя степень и *медианная степень* – среднее и медианное число соседей вершины соответственно. Поскольку все наши графы связны (при рассмотрении как неориентированных), для любых двух вершин существует путь между ними. *Среднее расстояние* – средняя длина кратчайших путей между всеми парами вершин; *диаметр* – максимальная длина кратчайших путей. Для наибольших графов – pokec-regions, web-traffic, web-fraud и web-topics – среднее расстояние аппроксимируется средним по расстояниям между 100 000 случайно выбранными парами вершин. *Глобальный коэффициент кластеризации* вычисляется как утроенное число треугольников, делённое на число пар смежных рёбер. *Средний локальный коэффициент кластеризации* сначала вычисляет локальный коэффициент кластеризации каждой вершины – долю связанных пар её соседей – а затем усредняет по всем вершинам. *Коэффициент ассортативности по степени* – корреляция Пирсона степеней пар связанных вершин. *Целевая ассортативность* для регрессионных наборов данных – корреляция Пирсона целевых значений пар связанных вершин. Для вычисления *несмещённой гомофилии* используется простейшая версия из [64] с параметром $\alpha = 0$.

Приложение C. Постановка экспериментов

C.1 Модели

Модели, не использующие структуру графа

Как простую базовую модель мы используем ResMLP – многослойный перцептрон (MLP) со скиповыми связями [41] и нормализацией слоёв [42]. Эта модель не имеет никакой информации о структуре графа и обрабатывает вершины как независимые объекты – такие модели мы называем *моделями, не использующими структуру графа*. Показано, что MLP-подобные модели со скиповыми связями могут служить очень сильными базовыми моделями для промышленных данных с числовыми и категориальными признаками [43]. Кроме того,

рассматриваются три наиболее распространённые реализации GBDT, широко применяемые в промышленности: XGBoost [46], LightGBM [47] и CatBoost [48]. Модели GBDT часто считаются превосходящими нейронные сети при работе с числовыми признаками [43-44] и задачами регрессии [45].

Перечисленные выше модели не используют структуру графа. Чтобы выяснить, можно ли улучшить их, предоставив доступ к части информации о графе, мы дополняем их *агрегацией признаков соседей* (NFA) – простой техникой обогащения признаков, расширяющей признаки каждой вершины графа агрегированной информацией о признаках её соседей (см. Приложение C.2). Мы добавляем NFA к ResMLP и LightGBM – нашим наиболее быстрым моделям, не использующим структуру графа. Версии с NFA обозначаются суффиксом -NFA. Если структура графа полезна для рассматриваемой задачи, то модели с NFA должны значительно превосходить их аналоги без NFA. В наших экспериментах это действительно наблюдается, подтверждая полезность структуры графа в предоставляемых наборах данных. Однако NFA предоставляет лишь ограниченный доступ к информации о графе – только агрегированные признаки соседей первого порядка. Значительно больший объём информации о графе доступен GNN, которые рассматриваются в следующем абзаце.

Графовые нейронные сети

Мы рассматриваем несколько репрезентативных архитектур GNN. В качестве простых классических моделей используются GCN [18] и GraphSAGE [40]. Для GraphSAGE применяется версия со средней агрегацией без сэмплирования соседей – модель обучается на полном графе. Также используются две GNN с агрегацией по соседям на основе механизма внимания: GAT [49] и Graph Transformer (GT) [50]. GT является *локальным* графовым трансформером: каждая вершина взаимодействует только со своими соседями в графе (в отличие от *глобальных* графовых трансформеров, где каждая вершина взаимодействует со всеми остальными и которые не являются примерами стандартных нейронных сетей с передачей сообщений (MPNN) из [65]). Следуя [8], все рассматриваемые GNN оснащены скиповыми связями и нормализацией слоёв, что оказалось важным для их высокого качества на наших наборах данных. После каждого блока агрегации информации по соседям добавляется двухслойный MLP с активацией GELU [66]. Все наши графовые модели реализованы в том же коде, что и ResMLP: каждый остаточный блок ResMLP просто заменяется остаточным блоком агрегации по соседям выбранной архитектуры GNN. Сравнение качества ResMLP и GNN – ещё один способ оценить полезность информации о графе для задачи. В наших экспериментах GNN значительно превосходят ResMLP на всех наборах данных, что ещё раз подтверждает полезность предоставляемой структуры графа.

Графовые фундаментальные модели

Большинство существующих GFM не поддерживают задачи предсказания свойств вершин в графах с произвольными признаками вершин. Среди тех, что поддерживают, нам удалось найти только две модели с открытыми предобученными параметрами моделей: OpenGraph [53] и AnyGraph [55]. OpenGraph и AnyGraph используют архитектуру трансформера и предобучаются на задаче предсказания связей на смеси различных графовых наборов данных. Модели различаются видами данных. OpenGraph использует только реляционную информацию и строит представления вершин на основе SVD-разложения матрицы смежности; AnyGraph также использует признаки вершин, комбинируя SVD-факторы матриц признаков и смежности. Обе модели предназначены для адаптации к новым наборам данных для классификации вершин без дообучения, в рамках обучения из контекста (ICL). Они решают любую задачу классификации вершин как задачу предсказания связей с виртуальными вершинами классов.

Также мы воспроизвели предобучение ещё двух GFM – TS-GNN [56] и GCOPE [52] (их веса не опубликованы, но код обучения доступен). GCOPE применяет проекцию признаков вершин (например, на основе SVD или механизма внимания) и использует дополнительные

виртуальные вершины-координаторы для одновременной обработки разных графовых наборов данных на этапе предобучения. В качестве целевой функции предобучения используются GraphCL [67] или SimGRACE [68]. В отличие от ICL-подходов, GCOPE использует дообучение для адаптации.

TS-GNN использует принципиально иной подход. Его авторы отмечают, что GFM должна быть инвариантна к перестановкам признаков, эквивариантна к перестановкам меток и эквивариантна к перестановкам вершин; предлагается линейное преобразование, обладающее этими свойствами, и используется как основной строительный блок GFM. TS-GNN требует решения нескольких задач наименьших квадратов перед предсказанием, что является формой обучения; однако, следуя авторам, мы классифицируем её как ICL, так как решение задач наименьших квадратов вычислительно значительно легче типичного дообучения. В оригинальной статье предложены два варианта: TS-Mean и TS-GAT; во всех экспериментах используется TS-Mean.

Ни одна из рассмотренных GFM не поддерживает регрессию вершин (для OpenGraph и AnyGraph это принципиальное ограничение архитектуры; для TS-GNN и GCOPE – лишь ограничение официальных реализаций). Наши эксперименты также показывают, что эти модели не масштабируются на большие наборы данных. Эти ограничения препятствуют успешной оценке рассмотренных GFM на значительной части наших наборов данных.

C.2 Агрегация признаков соседей

Агрегация признаков соседей (NFA) – это метод обогащения признаков вершины информацией из её локального окружения в графе. В наших экспериментах он значительно улучшает качество моделей, которые изначально не используют структуру графа.

Суть метода в том, что для каждой вершины $v \in V$ и каждого её признака $x \in X$ вычисляются статистики по значениям этого признака у её соседей. Для числовых признаков в качестве статистик используются среднее (mean), максимум (max) и минимум (min). Это даёт три новых признака. Для категориальных признаков сначала применяется one-hot-кодирование, а затем для каждого полученного бинарного признака вычисляется среднее (mean) в окрестности, что порождает столько новых признаков, сколько у исходного категориального признака было уникальных значений.

Формально, для вершины v, её 1-окрестности $\mathcal{N}_G(v)$ и признака x сначала собирается множество S значений этого признака у соседей, за исключением значений, являющихся NaN (т.е. соседи, у которых признак не равен NaN). Если все значения соседей равны NaN, соответствующая статистика также полагается равной NaN. Затем к нему применяется инвариантная к перестановкам функция агрегации f:

$$\boldsymbol{S} = \{\boldsymbol{x_u}: \boldsymbol{u} \in \boldsymbol{\mathcal{N}_G(v)} \wedge \boldsymbol{x_u} \text{ is not NaN}\}, \quad \boldsymbol{f(S) = h}.$$

Если множество S пусто (например, у вершины нет соседей или у всех соседей значение признака отсутствует), результат полагается равным NaN, т.е. $f(\varnothing) = \text{NaN}$. Полученный в результате агрегации NFA-вектор конкатенируется с исходным вектором признаков вершины v. Процедура выполняется для всех вершин и всех исходных признаков.

На рис. C.1 показан простой пример работы NFA. Центральная вершина имеет четырёх соседей, у каждого из которых есть один числовой признак (синий) и один категориальный (зелёный). Чтобы построить NFA-вектор для центральной вершины, вычисляются агрегированные значения признаков её соседей: mean, max и min для числового признака, а также mean для one-hot-представления категориального признака.

C.3 Постановка экспериментов и выбор гиперпараметров

Часть графов в собрании данных GraphLand ориентированная. Для экспериментов все ориентированные графы преобразовывались в неориентированные (каждое ребро заменялось

неориентированным, дублирующиеся рёбра удалялись). Изучение различных способов учёта направленности рёбер оставлено для будущих исследований.

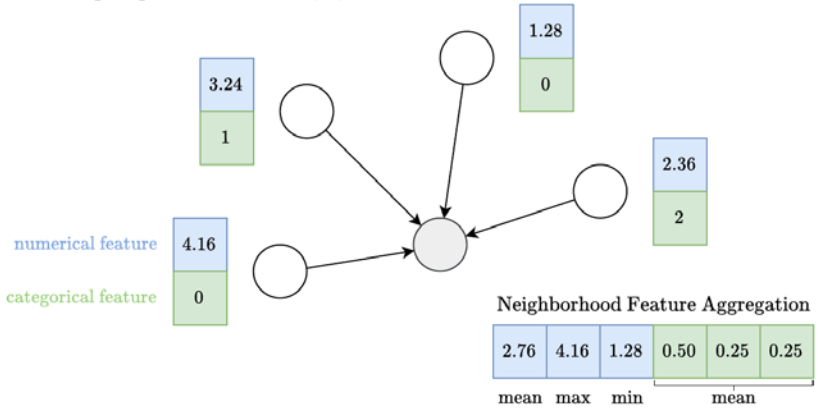

Рис. C.1. Пример применения агрегации признаков соседей (NFA).
Fig. C.1. Example of applying neighborhood feature aggregation (NFA).

Каждая модель обучается 10 раз с различными случайными начальными значениями для вычисления среднего и стандартного отклонения качества; для наибольших наборов данных pokec-regions, web-traffic, web-fraud, web-topics – 5 раз. Все GNN обучаются в полнопакетном режиме без использования методов сэмплирования подграфов. Базовая модель ResMLP реализована в том же коде и также обучается в полнопакетном режиме. Все эксперименты выполнялись на GPU NVIDIA Tesla A100 80 ГБ, за исключением моделей GBDT, обученных на процессорах AMD EPYC.

Выбор гиперпараметров принципиально важен для качества как GNN, так и моделей GBDT. Проводился обширный поиск гиперпараметров на валидационной выборке для всех моделей. Для рассматриваемых моделей GBDT выполнялось 100 итераций байесовской оптимизации с помощью Optuna [69]. Распределения гиперпараметров, использовавшиеся для этих моделей, приведены в табл. C.1, табл. C.2 и табл. C.3. Для GNN использовалась иная стратегия поиска гиперпараметров. Обнаружено, что скорость обучения и вероятность отключения нейронов [70] – наиболее важные гиперпараметры для наших реализаций GNN. Поэтому проводился сеточный поиск: скорость обучения выбиралась из $\{3 \times 10^{-5}, 1 \times 10^{-4}, 3 \times 10^{-4}, 1 \times 10^{-3}, 3 \times 10^{-3}\}$, вероятность отключения нейронов – из $\{0, 0.1, 0.2\}$ (наибольшая скорость обучения 3×10^{-3} нередко приводила к проблемам с NaN, однако включалась в поиск, поскольку в предварительных экспериментах показала преимущество для некоторых комбинаций наборов данных и моделей). Наши предварительные эксперименты показали, что качество моделей GNN достаточно стабильно в широком диапазоне архитектурных гиперпараметров (важными оказались только пропускные соединения и нормализация слоёв). Поэтому в итоговых экспериментах мы зафиксировали эти гиперпараметры: число блоков агрегации по соседям – 3, размер скрытого представления – 512. Исключения сделаны для крупнейших наборов данных, где этот размер был уменьшен до 400 для pokec-regions и до 200 для web-traffic, web-fraud и web-topics, чтобы избежать нехватки памяти GPU. Для GNN с механизмом внимания (GAT и GT) мы использовали 4 головы внимания. Во всех экспериментах с GNN применялся оптимизатор Adam [71]. Каждая модель обучалась 1000 шагов, а лучший шаг выбирался по качеству на валидационной выборке.

Предобработка числовых признаков крайне важна для моделей глубокого обучения. В поиск гиперпараметров для ResMLP и GNN мы включили два варианта: стандартизацию (standard

scaling) и квантильное преобразование к стандартному нормальному распределению. Модели GBDT, напротив, не требуют специальной предобработки числовых признаков. Категориальные признаки кодировались с помощью one-hot-кодирования для всех моделей, за исключением LightGBM и CatBoost, которые обрабатывают их напрямую. На задачах регрессии нейронные модели могут выиграть от преобразования целевой переменной, поэтому для ResMLP и GNN в таких задачах мы также добавили в поиск гиперпараметров и стандартизацию целевой переменной наряду с её исходными значениями.

Мы реализовали GNN на PyTorch [72] и DGL [73].

Табл. C.1. Пространство гиперпараметров для поиска с помощью Optuna для модели XGBoost.
Table C.1. Optuna hyperparameter search distributations for XGBoost.

Параметр	Распределение
colsample_bytree	Uniform[0.5, 1.0]
gamma	{0.0, LogUniform[0.001, 100.0]}
lambda	{0.0, LogUniform[0.1, 10.0]}
learning_rate	LogUniform[0.001, 1.0]
max_depth	UniformInt[3, 14]
min_child_weight	LogUniform[0.0001, 100.0]
subsample	Uniform[0.5, 1.0]

Табл. C.2. Пространство гиперпараметров для поиска с помощью Optuna для модели LightGBM.
Table C.2. Optuna hyperparameter search distributations for LightGBM.

Параметр	Распределение
feature_fraction	Uniform[0.5, 1.0]
lambda_l2	{0.0, LogUniform[0.1, 10.0]}
learning_rate	LogUniform[0.001, 1.0]
num_leaves	UniformInt[4, 768]
min_sum_hessian_in_leaf	LogUniform[0.0001, 100.0]
bagging_fraction	Uniform[0.5, 1.0]

Табл. C.3. Пространство гиперпараметров для поиска с помощью Optuna для модели CatBoost.
Table C.3. Optuna hyperparameter search distributations for CatBoost.

Параметр	Распределение
bagging_temperature	Uniform[0.0, 1.0]
depth	UniformInt[3, 14]
l2_leaf_reg	Uniform[0.1, 10.0]
leaf_estimation_iterations	Uniform[1, 10]
learning_rate	LogUniform[0.001, 1.0]

Приложение D. Полные результаты экспериментов

В основном тексте приводятся полные результаты для разбиения RL; для разбиений RH, TH и THI из-за ограничений объёма опускаются результаты для ряда базовых моделей (ResMLP, XGBoost, CatBoost, ResMLP-NFA) и для наборов данных без временно́го разбиения (city-roads-M, city-roads-L, city-reviews, web-traffic). Полные результаты для разбиения RH

приведены в табл. D.1 и D.2, для TH – в табл. D.3 и D.4, для THI – в табл. D.5 и D.6. Во всех таблицах с результатами для каждого набора данных цветом выделяется лучший результат, а также те результаты, у которых среднее отличается от лучшего не более чем на сумму стандартных отклонений двух результатов.

Табл. D.1. Результаты для задач классификации при разбиении RH. Для многоклассовой классификации приводится Accuracy, для бинарной – Average Precision.
Table D.1. Results for classification datasets under the RH setting. Accuracy is reported for multiclass classification datasets and Average Precision is reported for binary classification datasets.

Модель / метод	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
XGBoost	46.68 ± 0.13	5.11 ± 0.01	TLE	49.31 ± 1.05	77.77 ± 0.13	45.34 ± 0.40	16.06 ± 0.21
LightGBM	47.28 ± 0.14	5.10 ± 0.01	TLE	49.92 ± 0.19	77.88 ± 0.13	44.98 ± 0.80	TLE
CatBoost	46.98 ± 0.24	TLE	TLE	50.52 ± 0.19	78.00 ± 0.05	45.50 ± 0.15	TLE
ResMLP-NFA	61.34 ± 0.34	12.24 ± 0.12	MLE	56.77 ± 1.15	79.80 ± 0.07	44.52 ± 0.44	MLE
LightGBM-NFA	67.09 ± 0.15	12.15 ± 0.02	TLE	**62.19 ± 0.35**	**81.63 ± 0.06**	49.26 ± 0.18	TLE
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

Табл.D.2. Результаты для задач регрессии при разбиении RH. Для всех наборов данных приводится значение R^2.
Table D.2. Results for regression datasets under the RH setting. R^2 is reported for regression datasets.

Модель / метод	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
XGBoost	74.49 ± 0.14	29.70 ± 0.05	70.93 ± 0.05	64.62 ± 0.07	13.34 ± 0.02	36.83 ± 0.06	TLE
LightGBM	73.99 ± 0.11	29.60 ± 0.02	70.01 ± 0.19	63.66 ± 0.09	13.37 ± 0.00	36.38 ± 0.08	TLE
CatBoost	74.92 ± 0.14	29.68 ± 0.10	69.32 ± 0.17	61.24 ± 0.11	13.25 ± 0.03	37.47 ± 0.05	TLE
ResMLP-NFA	74.99 ± 0.50	34.93 ± 0.21	65.87 ± 0.31	57.96 ± 0.15	56.97 ± 0.28	54.95 ± 0.55	MLE
LightGBM-NFA	79.78 ± 0.09	35.35 ± 0.04	**72.09 ± 0.07**	**66.24 ± 0.08**	62.14 ± 0.01	60.30 ± 0.07	TLE
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**

Табл. D.3. Результаты для задач классификации при разбиении TH. Для многоклассовой классификации приводится Accuracy, для бинарной – Average Precision.
Table D.3. Results for classification datasets under the TH setting. Accuracy is reported for multiclass classification datasets and Average Precision is reported for binary classification datasets.

Модель / метод	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
XGBoost	31.95 ± 0.15	4.13 ± 0.02	TLE	18.37 ± 6.34	37.48 ± 0.28	TLE
LightGBM	32.08 ± 0.25	4.13 ± 0.01	TLE	15.85 ± 3.89	37.34 ± 0.18	TLE
CatBoost	33.24 ± 0.16	TLE	TLE	13.87 ± 1.55	38.56 ± 0.10	TLE
ResMLP-NFA	45.04 ± 0.26	6.07 ± 0.07	MLE	38.50 ± 2.47	36.13 ± 0.41	MLE
LightGBM-NFA	52.45 ± 0.18	5.54 ± 0.01	TLE	31.81 ± 7.51	39.89 ± 0.37	TLE
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

Табл. D.4. Результаты для задач регрессии при разбиении TH. Для всех наборов данных приводится значение R^2.
Table D.4. Results for regression datasets under the TH setting. R^2 is reported for regression datasets.

Модель / метод	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
XGBoost	63.96 ± 0.36	25.87 ± 0.14	-8.84 ± 0.24	42.45 ± 0.09
LightGBM	63.56 ± 0.39	26.23 ± 0.04	-9.60 ± 0.03	42.87 ± 0.08
CatBoost	62.05 ± 0.68	25.49 ± 0.09	-8.76 ± 0.23	42.83 ± 0.06
ResMLP-NFA	63.70 ± 1.22	38.00 ± 0.28	44.38 ± 1.65	47.17 ± 0.59
LightGBM-NFA	**71.36 ± 0.41**	38.69 ± 0.03	43.60 ± 0.03	52.42 ± 0.06
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

Табл. D.5. Результаты для задач классификации при разбиении THI. Для многоклассовой классификации приводится Accuracy, для бинарной – Average Precision.
Table D.5. Results for classification datasets under the THI setting. Accuracy is reported for multiclass classification datasets and Average Precision is reported for binary classification datasets.

Модель / метод	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
XGBoost	31.95 ± 0.15	4.13 ± 0.02	TLE	18.37 ± 6.34	37.48 ± 0.28	TLE
LightGBM	32.08 ± 0.25	4.13 ± 0.01	TLE	15.85 ± 3.89	37.34 ± 0.18	TLE
CatBoost	33.24 ± 0.16	TLE	TLE	13.87 ± 1.55	38.56 ± 0.10	TLE
ResMLP-NFA	44.99 ± 0.86	4.95 ± 0.09	MLE	43.43 ± 2.04	36.36 ± 0.49	MLE
LightGBM-NFA	47.46 ± 0.28	4.58 ± 0.02	TLE	**45.45 ± 0.82**	39.91 ± 0.25	TLE
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

Табл. D.6. Результаты для задач регрессии при разбиении THI. Для всех наборов данных приводится значение R^2.
Table D.6. Results for regression datasets under the THI setting. R^2 is reported for regression datasets.

Модель / метод	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
XGBoost	63.96 ± 0.36	25.87 ± 0.14	-8.84 ± 0.24	42.45 ± 0.09
LightGBM	63.56 ± 0.39	26.23 ± 0.04	-9.60 ± 0.03	42.87 ± 0.08
CatBoost	62.05 ± 0.68	25.49 ± 0.09	-8.76 ± 0.23	42.83 ± 0.06
ResMLP-NFA	65.66 ± 1.04	35.81 ± 0.56	36.98 ± 1.44	48.26 ± 0.82
LightGBM-NFA	**68.88 ± 0.11**	33.04 ± 0.17	24.81 ± 0.11	51.95 ± 0.05
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