# RELBENCH V2: A LARGE-SCALE BENCHMARK AND REPOSITORY FOR RELATIONAL DATA

**Justin Gu**[1], **Rishabh Ranjan**[1], **Charilaos Kanatsoulis**[1], **Haiming Tang**[2], **Martin Jurkovic**[3], **Valter Hudovernik**[4], **Mark Znidar**[5], **Pranshu Chaturvedi**[1], **Parth Shroff**[1], **Fengyu Li**[1], **Jure Leskovec**[1]
[1]Stanford University, [2]National University of Singapore, [3]University of Ljubljana, [4]Kumo AI, [5]University of Oxford
{justingu,ranjanr,jure}@stanford.edu
Website: https://relbench.stanford.edu

## ABSTRACT

Relational deep learning (RDL) has emerged as a powerful paradigm for learning directly on relational databases by modeling entities and their relationships across multiple interconnected tables. As this paradigm evolves toward larger models and relational foundation models, scalable and realistic benchmarks are essential for enabling systematic evaluation and progress. In this paper, we introduce RELBENCH v2, a major expansion of the RELBENCH benchmark for RDL. RELBENCH v2 adds four large-scale relational datasets spanning scholarly publications, enterprise resource planning, consumer platforms, and clinical records, contributing over 22 million rows across 29 tables and growing the benchmark to 11 datasets. We further introduce autocomplete tasks, a new class of predictive objectives that require models to infer missing attribute values directly within relational tables while respecting temporal constraints, expanding beyond traditional forecasting tasks constructed via SQL queries. In addition, RELBENCH v2 expands beyond its native datasets by integrating external benchmarks and evaluation frameworks: we translate event streams from the Temporal Graph Benchmark into relational schemas for unified relational–temporal evaluation, interface with ReDeLEx to provide uniform access to 70+ real-world databases suitable for pretraining, and incorporate 4DBInfer datasets and tasks to broaden multi-table evaluation coverage. Experimental results demonstrate that RDL models consistently outperform single-table baselines across autocomplete, forecasting, and recommendation tasks, highlighting the importance of modeling relational structure explicitly.

## 1 INTRODUCTION

Relational databases are the primary storage abstraction for structured data across enterprise, scientific, and healthcare systems. They organize information across multiple tables interconnected via primary–foreign key relationships, capturing rich structural and temporal dependencies between entities. Forecasting tasks such as predicting customer churn, estimating sales, recommending products, and anticipating system or patient outcomes are central to real-world decision making and have driven significant interest in applying machine learning to relational data.

Relational deep learning (RDL) (Robinson et al., 2024; Fey et al., 2024) has emerged as a powerful paradigm for learning directly on relational databases, reducing the human effort and engineering complexity of traditional machine learning pipelines that rely on manually flattening relational schemas into single tables through feature engineering and aggregation. Instead, RDL treats relational databases as heterogeneous graphs and applies graph neural networks (Robinson et al., 2024; Chen et al., 2025) and other relational representation learning architectures (Dwivedi et al., 2026; 2025) to model entities and their relationships directly. RDL models have demonstrated state-of-the-art performance in forecasting tasks over relational data.

At the same time, the machine learning landscape has shifted toward foundation models, which are pretrained on large and diverse datasets to learn transferable representations. This paradigm has

transformed natural language processing and computer vision, and has recently extended to tabular data, where pretrained models achieve strong performance across predictive tasks (Hollmann et al., 2023; 2025; Qu et al., 2025). However, these approaches focus on single-table data and do not capture the multi-table structure of relational databases. To address this limitation, recent efforts (Fey et al., 2025; Wang et al., 2025; Ranjan et al., 2026) develop foundation models for relational databases, leveraging RDL architectures to model entities and their interdependencies across tables. Given their ubiquity and structural richness, relational databases represent a natural next frontier for foundation models, motivating benchmarks that capture realistic relational structure and predictive tasks.

To support machine learning research for forecasting on relational databases, Robinson et al. (2024) introduced RELBENCH, the first benchmark for RDL. RELBENCH v1 provided a collection of real-world relational databases along with forecasting tasks that require predicting future outcomes such as entity churn, sales, or recommendations. These tasks enabled standardized evaluation of relational learning models and demonstrated the effectiveness of RDL compared to traditional approaches.

In this work, we introduce RELBENCH v2, a significant expansion of the benchmark that presents new datasets, new task types, and a new paradigm for relational prediction. RELBENCH v2 adds four new large-scale relational datasets spanning diverse domains, increasing the total number of datasets to eleven. These include `rel-arxiv`, a scholarly publication database capturing papers, authors, categories, and citation relationships; `rel-salt`, an enterprise resource planning dataset modeling sales orders and business workflows; `rel-ratebeer`, a consumer platform dataset containing user interactions, product information, and reviews; and `rel-mimic`, a clinical dataset derived from electronic health records. Together, these datasets introduce new relational structures, domains, and predictive challenges, and collectively contain over 22 million rows across 29 tables.

In addition to expanding the datasets, RELBENCH v2 introduces new predictive tasks and extends the benchmark with multiple strategic integrations of diverse external benchmarks and diagnostic frameworks. Most notably, RELBENCH v2 introduces autocomplete tasks, where the objective is to predict values of existing columns in relational tables at a given timestamp, requiring models to infer missing values from relational and temporal context while preventing information leakage. In addition, as subsequent efforts following RELBENCH v1 have expanded the scale and diversity of relational benchmarks, we integrate several of these into RELBENCH v2. These include temporal interaction datasets from the Temporal Graph Benchmark (TGB) (Rossi et al., 2020), widespread RDL evaluation on 70+ relational databases via ReDeLEx (Peleška & Šír, 2025b), and the 4D design space for graph-centric relational modeling from 4DBInfer (Wang et al., 2024). Additional discussion on RDL and relational foundation models can be found in Appendix A.

## 2 OVERVIEW AND DESIGN

RELBENCH v1 (Robinson et al., 2024) laid the framework for creating a benchmark for deep learning on relational databases. The benchmark consists of two key components: a collection of diverse real-world relational databases, and each database's corresponding set of realistic predictive tasks.

- **Relational databases**, consisting of a set of tables connected via primary–foreign key relationships. Tables store diverse information about entities, and some include time columns indicating when rows are created (e.g., transaction date). Each database is associated with fixed VAL_TIMESTAMP and TEST_TIMESTAMP cutoffs: models are trained on data up to VAL_TIMESTAMP, validated on rows between VAL_TIMESTAMP and TEST_TIMESTAMP, and tested on rows after TEST_TIMESTAMP. Data beyond the test cutoff is hidden during inference to prevent test-time leakage (Kapoor & Narayanan, 2023), using the temporal neighbor sampling strategy of Fey et al. (2024).
- **Predictive tasks** are defined per database via a training table (Fey et al., 2024). Each training table specifies an entity ID, a seed time, and target labels. The seed time determines when the prediction is made and filters out future information. Importantly, the VAL_TIMESTAMP and TEST_TIMESTAMP cutoffs are shared across all tasks within a dataset, enabling multi-task learning and pre-training across predictive tasks defined on the same relational database.

**Autocomplete tasks**: RELBENCH v2 introduces autocomplete tasks, a new paradigm of predictive tasks. These tasks allow for making predictions on the existing columns within tables in the dataset,

Table 1: **Statistics of new RELBENCH datasets.** Datasets vary significantly in the number of tables, total number of rows, and number of columns. In this table, we only count rows available for test inference, i.e., rows up to the test time cutoff. Note: `rel-mimic` shifts dates for patient privacy.

| Name | Domain | #Tasks | Tables | | | Timestamp (year-mon-day) | | |
|---|---|---|---|---|---|---|---|---|
| | | | #Tables | #Rows | #Cols | Start | Val | Test |
| `rel-salt` | Enterprise | 8 | 4 | 4,257,145 | 31 | 2018-01-02 | 2020-02-01 | 2020-07-01 |
| `rel-arxiv` | Academic | 4 | 6 | 2,146,112 | 21 | 2018-01-01 | 2022-01-01 | 2023-01-01 |
| `rel-ratebeer` | Consumer | 8 | 13 | 13,787,005 | 221 | 2000-04-02 | 2018-09-01 | 2020-01-01 |
| `rel-mimic` | Medical | 1 | 6 | 2,424,751 | 54 | 2110-01-14 | 2168-09-06 | 2180-10-26 |
| Total | | 21 | 29 | 22,615,013 | 327 | / | / | / |

as opposed to the previous **forecasting tasks** that predict on target labels constructed via SQL queries. However, like forecasting tasks, autocomplete tasks are still temporal in nature. Autocomplete tasks can be thought of as adding a new row to the database, filling some columns, and then trying to predict the remaining columns without access to future data. For each autocomplete task, we define a fixed set of observed columns and a target column to predict, where models receive the observed values for a row at a given seed time and must infer the target using only relational and temporal context available up to that time.

Autocomplete tasks expand the utility of the benchmark and widen the scope of real-world RDL applications; successfully predicting on autocomplete tasks requires models to deeply understand the relational context of the data. They have many real-world applications. In fact, these tasks were inspired by the SALT (Klein et al., 2024) sales order autocomplete task (see Figure 5), where the SAP S/4HANA Sales Order user interface recommends a payment category based on answers to other data fields and contextual knowledge from the relational schema.

A key consideration for autocomplete tasks involves preventing information leakage, as some columns in a table may be highly correlated. Therefore, designing autocomplete tasks requires dropping the columns that correlate with the specified target column. To prevent information leakage, we manually identify and remove columns that are highly correlated with the target column based on domain knowledge and inspection of the training data. For instance, in the `review-rating` task for the `rel-amazon` dataset, we must drop the 'review_text' column, which otherwise would provide intertwined signals with the review ratings. This prevents such interdependent columns from guiding predictions on the target column, hence preserving the authenticity of each predictive task and maintaining its real-world applicability.

**RDL implementation**: RELBENCH v2 utilizes the same RDL framework defined by Robinson et al. (2024). To reiterate, we first encode raw row-level data into initial node embeddings via PyTorch Frame (Hu et al., 2024), specifically with the ResNet tabular model (Gorishniy et al., 2021). We perform temporal-aware subgraph sampling (Fey et al., 2024) around each entity node at a given seed time, where the embeddings are passed into a heterogeneous GraphSAGE model (Hamilton et al., 2017; Fey & Lenssen, 2019) with sum-based neighbor aggregation to iteratively update node embeddings. Finally, task-specific prediction heads turn output embeddings into predictions.

The rest of this paper is organized as follows. Section 3 describes the new RELBENCH relational databases. Sections 4 and 5 introduce the new autocomplete and forecasting predictive tasks for each RELBENCH dataset, including results from benchmarking our RDL implementation against baselines. Finally, Section 6 describes the expansion of the RELBENCH ecosystem through the integration of external datasets and multi-dimensional RDL benchmarking tools.

## 3 RELBENCH DATASETS

In RELBENCH v2, we introduce four new datasets, bringing the total number of datasets in the benchmark to eleven. These new datasets expand RELBENCH into new domains such as scholarly citations and enterprise operations, widening the breadth of data the benchmark covers and strengthening its position as a core benchmark for foundation models in RDL. Each dataset's predictive autocomplete and forecasting tasks are explained in more detail in Sections 4 and 5, respectively. Detailed statistics for the new datasets can be found in Table 1.

### 3.1 REL-ARXIV

The arXiv-physics dataset (Tang et al., 2024) is a large-scale relational benchmark of over 222,000 research papers published between 2018 and 2023, designed to expand RELBENCH into the domain of scholarly network analysis. It captures the complex evolution of scientific research through 1.5 million directed citation links, paper-author relationships mapped via unique ORCID identifiers, and a hierarchical taxonomy of 53 physics categories. By modeling these dense many-to-many relations between 143,000 authors and their respective research areas, the dataset provides a rich structure for evaluating RDL models within the academic citation ecosystem.

### 3.2 REL-SALT

The Sales Autocompletion Linked Business Tables (SALT) (Klein et al., 2024) database, released by SAP AI Research, provides an authentic relational dataset of end-to-end business transactions from an enterprise resource planning (ERP) system. Centered on sales document headers and line items linked to customer and address master data, SALT models internal enterprise workflows including sales offices, shipping points, and payment terms. Predictive tasks focus on multiclass classification of operational variables in real-world supply chain and order fulfillment settings. By contributing minimally-processed industry data, SALT offers a unique business perspective to RELBENCH.

### 3.3 REL-RATEBEER

The RateBeer dataset provides over two decades of user interactions across distinct tables for beers, places, users, and brewers. Linked through well-defined foreign keys, these attribute-rich tables contain over 30 columns of multi-modal features—including text, categorical, and temporal data—while interaction tables offer granular feedback through multi-aspect sub-scores and textual reviews. `rel-ratebeer` contributes a dataset with strong potential for capturing user preferences in multiple ways; by mapping users to beers, the dataset provides powerful signals for modeling preferences via both explicit rating scores and implicit "Favorites" lists.

### 3.4 REL-MIMIC

The Medical Information Mart for Intensive Care IV (MIMIC-IV) (Johnson et al., 2024) is a large, deidentified electronic health record (EHR) dataset containing clinical data from patients at the Beth Israel Deaconess Medical Center. Designed to support clinical research and machine learning, the RELBENCH implementation allows for deep customization, including parameters to limit patient or table counts, drop specific columns, and filter by features like age. While the standard RELBENCH download utilizes a subset of 20,000 patients, the dataset also supports integration with Google BigQuery for accessing the full MIMIC-IV v3.1 data. Due to the sensitive nature of real-world medical data, users must obtain proper credentials through PhysioNet to access the dataset. MIMIC-IV also applies patient-level date shifting for privacy, offsetting all timestamps by a random amount so dates appear in the 2110–2200 range rather than the original 2008–2022 collection period.

## 4 AUTOCOMPLETE TASKS

RELBENCH v2 introduces 23 autocomplete tasks for both existing and new datasets, grouped into two task types: autocomplete classification (Section /refsec:autocomplete-classification) and autocomplete regression (Section 4.2). Tasks are named based on the table and column used as the target labels for the predictions. A full list of autocomplete tasks is given in Table 2, with high-level descriptions given in Appendix C.

### 4.1 AUTOCOMPLETE CLASSIFICATION

Autocomplete classification tasks aim to predict labels for existing categorical columns in a dataset. Because categorical data often comes in both binary and multiclass situations, autocomplete tasks support both binary classification and multiclass classification. For binary classification, we use the ROC-AUC (Hanley & McNeil, 1983) metric for evaluation, where higher scores are better. For multiclass classification, we use accuracy as the evaluation metric, where higher is also better. As a

Table 2: **Full list of new autocomplete tasks.** Autocomplete tasks aim to make predictions on existing columns in the dataset.

| Dataset | Task name | Task type | #Rows of training table | | | #Unique Entities | %train/test Entity Overlap | #Dst Entities |
|---|---|---|---|---|---|---|---|---|
| | | | Train | Validation | Test | | | |
| rel-amazon | review-rating | auto-reg | 11,822,796 | 806,355 | 8,217,532 | 17,255,399 | 40.5 | – |
| rel-avito | searchstream-click | auto-bcls | 2,212,750 | 1,177,380 | 924,990 | 3,976,413 | 23.8 | – |
| | searchinfo-isuserloggedon | auto-bcls | 1,291,566 | 695,590 | 592,133 | 2,579,289 | 0.0 | – |
| rel-event | event_interest-interested | auto-bcls | 14,442 | 536 | 420 | 14,992 | 94.5 | – |
| | event_interest-not_interested | auto-bcls | 14,442 | 536 | 420 | 14,992 | 94.5 | – |
| | users-birthyear | auto-reg | 33,937 | 1,731 | 1,002 | 36,670 | 0.0 | – |
| rel-f1 | results-position | auto-reg | 8,997 | 1,400 | 4,798 | 15,195 | 0.0 | – |
| | qualifying-position | auto-reg | 2,228 | 1,854 | 5,733 | 9,815 | 0.0 | – |
| rel-hm | transactions-price | auto-reg | 14,844,291 | 235,662 | 266,364 | 15,346,317 | 0.0 | – |
| rel-ratebeer | beer_ratings-total_score | auto-reg | 10,620,177 | 1,227,702 | 2,495,360 | 14,343,239 | 0.0 | – |
| rel-salt | item-plant | auto-mcls | 1,622,787 | 293,823 | 400,206 | 2,316,816 | 0.0 | – |
| | item-shippoint | auto-mcls | 1,622,787 | 293,780 | 398,536 | 2,315,103 | 0.0 | – |
| | item-incoterms | auto-mcls | 1,622,787 | 293,891 | 402,835 | 2,319,513 | 0.0 | – |
| | sales-office | auto-mcls | 340,491 | 71,474 | 88,942 | 500,907 | 0.0 | – |
| | sales-group | auto-mcls | 340,491 | 70,224 | 83,193 | 493,908 | 0.0 | – |
| | sales-payterms | auto-mcls | 340,491 | 71,472 | 88,831 | 500,794 | 0.0 | – |
| | sales-shipcond | auto-mcls | 340,491 | 71,398 | 88,422 | 500,311 | 0.0 | – |
| | sales-incoterms | auto-mcls | 340,491 | 71,470 | 88,925 | 500,886 | 0.0 | – |
| rel-stack | badges-class | auto-mcls | 448,358 | 15,105 | 127,370 | 590,833 | 0.0 | – |
| rel-trial | studies-enrollment | auto-reg | 233,072 | 14,470 | 23,430 | 270,972 | 0.0 | – |
| | studies-has_dmc | auto-bcls | 202,840 | 11,983 | 18,944 | 233,767 | 0.0 | – |
| | eligibilities-adult | auto-bcls | 234,366 | 14,470 | 23,430 | 272,266 | 0.0 | – |
| | eligibilities-child | auto-bcls | 234,366 | 14,470 | 23,430 | 272,266 | 0.0 | – |

baseline to compare our heterogeneous GraphSAGE model against, we utilize a LightGBM classifier baseline over the raw entity table features.

**Experimental results**. Classification results for the new autocomplete tasks are given in Table 3 for binary classification and Table 4 for multiclass classification. In both types of classification, RDL strongly outperforms the LightGBM baseline in all cases. On some tasks, such as `item-shippoint` from `rel-salt` or `badges-class` from `rel-stack`, RDL's high accuracy indicates that relational context gives highly informative signal to predict missing attributes.

For other tasks such as `sales-office` from `rel-salt`, comparison with the majority-class baseline suggests a strong class imbalance in the target labels, with the majority baseline already achieving very high accuracy. Despite this, the GNN matches or slightly improves upon the majority baseline while maintaining strong performance on other tasks, whereas LightGBM shows unstable behavior and substantially worse test performance, suggesting limited generalization. Overall, these results highlight the ability of RDL to exploit relational structure for autocomplete tasks, even in the presence of class imbalance or sparse feature information.

### 4.2 AUTOCOMPLETE REGRESSION

Autocomplete regression tasks involve predicting numerical labels of an entity at a given seed time. For our evaluation metric, we use $R^2$, where higher values are better. We compare our RDL approach against four baselines. Global zero predicts zero for all entities. Global mean/median calculates the global mean/median label value for the target column in the training data and predicts that mean/median for every entity. Entity mean/median calculates the mean/median value with respect to each entity, and predicts that mean/median for the entity. LightGBM joins the entity table with the task table to get raw features from both, then learns a LightGBM (Ke et al., 2017) regressor over those raw features to predict numerical targets.

**Experimental results**. Autocomplete regression results are reported in Table 5. Additionally, MAE metrics for autocomplete regression tasks are reported in Table 15 in Appendix D. Across most tasks, RDL achieves higher $R^2$ values, outperforming both feature-based and aggregation baselines, indicating that capturing relational context improves explanatory power.

Table 3: **Autocomplete binary classification results on RELBENCH.** Binary classification uses the AUROC metric (higher is better). Best values are in bold. Standard baselines of random choice and majority class both correspond to AUROC values of approximately 50.00, so we exclude them below. See Table 12 for standard deviations.

| Dataset | Task | Split | LightGBM | GNN |
|---|---|---|---|---|
| rel-avito | searchinfo-isuserloggedon | Val | 59.09 | **82.57** |
| | | Test | 50.00 | **73.00** |
| | searchstream-click | Val | **68.33** | 50.39 |
| | | Test | 49.92 | **55.92** |
| rel-event | event_interest-interested | Val | 51.25 | **54.16** |
| | | Test | 49.57 | 47.64 |
| | event_interest-not_interested | Val | 51.98 | 49.74 |
| | | Test | 52.88 | **60.40** |
| rel-trial | eligibilities-adult | Val | 58.10 | **94.91** |
| | | Test | 50.00 | **93.73** |
| | eligibilities-child | Val | 59.78 | **85.91** |
| | | Test | 50.00 | **87.25** |
| | studies-has_dmc | Val | 76.47 | **78.21** |
| | | Test | 50.00 | **75.72** |
| Average | | Val | 60.71 | **70.84** |
| | | Test | 50.34 | **70.52** |

Table 4: **Autocomplete multiclass classification results on RELBENCH.** Multiclass classification uses the accuracy metric (higher is better). Best values are in bold. See Table 13 for std. devs.

| Dataset | Task | Split | Random | Majority | LightGBM | GNN |
|---|---|---|---|---|---|---|
| rel-salt | item-incoterms | Val | 34.49 | 66.46 | 66.43 | **80.23** |
| | | Test | 30.33 | 58.05 | 58.05 | **69.36** |
| | item-plant | Val | 33.19 | 60.95 | 60.97 | **99.70** |
| | | Test | 32.38 | 59.69 | 59.69 | **99.46** |
| | item-shippoint | Val | 8.20 | 2.34 | 4.72 | **98.54** |
| | | Test | 6.53 | 1.99 | 5.67 | **98.39** |
| | sales-group | Val | 0.90 | 0.86 | 0.70 | **18.43** |
| | | Test | 0.85 | 0.75 | 0.94 | **15.76** |
| | sales-incoterms | Val | 31.83 | 61.00 | 60.53 | **69.07** |
| | | Test | 29.39 | 56.63 | 56.63 | **62.23** |
| | sales-office | Val | 50.01 | **99.91** | 99.90 | 99.91 |
| | | Test | 49.71 | **99.88** | 59.93 | 99.88 |
| | sales-payterms | Val | 0.32 | 0.65 | 1.85 | **39.88** |
| | | Test | 0.24 | 0.47 | 5.64 | **37.47** |
| | sales-shipcond | Val | 16.49 | 27.61 | 31.92 | **59.21** |
| | | Test | 15.64 | 26.30 | 4.91 | **56.85** |
| rel-stack | badges-class | Val | 11.59 | 20.68 | 1.93 | **79.97** |
| | | Test | 10.49 | 18.34 | 2.51 | **82.83** |
| Average | | Val | 20.78 | 37.83 | 36.55 | **71.66** |
| | | Test | 19.51 | 35.79 | 28.22 | **69.14** |

## 5 NEW FORECASTING TASKS

Forecasting tasks were the original predictive tasks defined in RELBENCH v1, and version 2 introduces 13 new ones. Split into entity classification, entity regression, and recommendation (link prediction), these tasks use SQL queries to construct new task target columns. While classification and regression are entity-level tasks, recommendation tasks aim to predict the next temporal links between two sets of entities for a given link type, such as whether users will purchase a certain product.

We define forecasting tasks for the new datasets `rel-arxiv`, `rel-ratebeer`, and `rel-mimic`, and we add a new recommendation task for `rel-f1`. New forecasting tasks are shown in Table 6.

### 5.1 ENTITY CLASSIFICATION

In RELBENCH v1, all entity-level classification tasks were binary classification. In RELBENCH v2, we broaden to the multiclass case with the first entity multiclass classification task, `rel-arxiv`'s `author-category` task. For entity-level forecasting tasks, both binary and multiclass classification are evaluated with the same metrics as their autocomplete counterparts, with entity binary

Table 5: **Autocomplete regression results on RELBENCH.** Regression uses the $R^2$ metric (higher is better). Best values are in bold. See Table 14 for standard deviations and Table 15 for MAE results.

| Dataset | Task | Split | Zero | Mean | Median | Ent. Mean | Ent. Med. | LightGBM | GNN |
|---|---|---|---|---|---|---|---|---|---|
| rel-amazon | review-rating | Val | −20.848 | **−0.006** | −0.364 | −20.848 | −20.848 | −0.364 | −0.356 |
| | | Test | −22.579 | **−0.014** | −0.341 | −13.313 | −13.313 | −0.341 | −0.331 |
| rel-event | users-birthyear | Val | −55012.323 | −0.047 | −0.216 | −55012.323 | −55012.323 | 0.004 | **0.008** |
| | | Test | −64803.758 | −0.121 | −0.395 | −64803.758 | −64803.758 | −0.192 | **−0.030** |
| rel-f1 | qualifying-position | Val | −3.267 | −0.018 | −0.030 | −3.267 | −3.267 | **0.153** | 0.015 |
| | | Test | −3.214 | −0.002 | −0.001 | −3.214 | −3.214 | −0.953 | **0.015** |
| | results-position | Val | −3.123 | −0.100 | −0.107 | −3.123 | −3.123 | 0.283 | **0.440** |
| | | Test | −3.148 | −0.176 | −0.219 | −3.148 | −3.148 | −2.437 | **0.394** |
| rel-hm | transactions-price | Val | −2.215 | −0.065 | −0.140 | −2.215 | −2.215 | −0.140 | **0.725** |
| | | Test | −2.329 | −0.075 | −0.159 | −2.329 | −2.329 | −0.160 | **0.736** |
| rel-ratebeer | beer_ratings-total_score | Val | −23.411 | −0.015 | −0.004 | −23.411 | −23.411 | −0.004 | **0.448** |
| | | Test | −34.352 | −0.031 | −0.003 | −34.352 | −34.352 | −0.014 | **0.394** |
| rel-trial | studies-enrollment | Val | −0.001 | **−0.000** | −0.001 | −0.001 | −0.001 | −0.001 | −0.000 |
| | | Test | −0.000 | **−0.000** | −0.000 | −0.000 | −0.000 | −0.000 | −0.000 |

Table 6: **Full list of new forecasting tasks**. Forecasting tasks make predictions on new target columns created using SQL queries.

| Dataset | Task name | Task type | #Rows of training table | | | #Unique Entities | %train/test Entity Overlap | #Dst Entities |
|---|---|---|---|---|---|---|---|---|
| | | | Train | Validation | Test | | | |
| rel-arxiv | paper-citation | entity-bcls | 534,233 | 155,845 | 193,696 | 136,183 | 70.31 | – |
| | author-category | entity-mcls | 210,769 | 39,015 | 39,655 | 126,219 | 62.15 | – |
| | author-publication | entity-reg | 210,769 | 39,015 | 39,655 | 101,886 | 62.15 | – |
| | paper-paper-cocitation | recommendation | 246,341 | 71,257 | 82,033 | 94,289 | 60.57 | 138,688 |
| rel-f1 | driver-circuit-compete | recommendation | 2,649 | 27 | 27 | 786 | 40.74 | 19,044 |
| rel-mimic | patient-iculengthofstay | entity-bcls | 13,816 | 2,699 | 2,445 | 13,816 | 0.00 | – |
| rel-ratebeer | beer-churn | entity-bcls | 2,470,686 | 92,367 | 79,927 | 516,368 | 45.46 | – |
| | user-churn | entity-bcls | 373,709 | 19,908 | 9,392 | 154,071 | 35.21 | – |
| | brewer-dormant | entity-bcls | 98,697 | 15,840 | 16,366 | 28,333 | 64.07 | – |
| | user-count | entity-reg | 373,709 | 19,908 | 9,392 | 154,071 | 35.21 | – |
| | user-beer-favorite | recommendation | 1,099 | 1,043 | 499 | 2,296 | 10.82 | 7,745 |
| | user-beer-liked | recommendation | 150,322 | 5,681 | 2,783 | 35,010 | 58.53 | 170,964 |
| | user-place-liked | recommendation | 38,444 | 547 | 351 | 7,425 | 81.77 | 46,814 |

classification using ROC-AUC (Hanley & McNeil, 1983) and multiclass classification using accuracy (for both, higher is better). We again compare to a LightGBM classifier baseline over the raw entity table features, but here only information from the single entity table is used.

**Experimental results**. Entity classification results for the new forecasting tasks are given in Table 7 for binary classification and Table 8 for multiclass classification, with RDL outperforming the LightGBM baseline in all cases. Notably, RDL vastly outperforms LightGBM on the multiclass `author-category` task, where predicting an author's research area benefits from aggregating relational signals from coauthorship, citation patterns, and publication context. This suggests that as classification tasks become more complex, leveraging relational context becomes more important.

### 5.2 ENTITY REGRESSION

Entity-level regression tasks involve predicting numerical labels of an entity at a given seed time. Like for autocomplete regression, our evaluation metric is $R^2$, where higher values are better. We compare our RDL approach against essentially the same baselines as described in Section 4.2. The only modification is that for the LightGBM baseline for entity regression, only the raw features from the single entity table are used to predict the numerical targets.

**Experimental results**. The entity regression results in Table 9 show our RDL implementation outperforms the baselines across all new forecasting regression tasks. RDL achieves higher $R^2$ values, indicating improved explanatory power when relational information is incorporated. Additionally, MAE metrics are reported in Appendix D (Table 19), where RDL consistently achieves lower errors. These results suggest that relational modeling provides consistent benefits for numeric prediction at the entity level, even when the target variable is defined on a single entity table.

Table 7: **Entity binary classification results on RELBENCH.** Binary classification uses the AUROC metric (higher is better). Best values are in bold. Standard baselines of random choice and majority class both correspond to AUROC values of approximately 50.00, so we exclude them below. See Table 16 for standard deviations.

| Dataset | Task | Split | LightGBM | GNN |
|---|---|---|---|---|
| rel-arxiv | paper-citation | Val | 71.94 | **82.45** |
| | | Test | 71.21 | **82.50** |
| rel-mimic | patient-iculengthofstay | Val | 53.64 | **56.52** |
| | | Test | 51.81 | **55.01** |
| rel-ratebeer | beer-churn | Val | 81.90 | **90.47** |
| | | Test | 76.21 | **78.67** |
| | brewer-dormant | Val | 76.39 | **82.10** |
| | | Test | 75.79 | **80.51** |
| | user-churn | Val | 87.02 | **96.85** |
| | | Test | 83.92 | **94.27** |
| Average | | Val | 74.18 | **81.68** |
| | | Test | 71.79 | **78.19** |

Table 8: **Entity multiclass classification results on RELBENCH.** Multiclass classification uses the accuracy metric (higher is better). Best values are in bold. See Table 17 for standard deviations.

| Dataset | Task | Split | Random | Majority | LightGBM | GNN |
|---|---|---|---|---|---|---|
| rel-arxiv | author-category | Val | 1.75 | 8.83 | 1.95 | **52.63** |
| | | Test | 1.77 | 9.09 | 2.01 | **50.74** |

### 5.3 RECOMMENDATION

Recommendation tasks involve predicting, for each source entity and seed time, a ranked list of the top-$K$ target entities. This requires calculating pairwise scores between source and target entities.

We evaluate two GNN-based models. In GraphSAGE (Hamilton et al., 2017), source and target embeddings are learned via message passing, and pairwise scores are computed using their inner product, with training performed using the Bayesian Personalized Ranking loss (Rendle et al., 2012). In ID-GNN (You et al., 2021), target embeddings are passed through a source-specific MLP prediction head to produce pairwise scores, and the model is trained with cross-entropy loss (You et al., 2021).

We report Mean Average Precision (MAP) @$K$ (higher is better), with $K$ set per task. Baselines include Past Visit (ranks targets prior per-entity visit frequency), Global Popularity (ranks targets by overall frequency in the training data), and LightGBM (Ke et al., 2017) (predicts source–target links using concatenated entity features, augmented with popularity and past-visit rank features).

**Experimental results**. Results are given in Table 10. In general, we observe that either the RDL implementation using GraphSAGE (Hamilton et al., 2017), or ID-GNN (You et al., 2021) as the GNN component performs best, often by a very significant margin. ID-GNN excels in settings where predictions are highly entity-specific, whereas the plain GNN performs better when such specificity is less critical. This behavior reflects the inductive biases of the two models: GraphSAGE primarily captures structural and neighborhood-based patterns, while ID-GNN explicitly incorporates node identity information. Additionally, increasing the number of layers in the RDL models from two layers to four tended to yield improved performance across both GNN-based models, although the four-layer GraphSAGE model encountered CUDA memory errors on an 80GB Nvidia A100. These results highlight the multi-hop nature of recommendation tasks.

## 6 INTEGRATING EXTERNAL BENCHMARKS INTO RELBENCH

RELBENCH (Robinson et al., 2024) introduced the first standardized benchmark for forecasting over relational databases, enabling end-to-end evaluation of RDL methods on real-world multi-table datasets. Subsequent efforts have expanded the scale and diversity of relational benchmarks. In addition to the new relational databases and tasks introduced in RELBENCH v2, we also extend RELBENCH with direct integration of external benchmarks and diagnostic frameworks. These include a suite of large-scale *temporal interaction* datasets sourced from the Temporal Graph Benchmark (TGB) (Rossi et al., 2020), widespread evaluation of RDL models on 70+ relational databases via

Table 9: **Entity regression results on RELBENCH.** Regression uses the $R^2$ metric (higher is better). Best values are in bold. See Table 18 for standard deviations and Table 19 for MAE results.

| Dataset | Task | Split | Zero | Mean | Median | Ent. Mean | Ent. Med. | LightGBM | GNN |
|---|---|---|---|---|---|---|---|---|---|
| rel-amazon | item-ltv | Val | −0.025 | −0.000 | −0.013 | −0.247 | −0.107 | 0.002 | **0.066** |
| | | Test | −0.013 | −0.000 | −0.007 | 0.030 | **0.099** | 0.001 | 0.032 |
| | user-ltv | Val | −0.084 | −0.003 | −0.084 | 0.053 | 0.095 | −0.084 | **0.195** |
| | | Test | −0.092 | −0.000 | −0.092 | 0.143 | 0.168 | −0.092 | **0.172** |
| rel-arxiv | author-publication | Val | −1.579 | −0.012 | −0.259 | 0.254 | 0.236 | −0.259 | **0.437** |
| | | Test | −1.572 | −0.000 | −0.210 | −0.010 | −0.064 | −0.210 | **0.249** |
| rel-avito | ad-ctr | Val | −0.238 | −0.002 | −0.095 | −0.224 | −0.224 | −0.032 | **0.030** |
| | | Test | −0.226 | −0.004 | −0.098 | −0.148 | −0.148 | −0.039 | **−0.001** |
| rel-event | user-attendance | Val | −0.249 | **−0.037** | −0.249 | −0.193 | −0.147 | −0.249 | −0.045 |
| | | Test | −0.168 | −0.019 | −0.168 | −0.065 | −0.043 | −0.168 | **0.003** |
| rel-f1 | driver-position | Val | −5.715 | −0.370 | −0.236 | −2.866 | −2.840 | 0.150 | **0.249** |
| | | Test | −5.239 | −0.119 | −0.042 | −2.841 | −2.849 | **0.068** | 0.039 |
| rel-hm | item-sales | Val | −0.017 | −0.000 | −0.017 | 0.065 | 0.053 | −0.017 | **0.187** |
| | | Test | −0.017 | −0.000 | −0.017 | 0.058 | 0.042 | −0.017 | **0.215** |
| rel-ratebeer | user-count | Val | −0.037 | −0.053 | −0.037 | 0.551 | 0.547 | **0.559** | 0.526 |
| | | Test | −0.071 | −0.025 | −0.071 | 0.264 | 0.285 | −0.170 | **0.625** |
| rel-stack | post-votes | Val | −0.028 | −0.007 | −0.028 | **0.306** | 0.285 | −0.028 | 0.122 |
| | | Test | −0.034 | −0.004 | −0.034 | **0.294** | 0.272 | −0.034 | 0.122 |
| rel-trial | site-success | Val | −0.988 | **−0.005** | −0.988 | −0.749 | −0.809 | −0.319 | −0.425 |
| | | Test | −0.923 | **−0.001** | −0.923 | −0.714 | −0.751 | −0.336 | −0.483 |
| | study-adverse | Val | −0.021 | −0.002 | −0.020 | −0.021 | −0.021 | **0.134** | 0.066 |
| | | Test | −0.054 | −0.005 | −0.050 | −0.054 | −0.054 | **0.307** | 0.177 |

Table 10: **Recommendation results on RELBENCH.** Recommendation uses the MAP metric, where higher values are better. Best values are in bold. See Table 20 for standard deviations.

| Dataset | Task | Split | Past Visit | Global Pop. | LightGBM | GNN (2) | GNN (4) | IDGNN (2) | IDGNN (4) |
|---|---|---|---|---|---|---|---|---|---|
| rel-arxiv | paper-paper-cocitation | Val | 19.01 | 1.25 | 12.49 | 12.19 | 12.96 | 25.22 | **35.76** |
| | | Test | 16.51 | 1.13 | 11.01 | 8.83 | 10.46 | 22.95 | **35.39** |
| rel-f1 | driver-circuit-compete | Val | 53.41 | 55.19 | 66.06 | 3.60 | 10.57 | 70.70 | **74.40** |
| | | Test | 20.76 | 50.12 | 57.77 | 9.67 | 16.57 | 62.32 | **76.18** |
| rel-ratebeer | user-beer-favorite | Val | 0.00 | 2.33 | 1.24 | 2.09 | – | 3.09 | **3.33** |
| | | Test | 0.00 | 1.10 | 0.67 | 0.56 | – | 1.21 | **1.89** |
| | user-beer-liked | Val | 0.00 | 0.77 | 0.43 | 0.77 | – | 0.21 | **1.48** |
| | | Test | 0.00 | 0.61 | 0.29 | 0.54 | – | 0.32 | **1.46** |
| | user-place-liked | Val | 0.00 | 0.24 | 0.24 | 1.06 | – | 0.88 | **2.20** |
| | | Test | 0.00 | 0.11 | 0.08 | 1.15 | – | 0.60 | **1.85** |
| | Average | Val | 14.48 | 11.96 | 16.09 | 3.94 | 11.76 | 20.02 | **23.43** |
| | | Test | 7.45 | 10.61 | 13.96 | 4.15 | 13.52 | 17.48 | **23.35** |

ReDeLex (Peleška & Šír, 2025b), and a 4D benchmarking toolbox spanning multiple datasets, tasks, graph construction strategies, and predictive models from 4DBInfer (Wang et al., 2024).

### 6.1 TEMPORAL GRAPH BENCHMARK (TGB)

The Temporal Graph Benchmark (TGB) is a benchmark centered on learning from time-stamped event streams (temporal edges), with evaluation protocols that enforce strict chronological generalization. By translating TGB datasets into the RELBENCH database and task abstraction, we enable direct comparisons between (i) *temporal GNN* baselines that operate on event streams and (ii) *relational deep learning* baselines that operate on a multi-table schema with explicit primary/foreign key structure. Following the principle of normalization in database theory, we translate each node and edge type into its own table. We focus on TGB datasets, excluding knowledge graphs that require additional adjustments, and naturally map the remaining datasets to relational event logs targeting the following downstream tasks: (i) Dynamic Link Property Prediction (`tgbl-*`), (ii) Dynamic Node Property Prediction (`tgbn-*`), and (iii) Temporal Heterogeneous Graph Link Prediction (`thgl-*`).

The converted TGB datasets cover diverse domains and scales, from small bipartite interaction graphs to multi-relational, multi-entity temporal databases with tens of millions of events. Each dataset becomes a RELBENCH `Database` (parquet tables plus schema metadata) together with temporal cutoffs and tasks, enabling training and evaluation under the same leakage-safe conventions used elsewhere in RELBENCH. The dataset statistics can be found in Table 11. Additional details about the RELBENCH TGB datasets and experiments can be found in App. E.

Table 11: **Statistics of TGB datasets translated into RELBENCH.** We report the relational size of each translated dataset as stored in parquet: number of tables, total number of rows (summed across all tables, up to the test-time cutoff), and total number of columns (summed across all tables).

| Task family | Dataset | #Tables | #Rows | #Cols |
|---|---|---|---|---|
| tgbl-* (link) | tgbl-wiki-v2 | 3 | 166,701 | 7 |
| | tgbl-review-v2 | 2 | 5,226,177 | 6 |
| | tgbl-coin | 2 | 23,447,972 | 6 |
| | tgbl-comment | 2 | 45,309,297 | 6 |
| | tgbl-flight | 2 | 67,187,713 | 6 |
| tgbn-* (node) | tgbn-trade | 5 | 934,072 | 14 |
| | tgbn-genre | 5 | 20,858,841 | 14 |
| | tgbn-reddit | 5 | 43,669,153 | 14 |
| | tgbn-token | 5 | 81,663,534 | 14 |
| thgl-* (hetero link) | thgl-software | 18 | 2,171,733 | 74 |
| | thgl-forum | 4 | 23,910,523 | 12 |
| | thgl-github | 18 | 23,356,342 | 74 |
| | thgl-myket | 4 | 55,163,623 | 12 |

### 6.2 RELATIONAL DEEP LEARNING EXPLORATION (REDELEX)

Relational Deep Learning Exploration (ReDeLEx) (Peleška & Šír, 2025b) is a large-scale experimental framework for systematically evaluating Relational Deep Learning (RDL) on real-world relational databases. From the CTU Relational Learning Repository (Motl & Schulte, 2025), ReDeLEx integrates over 70 datasets into a unified pipeline that connects directly to SQL databases, infers attribute semantics, and represents relational schemas as heterogeneous graphs. These datasets span domains such as healthcare, government, education, sports, and business applications, vastly increasing RELBENCH's coverage of data. RDL models in ReDeLEx follow a modular design that combines attribute encoders, optional tabular models, graph neural network layers, and task-specific prediction heads. These models enable controlled comparisons across architectures such as linear GraphSAGE, Tabular ResNet–augmented GNNs, and Transformer-based models.

### 6.3 4DBINFER

4DBInfer (Wang et al., 2024) is a large-scale benchmarking effort focused on predictive modeling over multi-table relational databases. 4DBInfer introduces an explicit four-dimensional evaluation framework: datasets, tasks, relational-to-graph construction strategies, and predictive model families. These are designed to expose how modeling choices across the full pipeline impact performance. While 4DBInfer explores this design space through extensive empirical comparisons, its datasets and task formulations provide a valuable foundation for unified evaluation. As part of RELBENCH v2, we incorporate 7 4DBInfer datasets and 12 tasks, allowing RELBENCH to inherit the scale and diversity of 4DBInfer while enforcing consistent experimental conventions across relational, temporal, and graph-based learning settings.

## 7 CONCLUSION

In this work, we introduced RELBENCH v2, a major expansion of the RELBENCH benchmark for relational deep learning (RDL). RELBENCH v2 adds four large-scale relational datasets spanning academic, enterprise, consumer, and clinical domains, substantially increasing the scale and diversity of real-world relational data. We further introduced autocomplete tasks, a new class of predictive objectives that require models to infer missing attribute values directly within relational tables under temporal constraints, complementing traditional forecasting and recommendation tasks. In addition, we integrated temporal interaction datasets from the Temporal Graph Benchmark (TGB), and relational databases from Relational Deep Learning Exploration (ReDeLEx) and 4DBInfer, enabling unified evaluation across relational and temporal learning settings. Experimental results show that RDL models consistently outperform single-table baselines across autocomplete, forecasting, and recommendation tasks, highlighting the importance of modeling relational structure explicitly. RELBENCH v2 provides a scalable and realistic benchmark to support the development and evaluation of RDL systems and relational foundation models.

## REFERENCES

Tianlang Chen, Charilaos Kanatsoulis, and Jure Leskovec. RelGNN: Composite message passing for relational deep learning. In *International Conference on Machine Learning (ICML)*, volume 267, pp. 8296–8312. PMLR, 2025.

Vijay Prakash Dwivedi, Charilaos Kanatsoulis, Shenyang Huang, and Jure Leskovec. Relational deep learning: Challenges, foundations and next-generation architectures. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, volume 2, pp. 5999–6009. ACM, 2025.

Vijay Prakash Dwivedi, Sri Jaladi, Yangyi Shen, Federico López, Charilaos I Kanatsoulis, Rishi Puri, Matthias Fey, and Jure Leskovec. Relational graph transformer. In *International Conference on Learning Representations (ICLR)*, 2026.

Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. In *ICLR 2019 Workshop on Representation Learning on Graphs and Manifolds*, 2019. URL `https://rlgm.github.io/papers/2.pdf`.

Matthias Fey, Weihua Hu, Kexin Huang, Jan Eric Lenssen, Rishabh Ranjan, Joshua Robinson, Rex Ying, Jiaxuan You, and Jure Leskovec. Position: Relational deep learning-graph representation learning on relational databases. In *International Conference on Machine Learning (ICML)*, volume 235, pp. 13592–13607. PMLR, 2024.

Matthias Fey, Vid Kocijan, Federico Lopez, Jan Eric Lenssen, and Jure Leskovec. KumoRFM: A foundation model for in-context learning on relational data, 2025.

Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 18932–18943, 2021.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

James A Hanley and Barbara J McNeil. A method of comparing the areas under receiver operating characteristic curves derived from the same cases. *Radiology*, 148(3):839–843, 1983.

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

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

Weihua Hu, Yiwen Yuan, Zecheng Zhang, Akihiro Nitta, Kaidi Cao, Vid Kocijan, Jure Leskovec, and Matthias Fey. PyTorch Frame: A modular framework for multi-modal tabular learning, 2024. URL `https://arxiv.org/abs/2404.00776`.

Valter Hudovernik, Minkai Xu, Juntong Shi, Lovro Šubelj, Stefano Ermon, Erik Štrumbelj, and Jure Leskovec. RelDiff: Relational data generative modeling with graph-based diffusion models, 2025. URL `https://arxiv.org/abs/2506.00710`.

Alistair Johnson, Lucas Bulgarelli, Tom Pollard, Brian Gow, Benjamin Moody, Steven Horng, Leo Anthony Celi, and Roger Mark. MIMIC-IV. *PhysioNet*, October 2024. doi: 10.13026/kpb9-mt58. URL `https://doi.org/10.13026/kpb9-mt58`. Version 3.1.

Charilaos Kanatsoulis, Evelyn Choi, Stefanie Jegelka, Jure Leskovec, and Alejandro Ribeiro. Learning efficient positional encodings with graph neural networks. In *International Conference on Learning Representations (ICLR)*, 2025.

Sayash Kapoor and Arvind Narayanan. Leakage and the reproducibility crisis in machine-learning-based science. *Patterns*, 4(9), 2023.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

Mohamed Amine Ketata, David Lüdke, Leo Schwinn, and Stephan Günnemann. Joint relational database generation via graph-conditional diffusion models, 2025. URL https://arxiv.org/abs/2505.16527.

Myung Jun Kim, Leo Grinsztajn, and Gael Varoquaux. CARTE: Pretraining and transfer for tabular learning. In *International Conference on Machine Learning (ICML)*, volume 235, pp. 23843–23866. PMLR, 2024.

Tassilo Klein, Clemens Biehl, Margarida Costa, Andre Sres, Jonas Kolk, and Johannes Hoffart. SALT: Sales autocompletion linked business tables dataset. In *NeurIPS 2024 Third Table Representation Learning Workshop*, 2024. URL https://openreview.net/forum?id=UZbELpkWIr.

Vignesh Kothapalli, Rishabh Ranjan, Valter Hudovernik, Vijay Prakash Dwivedi, Johannes Hoffart, Carlos Guestrin, and Jure Leskovec. PluRel: Synthetic data unlocks scaling laws for relational foundation models, 2026. URL https://arxiv.org/abs/2602.04029.

Jan Motl and Oliver Schulte. The CTU prague relational learning repository, 2025. URL https://arxiv.org/abs/1511.03086.

Jakub Peleška and Gustav Šír. Tabular transformers meet relational databases. *ACM Transactions on Intelligent Systems and Technology*, 16(5), 2025a.

Jakub Peleška and Gustav Šír. REDELEX: A framework for relational deep learning exploration. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECMLPKDD)*, pp. 438–456, 2025b.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICL: A tabular foundation model for in-context learning on large data. In *International Conference on Machine Learning (ICML)*, volume 267, pp. 50817–50847, 2025.

Rishabh Ranjan, Valter Hudovernik, Mark Znidar, Charilaos Kanatsoulis, Roshan Upendra, Mahmoud Mohammadi, Joe Meyer, Tom Palczewski, Carlos Guestrin, and Jure Leskovec. Relational transformer: Toward zero-shot foundation models for relational data. In *International Conference on Learning Representations (ICLR)*, 2026.

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback, 2012. URL https://arxiv.org/abs/1205.2618.

Joshua Robinson, Rishabh Ranjan, Weihua Hu, Kexin Huang, Jiaqi Han, Alejandro Dobles, Matthias Fey, Jan Eric Lenssen, Yiwen Yuan, Zecheng Zhang, Xinwei He, and Jure Leskovec. Relbench: A benchmark for deep learning on relational databases. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 21330–21341, 2024.

Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael Bronstein. Temporal graph networks for deep learning on dynamic graphs. In *ICML 2024 Workshop on Graph Representation Learning and Beyond*, 2020. URL https://grlplus.github.io/papers/58.pdf.

Marco Spinaci, Marek Polewczyk, Johannes Hoffart, Markus C. Kohler, Sam Thelin, and Tassilo Klein. PORTAL: Scalable tabular foundation models via content-specific tokenization. In *NeurIPS 2024 Third Table Representation Learning Workshop*, 2024. URL https://openreview.net/forum?id=TSZQvknbLO.

Haiming Tang, Sirui He, Mengjie Li, and Zhimao Guo. arXiv-physics: A large-scale physics citation and authorship dataset, 2024. URL https://github.com/PKUTHM/arxiv-physics.

Minjie Wang, Quan Gan, David Wipf, Zhenkun Cai, Ning Li, Jianheng Tang, Yanlin Zhang, Zizhao Zhang, Zunyao Mao, Yakun Song, Yanbo Wang, Jiahang Li, Han Zhang, Guang Yang, Xiao Qin, Chuan Lei, Muhan Zhang, Weinan Zhang, Christos Faloutsos, and Zheng Zhang. 4DBInfer: A 4D benchmarking toolbox for graph-centric predictive modeling on relational DBs, 2024. URL https://arxiv.org/abs/2404.18209.

Yanbo Wang, Xiyuan Wang, Quan Gan, Minjie Wang, Qibin Yang, David Wipf, and Muhan Zhang. Griffin: Towards a graph-centric relational database foundation model. In *International Conference on Machine Learning (ICML)*, volume 267, pp. 64604–64627. PMLR, 2025.

Jiaxuan You, Jonathan M Gomes-Selman, Rex Ying, and Jure Leskovec. Identity-aware graph neural networks. In *Association for the Advancement of Artificial Intelligence (AAAI)*, volume 35, pp. 10737–10745, 2021.

## A Related Work

**Relational deep learning (RDL).** RDL studies how to train neural models directly on relational databases by leveraging their multi-table structure. RDL represents a relational database as a heterogeneous graph, where rows correspond to entities and foreign-key relationships define edges between them (Fey et al., 2024). Early works applied graph neural networks to such relational graphs and demonstrated substantial improvements over feature-engineered baselines on forecasting, recommendation, and prediction tasks (Robinson et al., 2024; Chen et al., 2025). More recent approaches have explored transformer-based architectures to better capture long-range and higher-order dependencies across tables (Peleška & Šír, 2025a; Dwivedi et al., 2026), as well as positional encoding methods designed to improve representation learning on relational graphs (Kanatsoulis et al., 2025).

**Foundation models for tabular and relational data.** Recent tabular foundation models demonstrate strong performance, including in-context learning (Hollmann et al., 2023; Qu et al., 2025) and efficient fine-tuning (Kim et al., 2024). These models leverage supervised (Hollmann et al., 2023; 2025) or self-supervised (Spinaci et al., 2024; Kim et al., 2024) pretraining on real and synthetic tabular datasets. Extending such models to relational databases is challenging due to the presence of multiple tables connected via foreign-key relationships. To address this, relational foundation models have recently been proposed. For example, Fey et al. (2025) introduce KumoRFM, a graph-transformer-based architecture capable of in-context learning and fine-tuning. Similarly, Wang et al. (2025) pretrain the Griffin model on both tabular and relational datasets, combining table-level encoders with graph neural networks for cross-table reasoning. More recent approaches operate directly at the cell level and use attention mechanisms to explicitly represent foreign-key relationships, enabling unified reasoning across the entire relational database without requiring intermediate aggregation (Ranjan et al., 2026). To circumvent real-data limitations for large-scale pretraining, recent works have explored generating privacy-preserving versions of real databases with diffusion models (Hudovernik et al., 2025; Ketata et al., 2025) as well as generating synthetic databases from scratch using random graphs and Structural Causal Models (SCMs) (Kothapalli et al., 2026). In contrast, in RelBench v2 we collect a large number of realistic databases in a uniformly accessible manner.

## B Dataset schemas

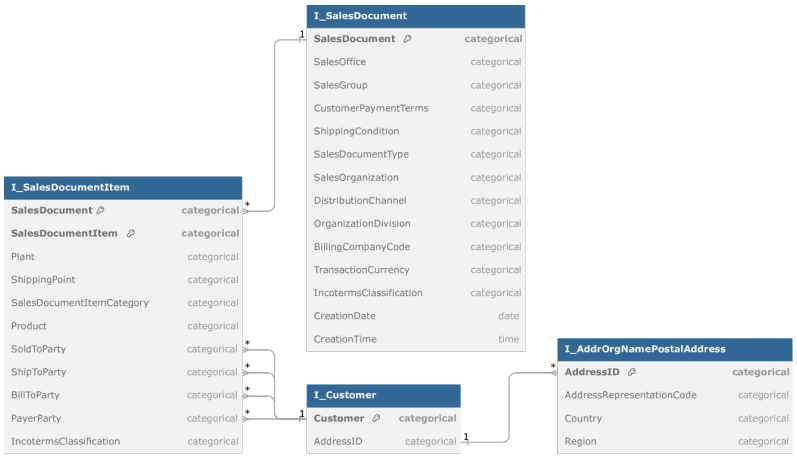


Figure 1: RelBench schema of the newly added Sales Autocompletion Linked Business Tables (SALT) dataset (Klein et al., 2024).

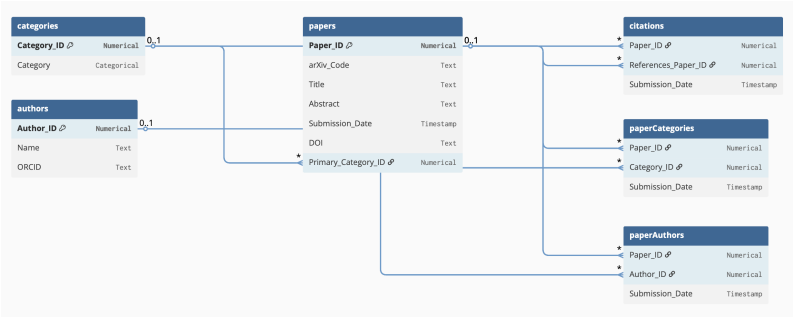


Figure 2: RELBENCH schema of the newly added arXiv-physics dataset (Tang et al., 2024).

# C ADDITIONAL TASK INFORMATION

## C.1 AUTOCOMPLETE TASK: MOTIVATION

Autocomplete tasks were inspired by the sales order autocomplete task from the SAP S/4HANA Sales Order User interface. In Figure 5, the response fields as a whole correspond to one row in a dataset. The user has filled in most of the response fields, allowing the interface to predict the terms of payment for this record.

## C.2 LIST OF PREDICTIVE TASK DESCRIPTIONS

1. `rel-amazon`
   Autocomplete Regression:
   - `review-rating`: For each review, predict the star rating.
2. `rel-arxiv`
   Forecasting Classification:
   - `paper-citation`: For each paper, predict whether it will receive at least one citation in the next 6 months.
   - `author-category`: For each author, predict the primary research category in which they will publish most in the next 6 months.

   Forecasting Regression:
   - `author-publication`: For each author, predict how many papers they will publish in the next 6 months.

   Recommendation:
   - `paper-paper-cocitation`: For each paper, predict which other papers will be co-cited with it in the next 6 months.
3. `rel-avito`
   Autocomplete Classification:
   - `searchstream-click`: For each search session, predict whether the user clicked on a result.
   - `searchinfo-isuserloggedon`: For each search, predict whether the user was logged in.
4. `rel-trial`
   Autocomplete Classification:
   - `studies-has_dmc`: For each study, predict whether it has a data monitoring committee.
   - `eligibilities-adult`: For each study, predict whether it includes adult participants.
   - `eligibilities-child`: For each study, predict whether it includes child participants.

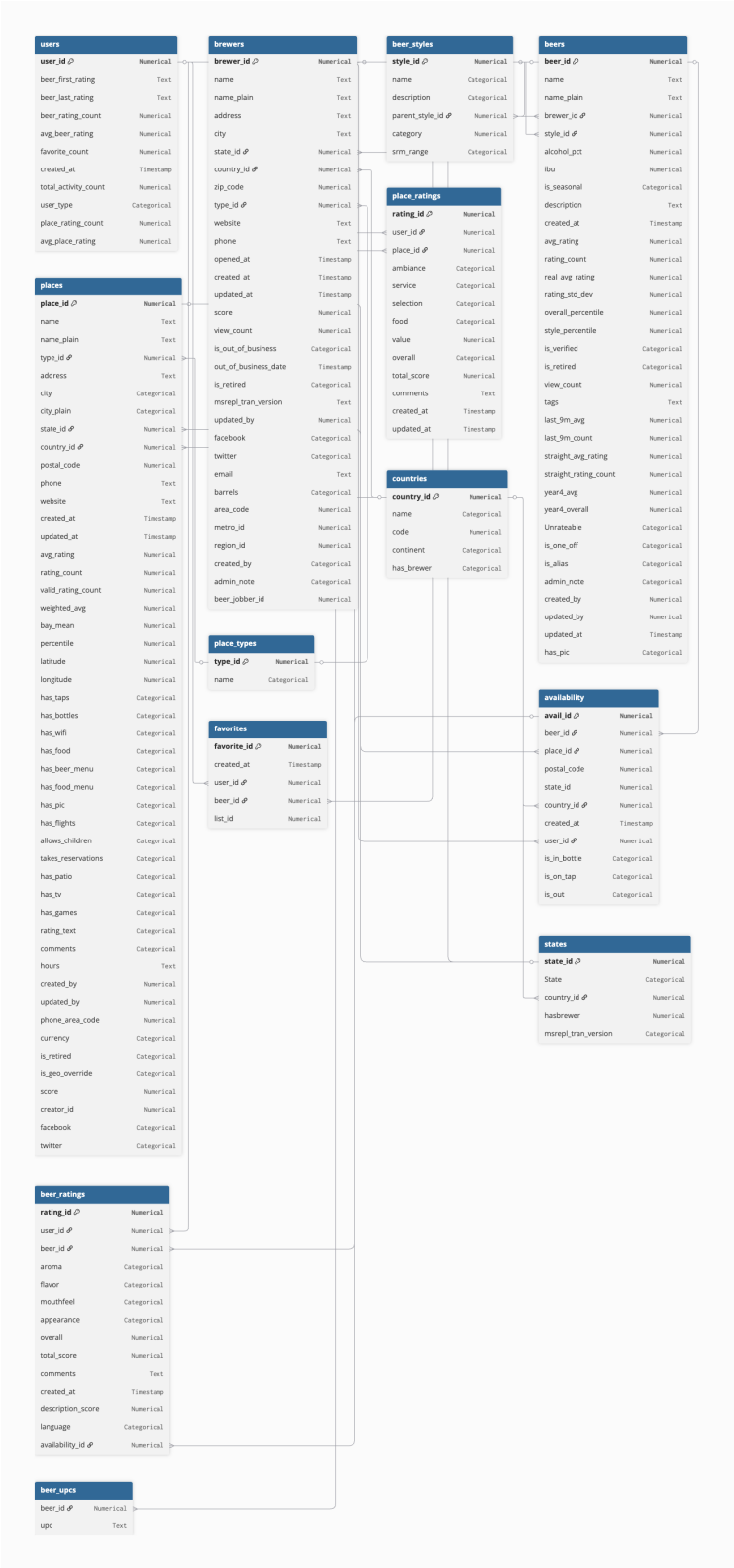


Figure 3: RELBENCH schema of the newly added RateBeer dataset.

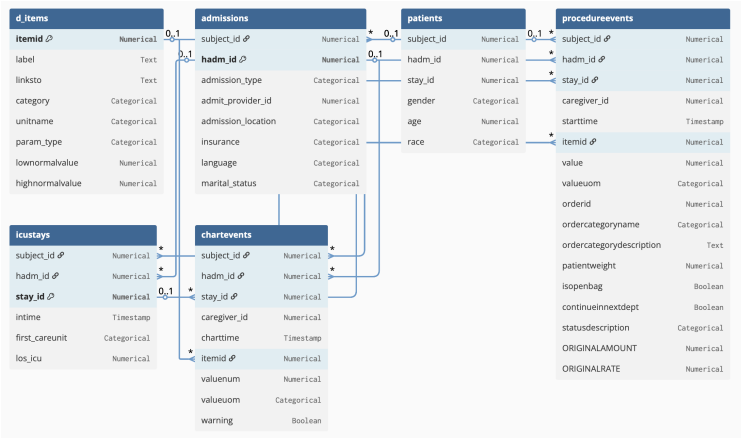


Figure 4: RELBENCH schema of the newly added MIMIC-IV v3.1 dataset (Johnson et al., 2024).

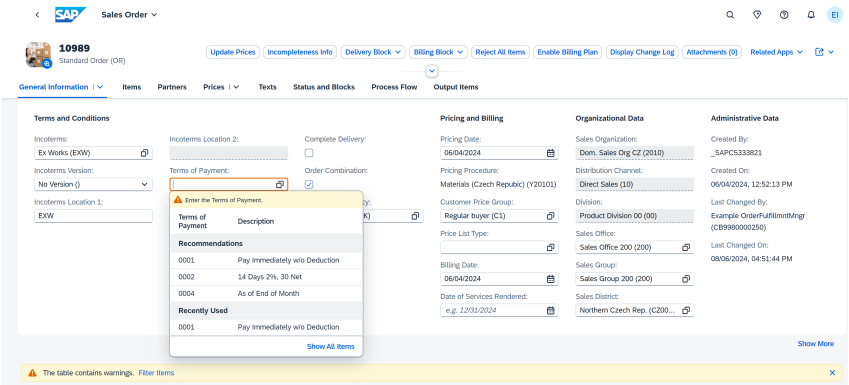


Figure 5: Illustrative example of a real-world autocomplete task, where the SAP S/4HANA Sales Order User interface (Klein et al., 2024) predicts payment terms based on other filled-in response fields.

Autocomplete Regression:

- `studies-enrollment`: For each study, predict the enrollment count.

5. `rel-event`

   Autocomplete Classification:

   - `event_interest-interested`: For each user–event interaction, predict whether the user marked the event as "interested."
   - `event_interest-not_interested`: For each user–event interaction, predict whether the user marked the event as "not interested."

   Autocomplete Regression:

   - `users-birthyear`: For each user, predict the user's birth year.

6. `rel-f1`

   Autocomplete Regression:

   - `results-position`: For each race result, predict the finishing position.
   - `qualifying-position`: For each qualifying entry, predict the qualifying position.

   Recommendation:

   - `driver-circuit-compete`: Predict in which circuits a driver will compete in the next year.

7. `rel-hm`
   Autocomplete Regression:
   - `transactions-price`: For each transaction, predict the item price.
8. `rel-ratebeer`
   Autocomplete Regression:
   - `beer_ratings-total_score`: For each user, given a beer, predict the total score rating the user will give to the beer.

   Forecasting Classification:
   - `beer-churn`: For each beer, predict if it will receive zero ratings in the next 90 days.
   - `user-churn`: For each active user, predict if they will rate zero beers in the next 90 days.
   - `brewer-dormant`: For each brewer, predict if it will release zero beers in the next year (risk of going dormant).

   Forecasting Regression:
   - `user-count`: Predict the number of ratings a user will give in the next 90 days.

   Recommendation:
   - `user-beer-favorite`: For each user, predict the top 10 beers they will next add to their Favorites list.
   - `user-beer-liked`: For each user, predict the top 10 beers they will rate at least 4.0 / 5.0.
   - `user-place-liked`: For each user, predict the top 10 places they will rate at least 80 / 100.
9. `rel-salt`
   Autocomplete Classification:
   - `item-plant`: For each sales order item, predict its plant (production/storage facility).
   - `item-shippoint`: For each sales order item, predict its shipping point (dispatch location).
   - `item-incoterms`: For each sales order item, predict its item-level international commercial terms.
   - `sales-office`: For each sales order, predict the sales office responsible for managing sales activities for the relevant products and geographic region.
   - `sales-group`: For each sales order, predict the sales group, i.e. the subdivision within the distribution chain that handles the customer and transaction.
   - `sales-payterms`: For each sales order, predict the customer payment terms (payment deadlines/discounts).
   - `sales-shipcond`: For each sales order, predict the shipping condition (logistics terms).
   - `sales-incoterms`: Predict the header-level Incoterms (international commercial terms) for each sales order.
10. `rel-stack`
    Autocomplete Classification:
    - `badges-class`: For each badge, predict the badge class.
11. `rel-mimic`
    Forecasting Classification:
    - `patient-iculengthofstay`: For each patient admitted into the ICU, predict whether their stay will last at least 3 days.

# D ADDITIONAL RESULTS AND EXPERIMENT DETAILS

## D.1 RESULTS WITH STANDARD DEVIATIONS

Tables 12 - 20 show mean and standard deviations over 5 runs for the autocomplete classification, autocomplete regression, entity classification, entity regression and link prediction results for all new tasks introduced in RELBENCH v2. For regression tasks, MAE metrics are reported below (the main paper reports $R^2$ metrics).

Table 12: **Autocomplete binary classification results on RELBENCH.** Binary classification uses the AUROC metric (higher is better). Standard baselines of random choice and majority class both correspond to AUROC values of approximately 50.00, so we exclude them below. Best values are in bold.

| Dataset | Task | Split | LightGBM | GNN |
|---|---|---|---|---|
| rel-avito | searchinfo-isuserloggedon | Val | $59.09_{\pm 0.37}$ | $\mathbf{82.57}_{\pm 0.64}$ |
| | | Test | $50.00_{\pm 0.00}$ | $\mathbf{73.00}_{\pm 0.79}$ |
| | searchstream-click | Val | $\mathbf{68.33}_{\pm 0.19}$ | $50.39_{\pm 0.22}$ |
| | | Test | $49.92_{\pm 0.17}$ | $\mathbf{55.92}_{\pm 14.04}$ |
| rel-event | event_interest-interested | Val | $51.25_{\pm 0.00}$ | $\mathbf{54.16}_{\pm 1.75}$ |
| | | Test | $49.57_{\pm 0.00}$ | $47.64_{\pm 3.44}$ |
| | event_interest-not_interested | Val | $51.98_{\pm 0.00}$ | $49.74_{\pm 13.71}$ |
| | | Test | $52.88_{\pm 0.00}$ | $\mathbf{60.40}_{\pm 19.57}$ |
| rel-trial | eligibilities-adult | Val | $58.10_{\pm 0.23}$ | $\mathbf{94.91}_{\pm 0.10}$ |
| | | Test | $50.00_{\pm 0.00}$ | $\mathbf{93.73}_{\pm 0.15}$ |
| | eligibilities-child | Val | $59.78_{\pm 0.15}$ | $\mathbf{85.91}_{\pm 0.20}$ |
| | | Test | $50.00_{\pm 0.00}$ | $\mathbf{87.25}_{\pm 0.10}$ |
| | studies-has_dmc | Val | $76.47_{\pm 0.26}$ | $\mathbf{78.21}_{\pm 0.12}$ |
| | | Test | $50.00_{\pm 0.00}$ | $\mathbf{75.72}_{\pm 0.11}$ |

Table 13: **Autocomplete multiclass classification results on RELBENCH.** Multiclass classification uses the accuracy metric (higher is better). Best values are in bold.

| Dataset | Task | Split | Random | Majority | LightGBM | GNN |
|---|---|---|---|---|---|---|
| rel-salt | item-incoterms | Val | 34.49 | 66.46 | $66.43_{\pm 0.01}$ | $\mathbf{80.23}_{\pm 0.48}$ |
| | | Test | 30.33 | 58.05 | $58.05_{\pm 0.00}$ | $\mathbf{69.36}_{\pm 0.77}$ |
| | item-plant | Val | 33.19 | 60.95 | $60.97_{\pm 0.04}$ | $\mathbf{99.70}_{\pm 0.16}$ |
| | | Test | 32.38 | 59.69 | $59.69_{\pm 0.00}$ | $\mathbf{99.46}_{\pm 0.12}$ |
| | item-shippoint | Val | 8.20 | 2.34 | $4.72_{\pm 0.02}$ | $\mathbf{98.54}_{\pm 0.13}$ |
| | | Test | 6.53 | 1.99 | $5.67_{\pm 5.18}$ | $\mathbf{98.39}_{\pm 0.08}$ |
| | sales-group | Val | 0.90 | 0.86 | $0.70_{\pm 0.12}$ | $\mathbf{18.43}_{\pm 0.22}$ |
| | | Test | 0.85 | 0.75 | $0.94_{\pm 1.32}$ | $\mathbf{15.76}_{\pm 0.30}$ |
| | sales-incoterms | Val | 31.83 | 61.00 | $60.53_{\pm 0.09}$ | $\mathbf{69.07}_{\pm 1.46}$ |
| | | Test | 29.39 | 56.63 | $56.63_{\pm 0.00}$ | $\mathbf{62.23}_{\pm 0.53}$ |
| | sales-office | Val | 50.01 | **99.91** | $99.90_{\pm 0.00}$ | $99.91_{\pm 0.00}$ |
| | | Test | 49.71 | **99.88** | $59.93_{\pm 54.71}$ | $99.88_{\pm 0.00}$ |
| | sales-payterms | Val | 0.32 | 0.65 | $1.85_{\pm 0.33}$ | $\mathbf{39.88}_{\pm 0.19}$ |
| | | Test | 0.24 | 0.47 | $5.64_{\pm 5.14}$ | $\mathbf{37.47}_{\pm 0.43}$ |
| | sales-shipcond | Val | 16.49 | 27.61 | $31.92_{\pm 0.04}$ | $\mathbf{59.21}_{\pm 1.87}$ |
| | | Test | 15.64 | 26.30 | $4.91_{\pm 0.00}$ | $\mathbf{56.85}_{\pm 1.32}$ |
| rel-stack | badges-class | Val | 11.59 | 20.68 | $1.93_{\pm 0.13}$ | $\mathbf{79.97}_{\pm 0.05}$ |
| | | Test | 10.49 | 18.34 | $2.51_{\pm 0.00}$ | $\mathbf{82.83}_{\pm 0.18}$ |

Table 14: **Autocomplete regression $R^2$ results on RELBENCH.** Regression uses the $R^2$ metric (higher is better). Best values are in bold.

| | | | Zero | Mean | Median | Ent. Mean | Ent. Med. | LightGBM | GNN |
|---|---|---|---|---|---|---|---|---|---|
| rel-amazon | review-rating | Val | −20.848 | **−0.006** | −0.364 | −20.848 | −20.848 | $-0.364_{\pm 0.000}$ | $-0.356_{\pm 0.012}$ |
| | | Test | −22.579 | **−0.014** | −0.341 | −13.313 | −13.313 | $-0.341_{\pm 0.000}$ | $-0.331_{\pm 0.013}$ |
| rel-event | users-birthyear | Val | −55012.323 | −0.047 | −0.216 | −55012.323 | −55012.323 | $0.004_{\pm 0.004}$ | $\mathbf{0.008}_{\pm 0.036}$ |
| | | Test | −64803.758 | −0.121 | −0.395 | −64803.758 | −64803.758 | $-0.192_{\pm 0.108}$ | $\mathbf{-0.030}_{\pm 0.040}$ |
| rel-f1 | qualifying-position | Val | −3.267 | −0.018 | −0.030 | −3.267 | −3.267 | $\mathbf{0.153}_{\pm 0.003}$ | $0.015_{\pm 0.009}$ |
| | | Test | −3.214 | −0.002 | −0.001 | −3.214 | −3.214 | $-0.953_{\pm 0.039}$ | $\mathbf{0.015}_{\pm 0.010}$ |
| | results-position | Val | −3.123 | −0.100 | −0.107 | −3.123 | −3.123 | $0.283_{\pm 0.006}$ | $\mathbf{0.440}_{\pm 0.026}$ |
| | | Test | −3.148 | −0.176 | −0.219 | −3.148 | −3.148 | $-2.437_{\pm 0.053}$ | $\mathbf{0.394}_{\pm 0.039}$ |
| rel-hm | transactions-price | Val | −2.215 | −0.065 | −0.140 | −2.215 | −2.215 | $-0.140_{\pm 0.000}$ | $\mathbf{0.725}_{\pm 0.003}$ |
| | | Test | −2.329 | −0.075 | −0.159 | −2.329 | −2.329 | $-0.160_{\pm 0.000}$ | $\mathbf{0.736}_{\pm 0.002}$ |
| rel-ratebeer | user-beer-rating | Val | −23.411 | −0.015 | −0.004 | −23.411 | −23.411 | $-0.004_{\pm 0.000}$ | $\mathbf{0.448}_{\pm 0.006}$ |
| | | Test | −34.352 | −0.031 | −0.003 | −34.352 | −34.352 | $-0.014_{\pm 0.000}$ | $\mathbf{0.394}_{\pm 0.010}$ |
| rel-trial | studies-enrollment | Val | −0.001 | **−0.000** | −0.001 | −0.001 | −0.001 | $-0.001_{\pm 0.000}$ | $-0.000_{\pm 0.000}$ |
| | | Test | −0.000 | **−0.000** | −0.000 | −0.000 | −0.000 | $-0.000_{\pm 0.000}$ | $-0.000_{\pm 0.000}$ |

Table 15: **Autocomplete regression results on RELBENCH.** Regression uses the MAE metric (lower is better). Best values are in bold.

| Dataset | Task | Split | Zero | Mean | Median | Ent. Mean | Ent. Med. | LightGBM | GNN |
|---|---|---|---|---|---|---|---|---|---|
| rel-amazon | review-rating | Val | 4.416 | 0.779 | **0.584** | 4.416 | 4.416 | $0.584_{\pm 0.000}$ | $0.586_{\pm 0.003}$ |
| | | Test | 4.453 | 0.762 | **0.547** | 2.907 | 2.907 | $0.547_{\pm 0.000}$ | $0.549_{\pm 0.004}$ |
| rel-event | users-birthyear | Val | 1987.058 | 5.668 | 5.885 | 1987.058 | 1987.058 | $\mathbf{5.144}_{\pm 0.006}$ | $5.193_{\pm 0.044}$ |
| | | Test | 1986.094 | 5.588 | 6.198 | 1986.094 | 1986.094 | $5.752_{\pm 0.164}$ | $\mathbf{5.224}_{\pm 0.046}$ |
| rel-f1 | qualifying-position | Val | 10.943 | 5.266 | 5.280 | 10.943 | 10.943 | $\mathbf{4.411}_{\pm 0.014}$ | $5.190_{\pm 0.023}$ |
| | | Test | 11.142 | 5.358 | 5.342 | 11.142 | 11.142 | $7.048_{\pm 0.067}$ | $\mathbf{5.312}_{\pm 0.023}$ |
| | results-position | Val | 8.587 | 4.251 | 4.257 | 8.587 | 8.587 | $3.167_{\pm 0.014}$ | $\mathbf{2.779}_{\pm 0.069}$ |
| | | Test | 9.504 | 4.822 | 4.877 | 9.504 | 9.504 | $8.377_{\pm 0.080}$ | $\mathbf{3.139}_{\pm 0.104}$ |
| rel-hm | transactions-price | Val | 0.034 | 0.015 | 0.015 | 0.034 | 0.034 | $0.015_{\pm 0.000}$ | $\mathbf{0.004}_{\pm 0.000}$ |
| | | Test | 0.034 | 0.015 | 0.015 | 0.034 | 0.034 | $0.015_{\pm 0.000}$ | $\mathbf{0.004}_{\pm 0.000}$ |
| rel-ratebeer | beer_ratings-total_score | Val | 3.444 | 0.530 | 0.520 | 3.444 | 3.444 | $0.520_{\pm 0.000}$ | $\mathbf{0.355}_{\pm 0.001}$ |
| | | Test | 3.470 | 0.457 | 0.431 | 3.470 | 3.470 | $0.447_{\pm 0.000}$ | $\mathbf{0.323}_{\pm 0.003}$ |
| rel-trial | studies-enrollment | Val | 3782.463 | 5893.531 | 3754.731 | 3782.463 | 3782.463 | $3734.843_{\pm 1.368}$ | $\mathbf{3709.484}_{\pm 1.698}$ |
| | | Test | 17660.073 | 19949.874 | 17635.281 | 17660.073 | 17660.073 | $17770.098_{\pm 72.701}$ | $\mathbf{17604.633}_{\pm 1.562}$ |

Table 16: **Entity binary classification results on RELBENCH.** Binary classification uses the AUROC metric (higher is better). Standard baselines of random choice and majority class both correspond to AUROC values of approximately 50.00, so we exclude them below. Best values are in bold.

| Dataset | Task | Split | LightGBM | GNN |
|---|---|---|---|---|
| rel-arxiv | paper-citation | Val | $71.94_{\pm 0.12}$ | $\mathbf{82.45}_{\pm 0.03}$ |
| | | Test | $71.21_{\pm 0.13}$ | $\mathbf{82.50}_{\pm 0.04}$ |
| rel-mimic | patient-iculengthofstay | Val | $53.64_{\pm 0.28}$ | $\mathbf{56.52}_{\pm 0.05}$ |
| | | Test | $51.81_{\pm 0.56}$ | $\mathbf{55.01}_{\pm 0.11}$ |
| rel-ratebeer | beer-churn | Val | $81.90_{\pm 0.06}$ | $\mathbf{90.47}_{\pm 0.06}$ |
| | | Test | $76.21_{\pm 0.12}$ | $\mathbf{78.67}_{\pm 0.60}$ |
| | brewer-dormant | Val | $76.39_{\pm 0.10}$ | $\mathbf{82.10}_{\pm 0.13}$ |
| | | Test | $75.79_{\pm 0.11}$ | $\mathbf{80.51}_{\pm 0.17}$ |
| | user-churn | Val | $87.02_{\pm 0.02}$ | $\mathbf{96.85}_{\pm 0.26}$ |
| | | Test | $83.92_{\pm 18.42}$ | $\mathbf{94.27}_{\pm 0.22}$ |

### D.2 ADDITIONAL REGRESSION METRICS

For regression tasks, the main paper reports $R^2$ to measure each method's predictive power, while MAE is reported in Tables 15 and 19 above.

Table 17: **Entity multiclass classification results on RELBENCH.** Multiclass classification uses the accuracy metric (higher is better). Best values are in bold.

| Dataset | Task | Split | Random | Majority | LightGBM | GNN |
|---|---|---|---|---|---|---|
| rel-arxiv | author-category | Val | 1.75 | 8.83 | $1.95_{\pm 0.16}$ | $\mathbf{52.63}_{\pm 0.08}$ |
| | | Test | 1.77 | 9.09 | $2.01_{\pm 0.21}$ | $\mathbf{50.74}_{\pm 1.01}$ |

Table 18: **Entity regression $R^2$ results on RELBENCH.** Regression uses the $R^2$ metric (higher is better). Best values are in bold.

| | | | Zero | Mean | Median | Ent. Mean | Ent. Med. | LightGBM | GNN |
|---|---|---|---|---|---|---|---|---|---|
| rel-amazon | item-ltv | Val | −0.025 | −0.000 | −0.013 | −0.247 | −0.107 | $0.002_{\pm 0.001}$ | $\mathbf{0.066}_{\pm 0.003}$ |
| | | Test | −0.013 | −0.000 | −0.007 | 0.030 | **0.099** | $0.001_{\pm 0.000}$ | $0.032_{\pm 0.001}$ |
| | user-ltv | Val | −0.084 | −0.003 | −0.084 | 0.053 | 0.095 | $-0.084_{\pm 0.000}$ | $\mathbf{0.195}_{\pm 0.006}$ |
| | | Test | −0.092 | −0.000 | −0.092 | 0.143 | 0.168 | $-0.092_{\pm 0.000}$ | $\mathbf{0.172}_{\pm 0.006}$ |
| rel-arxiv | author-publication | Val | −1.579 | −0.012 | −0.259 | 0.254 | 0.236 | $-0.259_{\pm 0.000}$ | $\mathbf{0.437}_{\pm 0.013}$ |
| | | Test | −1.572 | −0.000 | −0.210 | −0.010 | −0.064 | $-0.210_{\pm 0.000}$ | $\mathbf{0.249}_{\pm 0.013}$ |
| rel-avito | ad-ctr | Val | −0.238 | −0.002 | −0.095 | −0.224 | −0.224 | $-0.032_{\pm 0.005}$ | $\mathbf{0.030}_{\pm 0.017}$ |
| | | Test | −0.226 | −0.004 | −0.098 | −0.148 | −0.148 | $-0.039_{\pm 0.004}$ | $\mathbf{-0.001}_{\pm 0.020}$ |
| rel-event | user-attendance | Val | −0.249 | **−0.037** | −0.249 | −0.193 | −0.147 | $-0.249_{\pm 0.001}$ | $-0.045_{\pm 0.116}$ |
| | | Test | −0.168 | −0.019 | −0.168 | −0.065 | −0.043 | $-0.168_{\pm 0.000}$ | $\mathbf{0.003}_{\pm 0.096}$ |
| rel-f1 | driver-position | Val | −5.715 | −0.370 | −0.236 | −2.866 | −2.840 | $0.150_{\pm 0.025}$ | $\mathbf{0.249}_{\pm 0.008}$ |
| | | Test | −5.239 | −0.119 | −0.042 | −2.841 | −2.849 | $\mathbf{0.068}_{\pm 0.049}$ | $0.039_{\pm 0.063}$ |
| rel-hm | item-sales | Val | −0.017 | −0.000 | −0.017 | 0.065 | 0.053 | $-0.017_{\pm 0.000}$ | $\mathbf{0.187}_{\pm 0.002}$ |
| | | Test | −0.017 | −0.000 | −0.017 | 0.058 | 0.042 | $-0.017_{\pm 0.000}$ | $\mathbf{0.215}_{\pm 0.002}$ |
| rel-ratebeer | user-count | Val | −0.037 | −0.053 | −0.037 | 0.551 | 0.547 | $\mathbf{0.559}_{\pm 0.014}$ | $0.526_{\pm 0.005}$ |
| | | Test | −0.071 | −0.025 | −0.071 | 0.264 | 0.285 | $-0.170_{\pm 0.684}$ | $\mathbf{0.625}_{\pm 0.003}$ |
| rel-stack | post-votes | Val | −0.028 | −0.007 | −0.028 | **0.306** | 0.285 | $-0.028_{\pm 0.000}$ | $0.122_{\pm 0.003}$ |
| | | Test | −0.034 | −0.004 | −0.034 | **0.294** | 0.272 | $-0.034_{\pm 0.000}$ | $0.122_{\pm 0.004}$ |
| rel-trial | site-success | Val | −0.988 | **−0.005** | −0.988 | −0.749 | −0.809 | $-0.319_{\pm 0.077}$ | $-0.425_{\pm 0.059}$ |
| | | Test | −0.923 | **−0.001** | −0.923 | −0.714 | −0.751 | $-0.336_{\pm 0.087}$ | $-0.483_{\pm 0.110}$ |
| | study-adverse | Val | −0.021 | −0.002 | −0.020 | −0.021 | −0.021 | $\mathbf{0.134}_{\pm 0.017}$ | $0.066_{\pm 0.003}$ |
| | | Test | −0.054 | −0.005 | −0.050 | −0.054 | −0.054 | $\mathbf{0.307}_{\pm 0.039}$ | $0.177_{\pm 0.009}$ |

Table 19: **Entity regression results on RELBENCH.** Regression uses the MAE metric (lower is better). Best values are in bold.

| Dataset | Task | Split | Zero | Mean | Median | Ent. Mean | Ent. Med. | LightGBM | GNN |
|---|---|---|---|---|---|---|---|---|---|
| rel-arxiv | author-publication | Val | 1.681 | 0.864 | 0.681 | 0.827 | 0.804 | $0.681_{\pm 0.000}$ | $\mathbf{0.435}_{\pm 0.008}$ |
| | | Test | 1.577 | 0.769 | 0.577 | 0.879 | 0.874 | $0.577_{\pm 0.000}$ | $\mathbf{0.513}_{\pm 0.008}$ |
| rel-ratebeer | user-count | Val | 11.255 | 28.892 | 11.255 | 8.363 | 7.866 | $7.065_{\pm 0.058}$ | $\mathbf{5.813}_{\pm 0.031}$ |
| | | Test | 15.124 | 29.050 | 15.124 | 13.883 | 13.079 | $20.350_{\pm 9.536}$ | $\mathbf{7.374}_{\pm 0.102}$ |

### D.3 RECOMMENDATION TASK ABLATIONS

**The benefit of node position**. ID-GNN, which strongly utilizes node position, excels in entity-specific tasks. For example, we focus on the `rel-ratebeer` dataset whose recommendation tasks strongly highlight user-specific preferences. We observed that ID-GNN significantly outperforms plain GraphSAGE while also being substantially faster to train. This advantage stems from the fact that recommendation tasks are inherently multi-hop link prediction problems, where node position and context are crucial, as nodes tend to connect with others in their community. ID-GNN is better at capturing such node-specific patterns than vanilla GraphSAGE, leading to its superior performance.

**Multi-hop recommendation**. Recommendation tasks naturally involve multi-hop reasoning, where predicting a user–item link often requires traversing intermediate tables in the relational schema. Increasing the number of GNN layers expands the receptive field, allowing the model to aggregate information from progressively richer neighborhoods. This is particularly beneficial when intermediate tables contain strong preference signals. In the `user-beer-liked` task, the Beer Ratings table includes explicit ratings, subscores, and review text, enabling deeper GNNs to capture collaborative

Table 20: **Recommendation results on RELBENCH.** Recommendation uses the MAP metric (higher is better). Best values are in bold. Results for 4-layer GNN and ID-GNN models on RELBENCH v1 recommendation tasks are also included below.

| Dataset | Task | Split | Past Visit | Global Pop. | LightGBM | GNN (2) | GNN (4) | IDGNN (2) | IDGNN (4) |
|---|---|---|---|---|---|---|---|---|---|
| rel-amazon | user-item-purchase | Val | 0.07 | 0.31 | $0.16_{\pm 0.02}$ | $1.56_{\pm 0.05}$ | $\mathbf{1.60}_{\pm 0.07}$ | $0.13_{\pm 0.00}$ | $0.13_{\pm 0.00}$ |
| | | Test | 0.06 | 0.24 | $0.14_{\pm 0.01}$ | $0.74_{\pm 0.06}$ | $\mathbf{0.91}_{\pm 0.09}$ | $0.10_{\pm 0.00}$ | $0.11_{\pm 0.00}$ |
| | user-item-rate | Val | 0.09 | 0.16 | $0.23_{\pm 0.04}$ | $1.42_{\pm 0.13}$ | $\mathbf{1.54}_{\pm 0.08}$ | $0.15_{\pm 0.00}$ | $0.15_{\pm 0.00}$ |
| | | Test | 0.07 | 0.15 | $0.18_{\pm 0.02}$ | $0.81_{\pm 0.06}$ | $\mathbf{0.98}_{\pm 0.09}$ | $0.12_{\pm 0.00}$ | $0.12_{\pm 0.00}$ |
| | user-item-review | Val | 0.05 | 0.18 | $0.16_{\pm 0.03}$ | $1.02_{\pm 0.03}$ | $\mathbf{1.19}_{\pm 0.06}$ | $0.11_{\pm 0.00}$ | $0.11_{\pm 0.00}$ |
| | | Test | 0.04 | 0.11 | $0.10_{\pm 0.01}$ | $0.46_{\pm 0.02}$ | $\mathbf{0.63}_{\pm 0.03}$ | $0.09_{\pm 0.00}$ | $0.09_{\pm 0.00}$ |
| rel-avito | user-ad-visit | Val | 3.66 | 0.01 | $0.17_{\pm 0.02}$ | $0.10_{\pm 0.03}$ | $0.12_{\pm 0.01}$ | $5.40_{\pm 0.02}$ | $\mathbf{5.78}_{\pm 0.04}$ |
| | | Test | 1.95 | 0.00 | $0.06_{\pm 0.01}$ | $0.04_{\pm 0.01}$ | $0.05_{\pm 0.01}$ | $3.65_{\pm 0.02}$ | $\mathbf{3.92}_{\pm 0.03}$ |
| rel-hm | user-item-purchase | Val | 1.07 | 0.36 | $0.43_{\pm 0.03}$ | $0.87_{\pm 0.06}$ | $0.68_{\pm 0.04}$ | $2.64_{\pm 0.01}$ | $\mathbf{2.70}_{\pm 0.01}$ |
| | | Test | 0.89 | 0.30 | $0.37_{\pm 0.03}$ | $0.80_{\pm 0.07}$ | $0.68_{\pm 0.06}$ | $2.80_{\pm 0.01}$ | $\mathbf{2.91}_{\pm 0.01}$ |
| rel-stack | user-post-comment | Val | 2.05 | 0.03 | $0.04_{\pm 0.02}$ | $0.40_{\pm 0.09}$ | $0.56_{\pm 0.08}$ | $15.17_{\pm 0.15}$ | $\mathbf{15.60}_{\pm 0.10}$ |
| | | Test | 1.42 | 0.02 | $0.03_{\pm 0.01}$ | $0.16_{\pm 0.06}$ | $0.16_{\pm 0.03}$ | $12.72_{\pm 0.22}$ | $\mathbf{13.75}_{\pm 0.36}$ |
| | post-post-related | Val | 0.00 | 0.47 | $0.86_{\pm 0.17}$ | $0.02_{\pm 0.03}$ | $0.00_{\pm 0.00}$ | $7.83_{\pm 0.17}$ | $\mathbf{8.35}_{\pm 0.22}$ |
| | | Test | 1.74 | 1.46 | $1.86_{\pm 0.56}$ | $0.03_{\pm 0.02}$ | $0.05_{\pm 0.03}$ | $10.71_{\pm 0.30}$ | $\mathbf{12.46}_{\pm 0.36}$ |
| rel-trial | condition-sponsor-run | Val | 8.58 | 2.63 | $4.64_{\pm 0.27}$ | $3.11_{\pm 0.31}$ | $2.84_{\pm 0.37}$ | $\mathbf{11.33}_{\pm 0.04}$ | $\mathbf{11.33}_{\pm 0.07}$ |
| | | Test | 8.42 | 2.52 | $4.53_{\pm 0.21}$ | $3.07_{\pm 0.12}$ | $2.69_{\pm 0.40}$ | $\mathbf{11.36}_{\pm 0.04}$ | $11.31_{\pm 0.23}$ |
| | site-sponsor-run | Val | 15.90 | 4.91 | $10.69_{\pm 0.69}$ | $14.17_{\pm 0.37}$ | $14.09_{\pm 0.46}$ | $\mathbf{17.43}_{\pm 0.07}$ | $17.41_{\pm 0.05}$ |
| | | Test | 17.31 | 3.75 | $8.21_{\pm 0.54}$ | $10.37_{\pm 0.92}$ | $11.05_{\pm 0.94}$ | $19.00_{\pm 0.12}$ | $\mathbf{19.04}_{\pm 0.05}$ |
| rel-arxiv | paper-paper-cocitation | Val | 19.01 | 1.25 | $12.49_{\pm 0.60}$ | $12.19_{\pm 0.23}$ | $12.96_{\pm 00.33}$ | $25.22_{\pm 0.09}$ | $\mathbf{35.76}_{\pm 0.09}$ |
| | | Test | 16.51 | 1.13 | $11.01_{\pm 0.43}$ | $8.83_{\pm 0.38}$ | $10.46_{\pm 0.56}$ | $22.95_{\pm 0.07}$ | $\mathbf{35.39}_{\pm 0.17}$ |
| rel-f1 | driver-circuit-compete | Val | 53.41 | 55.19 | $66.06_{\pm 2.68}$ | $3.60_{\pm 2.11}$ | $10.57_{\pm 8.35}$ | $70.70_{\pm 0.00}$ | $\mathbf{74.40}_{\pm 1.03}$ |
| | | Test | 20.76 | 50.12 | $57.77_{\pm 2.95}$ | $9.67_{\pm 10.56}$ | $16.57_{\pm 11.17}$ | $62.32_{\pm 0.00}$ | $\mathbf{76.18}_{\pm 6.59}$ |
| rel-ratebeer | user-beer-favorite | Val | 0.00 | 2.33 | $1.24_{\pm 0.15}$ | $2.09_{\pm 0.15}$ | – | $3.09_{\pm 0.14}$ | $\mathbf{3.33}_{\pm 0.09}$ |
| | | Test | 0.00 | 1.10 | $0.67_{\pm 0.13}$ | $0.56_{\pm 0.22}$ | – | $1.21_{\pm 0.93}$ | $\mathbf{1.89}_{\pm 0.09}$ |
| | user-beer-liked | Val | 0.00 | 0.77 | $0.43_{\pm 0.02}$ | $0.77_{\pm 0.06}$ | – | $0.21_{\pm 0.01}$ | $\mathbf{1.48}_{\pm 0.06}$ |
| | | Test | 0.00 | 0.61 | $0.29_{\pm 0.04}$ | $0.54_{\pm 0.22}$ | – | $0.32_{\pm 0.03}$ | $\mathbf{1.46}_{\pm 0.19}$ |
| | user-place-liked | Val | 0.00 | 0.24 | $0.24_{\pm 0.07}$ | $1.06_{\pm 0.17}$ | – | $0.88_{\pm 0.09}$ | $\mathbf{2.20}_{\pm 0.24}$ |
| | | Test | 0.00 | 0.11 | $0.08_{\pm 0.03}$ | $1.15_{\pm 0.34}$ | – | $0.60_{\pm 0.50}$ | $\mathbf{1.85}_{\pm 0.30}$ |

Table 21: Task-specific RDL default hyperparameters.

| **Hyperparameter** | **Task type** | | | |
|---|---|---|---|---|
| | Autocomplete | Entity classification | Entity regression | Recommendation |
| Learning rate | 0.005 | 0.005 | 0.005 | 0.001 |
| Maximum epochs | 10 | 10 | 10 | 20 |
| Batch size | 512 | 512 | 512 | 512 |
| Hidden feature size | 128 | 128 | 128 | 128 |
| Aggregation | summation | summation | summation | summation |
| Number of layers | 2 | 2 | 2 | 2 |
| Number of neighbors | 128 | 128 | 128 | 128 |
| Temporal sampling strategy | uniform | uniform | uniform | uniform |

filtering effects by traversing paths such as User → Ratings → Beer → Ratings → Beer. In such cases, increasing the number of layers yields substantial performance gains. In contrast, tasks with feature-sparse intermediate tables (e.g., Favorites, which contains the implicit link between users and beers with timestamps) tend to benefit less from additional hops, as deeper neighborhoods introduce weaker or noisier signals.

### D.4 EXPERIMENT HYPERPARAMETERS

All experiments for RELBENCH v2 were conducted with minimal parameter tuning. The default hyperparameters are specified in Table 21. The consistency of these defaults across task types highlights the robustness of RDL models, which perform well without extensive hyperparameter optimization.

For the `rel-ratebeer` recommendation tasks, we adjusted the batch size to 64 when training two-layer GraphSAGE, two-layer ID-GNN, and four-layer ID-GNN models to accommodate GPU memory constraints. While four-layer GraphSAGE remains prohibitively memory-intensive under this setting, all other models were trained successfully with the adjusted configuration.

# E BRIDGING TEMPORAL GRAPH BENCHMARK (TGB) AND RELBENCH

**Dataset descriptions.** `tgbl-wiki-v2` captures Wikipedia co-edit interactions over a short horizon and is naturally bipartite; `tgbl-review-v2` is a long-range e-commerce interaction graph derived from Amazon review activity; `tgbl-coin` consists of cryptocurrency transactions; `tgbl-comment` models a large-scale Reddit comment interaction stream; and `tgbl-flight` is a decades-long flight network event log. For node property prediction, `tgbn-trade` models annual UN trade interactions, `tgbn-genre` is a LastFM user–genre interaction stream, `tgbn-reddit` is a temporal Reddit hyperlink network, and `tgbn-token` represents blockchain user–token interactions. Finally, the heterogeneous family `thgl-*` includes GitHub interaction streams (`thgl-software`, `thgl-github`), a Reddit interaction stream (`thgl-forum`), and an app-market interaction stream (`thgl-myket`), each with explicit node/edge type constraints.

## E.1 TRANSLATION: TEMPORAL EVENT STREAMS TO RELATIONAL SCHEMAS

Each TGB dataset is provided as a chronological stream of interactions. Our translation materializes a RELBENCH-style normalized schema in which *entities* become tables with primary keys and *events* become tables whose rows reference entities via foreign keys and include a time column. Across all translated datasets, we (i) include an explicit timestamp column (`event_ts`) on time-varying tables; (ii) avoid storing train/validation/test split columns, and instead store dataset-level cutoffs (VAL_TIMESTAMP, TEST_TIMESTAMP) as metadata so splits can be derived from timestamps at load/evaluation time; and (iii) keep only low-dimensional attributes as relational columns (e.g., scalar `weight`), while omitting high-dimensional edge message vectors that would otherwise expand into hundreds of sparse columns.

For dynamic link prediction datasets (`tgbl-*`), we represent the interaction stream as a single event table referencing a node table:

```
nodes(node_id)
events(event_id, src_id, dst_id, event_ts, weight)
```

For bipartite datasets (e.g., `tgbl-wiki-v2`), we use separate entity tables `src_nodes(src_id)` and `dst_nodes(dst_id)` to preserve type constraints and enable type-correct negative sampling.

For temporal heterogeneous datasets (`thgl-*`), we materialize one entity table per node type and one event table per edge type:

`nodes_type_t(node_type_t_id)` for each node type $t$.
`events_edge_type_e(event_id, src_id, dst_id, event_ts, weight)` for each edge type $e$.

This schema-first representation preserves the semantics of typed relations and makes relation-conditioned tasks and evaluation natural in RELBENCH.

For dynamic node property prediction datasets (`tgbn-*`), targets are time-varying and often sparse. To avoid storing dense label vectors, we represent supervision as normalized label events:

```
labels(label_id)
label_events(label_event_id, src_id, label_ts)
label_event_items(item_id, label_event_id, label_id,
label_weight)
```

This preserves the full supervision signal while keeping storage proportional to the number of non-zeros.

## E.2 EFFICIENT STORAGE AND LOADING: PARQUET + DISK-BACKED CSR

A central engineering challenge in translating TGB to RELBENCH is scale: several datasets contain on the order of $\mathcal{O}(10^7\text{–}10^8)$ temporal events. Naïvely materializing the full event log as an in-memory edge list (e.g., a global `edge_index`) is prohibitive on commodity CPU machines, so we rely on columnar storage (parquet) together with disk-backed sparse graph caches. We store each table as a separate parquet file in the standard RELBENCH `Database.save()` layout. This provides

**Algorithm 1** Streaming CSR construction from parquet event logs (sketch).

**Require:** Parquet event table with columns `src_id`, `dst_id`, `event_ts`; cutoff time $\tau$.
**Ensure:** CSR adjacency arrays `indptr`, `indices` for events with `event_ts` $\leq \tau$.

1: **Pass 1:** Scan parquet in large row batches; filter rows by `event_ts` $\leq \tau$; accumulate per-node degree counts.
2: Build `indptr` by prefix-summing degrees.
3: **Pass 2:** Re-scan parquet in batches; filter by `event_ts` $\leq \tau$; write neighbors into `indices` using `indptr` offsets.
4: Optionally: symmetrize for undirected message passing; cache per-event arrays (`event_ts`, `weight`) aligned with `indices`.

columnar storage for fast projection to only the columns required by a given model, and row-group metadata that supports efficient sequential scans without loading entire tables into memory. For `thgl-*` datasets with many edge types, we additionally export each `events_edge_type_*` table using a streaming/chunked parquet writer. This avoids materializing massive intermediate dataframes per relation and keeps the conversion pipeline memory-bounded even when the total number of events is large.

To train sampled GNN baselines at scale, we build and cache CSR (compressed sparse row) adjacency representations directly from parquet scans. Concretely, we store `indptr` and `indices` arrays (and, when needed, aligned per-event arrays such as timestamps and weights) as memory-mappable `.npy` artifacts. With CSR, neighbor sampling reduces to lightweight slicing operations over these arrays, enabling mini-batch training without ever holding the full graph in system memory. For relational baselines, we use an "event-as-node" message passing graph: each event row becomes a node connected (via PK/FK edges) to its incident entity rows. We build node→event CSR adjacencies per entity table and unify per-event arrays. This makes relational message passing scalable while remaining faithful to the normalized schema.

### E.3 Baselines and evaluation protocol

We benchmark three complementary model families on the translated TGB datasets, spanning graph-native GNNs, relational (PK/FK) GNNs, and temporal sequence models.

**GraphSAGE (projected-edge graph; TGB-style).** We treat each interaction event as an edge from `src` to `dst` and train a sampled two-layer GraphSAGE encoder using disk-backed CSR adjacency built from parquet scans. This baseline is graph-native in the sense that the event log is directly interpreted as a temporal edge stream over a single (or bipartite) node set.

**GraphSAGE (event-as-node relational graph; RDL-style).** We represent each event row as its own node and perform message passing over the induced PK/FK graph that connects entity rows to event rows ("RelEventSAGE"). This baseline is schema-faithful: neighborhoods and aggregation follow the relational structure rather than collapsing the database into a single projected graph.

**TGN + attention (temporal baseline).** We train a temporal graph network with a lightweight attention-based embedding module that consumes the chronological event stream (or equivalently, the exported parquet event tables) under controlled compute budgets. This baseline captures temporal dynamics via memory and time encoding, and is directly comparable to the GNN baselines under the same split and leakage constraints.

**Metrics.** For `tgbl-*` and `thgl-*` we report sampled-negative MRR@100 (higher is better) to keep a single, consistent ranking metric across model families and schema variants. For `tgbn-*` we report the official NDCG@10 (higher is better), and additionally report sampled-negative MRR for consistency across task families. For existing RELBENCH recommendation tasks we report the official MAP@10 used in the RELBENCH evaluation code.

**Leakage control.** All splits are derived from dataset-level VAL_TIMESTAMP and TEST_TIMESTAMP cutoffs, and we do not store split membership as a database column. When comparing baselines, we ensure that message passing and neighborhood construction only use historical events up to the validation cutoff (i.e., `adj=val`) unless explicitly stated otherwise, preventing test evaluation from incorporating post-cutoff interactions.

### E.4 RESULTS

Tables 22, 23, and 24 summarize baseline performance on the translated TGB datasets. Table 25 reports a small reference point on existing RELBENCH recommendation tasks using GraphSAGE and the attention-based TGN baseline.

Table 22: **Dynamic node property prediction (`tgbn-*`) on translated TGB datasets.** We report validation and test performance for GraphSAGE and a sampled neighbor-attention variant ("Attn"), using NDCG@10 (official) and sampled-negative MRR.

| | GraphSAGE | | | | Attn (sampled neighbor attention) | | | |
|---|---|---|---|---|---|---|---|---|
| Dataset | Val MRR | Val NDCG@10 | Test MRR | Test NDCG@10 | Val MRR | Val NDCG@10 | Test MRR | Test NDCG@10 |
| `tgbn-trade` | 0.9522 | **0.3769** | 0.9393 | **0.3765** | 0.9102 | 0.3849 | 0.8635 | 0.3401 |
| `tgbn-genre` | **0.8682** | **0.6169** | **0.8591** | **0.6054** | 0.6664 | 0.4263 | 0.6552 | 0.4139 |
| `tgbn-reddit` | **0.7804** | **0.6474** | **0.7555** | **0.6146** | 0.4326 | 0.3413 | 0.4067 | 0.3098 |
| `tgbn-token` | **0.4043** | **0.3763** | **0.3405** | **0.3098** | 0.2451 | 0.2303 | 0.2101 | 0.1935 |

Table 23: **Dynamic link prediction (`tgbl-*`) on translated TGB datasets (sampled-negative MRR@100).** We compare (i) a graph-native projected-edge GraphSAGE baseline, (ii) an event-as-node relational GraphSAGE baseline (RelEventSAGE), and (iii) a TGN + attention baseline trained from exported parquet under bounded budgets. Best test results per dataset are in bold.

| | GraphSAGE (projected edges) | | GraphSAGE (event-as-node) | | TGN+Attn | |
|---|---|---|---|---|---|---|
| Dataset | Val | Test | Val | Test | Val | Test |
| `tgbl-wiki-v2` | **0.4203** | **0.3782** | 0.2757 | 0.2517 | 0.3998 | 0.3384 |
| `tgbl-review-v2` | 0.0932 | 0.0852 | 0.2596 | 0.2317 | **0.2528** | **0.2457** |
| `tgbl-coin` | 0.4541 | 0.3932 | **0.6064** | **0.5554** | 0.5604 | 0.5067 |
| `tgbl-comment` | 0.2089 | 0.1536 | **0.2896** | **0.2305** | 0.2960 | 0.2098 |
| `tgbl-flight` | **0.7082** | **0.6737** | 0.6357 | 0.5915 | 0.4838 | 0.4566 |

Table 24: **Temporal heterogeneous link prediction (`thgl-*`) on translated TGB datasets (sampled-negative MRR@100).** We compare event-as-node relational GraphSAGE (RelEventSAGE) against TGN + attention. Best test results per dataset are in bold.

| | GraphSAGE (event-as-node) | | TGN+Attn | |
|---|---|---|---|---|
| Dataset | Val | Test | Val | Test |
| `thgl-software` | 0.1388 | 0.1206 | 0.1367 | **0.1290** |
| `thgl-forum` | **0.4635** | **0.4401** | 0.3452 | 0.3527 |
| `thgl-myket` | **0.7264** | **0.7084** | 0.6614 | 0.6648 |
| `thgl-github` | 0.0725 | 0.0666 | **0.0782** | **0.0767** |

Table 25: **Reference recommendation results on existing RELBENCH datasets (smoke runs).** We report validation MAP@10 for GraphSAGE and TGN+Attn on three RELBENCH recommendation tasks.

| Dataset | Task | Metric | GraphSAGE | TGN+Attn |
|---|---|---|---|---|
| `rel-f1` | `driver-race-compete` | val MAP@10 | 0.06048 | **0.27821** |
| `rel-hm` | `user-item-purchase` | val MAP@10 | 0.0006649 | **0.0009053** |
| `rel-stack` | `post-post-related` | val MAP@10 | **0.0024797** | 0.00017857 |

**Discussion.** Across `tgbl-*`, the comparison between projected-edge GraphSAGE and event-as-node relational GraphSAGE highlights when the relationalization of events is beneficial: datasets with informative event records (e.g., heavy-tailed `weight` and rich event semantics) tend to favor the event-as-node representation (`tgbl-coin`, `tgbl-comment`), while datasets that are closer to "pure" structural proximity under our compact schema (and without high-dimensional message features) can favor the projected-edge baseline (`tgbl-wiki-v2`, `tgbl-flight`). On `thgl-*`, the attention-based temporal baseline is competitive and sometimes best (`thgl-software`, `thgl-github`), while relational GraphSAGE remains strong on datasets where type-correct relational neighborhoods

provide a high-signal inductive bias (`thgl-forum`, `thgl-myket`). Finally, Table 25 provides a small anchor on existing RELBENCH recommendation tasks: even with minimal tuning, the attention-based temporal baseline can substantially improve MAP on some datasets (`rel-f1`), but does not uniformly dominate across domains (`rel-stack`).