Relational Transformer: Toward Zero-Shot Foundation Models for Relational Data

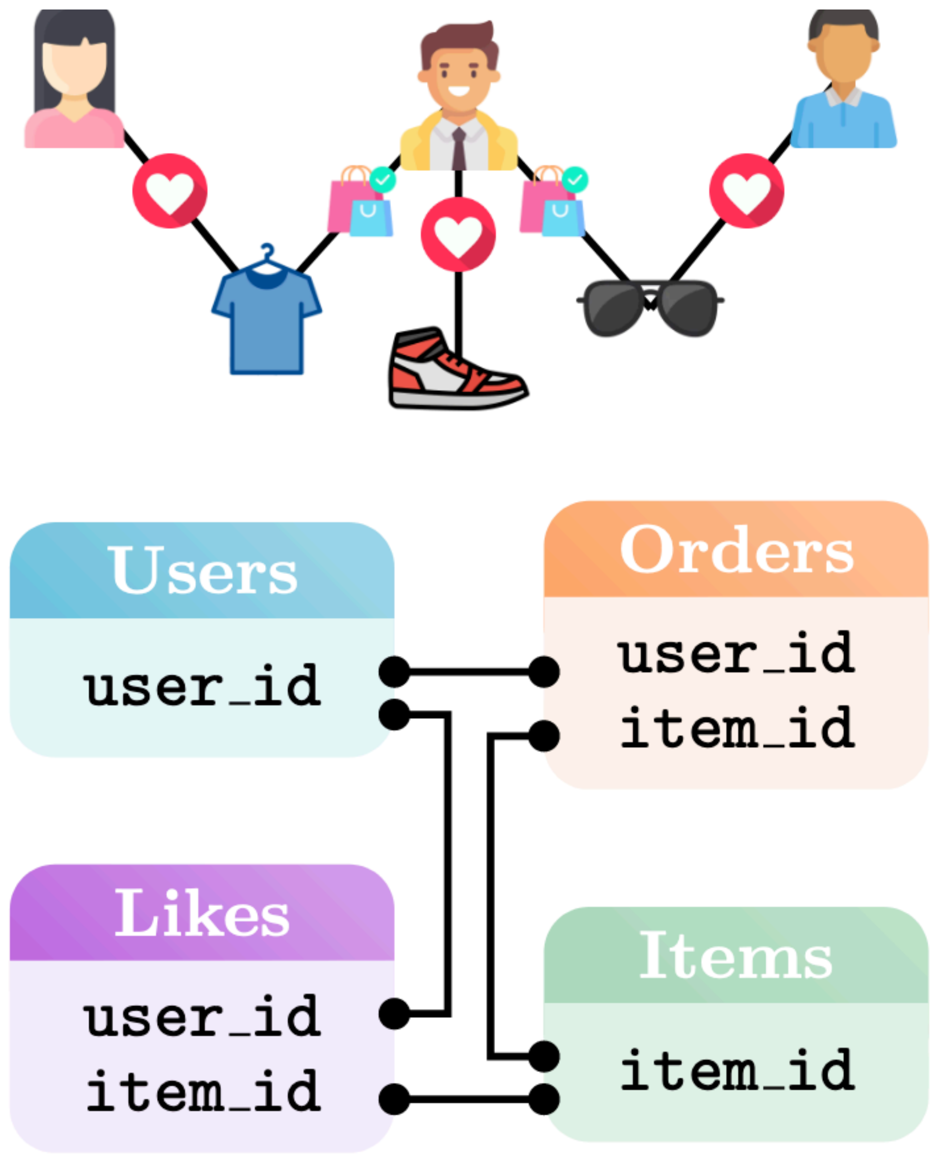
Rishabh Ranjan[01*], **Valter Hudovernik**[0], **Mark Znidar**[0], **Charilaos Kanatsoulis**[0],
Roshan Upendra[1], **Mahmoud Mohammadi**[1], **Joe Meyer**[1], **Tom Palczewski**[1],
Carlos Guestrin[0], **Jure Leskovec**[0]
[0]Stanford University, [1]SAP Labs LLC
{ranjanr,guestrin,jure}@stanford.edu

Pre-print: https://arxiv.org/abs/2510.06377
Accepted to **ICLR 2026**
Oral @ AI for Tabular Data (AI4TD) Workshop, NeurIPS 2025

Rishabh Ranjan | Temporal Graph Reading Group | 2026 Feb 19

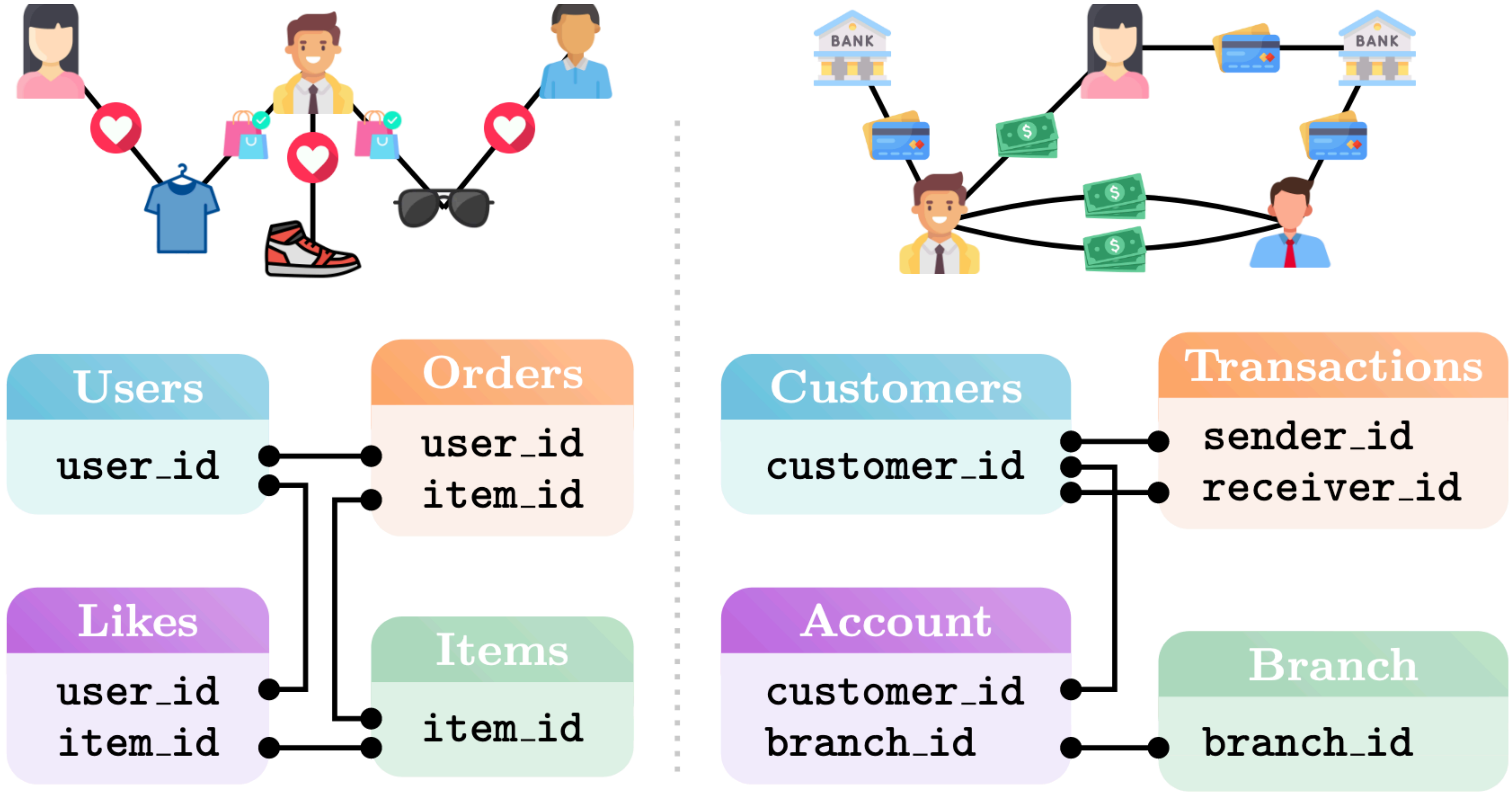
Relational Foundation Models

The world runs on relational data



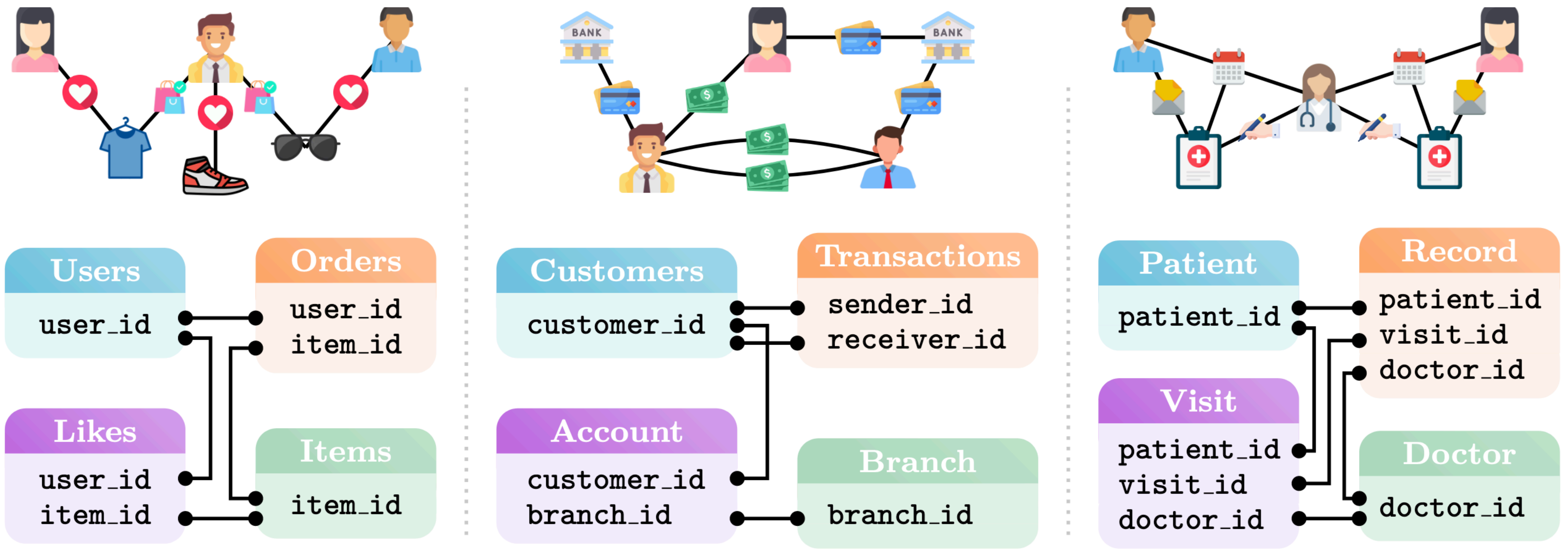
(a) E-Commerce

The world runs on relational data



(a) E-Commerce **(b)** Finance

The world runs on relational data



(a) E-Commerce **(b)** Finance **(c)** Health-Care

Predictions on relational data are critical

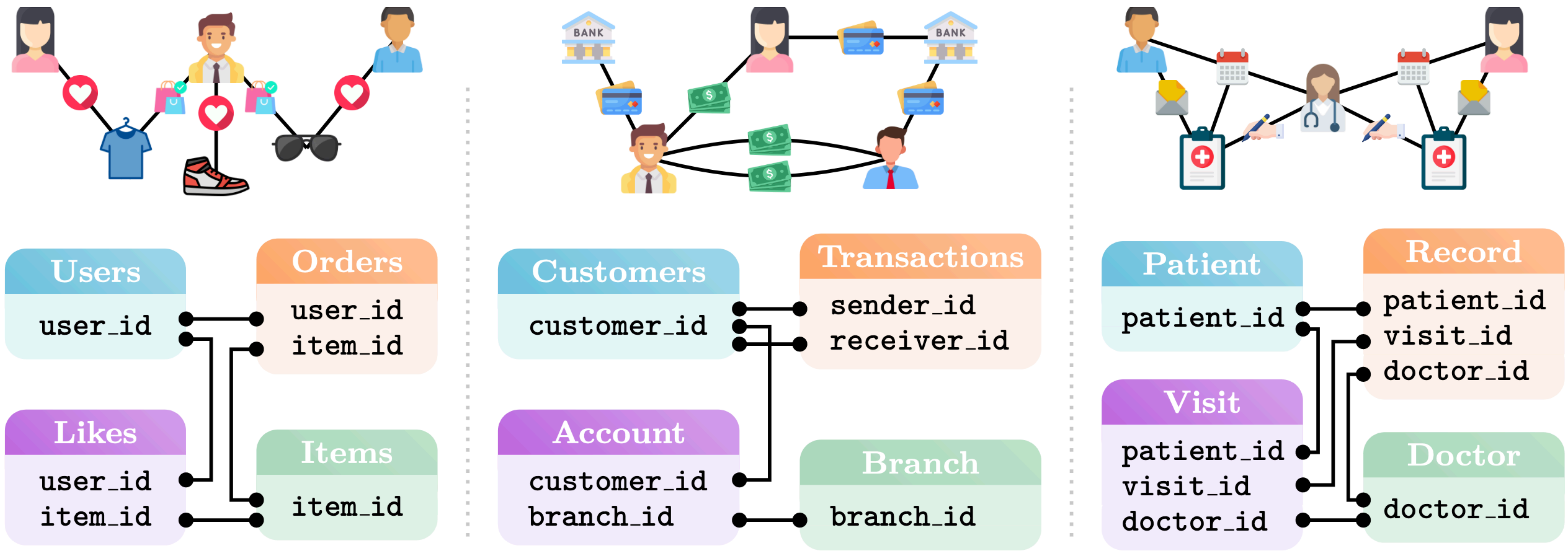
What will user *123* buy next?

Will user *321* churn soon?

Is transaction *567* fraudulent?

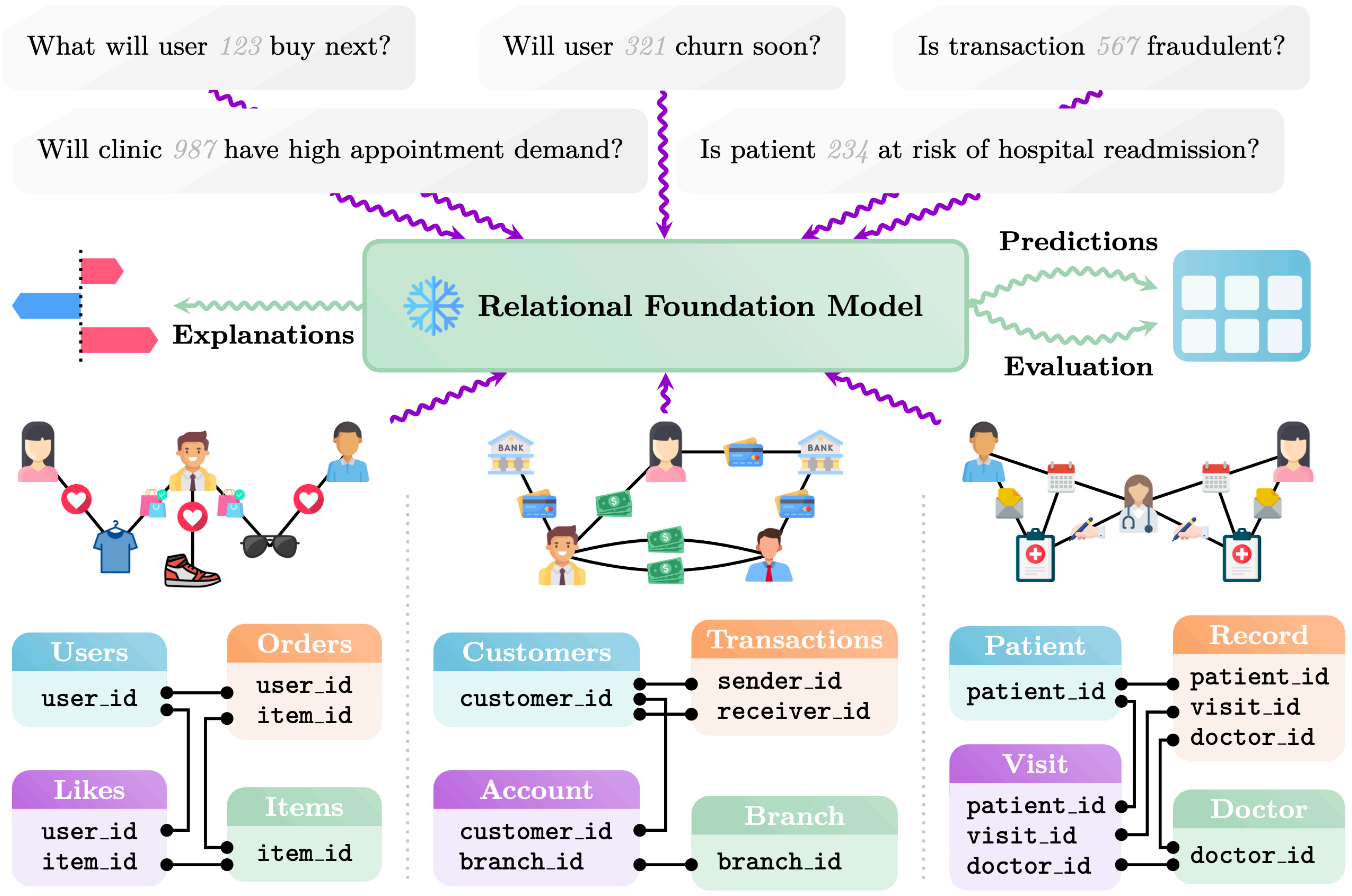
Will clinic *987* have high appointment demand?

Is patient *234* at risk of hospital readmission?



(a) E-Commerce **(b)** Finance **(c)** Health-Care

~~Custom ML models~~ One model to rule them all



(a) E-Commerce (b) Finance (c) Health-Care

Non-Relational Foundation Models for Enterprise Data

Single tables, time series, graphs

PRIOR

TabPFN-2.5: Advancing the State of the Art in Tabular Foundation Models

Prior Labs Team[1]

[1]The list of contributors can be found in the appendix.

21 May 2025

GRAPH FOUNDATION MODELS: A COMPREHENSIVE SURVEY

Zehong Wang[†,*,®,1] **Zheyuan Liu**[*,1] **Tianyi Ma**[*,1] **Jiazheng Li**[*,2] **Zheyuan Zhang**[*,1]
Xingbo Fu[*,3] **Yiyang Li**[*,1] **Zhengqing Yuan**[*,1] **Wei Song**[1] **Yijun Ma**[1] **Qingkai Zeng**[1]
Xiusi Chen[4] **Jianan Zhao**[6,7] **Jundong Li**[3] **Meng Jiang**[1] **Pietro Liò**[5]
Nitesh Chawla[1] **Chuxu Zhang**[2] **Yanfang Ye**[®,1]

[†]**Project Leader** [*]**Major Student Contributors**
[®]**Correspondance**: Zehong Wang <zwang43@nd.edu>, Yanfang Ye <yye7@nd.edu>

[1]University of Notre Dame, [2]University of Connecticut, [3]University of Virginia,
[4]University of Illinois Urbana-Champaign, [5]University of Cambridge,
[6]Mila - Québec AI Institute, [7]Université de Montréal

Google BigQuery

Home › Documentation › Data analytics › BigQuery › Guides

The TimesFM model

This document describes BigQuery ML's built-in TimesFM time series forecasting model.

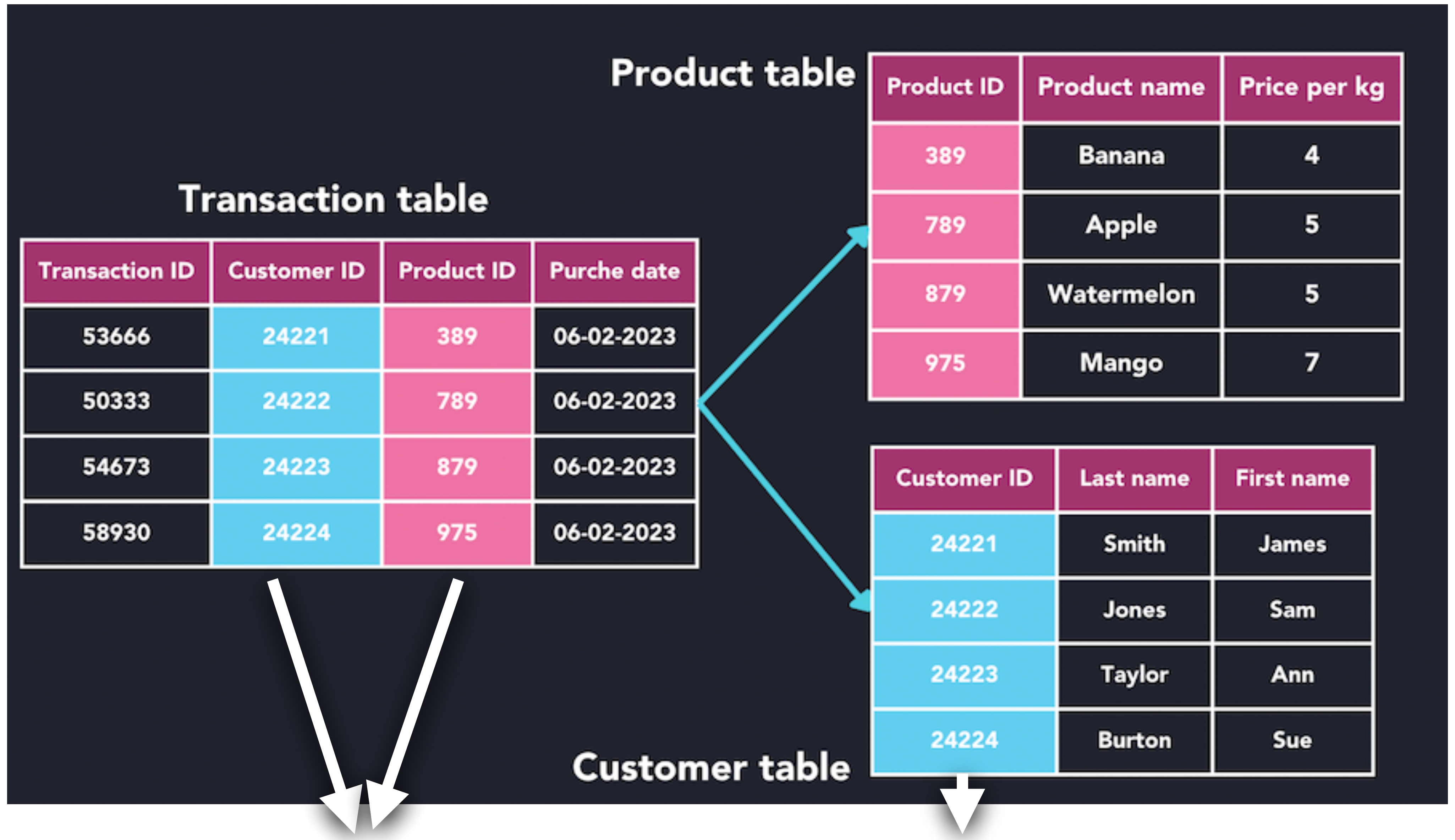
aws

Fast and accurate zero-shot forecasting with Chronos-Bolt and AutoGluon

by Abdul Fatir Ansari, Caner Turkmen, Oleksandr Shchur, and Lorenzo Stella | on 02 DEC 2024 | in Announcements, Generative AI, Intermediate (200), Technical How-to | Permalink | Comments | Share

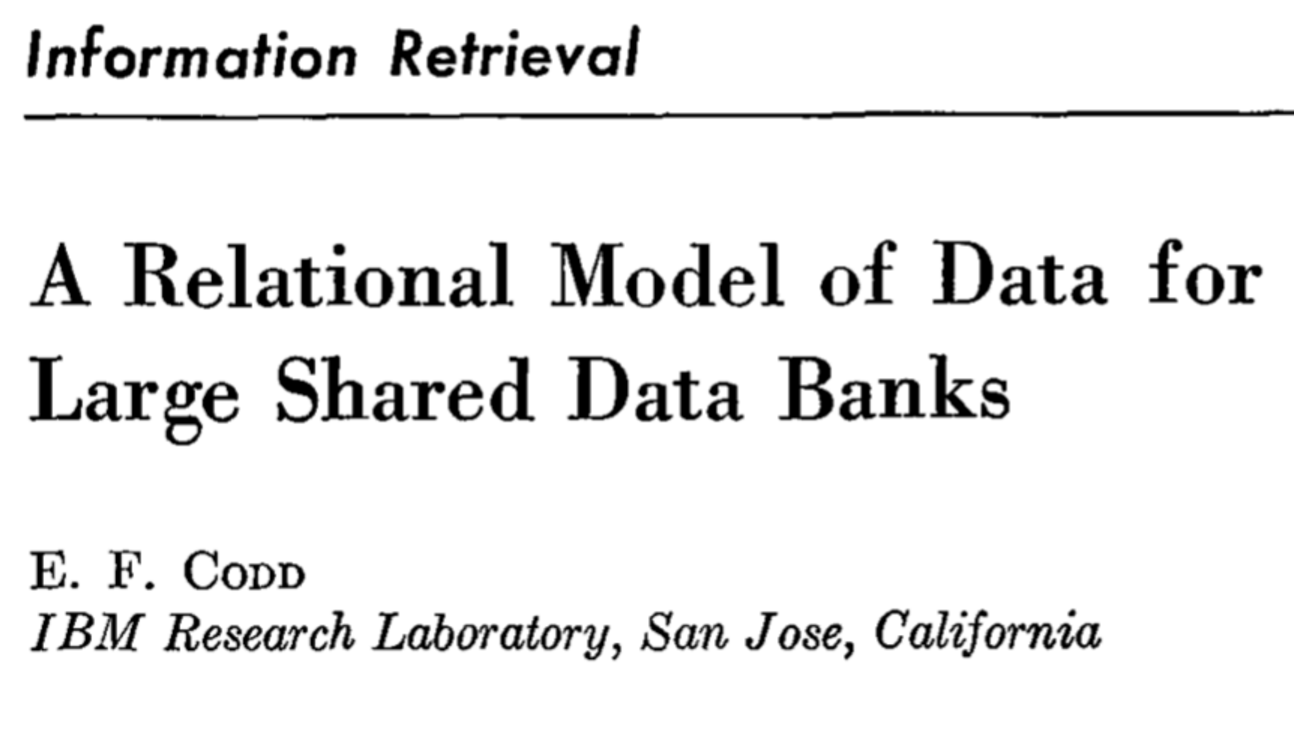
The **Relational Data** Model

A powerful, general, widely-used representation of enterprise data



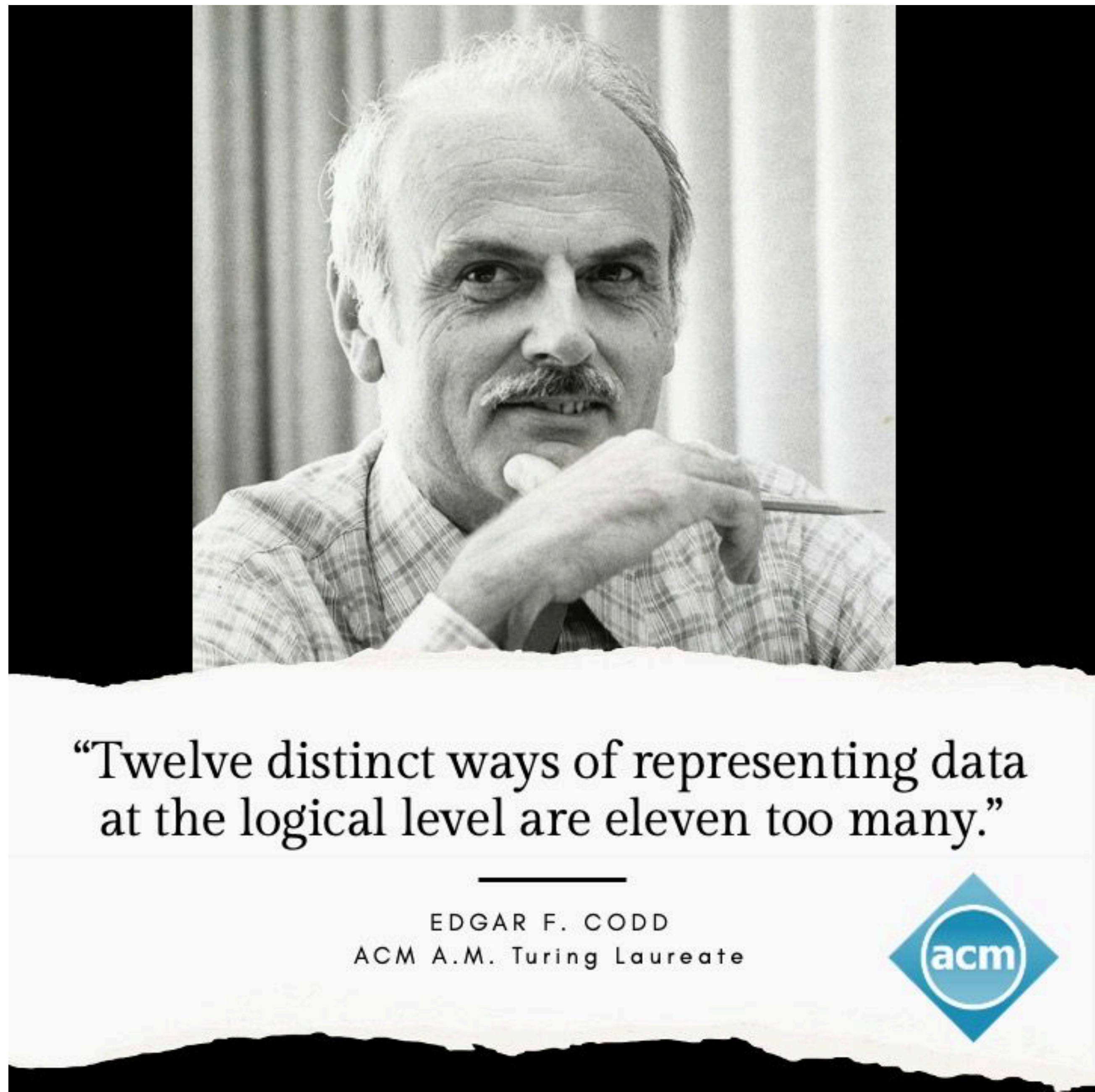
Foreign keys

Primary keys


Information Retrieval

A Relational Model of Data for Large Shared Data Banks

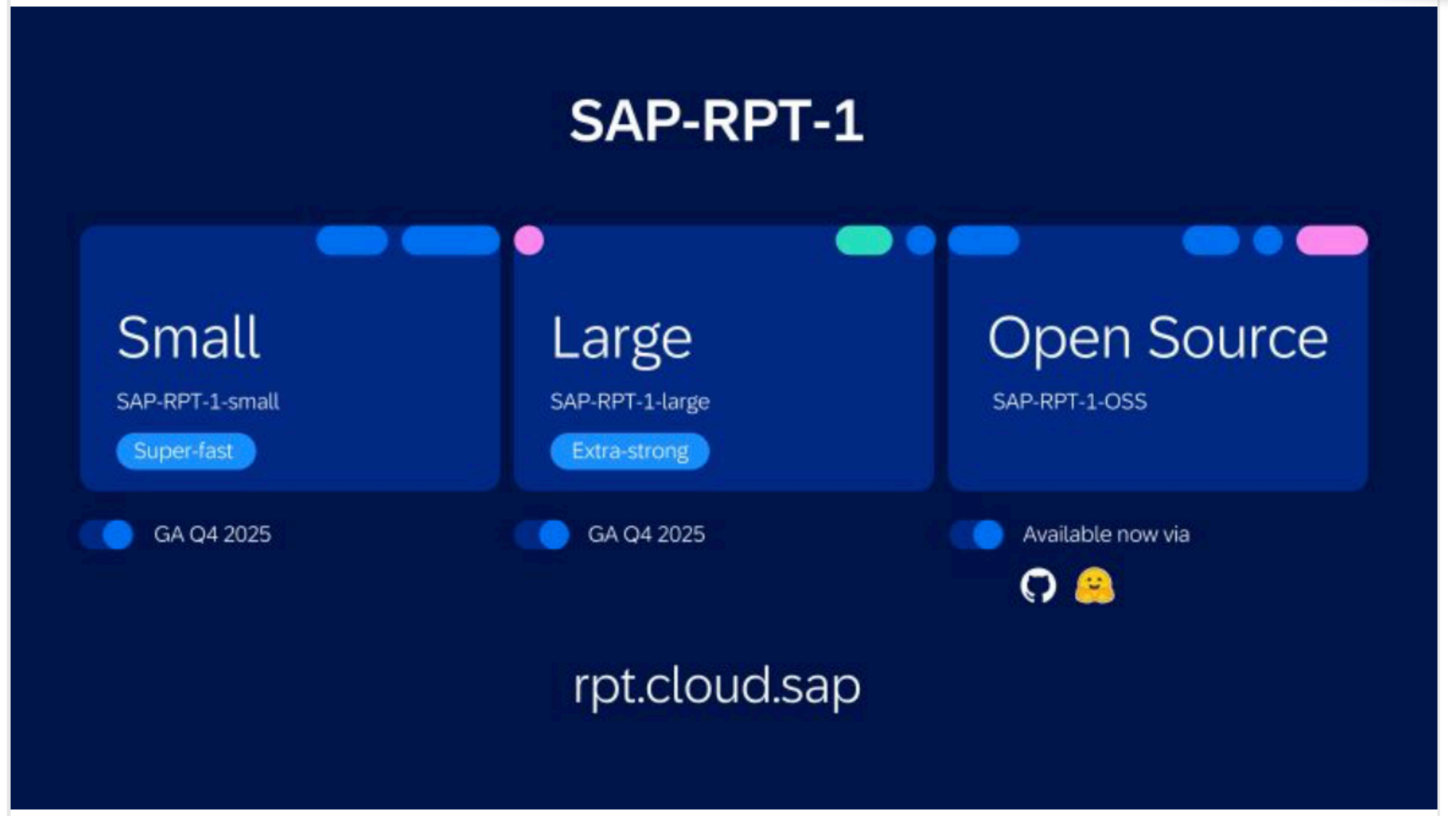
E. F. Codd
IBM Research Laboratory, San Jose, California



- Relational data ⊇ tabular data, time series data, graph data
- Existing FMs for tabular, time-series or graph data don't work

Recent Relational Foundation Model Works

RELATIONAL TRANSFORMER: TOWARD ZERO-SHOT FOUNDATION MODELS FOR RELATIONAL DATA



KumoRFM: A Foundation Model for In-Context Learning on Relational Data

Griffin: Towards a Graph-Centric Relational Database Foundation Model

Yanbo Wang [1 †] Xiyuan Wang [1] Quan Gan [2] Minjie Wang [2] Qibin Yang [1 †] David Wipf [2] Muhan Zhang [1]

Dr. Philipp Herzig · Following
Chief Technology Officer at SAP SE
4w · Edited ·

SAP just dropped a new category of AI models 🤯

Introducing: SAP RPT-1 (pronounced: /ˈræp.ɪd wʌn/)!

RPT stands for Relational Pretrained Transformer to capture its true nature: built specifically on relational data structures for business.

Why is this groundbreaking? Large Language Models struggle with a general understanding of table structures and associated semantics which requires us to use good-old machine learning (aka "narrow AI") for tasks like classification, regression and more. But classical machine learning requires to train a model per task which easily can lead to hundreds of separate models.

RPT-1 puts them all into one single, pre-trained model that understands relational business data, predicts business outcomes, and reduces effort from weeks to days (and we're working to get it down to just minutes.)

At #SAPTechEd, I was excited to announce that we're creating three versions tailored to different business needs:

✔️ SAP RPT-1 Small – optimized for speed and efficiency
✔️ SAP RPT-1 Large – designed for maximum accuracy.
✔️ SAP RPT-1 OSS (Open Source) – available now, giving you full access to explore, learn, and build.

Zero-Shot Relational Learning

Motivations and Examples

- Not every database user can afford custom model development:
  - resources (time, compute)
  - expertise (skilled data scientists)

- E.g., restaurant wants to *estimate future in-stock ingredients*,
  school wants to *flag students at risk of failing*,
  developer wants to add *autocomplete feature* to DB-backed web forms

- Even at larger orgs, data scientists can prototype modeling approaches,
  e.g., *does removing location data hurt sales forecasting?*

- Teams can prioritize high-stakes models (e.g., *loan defaults*),
  leaving less critical ones (e.g., *targeted offers*) to zero-shot learning

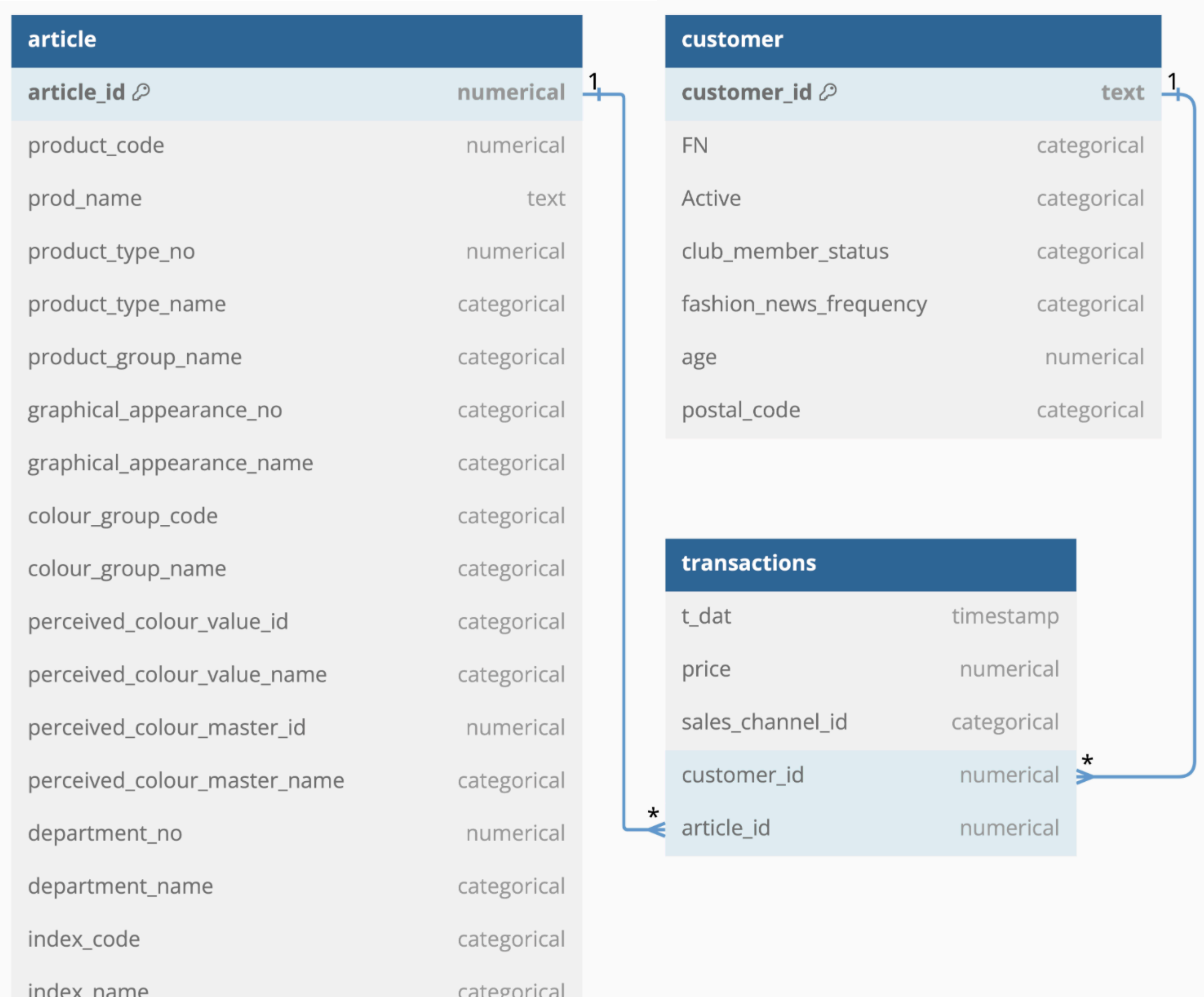
Problem Formulation

- Definition:
  - Predict a **new target** on **new DB** (with **new schema**)
  - Without any DB- or task-specific training
  - All info about new DB and task are supplied via the context window

- Why?
  - Company data is **proprietary** and **sensitive**
  - Company-specific customers, products, etc. $\notin$ pretraining data
  - **Prediction targets** are specific to the **domain** and **application**

Running Example Database and Task

`rel-hm` H&M Database



Num of Tables	3
Num of Rows	33,265,846
Num of Columns	37
Starting Time	2018-09-20
Validation timestamp	2020-09-07
Testing timestamp	2020-09-14
Time window	7 days

Node Regression Tasks

`item-sales`

Task Description: Predict the total sales for an article (the sum of prices of the associated transactions) in the next week

RELBENCH
RELATIONAL DEEP LEARNING BENCHMARK

kaggle

H&M

H&M GROUP · FEATURED PREDICTION COMPETITION · 4 YEARS AGO

H&M Personalized Fashion Recommendations

Provide product recommendations based on previous purchases

Context Window for Zero-Shot Prompting

(a) Tabular visualization for a real example from rel-hm/item-sales

item-sales (task) (39 rows x 3 cols)

timestamp	article_id	sales
2020-06-01	104264	[MASK]
2020-05-25	104264	0.337034
...	...	...
2019-10-07	104264	0.000000
2020-01-13	104264	0.000000

transactions (21 rows x 5 cols)

t_dat	customer_id	article_id	price	sales_channel_id
2020-02-20	29049	2251	0.033881	2
2020-06-01	29049	104264	0.041763	2
...	...	...	...	...
2020-06-01	1247634	104264	0.042356	2
2020-06-01	1247634	104264	0.042356	2

article (14 rows x 25 cols)

article_id	product_code	prod_name	...	garment_group_no	garment_group_name	detail_desc
2251	399223	Curvy Jeggings H...	...	1016	Trousers Denim	Jeggings in wash...
48500	692202	SPEED JAM SHIRT	...	1010	Blouses	Straight-cut blo...
...	...	...	...	...	...	...
101044	893796	Nejlika	...	1005	Jersey Fancy	Body in soft jer...
104264	920700	Dazzle top	...	1005	Jersey Fancy	Wide, slightly s...

customer (7 rows x 7 cols)

customer_id	FN	Active	club_member_status	fashion_news_frequency	age	postal_code
29049	NaN	NaN	ACTIVE	NONE	26.0	5cbf988955d931a7...
178380	NaN	NaN	ACTIVE	NONE	29.0	e777db329cc6dfe3...
...	...	...	...	...	...	...
856256	1.0	1.0	ACTIVE	Regularly	26.0	97cccf90aba93a67...
1247634	NaN	NaN	ACTIVE	NONE	24.0	0c0aaee59a3e86f5...

Context Window for Zero-Shot Prompting

(a) Tabular visualization for a real example from rel-hm/item-sales

item-sales (task) (39 rows x 3 cols)

timestamp	article_id	sales
2020-06-01	104264	[MASK]

Context Window for Zero-Shot Prompting

(a) Tabular visualization for a real example from rel-hm/item-sales

item-sales (task) (39 rows x 3 cols)

timestamp	article_id	sales
2020-06-01	104264	[MASK]

article (14 rows x 25 cols)

article_id	product_code	prod_name	...	garment_group_no	garment_group_name	detail_desc
2251	399223	Curvy Jeggings H...	...	1016	Trousers Denim	Jeggings in wash...
48500	692202	SPEED JAM SHIRT	...	1010	Blouses	Straight-cut blo...
...	...	...	...	...	...	...
101044	893796	Nejlika	...	1005	Jersey Fancy	Body in soft jer...
104264	920700	Dazzle top	...	1005	Jersey Fancy	Wide, slightly s...

Context Window for Zero-Shot Prompting

(a) Tabular visualization for a real example from rel-hm/item-sales

item-sales (task) (39 rows x 3 cols)

timestamp	article_id	sales
2020-06-01	104264	[MASK]
2020-05-25	104264	0.337034
...	...	...
2019-10-07	104264	0.000000
2020-01-13	104264	0.000000

transactions (21 rows x 5 cols)

t_dat	customer_id	article_id	price	sales_channel_id
2020-02-20	29049	2251	0.033881	2
2020-06-01	29049	104264	0.041763	2
...	...	...	...	...
2020-06-01	1247634	104264	0.042356	2
2020-06-01	1247634	104264	0.042356	2

article (14 rows x 25 cols)

article_id	product_code	prod_name	...	garment_group_no	garment_group_name	detail_desc
2251	399223	Curvy Jeggings H...	...	1016	Trousers Denim	Jeggings in wash...
48500	692202	SPEED JAM SHIRT	...	1010	Blouses	Straight-cut blo...
...	...	...	...	...	...	...
101044	893796	Nejlika	...	1005	Jersey Fancy	Body in soft jer...
104264	920700	Dazzle top	...	1005	Jersey Fancy	Wide, slightly s...

Context Window for Zero-Shot Prompting

(a) Tabular visualization for a real example from rel-hm/item-sales

item-sales (task) (39 rows x 3 cols)

timestamp	article_id	sales
2020-06-01	104264	[MASK]
2020-05-25	104264	0.337034
...	...	...
2019-10-07	104264	0.000000
2020-01-13	104264	0.000000

transactions (21 rows x 5 cols)

t_dat	customer_id	article_id	price	sales_channel_id
2020-02-20	29049	2251	0.033881	2
2020-06-01	29049	104264	0.041763	2
...	...	...	...	...
2020-06-01	1247634	104264	0.042356	2
2020-06-01	1247634	104264	0.042356	2

article (14 rows x 25 cols)

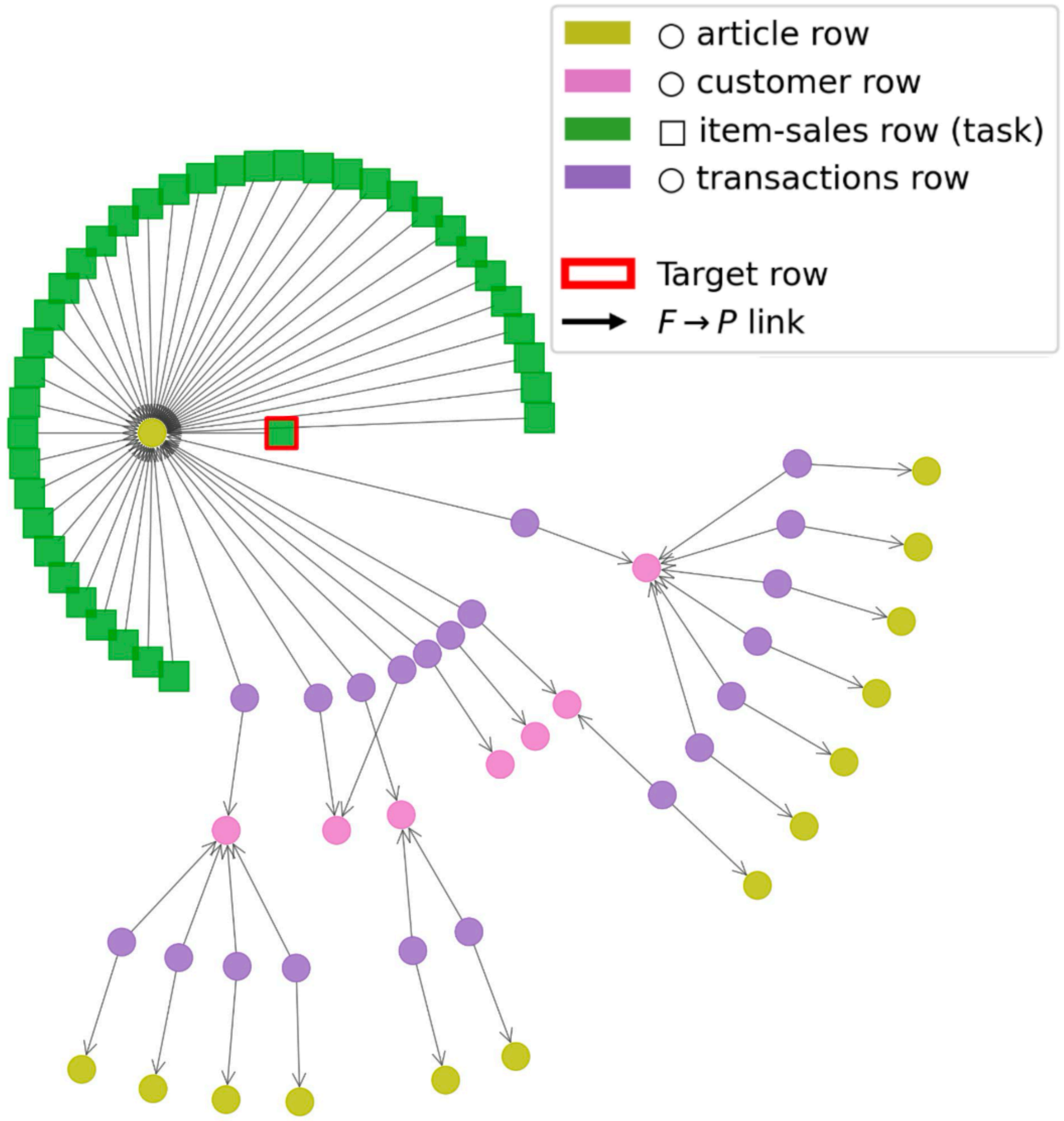
article_id	product_code	prod_name	...	garment_group_no	garment_group_name	detail_desc
2251	399223	Curvy Jeggings H...	...	1016	Trousers Denim	Jeggings in wash...
48500	692202	SPEED JAM SHIRT	...	1010	Blouses	Straight-cut blo...
...	...	...	...	...	...	...
101044	893796	Nejlika	...	1005	Jersey Fancy	Body in soft jer...
104264	920700	Dazzle top	...	1005	Jersey Fancy	Wide, slightly s...

customer (7 rows x 7 cols)

customer_id	FN	Active	club_member_status	fashion_news_frequency	age	postal_code
29049	NaN	NaN	ACTIVE	NONE	26.0	5cbf988955d931a7...
178380	NaN	NaN	ACTIVE	NONE	29.0	e777db329cc6dfe3...
...	...	...	...	...	...	...
856256	1.0	1.0	ACTIVE	Regularly	26.0	97cccf90aba93a67...
1247634	NaN	NaN	ACTIVE	NONE	24.0	0c0aaee59a3e86f5...

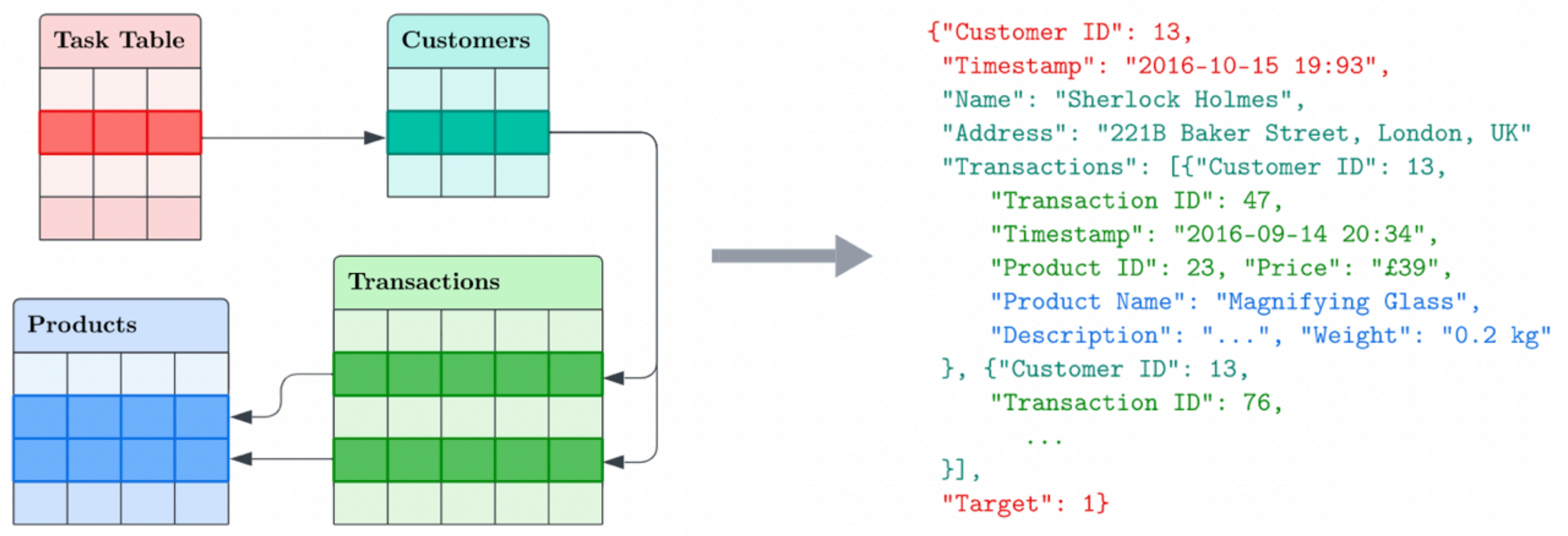
Context Window for Zero-Shot Prompting

(b) Graph visualization for the same example



Wait, let's just use LLMs?

- Database prediction is NOT sequence modeling
- LLMs don't capture relational structure effectively
- Text serialization requires large context lengths



from the paper "Tackling prediction tasks in relational databases with LLMs"

Example prompt using above serialization ↓
(context from last slide ~ 10k tokens)

{"timestamp": "2020-09-14T00:00:00", "article_id": 14075, "product_code": 561445, "prod_name": "Billie.", "product_type_no": 255, "product_type_name": "T-shirt", "product_group_name": "Garment Upper body", "graphical_appearance_no": 1010010, "graphical_appearance_name": "Melange", "colour_group_code": 71, "colour_group_name": "Light Blue", "perceived_colour_value_id": 1, "perceived_colour_value_name": "Dusty Light", "perceived_colour_master_id": 2, "perceived_colour_master_name": "Blue", "department_no": 1676, "department_name": "Jersey Basic", "index_code": "A", "index_name": "Ladieswear", "index_group_no": 1, "index_group_name": "Ladieswear", "section_no": 16, "section_name": "Womens Everyday Basics", "garment_group_no": 1002, "garment_group_name": "Jersey Basic", "detail_desc": "Short-sleeved, fitted jersey top with a slightly deeper neckline.", "add_transactions": [{"t_dat": "2019-09-18T00:00:00", "customer_id": 504738, "article_id": 14075, "price": 0.0067627118644067, "sales_channel_id": 1, "FN": 1.0, "Active": 1.0, "club_member_status": "ACTIVE", "fashion_news_frequency": "Regularly", "age": 55.0, "postal_code": "0ea90102716f47a3b56a635d91b086c25c3e8e125ff32c662683da50dd9afc77", "add_transactions": [{"t_dat": "2020-02-10T00:00:00", "customer_id": 504738, "article_id": 55514, "price": 0.036, "sales_channel_id": 1, "product_code": 710876, "prod_name": "Kennedy skinny stretch chinos", "product_type_no": 272, "product_type_name": "Trousers", "product_group_name": "Garment Lower body", "graphical_appearance_no": 1010016, "graphical_appearance_name": "Solid", "colour_group_code": 13, "colour_group_name": "Beige", "perceived_colour_value_id": 2, "perceived_colour_value_name": "Medium Dusty", "perceived_colour_master_id": 11, "perceived_colour_master_name": "Beige", "department_no": 5656, "department_name": "Trouser", "index_code": "F", "index_name": "Menswear", "index_group_no": 3, "index_group_name": "Menswear", "section_no": 21, "section_name": "Contemporary Casual", "garment_group_no": 1009, "garment_group_name": "Trousers", "detail_desc": "Chinos in stretch cotton twill with a zip fly and button, diagonal side pockets and welt back pockets with a button. Skinny Fit – a fit with slightly "}]}, {"t_dat": "2020-07-28T00:00:00", "customer_id": 1142649, "article_id": 14075, "price": 0.0067627118644067, "sales_channel_id": 1, "FN": 1.0, "Active": 1.0, "club_member_status": "ACTIVE", "fashion_news_frequency": "Regularly", "age": 39.0, "postal_code": "e8a7c9a0cad71040559431af44f21252133e48e28a93f431095a3d54f43397b6", "add_transactions": [{"t_dat": "2020-05-29T00:00:00", "customer_id": 1142649, "article_id": 92085, "price": 0.0355762711864406, "sales_channel_id": 1, "product_code": 852420, "prod_name": "Jalapeno trousers", "product_type_no": 272, "product_type_name": "Trousers", "product_group_name": "Garment Lower body", "graphical_appearance_no": 1010023, "graphical_appearance_name": "Denim", "colour_group_code": 71, "colour_group_name": "Light Blue", "perceived_colour_value_id": 1, "perceived_colour_value_name": "Dusty Light", "perceived_colour_master_id": 2, "perceived_colour_master_name": "Blue", "department_no": 1710, "department_name": "Trouser", "index_code": "A", "index_name": "Ladieswear", "index_group_no": 1, "index_group_name": "Ladieswear", "section_no": 6, "section_name": "Womens Casual", "garment_group_no": 1009, "garment_group_name": "Trousers", "detail_desc": "Ankle-length trousers in soft Tencel™ lyocell denim with a high waist, detachable tie belt, zip fly with a concealed hook-and-eye fastening and wide, "}, {"t_dat": "2020-02-15T00:00:00", "customer_id": 1142649, "article_id": 47294, "price": 0.0106610169491525, "sales_channel_id": 2, "product_code": 688537, "prod_name": "Simple as that Cheeky Tanga", "product_type_no": 59, "product_type_name": "Swimwear bottom", "product_group_name": "Swimwear", "graphical_appearance_no": 1010026, "graphical_appearance_name": "Other structure", "colour_group_code": 42, "colour_group_name": "Red", "perceived_colour_value_id": 5, "perceived_colour_value_name": "Bright", "perceived_colour_master_id": 18, "perceived_colour_master_name": "Red", "department_no": 4242, "department_name": "Swimwear", "index_code": "B", "index_name": "Lingeries/Tights", "index_group_no": 1, "index_group_name": "Ladieswear", "section_no": 60, "section_name": "Womens Swimwear, beachwear", "garment_group_no": 1018, "garment_group_name": "Swimwear", "detail_desc": "Fully lined bikini bottoms with a mid waist and cutaway coverage at the back."}, {"t_dat": "2020-06-04T00:00:00", "customer_id": 1142649, "article_id": 85156, "price": 0.0129491525423728, "sales_channel_id": 1, "product_code": 818162, "prod_name": "Pat ls bd plaid", "product_type_no": 259, "product_type_name": "Shirt", "product_group_name": "Garment Upper body", "graphical_appearance_no": 1010004, "graphical_appearance_name": "Check", "colour_group_code": 23, "colour_group_name":

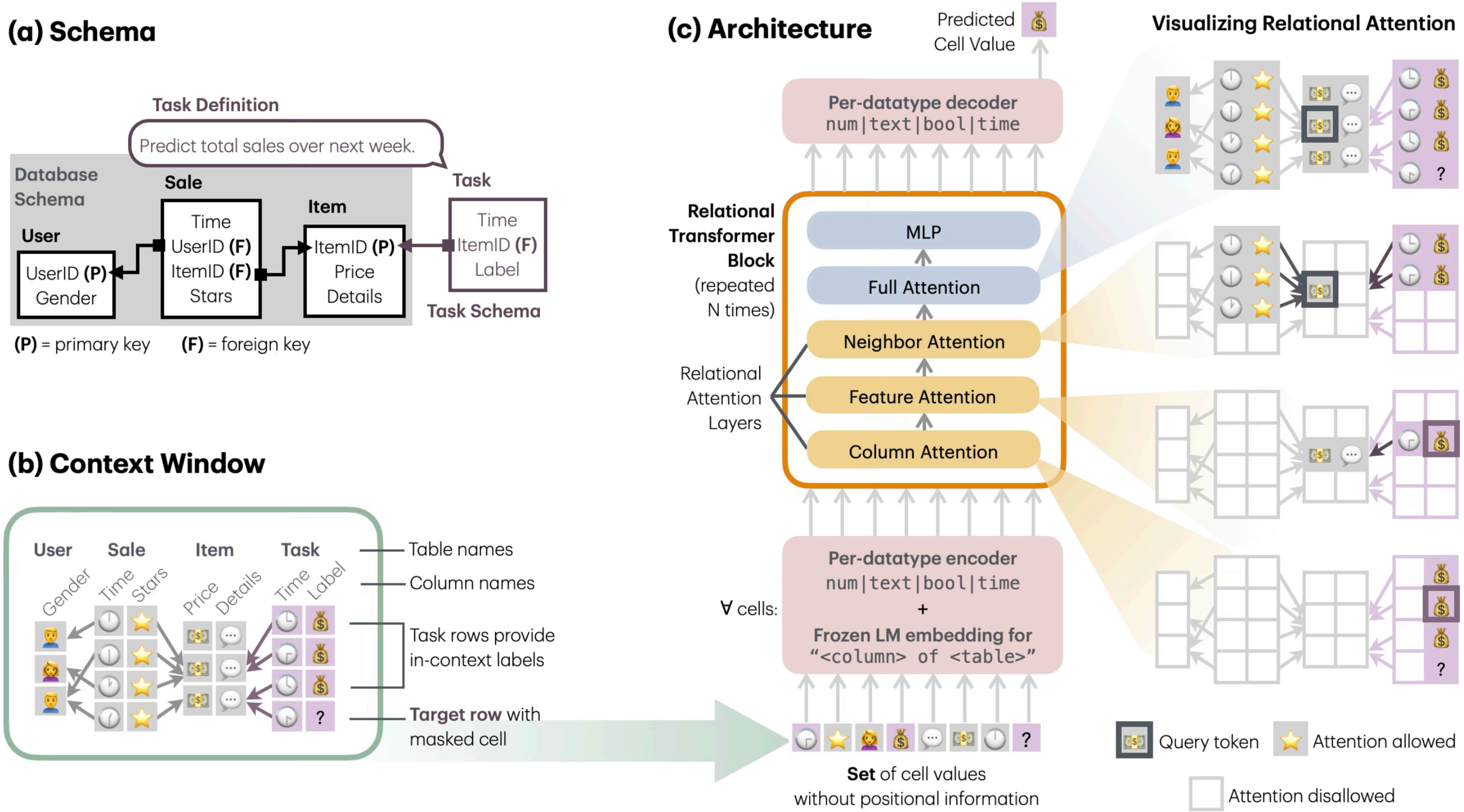
A Transformer designed for Relational Data

Design Principles

- Effectively capture **relational structure**
- Support flexible **self-supervised pretraining**
- Enable **schema-agnostic** zero-shot generalization

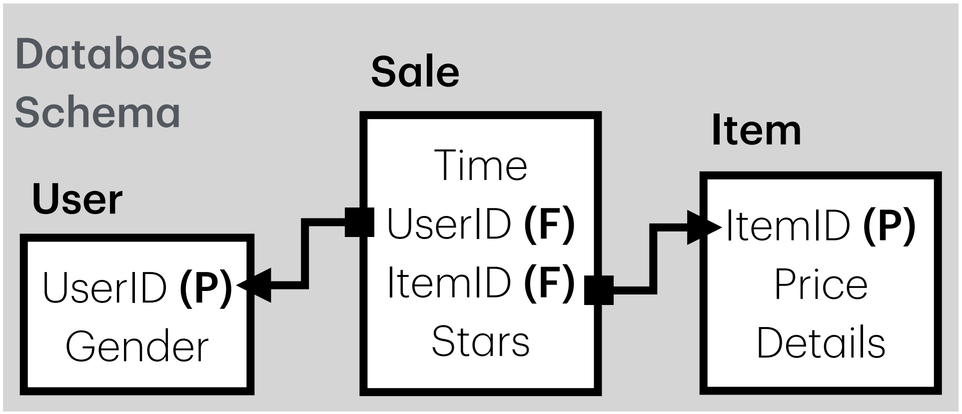
Our Innovations

- Effectively capture **relational structure**:
  - novel **Relational Attention** layers
- Support flexible **self-supervised pretraining**
  - **cell-level tokenization**
  - **masked token prediction** pretraining
- Enable **schema-agnostic** zero-shot generalization
  - **task table prompting** frames any task as masked token prediction
  - cell datatypes—**num/text/bool/datetime**—are universal
  - Relational Attention primitives—**rows, cols, foreign keys, primary keys**—are universal



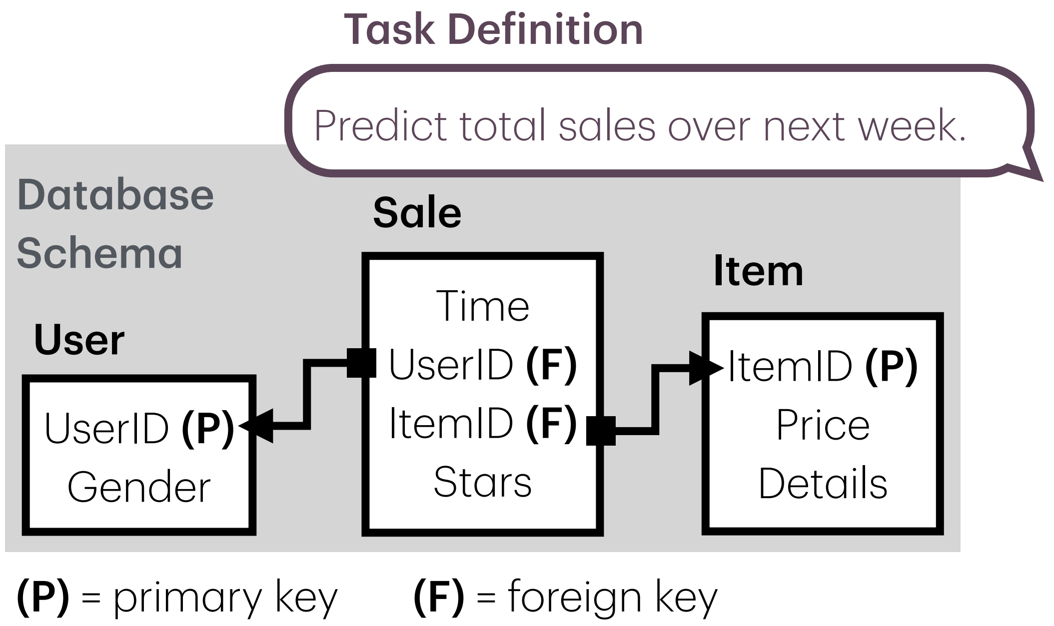
Overview of the ***Relational Transformer (RT)*** *architecture*

(a) Schema

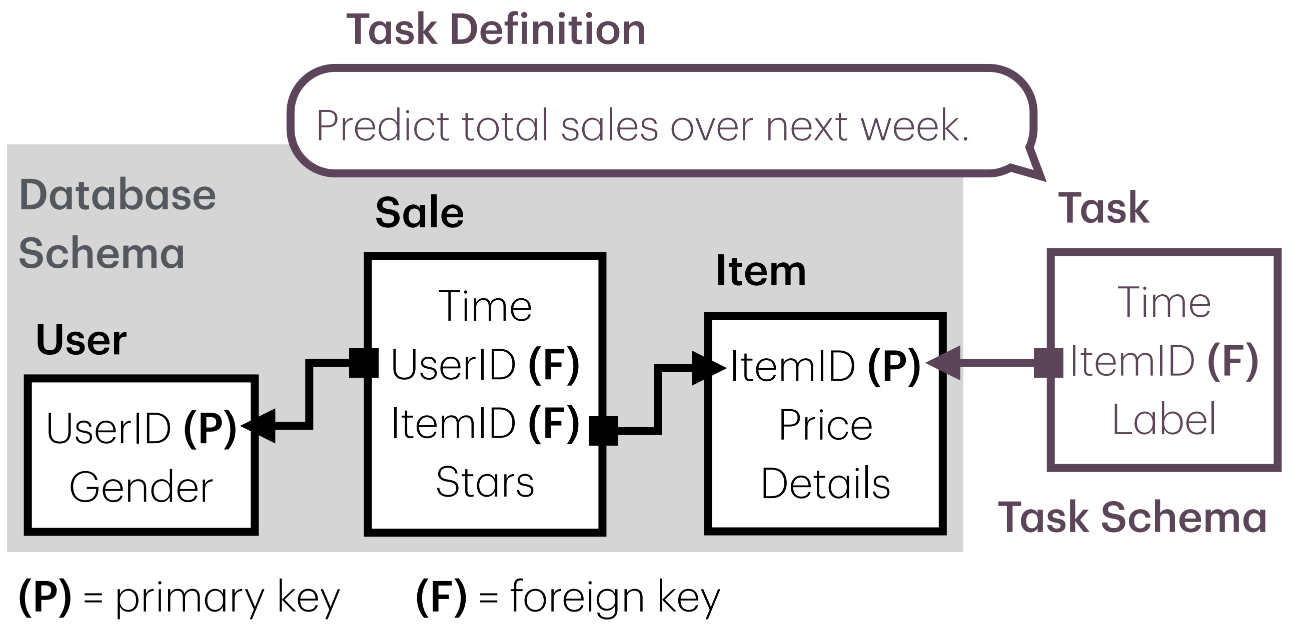


(P) = primary key **(F)** = foreign key

(a) Schema


Task Definition
Predict total sales over next week.
Database
Schema
User
UserID (P)
Gender
Sale
Time
UserID (F)
ItemID (F)
Stars
Item
ItemID (P)
Price
Details
(P) = primary key
(F) = foreign key

(a) Schema



Task Table Prompting: Task is specified by adding a new table to the DB
(Model treats task table as any other table)

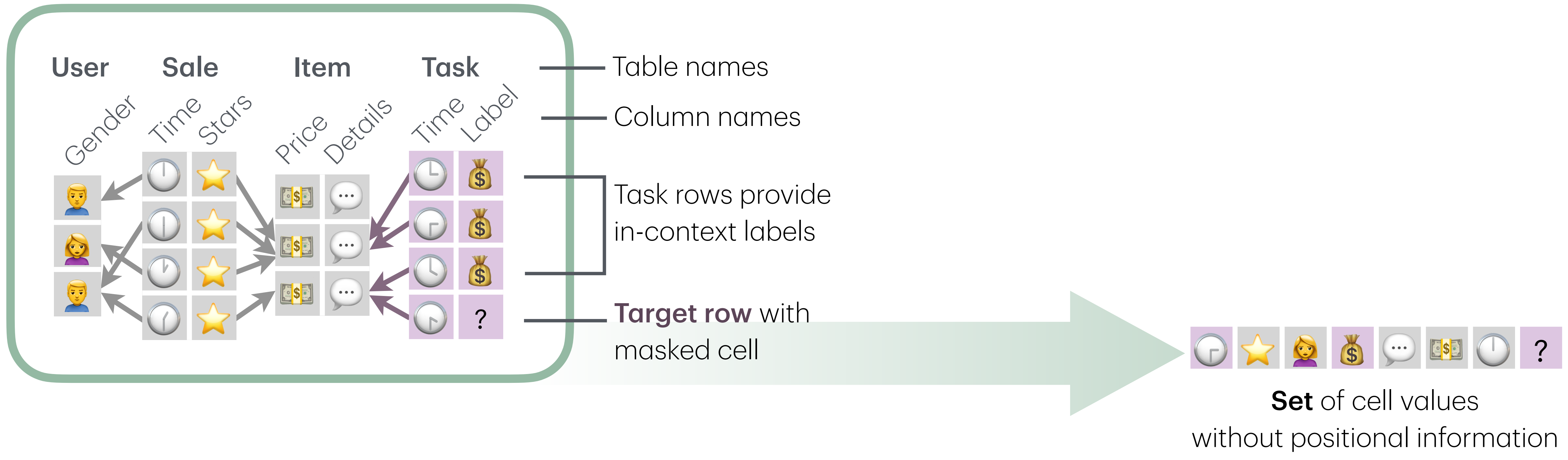

(a) Schema
Task Definition
Predict total sales over next week.
Database Schema
Sale
Time
UserID (F)
ItemID (F)
Stars
Item
ItemID (P)
Price
Details
Task
Time
ItemID (F)
Label
User
UserID (P)
Gender
Task Schema
(P) = primary key
(F) = foreign key
(b) Context Window
User
Sale
Item
Task
Gender
Time
Stars
Price
Details
Time
Label
?
Target row with masked cell

(a) Schema
Task Definition
Predict total sales over next week.
Database Schema
User
UserID (P)
Gender
Sale
Time
UserID (F)
ItemID (F)
Stars
Item
ItemID (P)
Price
Details
Task
Time
ItemID (F)
Label
Task Schema
(P) = primary key
(F) = foreign key
(b) Context Window
User
Sale
Item
Task
Gender
Time
Stars
Price
Details
Time
Label
?
Table names
Column names
Target row with masked cell

(a) Schema
Task Definition
Predict total sales over next week.
Database Schema
Sale
Item
Task
User
Time
UserID (F)
ItemID (F)
Stars
ItemID (P)
Price
Details
Time
ItemID (F)
Label
UserID (P)
Gender
Task Schema
(P) = primary key
(F) = foreign key
(b) Context Window
User
Sale
Item
Task
Gender
Time
Stars
Price
Details
Time
Label
?
Table names
Column names
Task rows provide in-context labels
Target row with masked cell

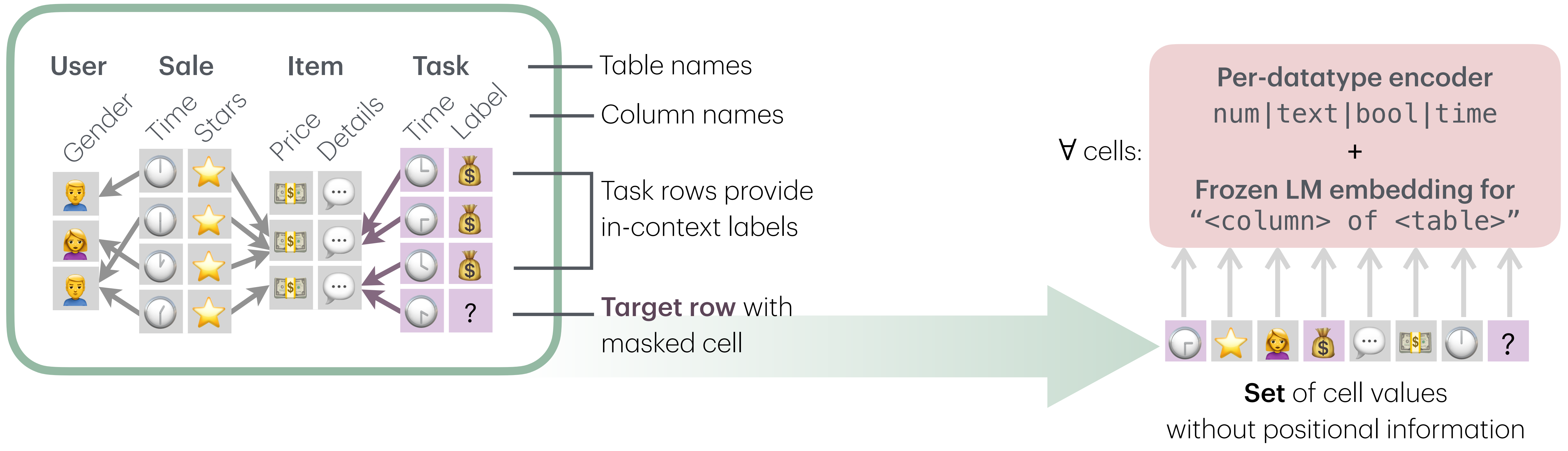
(c) Architecture

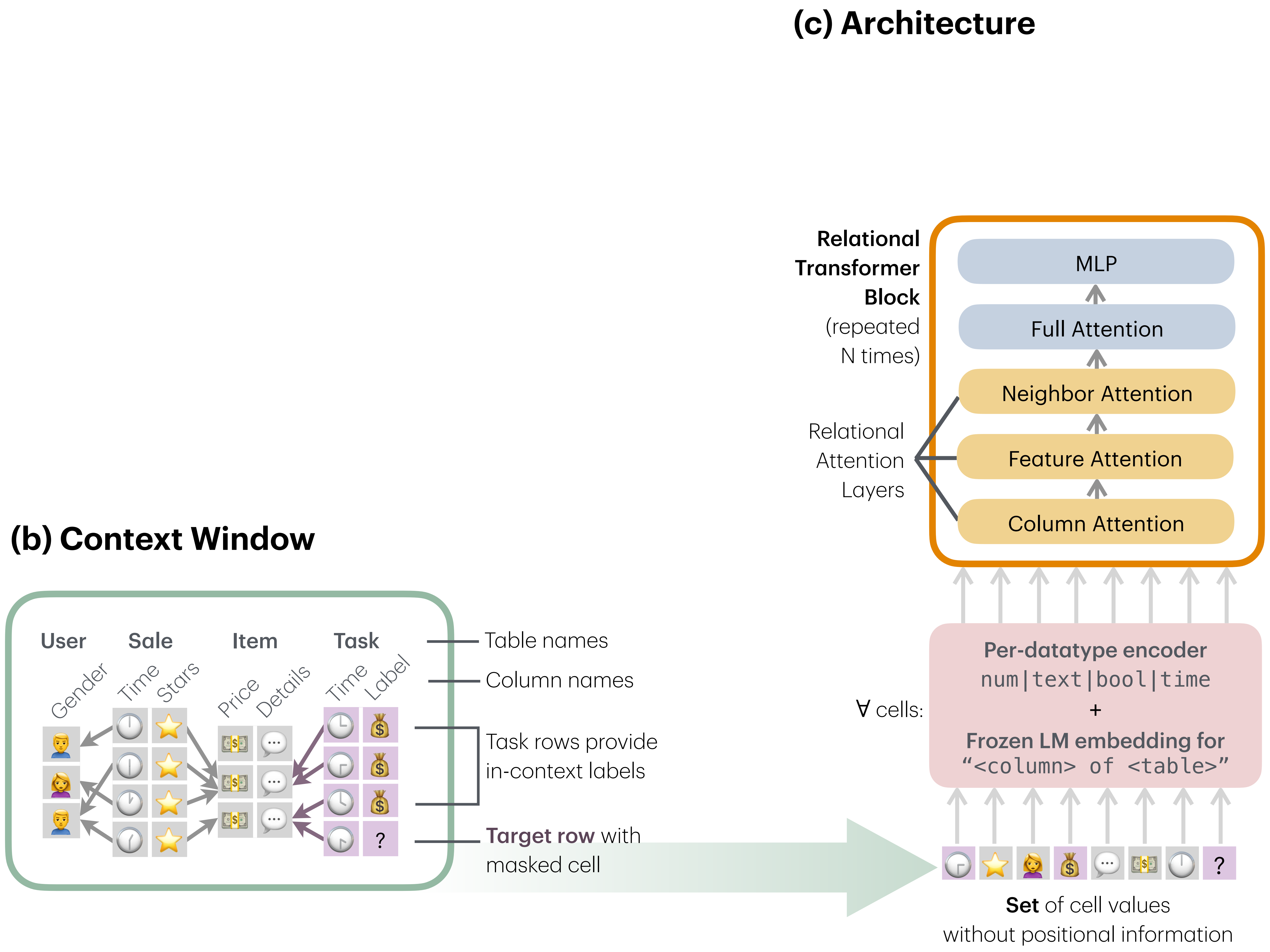
(b) Context Window

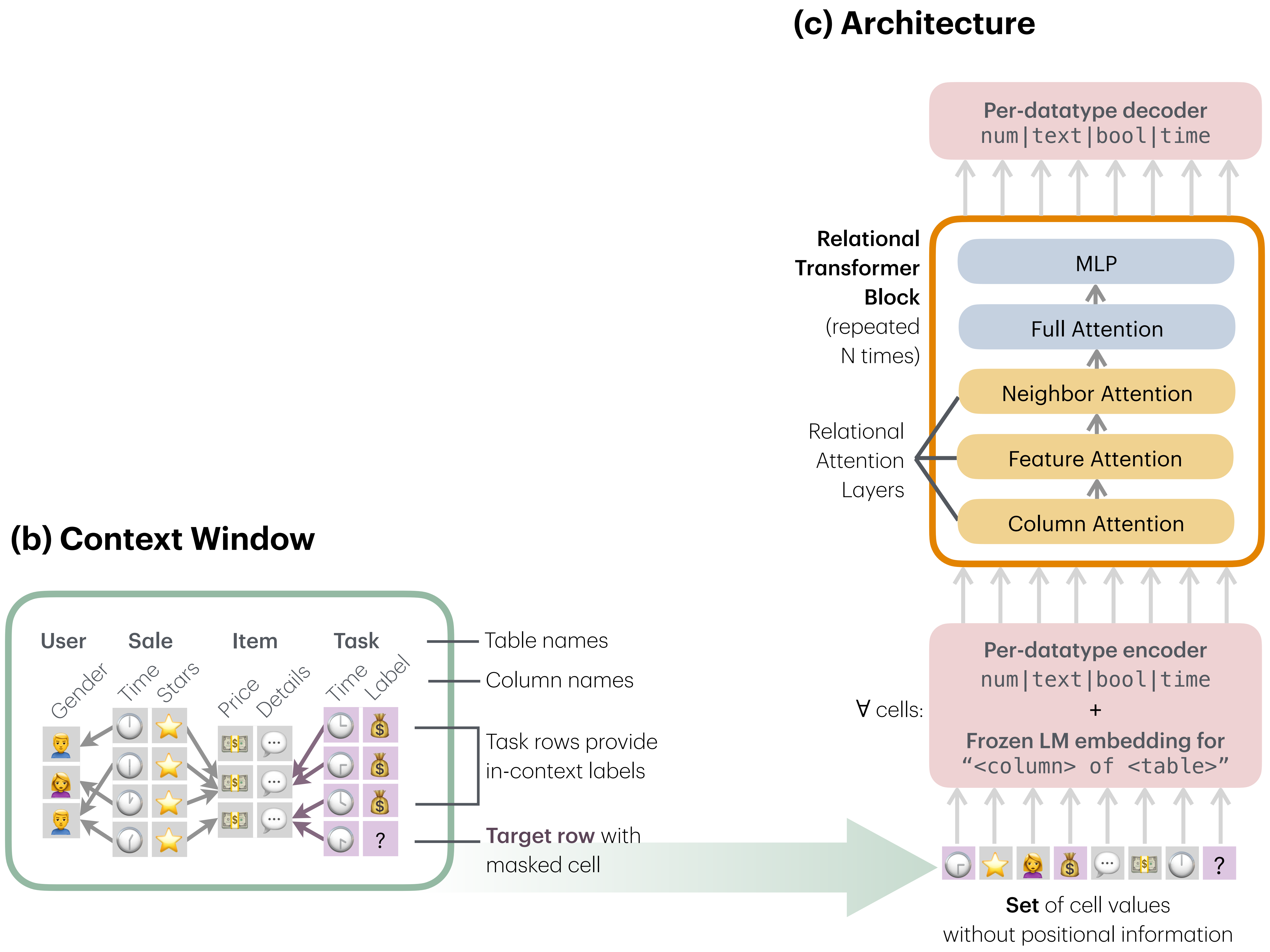


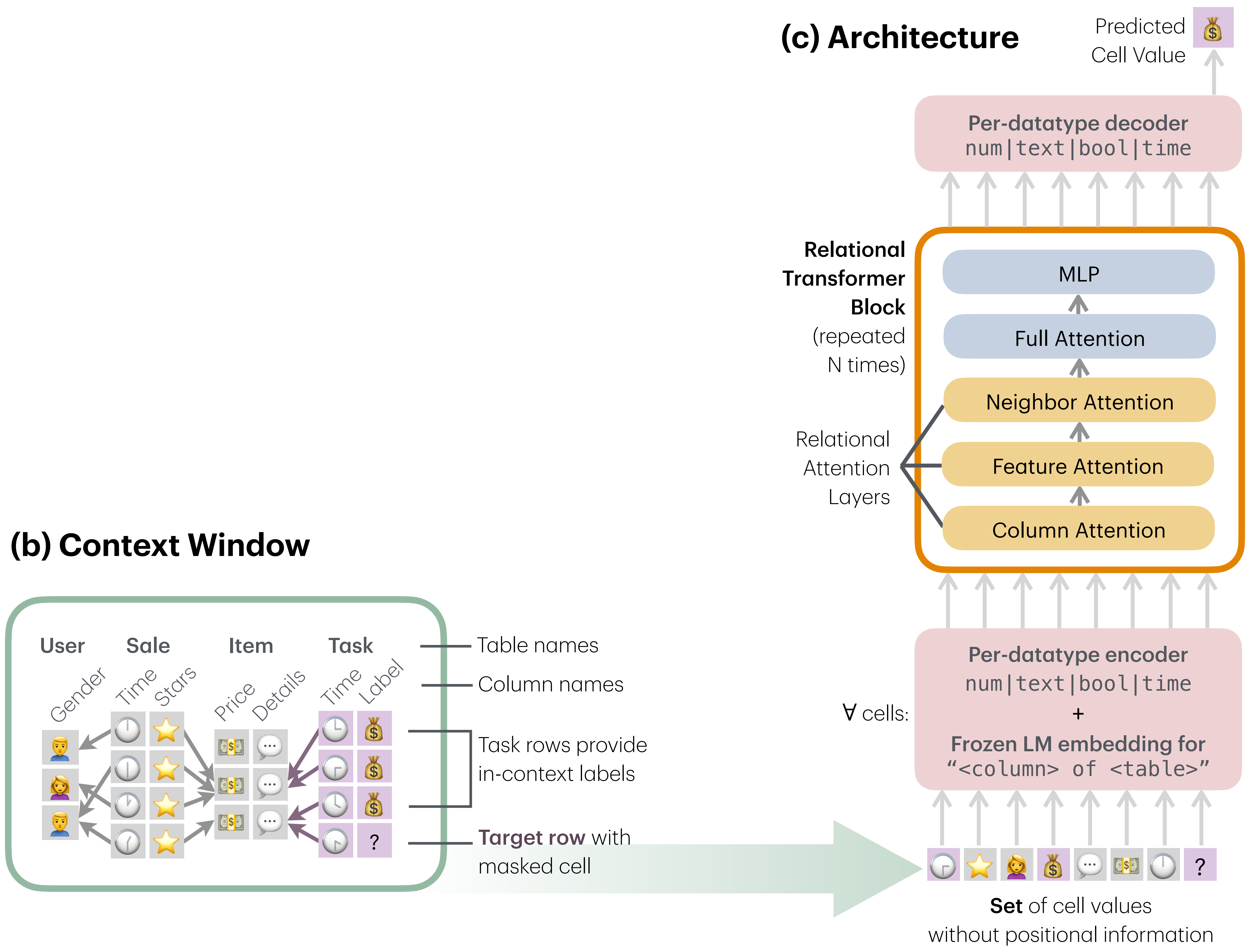
(c) Architecture

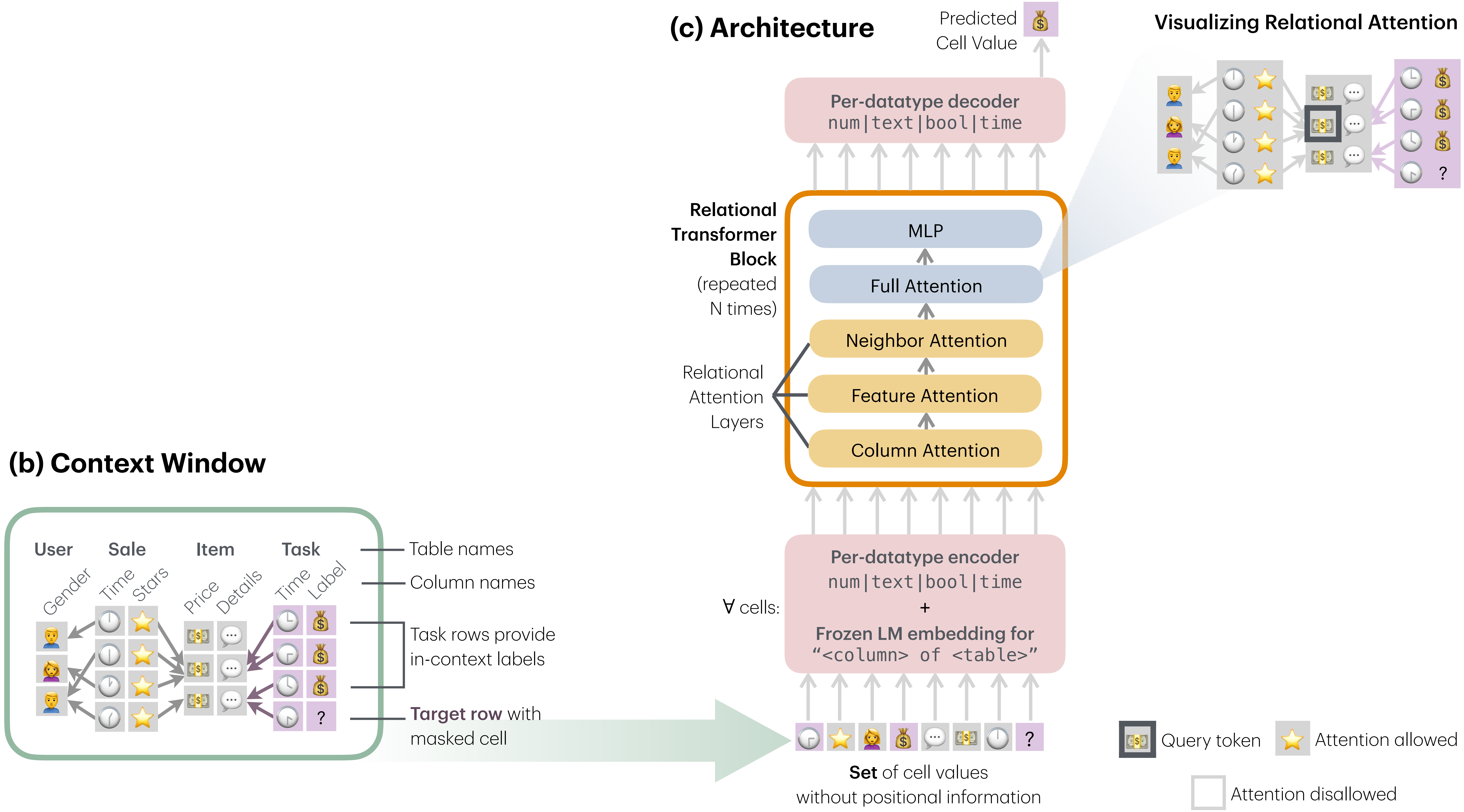
(b) Context Window

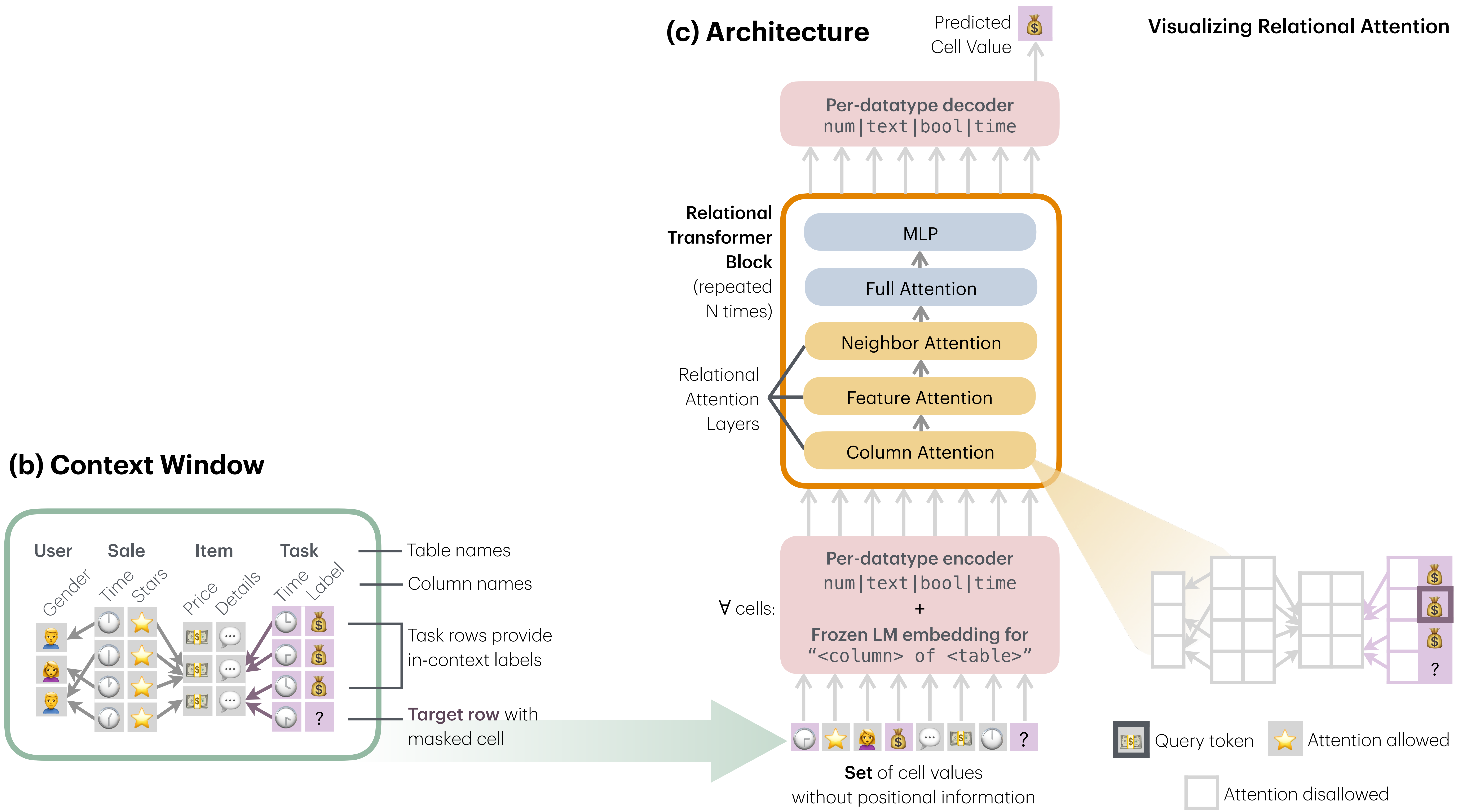



(c) Architecture
Relational Transformer Block (repeated N times)
MLP
Full Attention
Neighbor Attention
Feature Attention
Column Attention
Relational Attention Layers
∀ cells:
Per-datatype encoder
num|text|bool|time
+
Frozen LM embedding for
“<column> of <table>”
Set of cell values without positional information
(b) Context Window
User
Sale
Item
Task
Gender
Time
Stars
Price
Details
Time
Label
?
Table names
Column names
Task rows provide in-context labels
Target row with masked cell

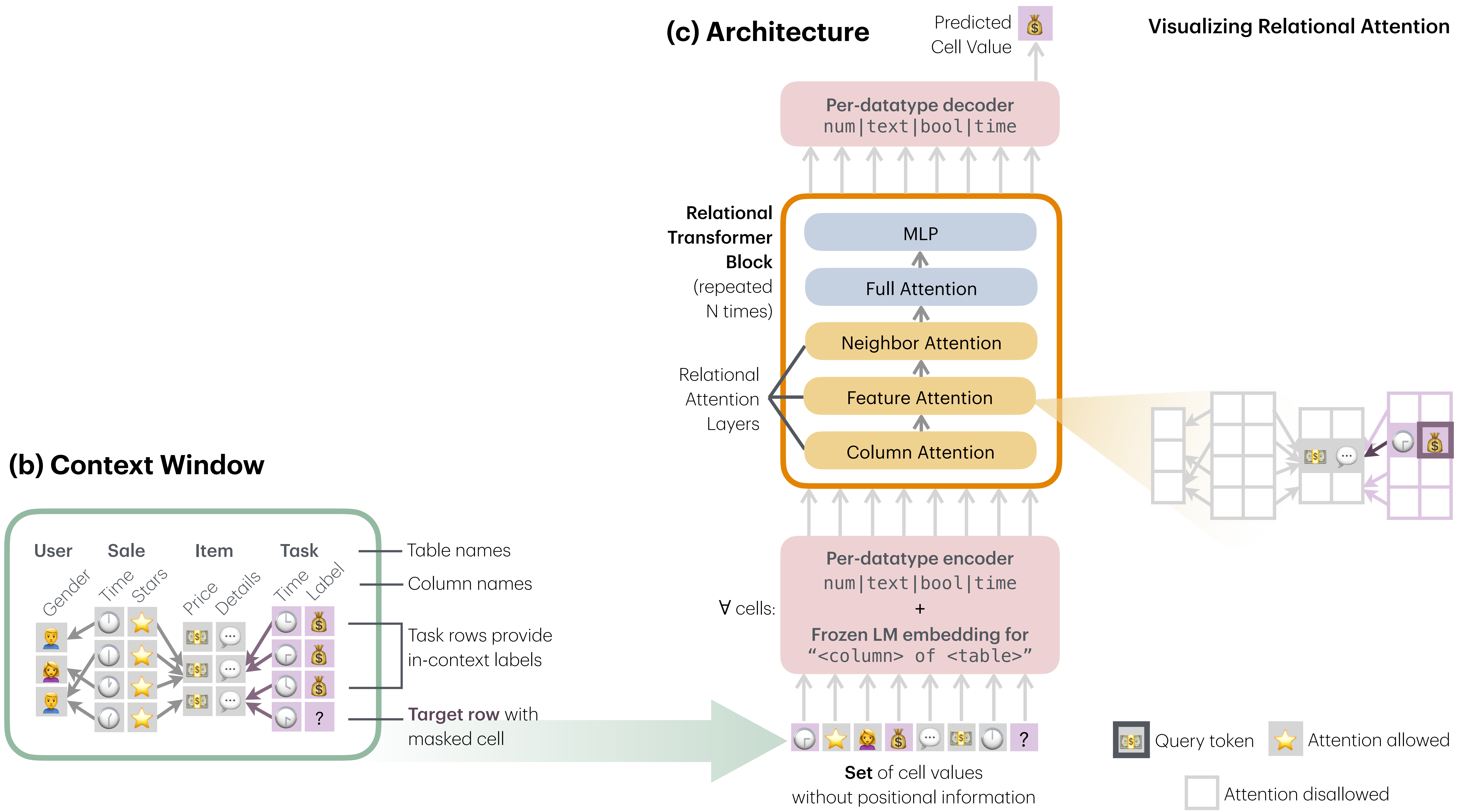

(c) Architecture
Per-datatype decoder
num|text|bool|time
Relational Transformer Block
(repeated N times)
MLP
Full Attention
Neighbor Attention
Feature Attention
Column Attention
Relational Attention Layers
Per-datatype encoder
num|text|bool|time
+
Frozen LM embedding for
“<column> of <table>”
∀ cells:
?
Set of cell values
without positional information
(b) Context Window
User
Sale
Item
Task
Gender
Time
Stars
Price
Details
Time
Label
?
Table names
Column names
Task rows provide
in-context labels
Target row with
masked cell

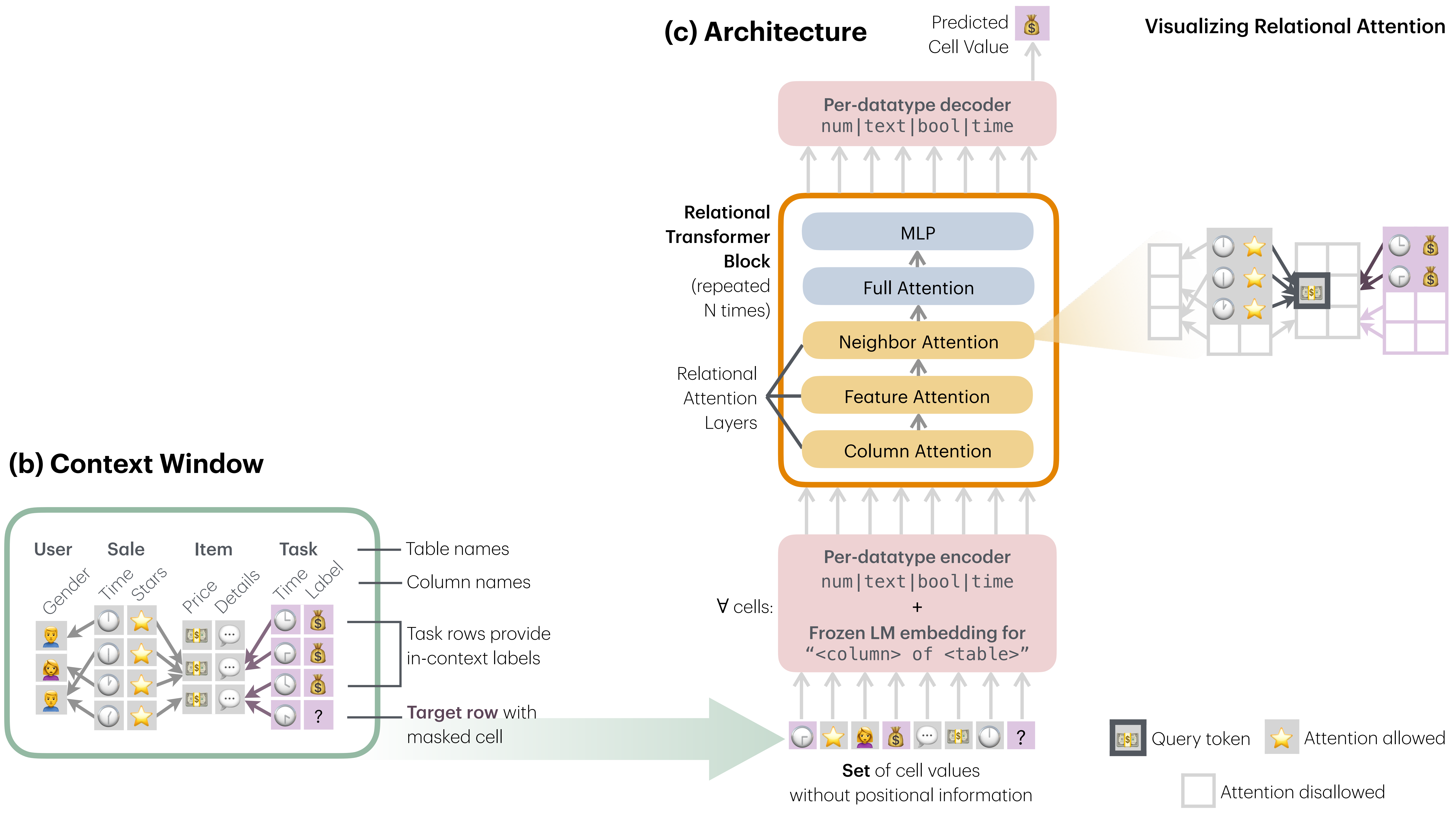

(c) Architecture
Predicted Cell Value
Per-datatype decoder
num|text|bool|time
Relational Transformer Block (repeated N times)
MLP
Full Attention
Neighbor Attention
Feature Attention
Column Attention
Relational Attention Layers
Per-datatype encoder
num|text|bool|time
+
Frozen LM embedding for
“<column> of <table>”
∀ cells:
Set of cell values without positional information
?
(b) Context Window
User
Sale
Item
Task
Gender
Time
Stars
Price
Details
Time
Label
?
Table names
Column names
Task rows provide in-context labels
Target row with masked cell


(c) Architecture
Predicted Cell Value
Visualizing Relational Attention
Per-datatype decoder
num|text|bool|time
Relational Transformer Block (repeated N times)
MLP
Full Attention
Neighbor Attention
Feature Attention
Column Attention
Relational Attention Layers
∀ cells:
Per-datatype encoder
num|text|bool|time
+
Frozen LM embedding for
“<column> of <table>”
Set of cell values
without positional information
Query token
Attention allowed
Attention disallowed
(b) Context Window
User
Sale
Item
Task
Gender
Time
Stars
Price
Details
Time
Label
?
Table names
Column names
Task rows provide
in-context labels
Target row with
masked cell

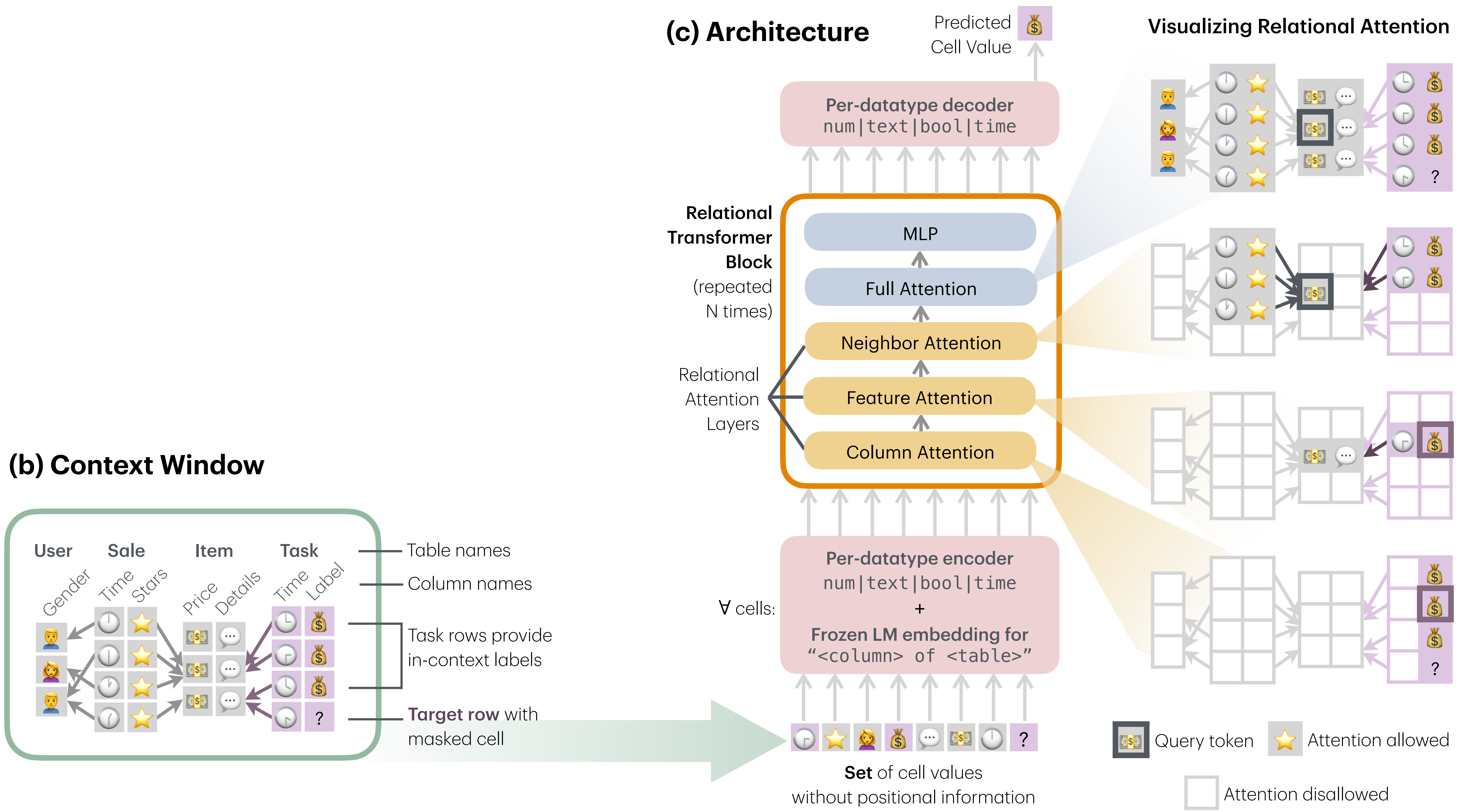


(c) Architecture
Predicted
Cell Value
Visualizing Relational Attention
Per-datatype decoder
num|text|bool|time
Relational
Transformer
Block
(repeated
N times)
MLP
Full Attention
Neighbor Attention
Relational
Attention
Layers
Feature Attention
Column Attention
(b) Context Window
User
Sale
Item
Task
Gender
Time
Stars
Price
Details
Time
Label
Table names
Column names
Task rows provide
in-context labels
Target row with
masked cell
Per-datatype encoder
num|text|bool|time
+
Frozen LM embedding for
“<column> of <table>”
∀ cells:
Set of cell values
without positional information
Query token
Attention allowed
Attention disallowed

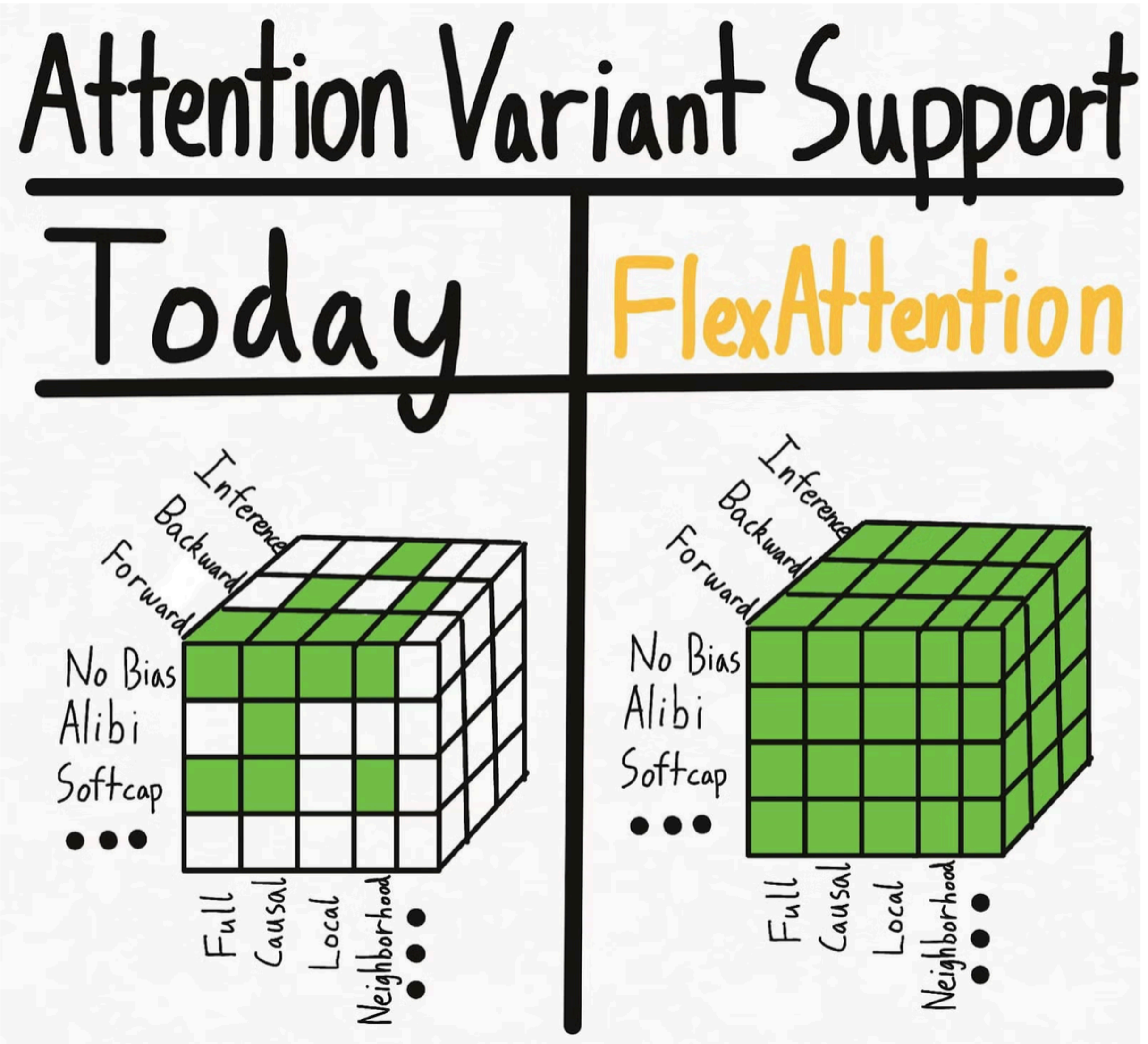

(c) Architecture
Predicted Cell Value
Visualizing Relational Attention
Per-datatype decoder
num|text|bool|time
Relational Transformer Block (repeated N times)
MLP
Full Attention
Neighbor Attention
Feature Attention
Column Attention
Relational Attention Layers
(b) Context Window
User
Sale
Item
Task
Gender
Time
Stars
Price
Details
Time
Label
Table names
Column names
Task rows provide in-context labels
Target row with masked cell
∀ cells:
Per-datatype encoder
num|text|bool|time
+
Frozen LM embedding for
“<column> of <table>”
Set of cell values without positional information
Query token
Attention allowed
Attention disallowed



(c) Architecture
Predicted Cell Value
Visualizing Relational Attention
Per-datatype decoder
num|text|bool|time
Relational Transformer Block (repeated N times)
MLP
Full Attention
Neighbor Attention
Feature Attention
Column Attention
Relational Attention Layers
(b) Context Window
User
Sale
Item
Task
Gender
Time
Stars
Price
Details
Time
Label
Table names
Column names
Task rows provide in-context labels
Target row with masked cell
Per-datatype encoder
num|text|bool|time
+
Frozen LM embedding for
“<column> of <table>”
∀ cells:
Set of cell values without positional information
Query token
Attention allowed
Attention disallowed


(b) Context Window
User
Sale
Item
Task
Gender
Time
Stars
Price
Details
Time
Label
?
Table names
Column names
Task rows provide
in-context labels
Target row with
masked cell
(c) Architecture
Predicted
Cell Value
Per-datatype decoder
num|text|bool|time
Relational
Transformer
Block
(repeated
N times)
MLP
Full Attention
Neighbor Attention
Feature Attention
Column Attention
Relational
Attention
Layers
∀ cells:
Per-datatype encoder
num|text|bool|time
+
Frozen LM embedding for
“<column> of <table>”
?
Set of cell values
without positional information
Visualizing Relational Attention
?
?
Query token
Attention allowed
Attention disallowed

Masked FlashAttention kernels using FlexAttention



FlexAttention: The Flexibility of PyTorch with the Performance of FlashAttention

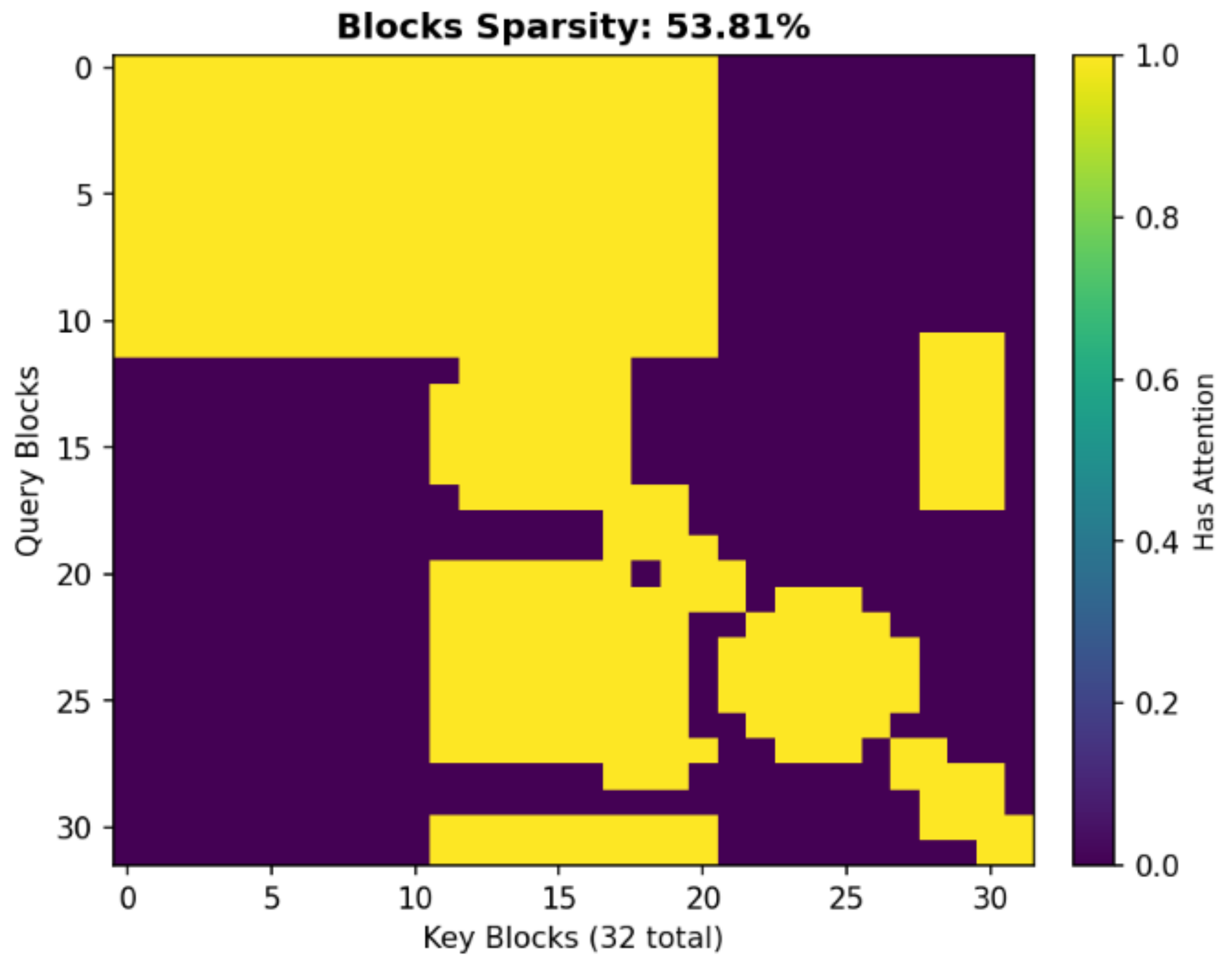
By Team PyTorch: Driss Guessous, Yanbo Liang, Joy Dong, Horace He | August 7, 2024

$$\text{FlexAttention}(Q, K, V) = \text{softmax}\left(\text{score\_mod}\left(\frac{QK^T}{\sqrt{d_k}}\right)\right)V$$

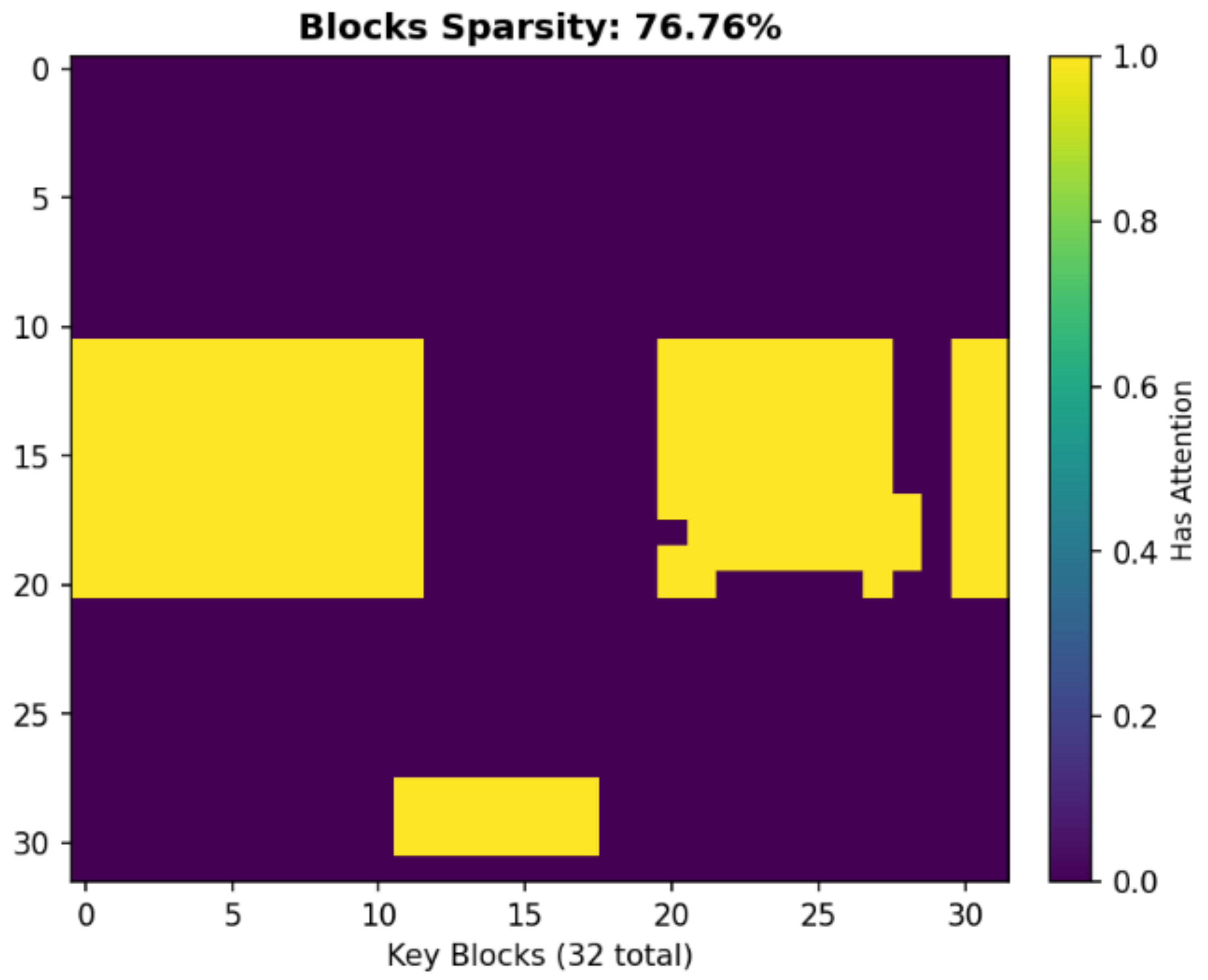
PyTorch

Block sparsity in attention masks

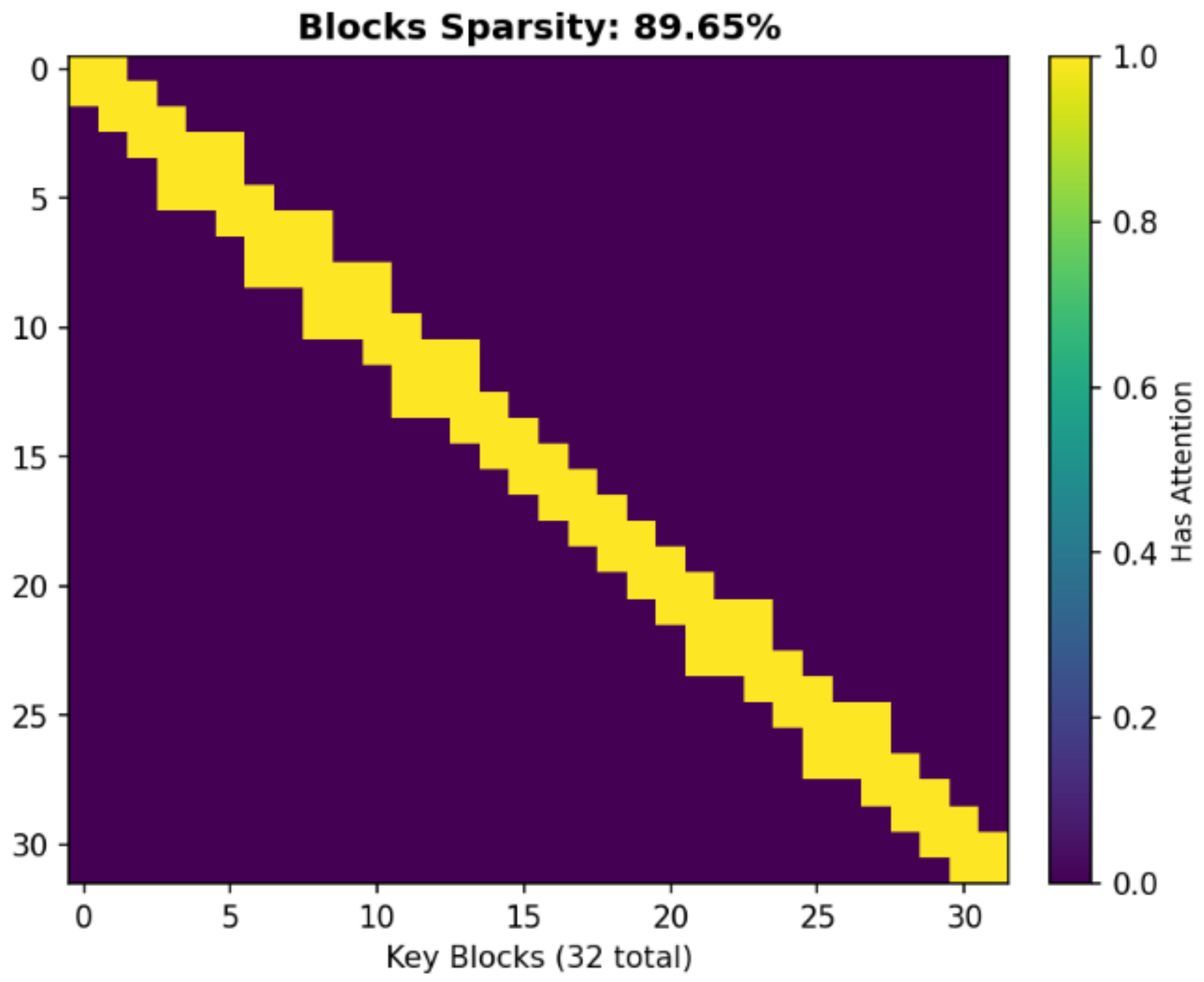
With column-major serialization of tokens



Feature Attention



Neighbor Attention



Column Attention

Pretraining Setup

- Leave-one-DB-out on RelBench
- Masked token prediction objective
- Combined training on classification and regression:
  - **classification loss:** BinaryCrossEntropyLoss
  - **regression loss:** HuberLoss
- Each run sees ~20 tasks from 6 DBs including:
  - **forecasting tasks:** predict a future derived quantity
  - **autocomplete tasks:** predict a future column value
- **22M** parameters, **13B** tokens, **2 hrs** on 8xA100, **1024** ctx_len

RELBENCH
RELATIONAL DEEP LEARNING BENCHMARK

7 Diverse Datasets

E-Commerce
- rel-amazon
- rel-avito
- rel-hm

Social
- rel-event
- rel-stack

Sports
- rel-f1

Medical
- rel-trial

3 to **15** tables

74k to **41M** rows in a DB

15 to **140** columns in a DB

Time span from **2 weeks** to **55 years**

Results

Zero-Shot Binary Classification

Target dataset ∈ pretraining? →		Maybe			No			Yes		
Dataset ↓	Task ↓	Gemma	Gemma	Gemma	Entity Mean	Griffin	RT (ours)	Rel LLM	Griffin	RT (ours)
	Parameter count →	4B	12B	27B	0	22M	22M	3B	22M	22M
`rel-amazon`	`item-churn`	62.1	55.0	42.1	73.0	69.0	70.2	64.1	71.9	**74.3**
`rel-amazon`	`user-churn`	58.1	54.7	50.5	64.4	62.3	63.9	60.1	64.1	**65.2**
`rel-avito`	`user-clicks`	54.5	59.5	59.8	44.7	45.9	59.5	**62.3**	45.9	60.8
`rel-avito`	`user-visits`	60.1	57.9	**62.7**	60.7	60.7	61.8	56.2	62.2	62.6
`rel-f1`	`driver-dnf`	56.2	54.6	75.8	75.4	57.7	**82.0**	71.8	57.7	**82.0**
`rel-f1`	`driver-top3`	84.6	90.5	**91.4**	85.0	82.5	89.1	70.6	81.8	89.3
`rel-hm`	`user-churn`	59.8	47.1	48.7	**64.4**	60.2	62.8	56.0	60.4	63.1
`rel-stack`	`user-badge`	79.1	79.8	80.0	66.2	73.5	80.0	62.1	82.3	**83.6**
`rel-stack`	`user-engage`	65.9	67.8	78.0	83.5	77.5	77.1	69.5	**89.4**	87.8
`rel-trial`	`study-out`	52.6	57.4	57.2	50.0	51.0	54.5	59.0	57.2	**60.1**
	Mean AUROC →	63.3	62.4	64.6	66.7	64.0	70.1	63.2	67.3	**72.9**

AUROC (%), higher is better.

Compared to Gemma3-27B, RT has
1,000x fewer parameters,
10-100x lower context requirements,
100,000x fewer inference FLOPs,
but **13%** better AUROC!

🤯

Zero-Shot Regression

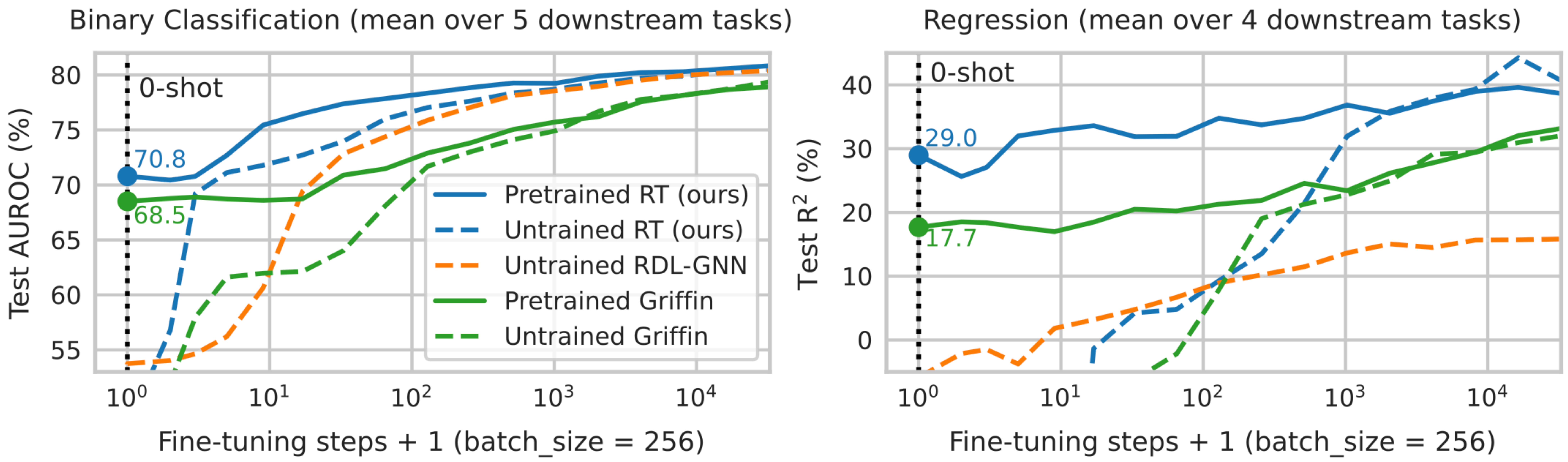
Target dataset ∈ pretraining? →		Maybe			No			Yes		
Dataset ↓	Task ↓	Gemma	Gemma	Gemma	Entity Mean	Griffin	RT (ours)	Rel LLM	Griffin	RT (ours)
	Parameter count →	4B	12B	27B	0	22M	22M	3B	22M	22M
rel-amazon	item-ltv	< -9	< -9	< -9	**54.2**	20.1	33.2	-	20.1	35.4
rel-amazon	user-ltv	< -9	< -9	< -9	19.9	20.6	36.4	-	24.4	**39.7**
rel-avito	ad-ctr	< -9	< -9	-8.2	3.4	2.4	4.5	-	2.4	**7.7**
rel-f1	driver-pos	35.2	43.4	52.4	38.2	-0.7	54.7	-	4.6	**58.4**
rel-hm	item-sales	< -9	< -9	< -9	1.8	2.7	14.0	-	2.5	**30.4**
rel-stack	post-votes	< -9	< -9	< -9	**43.7**	27.4	32.4	-	27.1	32.7
rel-trial	site-succ	< -9	< -9	< -9	-6.4	1.4	**5.2**	-	2.6	3.5
rel-trial	study-adv	< -9	< -9	-7.1	-0.5	-2.5	2.1	-	-2.5	**3.4**
	Mean R^2 →	< -9	< -9	< -9	19.3	8.9	22.8	-	10.1	**26.4**

Regression R^2 (%), higher is better.

LLMs are unable to do regression ‼️
(worse than the GlobalMean baseline)

Supervised Fine-Tuning

Best is **low-**, **medium-** and **high-**resource settings



Relational Feature Extraction

Pretrained **RT Embeddings + MLP Fine-Tuning** gives **Full Fine-Tuning quality** at much lower cost

Dataset ↓	Task ↓	MLP-only FT			Full FT (RT)
	Embedder (frozen) →	GNN	RT(U)	RT(P)	
rel-amz	item-churn	79.6	79.9	81.6	83.4
rel-amz	user-churn	67.6	67.0	68.5	70.8
rel-hm	user-churn	67.5	68.6	69.3	70.5
rel-stk	user-badge	85.5	86.9	87.0	88.7
rel-stk	user-engag	89.1	87.8	89.3	90.2
	Mean AUROC →	77.9	78.0	79.1	80.7
rel-amz	item-ltv	1.6	43.3	56.5	36.8
rel-amz	user-ltv	13.6	31.8	46.8	47.9
rel-hm	item-sales	14.1	17.5	38.8	45.7
rel-stk	post-votes	14.1	37.4	42.6	37.1
	Mean R^2 →	10.9	32.5	46.2	41.9

Table 3: RT as feature extractor. **GNN, RT(U)** are untrained, **RT(P)** is pretrained.

Test-Time Compute Scaling

or, Cost-Quality Trade-Off at Inference

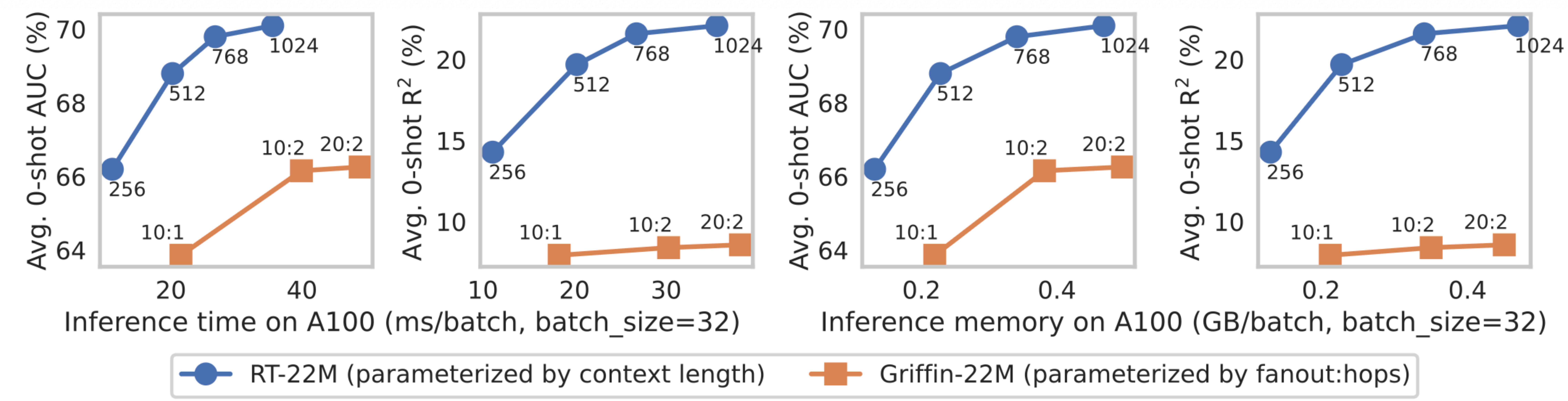


Figure 4: Cost-quality trade-off w.r.t. with varying context sizes at inference. Pretraining was done at the max context size. Average is over 10 classification and 8 regression tasks.

What leads to transfer?
(Ablation Studies)

A1: **In-Context Labels** and **Schema Semantics**

conclusion from context window ablations

Table 3: Mean AUROC (%) and R^2 (%) on ablating context window components for classification (clf) and regression (reg) tasks. Individual numbers are in App. E.

Ablated ↓	Zero-shot		Fine-tuned	
	clf	reg	clf	reg
none	70.1	22.8	77.2	33.2
col names	69.5	20.5	77.5	33.2
self labels	53.8	−5.5	77.1	26.7
other labels	70.6	22.9	77.4	31.0

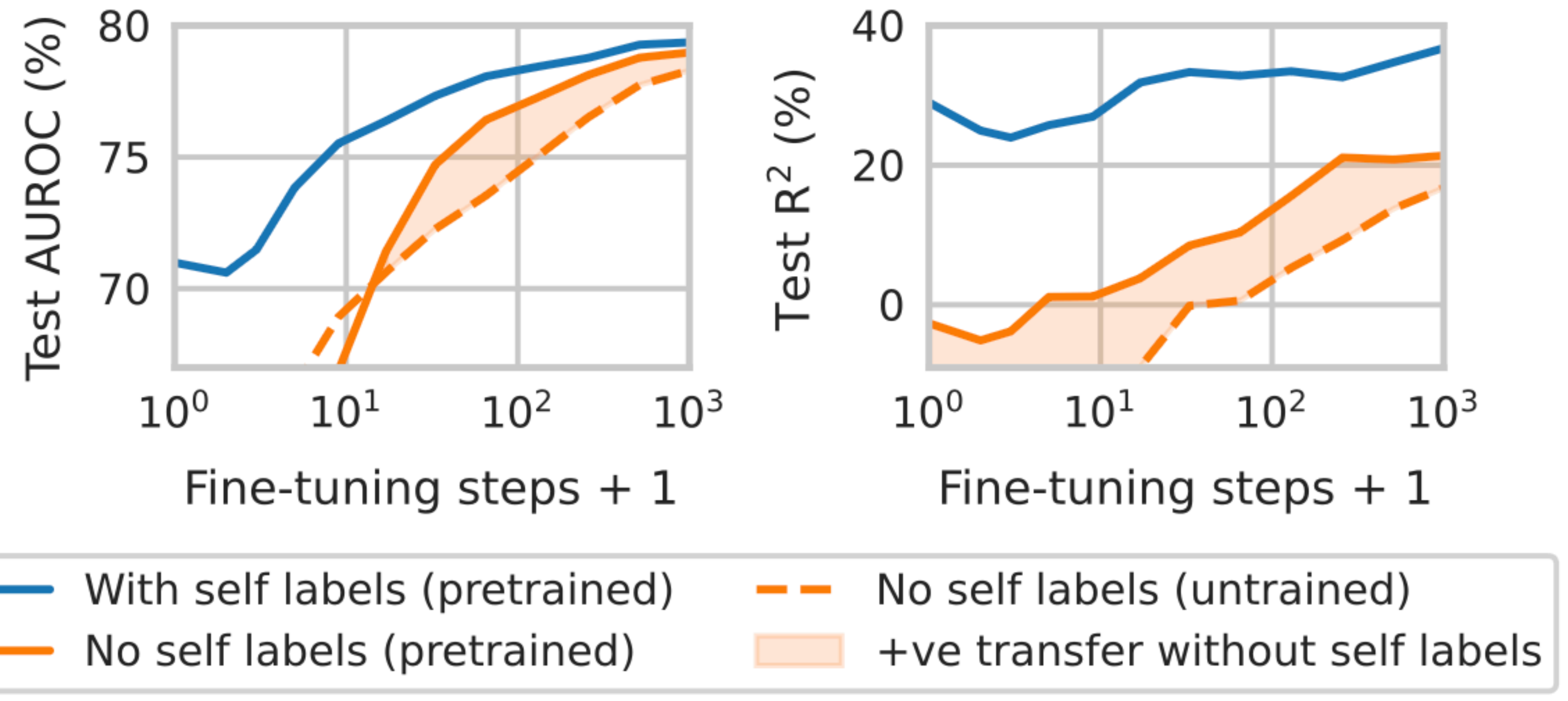


Figure 4: Pretrained RT shows transfer even without self labels. Setup is same as in Fig. 3.

A2: **Column** and **Feature Attention**

conclusion from relational layer ablations

Dataset ↓	Task ↓	Zero-shot					Fine-tuned				
	Ablated attention →	none	col	feat	nbr	full	none	col	feat	nbr	full
`rel-amazon`	`item-ltv`	33.2	5.6	29.8	33.2	44.9	36.8	34.5	34.7	32.9	33.3
`rel-amazon`	`user-ltv`	36.4	6.5	30.4	34.6	35.0	47.4	47.7	45.6	46.3	46.4
`rel-avito`	`ad-ctr`	4.5	-1.4	9.2	8.5	5.3	4.5	-7.1	16.6	2.1	10.6
`rel-f1`	`driver-pos`	54.7	36.9	37.6	49.4	50.1	51.6	51.4	42.0	48.9	39.5
`rel-hm`	`item-sales`	14.0	11.0	4.9	10.5	13.9	39.0	38.1	34.8	37.0	39.6
`rel-stack`	`post-votes`	32.4	28.7	29.2	31.2	30.8	36.5	37.2	35.9	36.9	36.4
`rel-trial`	`site-succ`	5.2	4.7	7.9	6.7	4.9	6.4	4.7	8.8	6.7	7.9
`rel-trial`	`study-adv`	2.1	3.0	1.6	1.8	5.1	43.4	43.0	39.0	40.3	48.4
	Mean R^2 →	22.8	11.9	18.8	22.0	23.7	33.2	31.2	32.2	31.4	32.7

Column attention helps to adapt to **new column distributions**;
Feature attention helps to adapt to **new entity types**.

What now?

Large-Scale Data Curation

https://relbench.stanford.edu

Jure Leskovec
@jure

🚀 Announcing RelBench V2, a major update to our benchmark for foundation models on relational data!

With V2, we are significantly expanding the benchmark's scope to catalyze further research in Relational Deep Learning (RDL) and Relational Foundation Models (RFMs).
Key features:
🍻 4 new databases, spanning domains like e-commerce and beer reviews to scientific research and clinical healthcare.
🧩 40 new predictive tasks, including 28 autocomplete tasks, across new and existing databases.
🔌 External data integrations: 70+ datasets from CTU, 7 datasets from 4DBInfer, and your own data via SQL connector, all in RelBench format.
🛠️ Bug fixes and performance improvements.

🔥 Introducing autocomplete tasks: As opposed to forecasting tasks, autocomplete tasks predict existing columns in the database. We found that models need to deeply understand the relational context to autocomplete database fields, a critical capability that expands the scope of real-world RDL applications.

Learn more:
🌐 Website: relbench.stanford.edu
💻 GitHub: github.com/snap-stanford/...
Huge thanks to @justingu32 @_rishabhranjan_ @jakub_peleska @VHudovernik @CKanatsoulis @fengyuli607, Tang Haiming, Alistiq and everyone else who contributed to our GitHub for making this possible!

RELBENCH
RELATIONAL DEEP LEARNING BENCHMARK

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

TGB is now in RelBench v2!

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 ReDeLex (Peleška & Šír, 2025), 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 make the decision to 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. The key outcome of the conversion is that 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

Synthetic Data Generation

snap-stanford.github.io/plurel

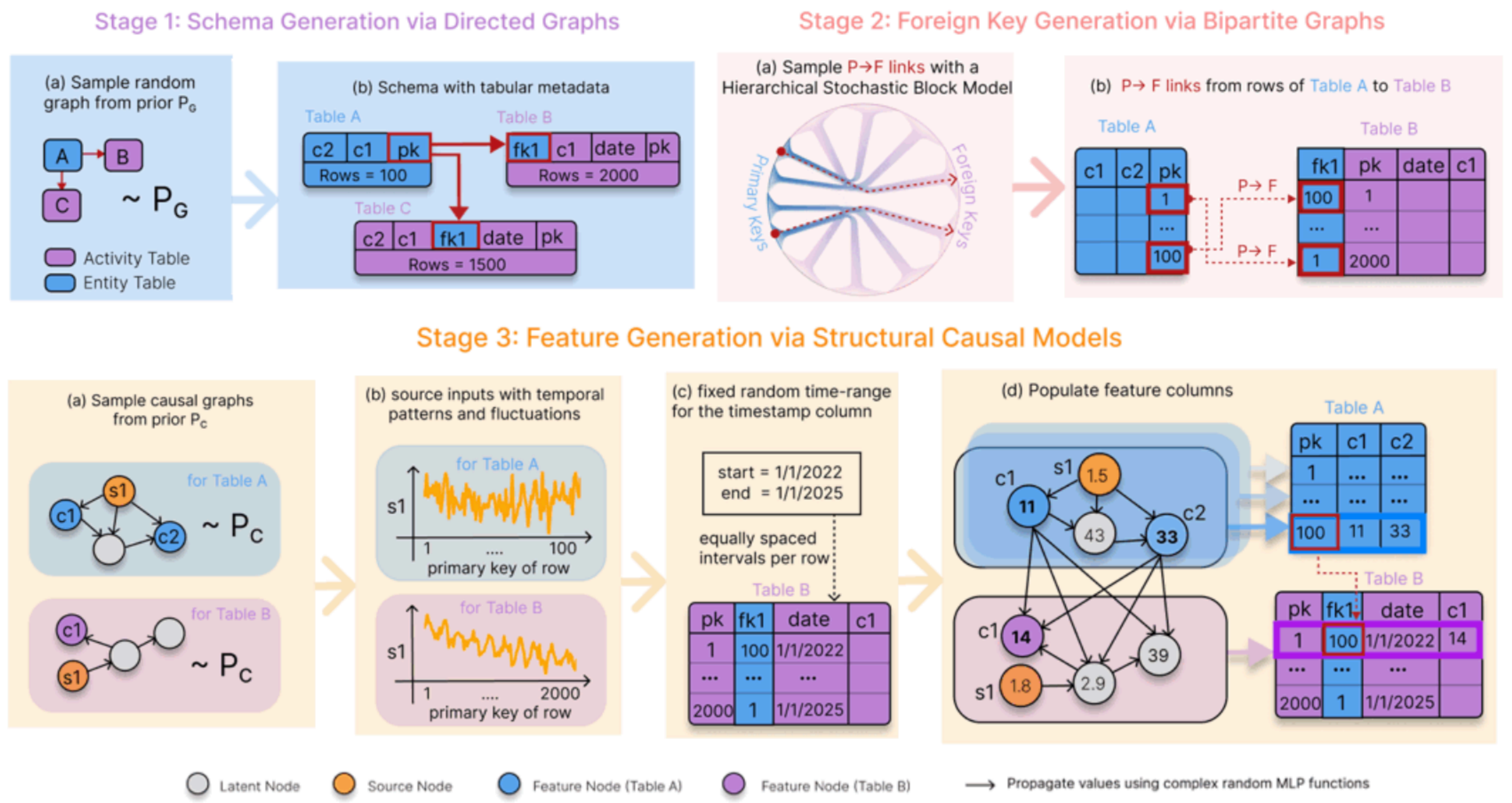
- Vignesh will present at TGL reading group on Feb 26 (next week)!

PluRel: Synthetic Data unlocks Scaling Laws for Relational Foundation Models

Vignesh Kothapalli[1], Rishabh Ranjan[1], Valter Hudovernik[2], Vijay Prakash Dwivedi[1], Johannes Hoffart[3], Carlos Guestrin[1], Jure Leskovec[1]

[1]Stanford University, [2]Kumo AI, [3]SAP

Paper | Code | Data | Models



PluRel synthesizes diverse multi-tabular relational databases using Structural Causal Models, enabling scaling laws for Relational Foundation Models.

Thank You!






Co-authors