



上海交通大学  
SHANGHAI JIAO TONG UNIVERSITY

# rLLM: Relational Data Learning with LLMs

Zheng Wang

Shanghai Jiao Tong University



# Outline

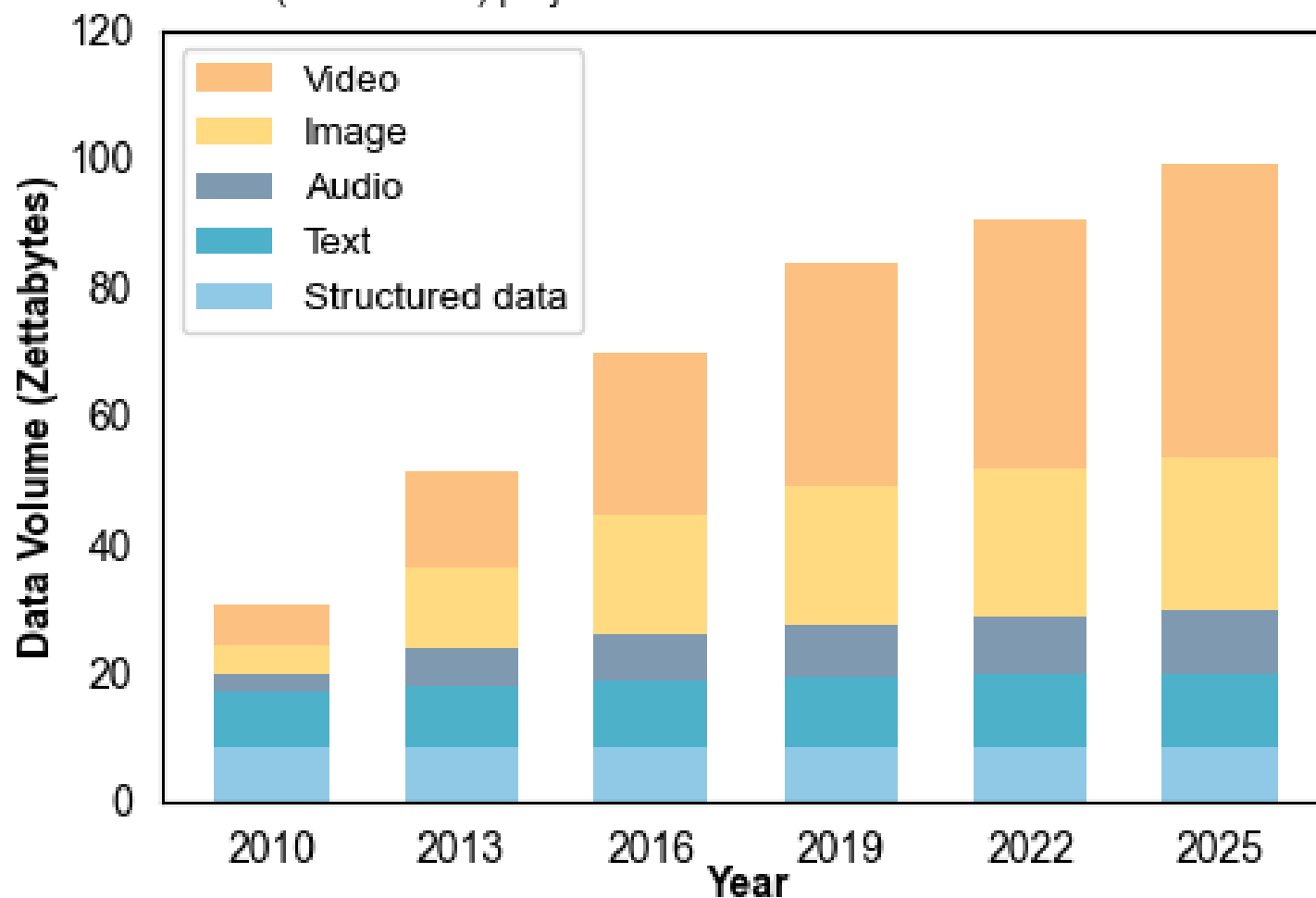
- **Background**
- rLLM (relationLLM)
- Use case

# Background: Big Data Trends



## Trends in Global Data Volume by Data Type (2010-2025)

Source: The rLLM (relationLLM) project.

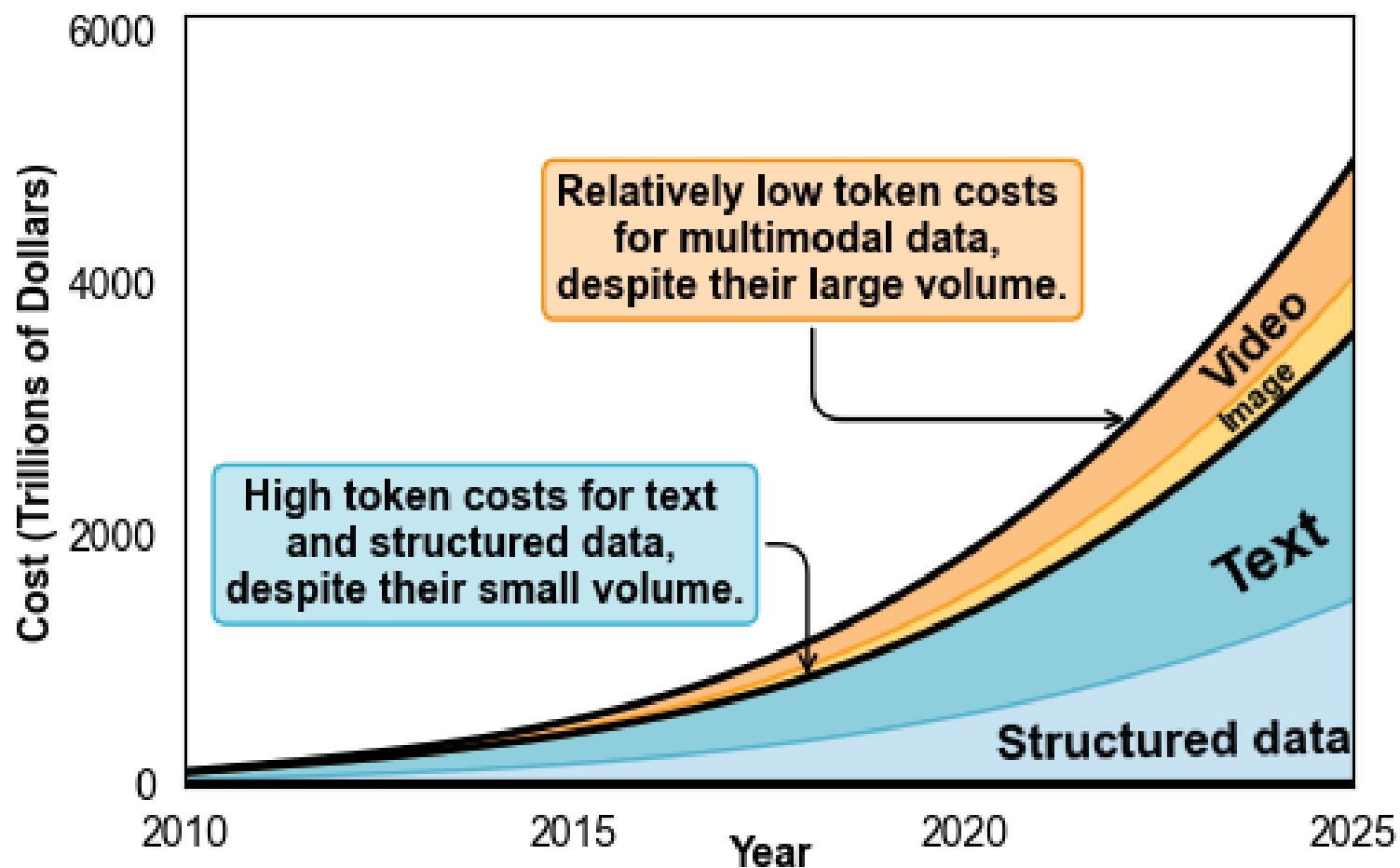


# Background: LLMs are Costly



## Trends in LLM Token Costs by Data Type (2010-2025)

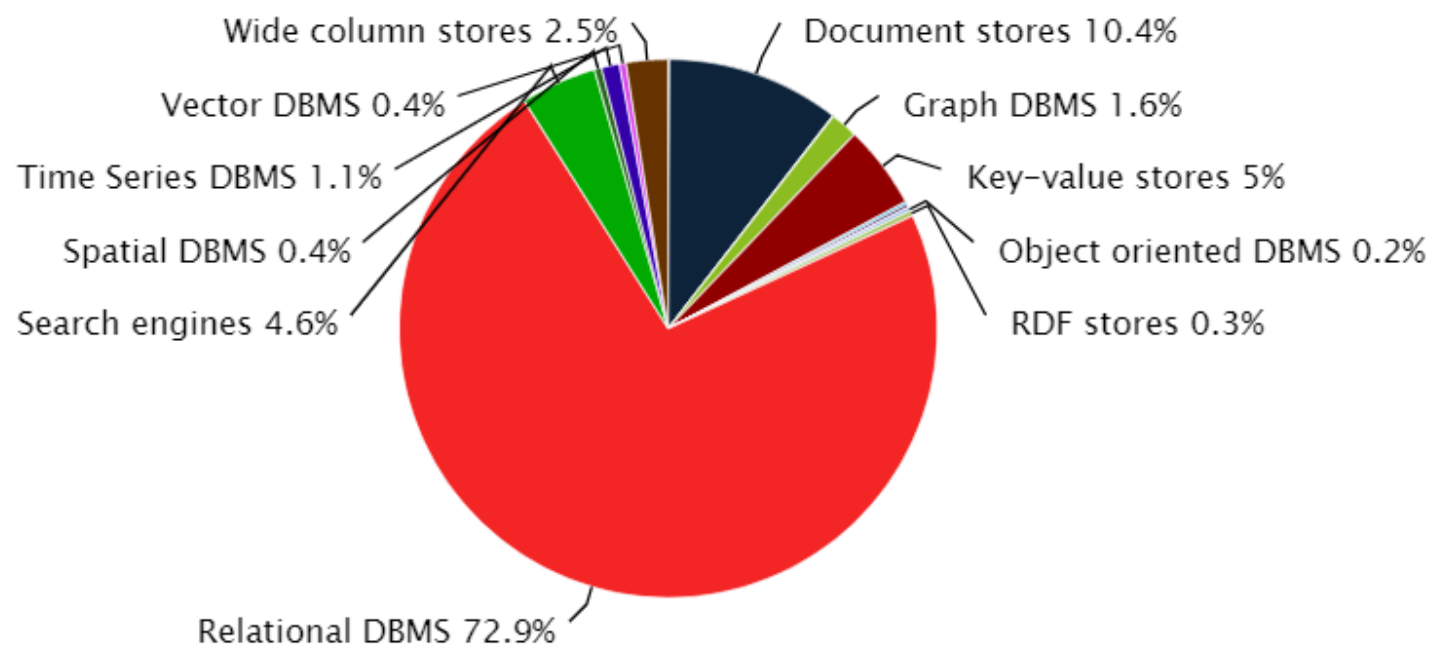
Source: The rLLM (relationLLM) project.



# Relational databases domain the world data



## Ranking scores per category in percent, July 2024



© 2024, DB-Engines.com

In the area of LLM, we need to pay great attention to **Relational Table Learning (RTL)**.



# Outline

- Background
- **rLLM (relationLLM)**
- Use case

# The Key of rLLM (relationLLM)



## rLLM: Relational Table Learning with LLMs

Table Learning  
(for every tables)

Graph Learning  
(for foreigner  
keys)

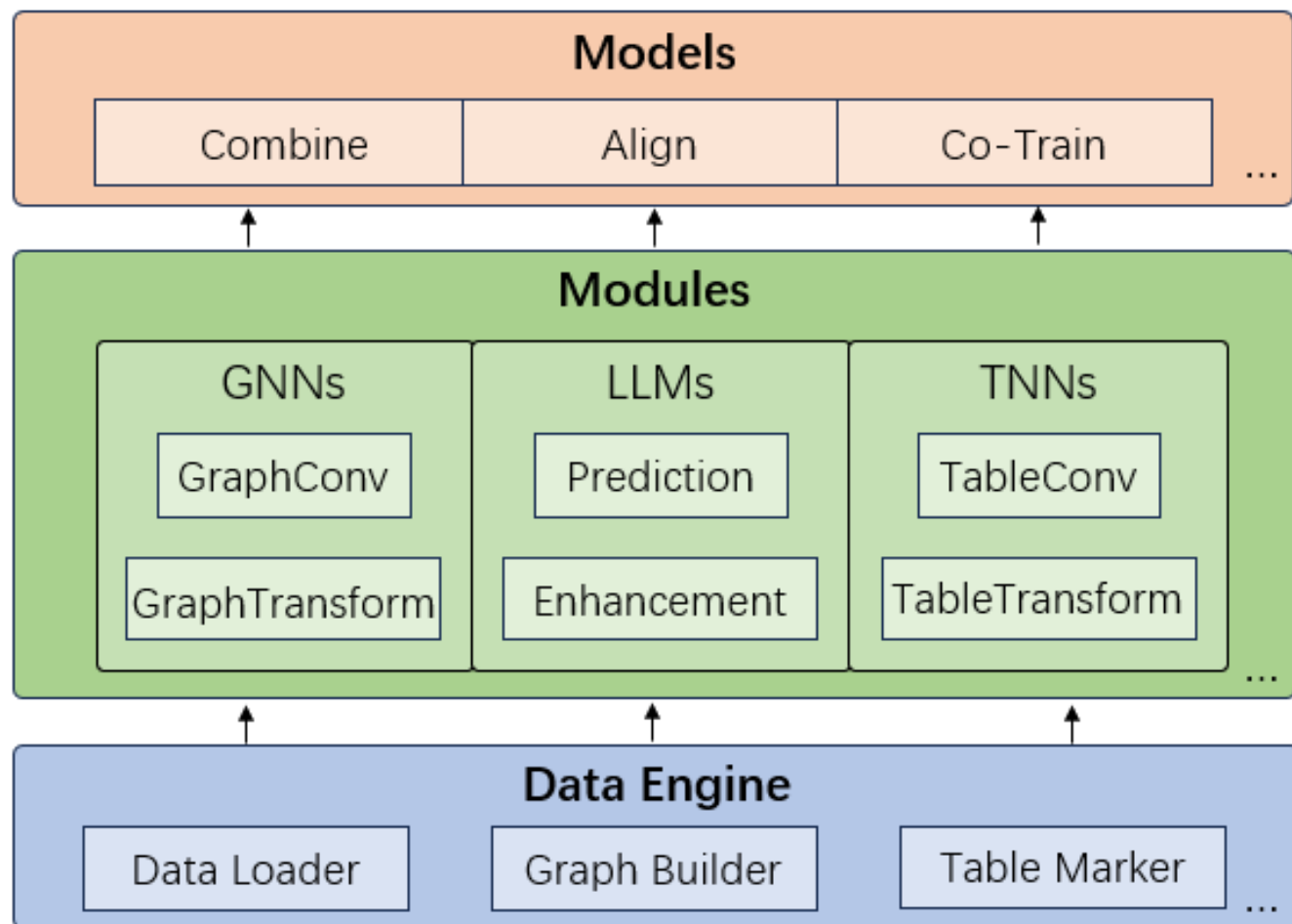
LLMs  
(for LLM-based  
learning)

rLLM (relationLLM) is an easy-to-use Pytorch library for Relational Table Learning with LLMs, by performing two key functions:

- Breaks down state-of-the-art GNNs, LLMs, and TNNs as standardized modules.
- Facilitates novel RTL model building in a "combine, align, and co-train" way using these modules.

Project page: <https://github.com/rllm-project/rllm>

# Overview of rLLM



**The rLLM Overall Architecture**



# Highlight Features:



- LLM-friendly: Modular interface designed for LLM-oriented applications, integrating smoothly with LangChain and Hugging Face transformers.
- One-Fit-All Potential: Processes various graphs (like social/citation/e-commerce graphs) by treating them as multiple tables linked by foreigner keys.
- Novel Datasets: Introduces three new relational table datasets useful for RTL model design. Includes the standard classification task, with examples.
- Community Support: Maintained by students and teachers from Shanghai Jiao Tong University and Tsinghua University. Supports the SJTU undergraduate course "Content Understanding (NIS4301)" and the graduate course "Social Network Analysis (NIS8023)".



# Outline

- Background
- rLLM (relationLLM)
- **Use case**

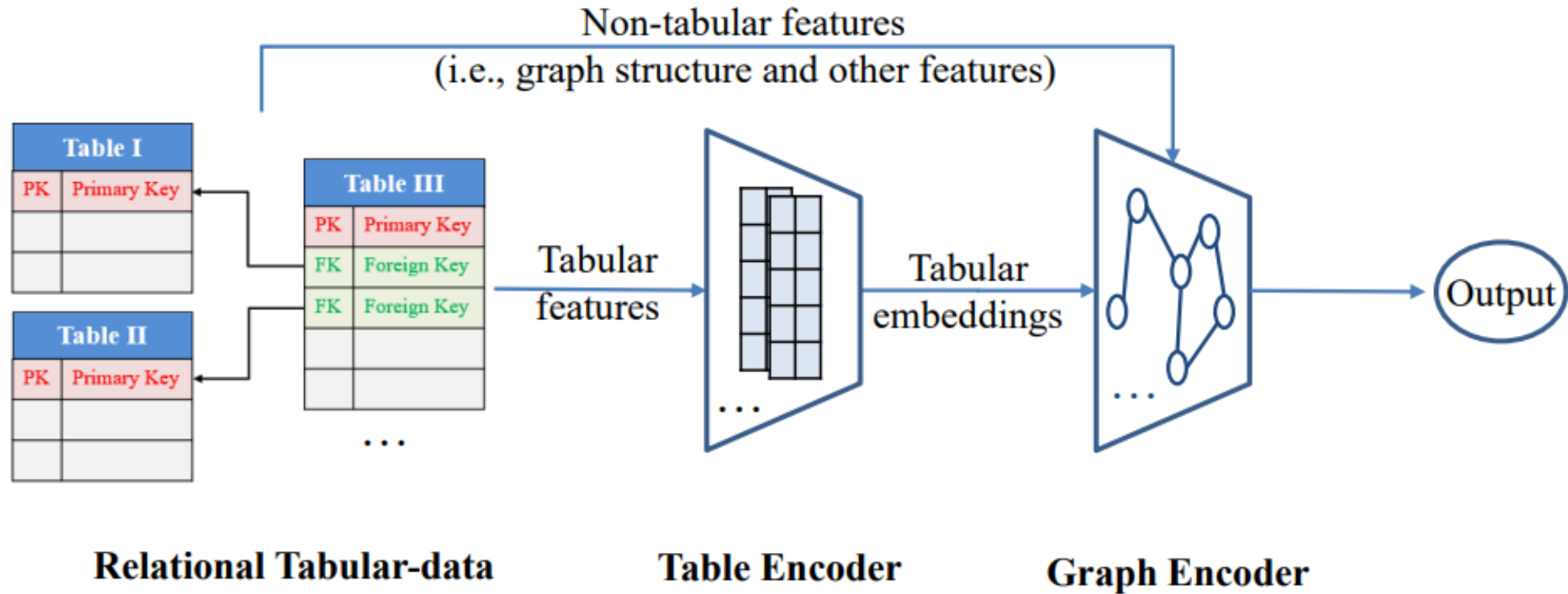


# Three relational table datasets with standard classification tasks

- TML1M is derived from the classical MovieLens 1M dataset.
- TLF2K is derived from the classical LastFM2K dataset.
- TACM12K is derived from the ACM heterogeneous graph dataset.

Dataset	Tables [#row/#col]	Relation Tables	Label	Classes	#Train/#Val/#Test
TML1M	users [6,040/5] movies [3,883/11] ratings [1,000,209/4]	ratings: user-movie	Age range of user	7	[140/500/1000]
TLF2K	artists [9,047/10] user_artists [80,009/3] user_friends [12,717/3]	user_artists: user-artist user_friends: user-user	Genre of artist	11	[220/500/1000]
TACM12K	papers [12,499/5] authors [17,431/3] citations [30,789/2] writings [37,055/2]	citations: paper-paper writings: paper-author	Conference of paper	14	[280/500/1000]

# An illustration RTL method - BRIDGE



**Figure 4: The architecture of BRIDGE**

BRIDGE utilizes TNNs to process table data and leverages the “foreign keys” in relational tables to construct relationships between table samples, which are then analyzed using GNNs.

# Pseudo-code of BRIDGE



```
class Bridge(torch.nn.Module):
    def __init__():
        self.table_encoder = TabTransformer()
        self.graph_encoder = GCN()

    def forward(self, table, x, adj):
        x_table = self.table_encoder(table)
        x = torch.cat([x, x_table])
        embedds = self.graph_encoder(x, adj)
        return embedds
```

# How to try



```
# cd ./examples
```

```
# set parameters if necessary
```

```
python bridge_tacm12k.py
```

```
python bridge_tlf2k.py
```

```
python bridge_tml1m.py
```

---

Thanks for your time.  
QA.