Exploring the Benefits of Graph Transformers in Relational Deep Learning

Student: Rafael Alani
Supervised by: Kubilay Atasu, Çağrı Bilgi
TU Delft

INTRODUCTION

- **Heterogeneous Datasets:** These datasets consist of diverse data types and structures, such as *relational databases*. Their complexity makes them *challenging to analyze* with traditional methods that often expect uniform, tabular data.
- **Intermediary representation:** To overcome limitations of flattening data into single tables, which results in losing rich relational information, **heterogeneous graphs** are used as an intermediary representation. This approach was used in Relbench[1] to help in *preserving complex* and *temporal relationships* within the data.
- **GNN Limitations:** Graph Neural Networks (GNNs) can suffer from "*oversmoothing*" where repeated message passing among nodes leads to indistinguishable node representations, thereby losing specific information.
- **Transformer Expressivity:** Transformers can enhance model expressivity, particularly through their global attention mechanism, allowing the model to weigh the importance of different parts of the input data, *capturing long-range dependencies*. Transformers adapted for graph tasks are called **Graph Transformers.**
- **Cascade architectures**(GNN layers followed by Transformer layers) are more simplistic while still using the *paradigm shift of the Graph Transformers*.
  Interleaved architectures(alternating GNN and Transformer layers) are on of the most promising aarchitectures. [2]

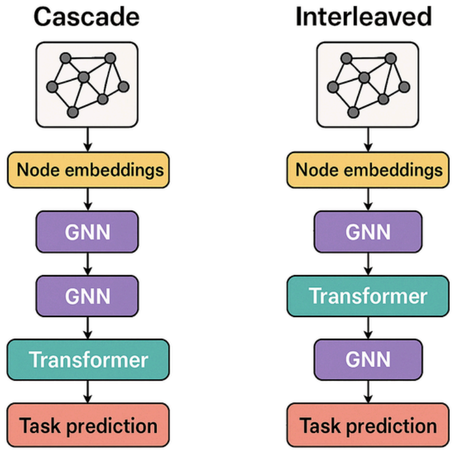

Figure 1: This figure illustrates three different schemes for combining Graph Neural Network (GNN) layers (GConv) and Transformer layers (TLayer) in Graph Transformer (GT) architectures [2]

> What is the most optimal architecture of combining the benefits of Graph Transformers and Relational Deep Learning between Cascade and Interleaved designs?
>
> What are the measurable effects of applying positional encodings in the context of Relational Deep Learning?

METHODOLOGY

Architectural choices and positional encodings are the principal adaptations through which GTs have been able to improve over their GNN counterparts.
Out of all the architecture, we have decided to explore the Cascade and Interleaved architectures. The Cascade model provides a simple implementation, while the Interleaved model has greater expressive power, rivaling SOTA GTs as shown by Yin and Zhong.[2]
The attention mechanism was applied on a node type basis. Two routes showed potential as we advanced: using a monolithic attention mechanism across all node types or using attention on a node-type basis. Applying a monolithic attention mechanism across all node types might dilute type-specific information, as such we opted for the latter.
The Random Walk Positional Encoding and Laplacian Eigenvector PE were selected due to their proven track record.

RESULTS & ANALYSIS

Cascade-GIN model generally achieves results comparable to or slightly varying from the RDL baseline. In the regression tasks a 2-6% improvement is achieved. As seen in both Tables the Interleaved-GIN architecture slightly underperforms the Cascade-GIN one.

Dataset	Task	Split	RDL	Cascade-GIN	Rel. Gain	Interleaved-GIN	Rel. Gain
rel-avito	user-visits	Test	66.20	69.41	4.85 %	66.2	−0.12 %
	user-clicks	Test	65.90	66.47	0.86 %	65.92	0.03 %
rel-f1	driver-dnf	Test	72.62	71.67	−1.31 %	65.16	−10.27 %
	driver-top3	Test	75.54	78.87	2.58 %	76.17	−.094 %
rel-hm	user-churn	Test	69.88	68.69	−1.7 %	67.92	−2.8 %
rel-stack	user-engagement	Test	90.59	90.8	0.23 %	89.63	−1.06 %
	user-badge	Test	88.86	88.15	−0.8 %	82.76	−6.82 %

Table 1: Entity classification results (AUROC, higher is better). Relative gains to RDL. Best values are highlighted.

Dataset	Task	Split	RDL	Cascade-GIN	Rel. Gain	Interleaved-GIN	Rel. Gain
rel-avito	ad-ctr	Test	0.041	0.038	6 %	0.039	3.41 %
rel-hm	item-sales	Test	0.056	0.053	4.17 %	0.052	6.2 %
rel-f1	driver-position	Test	4.022	3.955	1.7 %	4.195	−4.1 %

Table 2: Entity regression results (MAE, lower is better). Relative gains to RDL. Best values are highlighted

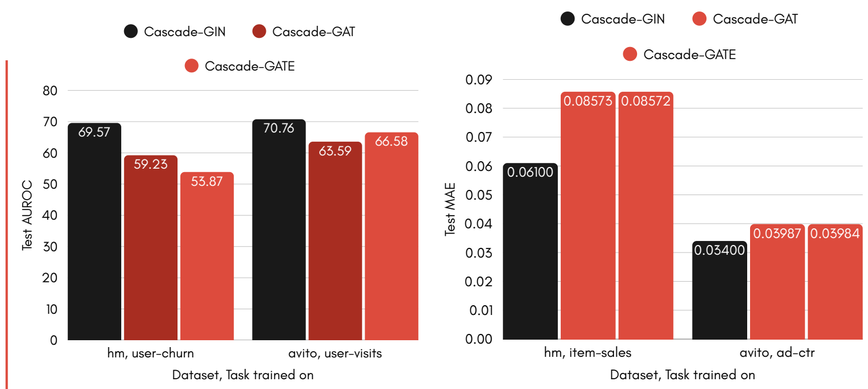

Figure 2: Ablation study of different GNN modules in the Cascade architecture

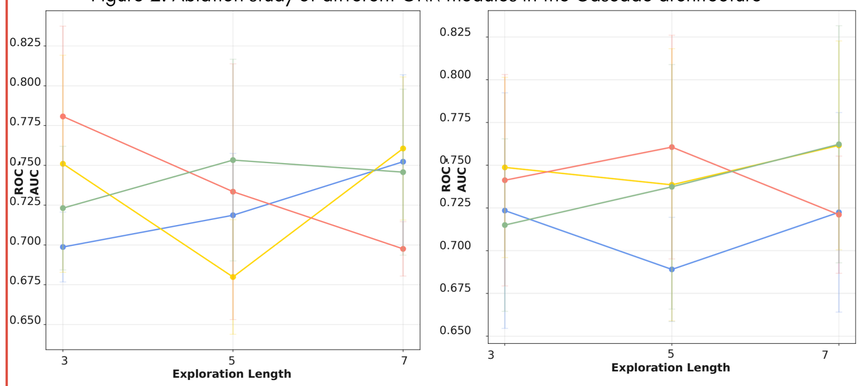

Figure 3: Explored Positional Encodings on driver-dnf, left LE, right RW

CONCLUSION

The choice of architectural design (Cascade vs. Interleaved) impacts predictive performance in Relational Deep Learning. Like the GNN module type, GIN performs significantly better. For shallower models and Relbench-type heterogeneous datasets, the cascade layering technique offers a simple yet effective solution, especially with GIN-based message passing and node-type-specific attention.
Positional encodings (Laplacian Eigenvectors and Random Walks), even with added multi-hop edges, did not consistently provide significant performance gains in the heterogeneous graph setting.

DISCUSSION

There were certain limitation in the way we used the PE, as we first transformed the graph into a homogeneous graph removing the node and edge types. These are likely to contribute to the null performance improvements.

FUTURE WORK

There is a strong need for positional encodings specifically designed for heterogeneous and large-scale graphs, and future research should explore these. As a clear trend was show when using different GNN modules, other more advanced GNN models like PNA should be explored.

REFERENCES

[1] Joshua Robinson et al. RelBench: A Benchmark for Deep Learning on Relational Databases. arXiv:2407.20060 [cs]. July 2024. DOI: 10.48550/arXiv.2407.20060
[2] Shuo Yin and Guoqiang Zhong. "LGI-GT: Graph Trans-formers with Local and Global Operators Interleaving". en. In: Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence. Macau, SAR China: International Joint Conferences on Artificial In- telligence Organization, Aug. 2023, pp. 4504-4512.
ISBN: 978-1-956792-03-4. DOI: 10.24963/ijcai.2023/501

AFFILIATIONS AND THANKS

Many thanks to Cagri Bilgi and Kubilay Atasu for all of the provided support, without them this wouldn't have been possible. Work done as a Bachelor's Thesis for the Computer Science and Engineering Major, TU Delft.