Representation Learning for Structured Data

Marek Dědič[a,b,*]

[a]Czech Technical University in Prague
[b]Cisco Systems, Inc.
ORCID (Marek Dědič): https://orcid.org/0000-0003-1021-8428

Abstract. Graph Neural Networks (GNNs) have become essential for processing graph-structured data, but their application in sensitive domains is often limited by a lack of interpretability and challenges with large-scale graphs. This abstract summarizes research addressing these challenges, focusing on the performance-complexity trade-off in GNNs, and introduces novel approaches for dealing with this trade-off. We also present a study on the effect of graph properties on model performance and outline future research directions in explainable AI and hyperparameter optimization. Our work aims to make GNNs more usable in real-world applications by providing methods to manage their computational cost and increase their transparency.

1 Introduction

Graphs are a powerful and flexible way to model complex systems, from social networks to molecular structures [1, 2, 7]. The rise of Graph Neural Networks (GNNs) [5] has provided a powerful tool for learning from such structured data. GNNs are capable of capturing both the features of individual nodes and the overall topology of the graph, which has led to state-of-the-art performance in a wide range of tasks, including node classification, link prediction, and graph classification. However, the very flexibility that makes GNNs so powerful also contributes to one of their biggest drawbacks: their "black-box" nature. The complexity of GNN models makes it difficult to understand how they arrive at their predictions. This lack of transparency is a major obstacle to their adoption in high-stakes environments, such as medical diagnosis or cybersecurity, where understanding the reasoning behind a decision is critical.

Furthermore, applying GNNs to very large graphs presents a significant computational challenge. As the size of the graph increases, the memory and processing power required to train a GNN can become prohibitive. This has led to an inherent trade-off between the performance of a GNN and its computational complexity. This research explores this trade-off in detail and proposes methods to find an optimal balance, enabling the use of GNNs in large-scale applications where resources are a constraint.

Our contributions to the field of graph machine learning focus on three main areas:

1. Novel methods for balancing the performance-complexity trade-off in large graphs, allowing for scalable GNN deployment.
2. A scalable and efficient algorithm for signal propagation in hypergraphs.
3. A meta-model approach for understanding the impact of graph properties on GNN performance, which can guide hyperparameter tuning.

Our work is motivated by the need for more practically usable and trustworthy GNNs that can be deployed in real-world scenarios. By addressing the challenges of scalability and interpretability, we hope to unlock the full potential of GNNs for a wider range of applications.

* Corresponding Author. Email: marek@dedic.eu.

2 Current Progress of Our Research

Our research has produced several key contributions to the field of graph machine learning, which we detail in the following sections.

2.1 *The Performance-Complexity Trade-off*

We formalize the performance-complexity trade-off problem for GNNs [8]. This framework allows us to systematically evaluate and compare different GNN models on large graph datasets. The core idea is to create a sequence of coarsened graphs, $G_0, G_1, \ldots, G_L$ with the purpose of selecting a graph that meets specific performance or complexity requirements.

2.2 *A HARP-based Method for Performance-Complexity Balancing*

Building on the HARP algorithm [3], which is a popular method for learning node representations, we developed a method that uses graph coarsening to balance performance and complexity [4]. Our approach modifies HARP and extends it with an "adaptive prolongation" algorithm that refines the embeddings from a coarsened graph back to the original graph structure. This allows us to leverage the computational efficiency of working with a smaller graph while still retaining much of the performance of the full model. This method was tested on 10 common graph datasets and showed that at 60% complexity, the models have a 99% probability of being within 10 percentage points of the performance on the full graph. This demonstrates the effectiveness of our approach in achieving a favorable balance between performance and complexity.

2.3 *A Direct Approach to Graph Coarsening*

As an alternative to the HARP-based method, we proposed a more direct approach to graph coarsening using edge contraction [8]. This method is conceptually simpler and offers higher resolution in the

trade-off curve, as each step in the sequence corresponds to a small change in the graph structure. However, our experiments show that its performance is not as consistent as the HARP-based approach.

2.4 *A Scalable Algorithm for Signal Propagation in Hypergraphs*

We developed a simple and scalable algorithm for signal propagation in hypergraphs called CSP [10]. Hypergraphs, where an edge can connect any number of nodes, are becoming increasingly common for modeling complex relationships in domains such as network security and bioinformatics. Our algorithm is parameter-free and easy to implement, making it a competitive baseline against more complex, state-of-the-art algorithms. In our experiments, CSP was shown to be several orders of magnitude faster than existing methods, making it suitable for large-scale hypergraph analysis where computational efficiency is a primary concern.

2.5 *The Effect of Graph Properties on Downstream Task Performance*

To better understand the relationship between graph structure and GNN performance, we conducted a large-scale study on how graph properties affect the performance of GNNs on downstream tasks [9]. We created a meta-dataset of over 15,000 combinations of graph properties, hyper-parameter values, and GNN performance metrics. Using this dataset, we trained a random forest meta-model to predict GNN performance based on graph properties and hyperparameters.

3 Ongoing Research

Building on the work presented in Section 2.5, a study of the interplay between graph properties, model hyperparameters and GNN performance was also conducted. We constructed a meta-dataset of 15 012 different combinations of graph properties, hyper-parameter values and associated GNN performance metrics. This meta-dataset was subsequently used to train a meta-model predicting GNN performance from the graph properties and hyper-parameter values. As the meta-model, we used a random forest regression model – the choice of a simple model was sufficient to achieve good predictive power and at the same time evaluating the meta-model is so computationally cheap when compared to the GNN that it is possible to carry out a total exploration of the hyper-parameter configuration space using the meta-model.

Figure 1 shows an example of the results of this experimental setup. We compared our approach to reference random search strategy for hyperparameter optimization. The proposed meta-model-based approach was used in two different fashions. In the first approach, the meta-model was trained only on GNN runs for the currently evaluated dataset. This approach (red line in the plot) approached the performance optimum faster or roughly as quickly as random search. Secondly, the cross-dataset approach used a meta-model that was pre-trained on GNN runs on other datasets, excluding the currently evaluated dataset. This approach (violet line in the plot) reached near-optimal performance without needing even one run of the GNN on the evaluated dataset.

The results of the cross-dataset hyperparameter optimization strategy are extraordinary in that they point to a possible “zero-shot” optimization strategy from GNNs – i.e. a strategy that produces competitive hyperparameter configurations for a given graph dataset without needing to run the GNN even once. These results, however, need further verification and development. We aim to further develop this method in the following ways:

1. Conduct a full experimental verification – the results from [9] are preliminary in that the selction of graph properties was not conducted thorougly and the random search method in not the state-of-the-art in hyper-parameter optimisation.
2. Pre-train on synthetic graphs – currently, the modified version of the method is pretrained on standard graphs, making it unusable for them. If the method were pretrained on synthetic graphs with similar performance, this would be a marginally stronger result.
3. Apply the method accross different GNN architectures – currently, the method was only tried with GraphSAGE.
4. Use the meta-model as a surrogate model for Bayesian hyper-parameter optimisation methods.
5. Invert the meta-model – The current design of the meta-model is a function predicting model performance from graph properties and hyper-parameters. However, the true aim is to predict the hyper-parameters from the graph properties. Choosing a different meta-model architecture could allow for such a model

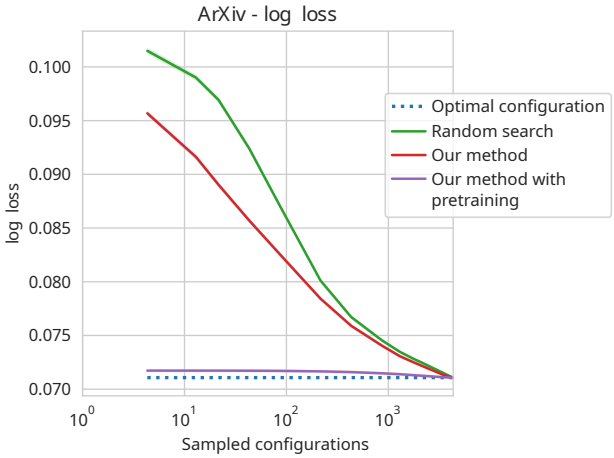

Figure 1. Comparison of reference random hyper-parameter search with our proposed solutions on the ArXiv dataset [6]. The plot shows the performance of each hyper-parameter optimisation method (measured by the log loss of the model) as a function of the number of hyper-parameter configurations sampled by the method.

4 Conclusion

This research addresses several challenges in the application of Graph Neural Networks, focusing on their scalability and usability. We have introduced effective methods for managing the performance-complexity trade-off in large graphs, developed a highly scalable algorithm for signal propagation in hypergraphs, and created a meta-model to predict GNN performance based on graph properties. Our ongoing work aims to develop explainable GNNs and advance a "zero-shot" hyperparameter optimization strategy. Collectively, these contributions work towards making GNNs more transparent, efficient, and reliable for deployment in sensitive, real-world applications.

References

[1] L. Akoglu, H. Tong, and D. Koutra. Graph based anomaly detection and description: a survey. *Data Mining and Knowledge Discovery*, 29(3):626–688, May 2015. ISSN 1573-756X.

doi: 10.1007/s10618-014-0365-y. URL https://doi.org/10.1007/s10618-014-0365-y.

[2] R. v. d. Berg, T. N. Kipf, and M. Welling. Graph Convolutional Matrix Completion, Oct. 2017. URL http://arxiv.org/abs/1706.02263. arXiv:1706.02263 [cs, stat].

[3] H. Chen, B. Perozzi, Y. Hu, and S. Skiena. HARP: Hierarchical Representation Learning for Networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018. ISSN 2374-3468. URL https://ojs.aaai.org/index.php/AAAI/article/view/11849. Number: 1.

[4] M. Dědič, L. Bajer, P. Prochazka, and M. Holena. Balancing performance and complexity with adaptive graph coarsening. In *The Second Tiny Papers Track at ICLR 2024*, Vienna, Austria, May 2024. URL https://openreview.net/forum?id=DrHwIzz93C.

[5] M. Gori, G. Monfardini, and F. Scarselli. A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pages 729–734 vol. 2, July 2005. doi: 10.1109/IJCNN.2005.1555942. ISSN: 2161-4407.

[6] W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren, B. Liu, M. Catasta, and J. Leskovec. Open Graph Benchmark: Datasets for Machine Learning on Graphs, Feb. 2021. URL http://arxiv.org/abs/2005.00687. arXiv:2005.00687 [cs, stat].

[7] B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.

[8] P. Procházka, M. Mareš, and M. Dědič. Scalable Graph Size Reduction for Efficient GNN Application. In *Proceedings of the 22nd Conference Information Technologies – Applications and Theory (ITAT 2022)*, volume 3226 of *CEUR Workshop Proceedings*, pages 75–84, Zuberec, Slovakia, Sept. 2022. CEUR-WS.org. URL http://ceur-ws.org/Vol-3226/#paper9.

[9] P. Procházka, M. Mareš, and M. Dědič. Which Graph Properties Affect GNN Performance for a Given Downstream Task? In *Proceedings of the 23nd Conference Information Technologies – Applications and Theory (ITAT 2023)*, volume 3498 of *CEUR Workshop Proceedings*, pages 58–66, Tatranské Matliare, Slovakia, Oct. 2023. CEUR-WS.org. URL https://ceur-ws.org/Vol-3498/#paper7.

[10] P. Procházka, M. Dědič, and L. Bajer. Convolutional Signal Propagation: A Simple Scalable Algorithm for Hypergraphs. In *21st International Workshop on Mining and Learning with Graphs @ECMLPKDD 2024*, Vilnius, Lithuania, Sept. 2024. URL https://mlg-europe.github.io/2024/papers/131/CameraReady/MLG-ECML-2024-paper.pdf.