MSc in Advanced Computer Science
FHS Computer Science; Mathematics and Computer Science;
Computer Science and Philosophy.

GEOMETRIC DEEP LEARNING

Hilary Term 2024

Submission deadline 12 noon, Wednesday 10th April 2024, via Inspera.

There is a total of 100 marks available for this paper, you should attempt all parts of the paper.

NB: You must not discuss this examination paper with anyone.

Geometric Deep Learning

Question 1

In this question, you are expected to investigate a research question (from the provided *list*) related to geometric deep learning models studied in the course (deepsets, graph neural networks, transformers, CNNs, group-equivariant CNNs, geometric GNNs, neural differential equations, intrinsic/mesh CNNs). You can choose one of the following:

- Implementation and detailed experimental study of an existing paper.
- Theoretical analysis on an existing model, e.g. focusing on some phenomenon such as over-smoothing or over-squashing.
- Propose an extension or improvement of an existing paper.

The report should not exceed *six* pages. If the study involves an implementation, please provide your code with the submission as a zip file. The study will be assessed based on the following criteria: (i) motivation, clarity, and presentation (20 marks), (ii) originality and novelty (20 marks), (iii) coherence and depth of the study (20 marks), (iv) scholarship, i.e., framing in the existing literature (15 marks) (v) relations to the concepts discussed throughout the course (10 marks), (vi) balanced critical self-evaluation (15 marks). (*The study will not be evaluated according to the number of experiments or datasets, but according to its merits in accordance to the outlined criteria. Please follow the notation from the lecture slides whenever applicable. As guidance, top marks will only be given for exceptionally written papers worth publishing that show novel theoretical results, propose a novel method or a significant improvement of an existing one, or report state-of-the-art experimental results.*)

(a) State a research question related to geometric deep learning and explain your motivation for the proposed study. *(Example: How can the use of graph direction help message passing?*

(b) Propose means to study the research question and give an outline of the overall approach, clearly identifying the goals of the chosen study. *(Example: An experimental setup which aims to test different ways of directed message passing on graphs, using a synthetic dataset or using existing datasets.)*

(c) Clearly state your methodology: theoretical framework and empirical setup, hyperparameters, and the assumptions underpinning the study.

(d) Report your empirical/theoretical/conceptual findings, identifying whether or not the results support the initial hypothesis. You can use visuals, figures, and tables to present your findings in a structured manner. You are expected to relate and compare your results to the relevant literature and to the concepts from the course.

(e) Provide a detailed discussion relating the results to the original motivation, as well as a critical perspective regarding the study.

(f) The report should conclude with an outlook stating any additional studies that need to be conducted to reach to more conclusive statements.

(100 marks)

List of Projects

This is a list of projects that can be selected by students for the examination in the Geometric Deep Learning course. Keep in mind when choosing that the projects span a range of difficulty, open-endedness, and may require different theoretical or coding focus. As in any research question, it is normal for the scope of the projects to be somewhat open-ended; part of the task will be to clarify the goals and scope of the project.

1.

Title: Evaluating the Impact of Linearized Attention vs Full-Attention in Graph Transformers

Description: Standard scaled-dot product attention computes attention coefficients using an all-to-all pairs key-query mechanism. This mechanism has been empirically proven to be highly effective and has yielded exceptional results across various domains. Scaled-dot product attention is indeed integral to the transformer architecture. However, its main drawback lies in its squared complexity in the number of nodes, making it impractical for application in graph transformers dealing with large graphs, where the number of nodes can be significantly high. Given this limitation, recent literature has proposed alternatives such as linearized attention or other approximations, which are computationally less extensive. Although this concept is not novel and has been extensively explored in the context of NLP, it has not gained widespread acceptance due to its performance degradation, leading to the exploration of alternatives like FlashAttention. Nonetheless, in the realm of large graphs, linearized attention and similar approximations remain pertinent. In this study, we will build upon a standard graph transformer architecture and investigate the impact of applying various approximations to the attention layer, drawing from the extensive literature in NLP. We invite both empirical and theoretical contributions to further explore this subject.

References: `https://www.researchgate.net/publication/378394991_Elucidating_Graph_Neural_Networks_Transformers_and_Graph_Transformers/stats` `https://arxiv.org/pdf/2205.12454.pdf` `https://arxiv.org/pdf/2006.04768.pdf` `https://arxiv.org/pdf/2009.14794.pdf` `https://arxiv.org/pdf/2007.14062.pdf` `https://github.com/rampasek/GraphGPS`

2.

Title: Improving Temporal Graph Networks for Dynamic Link Prediction

Description: Node-based models for temporal graphs typically begin by utilizing node information, such as temporal neighborhood or previous node history, to create node embeddings. These embeddings are then aggregated from both the source and destination nodes of an edge to predict its existence. Notable examples of such methods include TGN, DyRep, and TCL. According to the recently released Temporal Graph Benchmark, the TGN model achieves the best results for medium and large temporal graphs. However, its temporal message aggregator relies on simple approaches like the most recent message and mean message aggregation. In this study, we aim to explore more expressive methods for temporal message aggregation, such as those based on RNNs and attention mechanisms.

References: `https://arxiv.org/pdf/2006.10637.pdf` `https://github.com/twitter-research/tgn` `https://arxiv.org/pdf/2307.01026.pdf`

3.

Title: Temporal Graph Networks with Persistent Forecasting Bias for Dynamic Node Property Prediction

Description: In the realm of dynamic node property prediction, heuristic methods like persistence forecast and moving averages pose significant competition to temporal graph neural network approaches such as DyRep and TGN. Notably, persistence forecasting achieves state-of-the-art results on recently released temporal graph benchmarks. Unlike link prediction, where the existence of a link is treated as binary classification, the node affinity prediction task assesses the likelihood or weight that the model assigns to various target nodes, typically positive links. In this project, we aim to tailor conventional Temporal Graph Networks designed for link prediction to suit dynamic node property prediction, potentially integrating a hybrid architecture that incorporates elements from persistent forecasting as an inductive bias.

References: `https://arxiv.org/pdf/2006.10637.pdf` `https://github.com/twitter-research/tgn` `https://arxiv.org/pdf/2307.01026.pdf` `https://arxiv.org/pdf/2205.02082.pdf`

4.

Title: Latent Graph Inference for NeuroImaging

Description: Machine learning provides a powerful tool for analyzing intricate functional neuroimaging data, showing promise in predicting various neurological conditions, psychiatric disorders, and cognitive patterns. However, real-world clinical data can often be noisy or contain incomplete graphs. In this study, we aim to harness techniques in latent graph inference and graph structure learning to construct robust Graph Neural Networks (GNNs) for neuroimaging analysis. Specifically, we will utilize the NeuroGraph benchmark to assess the resilience of our methods against varying levels of neuroimaging corruption and missing links commonly encountered in clinical scenarios.

References: `https://arxiv.org/pdf/2306.06202.pdf` `https://arxiv.org/pdf/2002.04999.pdf` `https://openreview.net/pdf?id=JLR_B7n_Wqr` `https://github.com/lcosmo/DGM_pytorch`

5.

Title: Graph Transformer Surgical Fine-tuning

Description: A prevalent strategy for transfer learning amidst distribution shifts involves fine-tuning the final layers of a pre-trained model while retaining acquired features and adjusting to the new task. Research indicates that, particularly in transformers for NLP, selectively fine-tuning a subset of layers—referred to as surgical fine-tuning—either matches or surpasses conventional fine-tuning methods. Furthermore, the effectiveness of tuning specific subsets depends on the nature of the distribution shift. For instance, in scenarios involving image corruptions, fine-tuning only the initial layers yields optimal results for CNNs and ViTs. Meanwhile, Graph Transformers represent an extension of transformers to graphs, typically integrating global attention with a local message-passing neural network. This project aims to investigate whether such fine-tuning behaviors are also evident for graph transformers across various graph distribution shifts, such as node-level feature corruption, missing links, or graph mislabeling.

References: https://openreview.net/pdf?id=APuPRxjHvZhttps://www.researchgate.net/publication/378394991_Elucidating_Graph_Neural_Networks_Transformers_and_Graph_Transformers/stats https://arxiv.org/pdf/2205.12454.pdf https://arxiv.org/pdf/2310.06417.pdf https://github.com/rampasek/GraphGPS

6.

Title: Investigating Discrepancies between ChebNet and GCN

Description: The release of ChebNet and GCN in 2016 marked significant advancements in graph neural networks. ChebNet employs Chebyshev polynomials to construct an orthogonal basis, effectively reducing computational complexity when computing powers of the adjacency matrix. In contrast, GCN utilizes a message-passing mechanism to compute these powers iteratively, updating node features at each step. Surprisingly, in Practical 2, we found that a network employing ChebConv with a filter size of 1, thus learning only the first power of the adjacency, outperformed GCN on the MNIST-pixels dataset. Despite mathematical equivalence between a ChebNet with filter size 1 and a GCN lacking activation functions and skip connections, our experiments revealed a performance gap. This project aims to delve deeper into this discrepancy, and explain it.

References: https://arxiv.org/abs/1606.09375 https://arxiv.org/abs/1609.02907

7.

Title: What types of homomorphism counts can we view through spectral lenses?

Description: The Weisfeiler-Lehman (WL) coloring algorithm is a well-established method in graph theory, known for its equivalence to counting homomorphisms of tree-like graphs into the two graphs being compared. When the homomorphism counts are equivalent, the algorithm will produce the same coloring for both graphs, indicating their structural similarity. However, one limitation of the WL algorithm is its inability to count cycles and determine cycle sizes, which can be crucial for capturing certain graph properties. Recent advancements in graph representation learning have introduced positional encoding techniques that leverage spectral decomposition as a form of feature augmentation. These techniques have shown promising results in improving performance in various graph-related tasks. In this project, we aim to explore the potential of using spectral decomposition to learn homomorphism counts beyond what is traditionally captured by the WL algorithm. Specifically, we are interested in investigating whether spectral decomposition can enable us to effectively count cycles and determine their sizes. Another direction for exploration, which is somewhat related to the aforementioned problem, is the connection between the WL algorithm and the spectrum of a graph. Previous research, such as [1], has provided insights into this connection. For example, Corollary 4.4 and Theorem 4.5 in [1] suggest that graphs with a small number of automorphisms (e.g., maximum two nodes with the same WL coloring) exhibit a graph Laplacian with no repeating eigenvalues. While this result may appear limited, it opens up avenues for further investigation into the relationship between WL and graph spectra.

References: [1] https://arxiv.org/abs/2205.11172

8.

Title: Graph Neural Networks through fixed diffusion operators

Description: Nowadays, most Graph Neural Networks (GNNs) rely on the encode-propagate-decode paradigm: input features and topology are first processed without exchanging information between different nodes to build input latent features (encode phase); next, latent features are updated through a message-passing scheme (e.g. for each layer features receive information from their neighbours), often resulting in a diffusion process over the graph according to a parametric operator (propagate); finally, a readout operation maps the ultimate-layer features to the label space. Despite the success of such a pipeline, there are theoretical and practical reasons to believe that the propagate step above could actually be replaced by a family of fixed (i.e. non parametric) diffusion processes that can represent the graph through different spatial (or spectral) resolutions. In fact, the known results limiting the expressive power of message-passing in terms of 1-WL test, only require injective maps acting on the 1-hop aggregation [1]; on a practical side, [2] showed that a simple GNN with fixed aggregation over multiple hops can lead to promising results (particularly on large graphs) [2]. Besides, another empirical work [4] showed that with sufficiently powerful structural encoding, even without message-passing layers, GNNs can often be competitive using only their encode and decode steps. More in general, many common message-passing neural networks are known to be discretization of dynamical systems on graphs [3], and it is not clear whether one can choose family of fixed dynamical systems to obtain an exhaustive representation of attributed graphs to be used for the downstream task. In this project we will investigate how much expressive power and performance can be retained from several GNN families—from simple 1-hop message-passing models, to more sophisticated ones involving rewiring and/or subgraphs—when we replace their “propagate” step with analogous families of fixed aggregation (diffusion) operators. Using fixed operators could lead to more transparent architectures, where the “propagation” step is more explainable, and, importantly, to architectures that can be quite more efficient during training.

References: [1] https://arxiv.org/abs/1810.00826 [2] https://arxiv.org/abs/2004.11198 [3] https://arxiv.org/abs/2206.10991 [4] https://openreview.net/pdf?id=mbgod4sDia

9.

Title: Graph-aware data pruning and data augmentation to understand generalization

Description: Graph Neural Networks (GNNs) are, currently, state-of-the-art on many common graph-benchmarks. However, in many such cases, we often have a large number of labels in the training set, which is often unrealistic, particularly when compared to scientific (quantum) settings, where labels are hard and expensive to acquire [1]. Motivated by that, in this project we will explore how pruning a training set made of attributed graphs affects the performance (and generalization) of GNNs. We will investigate whether there exist “harmful” prunings, e.g. if we remove all labelled graphs such that a certain quantity M is larger (smaller) than ϵ, then the performance deteriorates faster than random pruning—confirming generalization failures of GNNs associated with the measure M. Conversely, we will also study whether there exists "smart" pruning, meaning that if the training set is "diverse" enough according to some metric/kernel K, then the performance of GNNs remains relatively stable even after (significantly) reducing the training set. The same considerations can be made about Graph-Transformers (GTs), and it would be interesting to explore whether GNNs are more stable than GTs in low-label regimes. If time allows, we will also explore the opposite directions of finding new ways to do data augmentation for inductive tasks on graphs.

References: [1] `https://arxiv.org/pdf/2306.10066.pdf` [2] Experiments in Figure 4 of `https://arxiv.org/pdf/2206.11140.pdf` [3] To get a picture of known theoretical results of generalization of GNNs on large (random) graphs, even with small training dataset (see also similar references from Ron Levie and Gitta Kutyniok) [4] `https://proceedings.neurips.cc/paper_files/paper/2022/file/1eeaae7c89d9484926db6974b6ece564-Paper-Conference.pdf`

10.

Title: Graph Neural Networks through multi-body potentials

Description: Despite their empirical successes, Message Passing Neural Networks (MPNNs) are known to have limited expressive power. One way to see this, is that MPNNs may struggle to capture higher-order interactions; in fact even in the case of 2-body (i.e. pairwise) interactions, these are limited by the commute time between the nodes [1]. This motivates the question: can we find more explicit ways to learn multi-body interactions? Moreover, are there bases of functions on graphs we can leverage to solve a downstream task? In this project we adopt an approach similar to MACE [2], and investigate how we can avoid the message-passing paradigm by parameterising functions on graphs directly. We do so by considering a hierarchical class of functions (potentials) of 1,2,.., K body terms. We will construct such potentials by relying on permutation-invariant architectures and separating the contribution of features and that of topology (in the form of distances, commute time, and/or positional encoding). Such an approach could result in a more transparent framework, where different k-body contributions are explicitly parameterised rather than implicitly (and often only partially) achieved after m message-passing layers.

References:

[1] `https://arxiv.org/abs/2306.03589` [2] `https://arxiv.org/pdf/2206.07697.pdf`

11.

Title: Local Reference Frames on Geometric Graphs

Description: Local Reference Frame (LRF) defined a local orthogonal basis for a set of point clouds that could be used to build invariant features with respect to 3D translations and rotations of shape.

Definition 1: For a point cloud P, the LRF at point $p \in P$ is defined as $L(p) = \{x(p), y(p), z(p)\}$ where $x(p), y(p)$, and $z(p)$ are the orthogonal axes of the coordinate system that follow the right-hand rule. The project aims to establish a Local Reference Frame (LRF) on geometric graphs. By projecting node coordinates into an orthogonal basis, we obtain new features that are invariant to graph rotations and translations. Then we can apply message-passing mechanisms using these transformed features to enhance geometric graphs related tasks (for example, molecules property prediction). There are several methods to build LRF on point clouds which could be classified into two main classes: covariance analysis and point distributions. Covariance methods attempt to identify a covariance matrix using all the point coordinates of a point cloud P. For example, let $p \in P$, [3] defined a [3 x 3] covariance matrix C, whose eigenvectors provide an orthogonal basis, which serves as our Local Reference Frame (LRF). Projecting the original coordinates into this eigenvector-based frame yields new features that remain invariant under 3D translations and rotations. However, the approach is not without its limitations, notably the sign ambiguity inherent in the eigenvector determination. You will investigate the advantages and disadvantages of this approach and implement different methods for calculating the covariance matrix on geometric graphs (see [4] and [1]).

References: [1] Yulan Guo, Ferdous Sohel, Mohammed Bennamoun, Min Lu, and Jianwei Wan. Rotational projection statistics for 3d local surface description and object recognition. International Journal of Computer Vision, 105(1), 2013.

[2] Simone Melzi, Riccardo Spezialetti, Federico Tombari, Michael M. Bronstein, Luigi Di Stefano, and Emanuele Rodola. Gframes: Gradient-based local reference frame for 3d shape matching. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 4624–4633, 2019.

[3] A. Mian, M. Bennamoun, and R. Owens. On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes. International Journal of Computer Vision, 89:348–361, 2010.

[4] F. Tombari, S. Salti, and L. Di Stefano. Unique signatures of histograms for local surface description. In European Conference on Computer Vision, pages 356–369. Springer, 2010.

12.

Title: Expressive Power of Graph Neural Network with Pair-wise Encoding for Link Prediction

Description: Graph Neural Networks are prominent models for tasks like node classification and graph classification. However, since they only encode unary invariants (node invariants), GNNs face notable limitations in performing the task of link predictions that encode binary invariants as they solely rely on a binary decoder to obtain the probability of the link. There has been a series of work focusing on obtaining pair-wise encoding directly with GNN for link prediction. For instance, Line graph neural networks [1] transformed the original graph into a line graph L(G), which represents each edge in the original graph as a node in the transformed line graph, and two nodes are connected in L(G) if the corresponding edges shared the same node in G. Message passing on L(G) will then directly output the pair-wise encoding, which can be passed through a unary decoder to obtain edge probability. INDIGO [2] considered a similar idea, but it adopted a slightly different approach to suit the nature of knowledge graphs, which additionally contain edge types. However, there has been limited study on the expressive power of these methods. How will these methods compare with k-GNN [3] when k = 2? How will the Whitney graph isomorphism theorem affect the expressivity of Line graph neural networks? Viewing as a binary classifier, what fragment of first-order logic can these models represent [4]? We aim to pick one or more of these models that compute pair-wise encoding and study their expressive power through the lens of the Weisfeiler-Leman test and how they are located in the current expressiveness hierarchy. We then empirically verify our findings of the expressive power by designing suitable experiments.

References: [1] https://arxiv.org/pdf/2010.10046.pdf [2] https://proceedings.neurips.cc/paper/2021/hash/0fd600c953cde8121262e322ef09f70e-Abstract.html [3] https://arxiv.org/abs/1810.02244 [4] https://arxiv.org/abs/2302.02209

13.

Title: The role of nonlinear encoder in Gradient Flow Framework (GRAFF)

Description: Di Giovanni et al. [1] showed that convolutional-type GNNs could be derived as gradient flows of parametric Dirichlet-type energy. They used an architecture consisting of a nonlinear Encoder (ENC), linear diffusion (without nonlinear activation, see eq. 16), and nonlinear Decoder (DEC). The aim of the project is to study the importance of ENC, and whether it is sufficient to have nonlinearity only in DEC. The implication of this will be the ability to pre-compute the diffused features, leading to very efficient scalable architecture (see Project 14).

References:

[1] https://arxiv.org/pdf/2206.10991.pdf

14.

Title: Scalable Gradient Flow Framework (GRAFF)

Description: Di Giovanni et al. [1] showed that convolutional-type GNNs could be derived as gradient flows of parametric Dirichlet-type energy. They used an architecture consisting of a nonlinear Encoder (ENC), linear diffusion (without nonlinear activation, see eq. 16), and nonlinear Decoder (DEC). In this project, we will eliminate ENC (see Project 13) and consider the linear diffusion part as a polynomial w.r.t adjacency matrix A and channel mixing matrix W. The terms of the form $A^k X$ can be pre-computed as done in the SIGN architecture [2], potentially leading to a scalable multilayer architecture.

References:

[1] https://arxiv.org/pdf/2206.10991.pdf [2] https://arxiv.org/abs/2004.11198