Extended Abstract Track

Towards Foundation Models on Graphs: An Analysis on Cross-Dataset Transfer of Pretrained GNNs

Abstract

To develop a preliminary understanding towards Graph Foundation Models, we study the extent to which pretrained Graph Neural Networks can be applied across datasets, an effort requiring to be agnostic to dataset-specific features and their encodings. We build upon a purely structural pretraining approach and propose an extension to capture feature information while still being feature-agnostic. We evaluate pretrained models on downstream tasks for varying amounts of training samples and choices of pretraining datasets. Our preliminary results indicate that embeddings from pretrained models improve generalization only with enough downstream data points and in a degree which depends on the quantity and properties of pretraining data. Feature information can lead to improvements, but currently requires some similarities between pretraining and downstream feature spaces.

Keywords: Graph Neural Networks; Foundation Models; Transfer Learning.

1. Introduction

Machine Learning is being revolutionized by Foundation Models (FMs), reference models that can be effectively adapted to diverse downstream tasks with a relatively small amount of data. Successful FMs abound in the textual and visual domains, but the development of FMs on graphs is still an open problem (Liu et al., 2023b; Mao et al., 2024) with a dangling fundamental question: *How to design an architecture that can be pretrained and, later, successfully applied across graph datasets?* This is challenging, as graphs representing different objects may exhibit different patterns in their structure and features; these last ones, across domains, may even represent semantically different spaces with distinct encodings.

If anything, such an architecture is required to be *feature-agnostic*: it may process features, but in a way that is independent of dataset-specific encodings and that encourages learning of transferable patterns. While researchers have recently started to develop feature-agnostic approaches, they are still plagued by various limitations (see Appendix A).

Towards a more foundational understanding of the matter, we find a constructive reference in the pipeline proposed by Cantürk et al. (2024). The authors propose GPSE, a method to *learn* graph positional and structural encodings (P/SEs) (Dwivedi et al., 2022a; Rampášek et al., 2022) that can be employed to augment any downstream Graph Neural Network (GNN). It consists of an expressive GNN pretrained on featureless graphs to predict P/SEs. Given a downstream graph task, the model can generate encodings that can be combined with explicit node features and fed into a GNN architecture. While pretrained only on a single molecular dataset, the authors show GPSE encodings transfer to different downstream datasets. GPSE represents an important step towards feature-agnostic approaches, but the pretraining phase disregards features, which may constitute an important source of information. Also, the authors do not explore settings of interest for FMs, viz.,

multi-dataset pretraining and downstream low-data regimes, and do not discuss label-free approaches to validate a pretrained model for a downstream target dataset.

In this work, we analyze the aforementioned aspects. We propose *Feature-Structuralization*[1] to encode feature information in structural form in a way that is compatible with structural pretraining strategies. Next, we consider three datasets that differ in their structural and feature patterns. We analyze the behavior and performance of structuralization and the original GPSE model on them, studying the impact of multi-dataset pretraining and of feature information in successfully transferring to downstream low-data regimes.

2. Feature-Structuralization

Structuralization encodes feature information into additional structure augmenting the original input graph: by only encoding abstract, logical relations induced by features, the pretraining approach remains feature-agnostic while still accessing meaningful information on how features interact with the original graph structure.
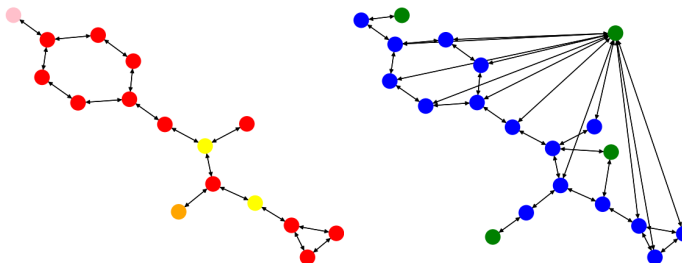

Figure 1: Left: graph from `zinc`; Right: structuralized form. Colors represent node features.

Consider a graph G with d different categorical node features[2]. For each channel $i = 1, \ldots d$, structuralization materializes an additional *feature-node* $u_{i,j}$ for any category j in i, and connects $u_{i,j}$ to the nodes $v \in G$ attributed with category j in channel i. In the transformed graph, all explicit features are discarded; nodes are only assigned a mark which allows to disambiguate between original and feature-nodes. An example of a single-channel structuralization is shown in Figure 1 where, on the transformed graph, feature-nodes are in green, original nodes in blue.

3. Experimental Setting

We seek to address the following questions: (**Q1**) *"Are embeddings from pretrained models beneficial in downstream low-data regimes?"* (**Q2**) *"How does the composition of the pretraining corpus affect downstream generalization performance?"* (**Q3**) *"Is pretraining with features information beneficial when these are incorporated via structuralization?"*. We now describe our experimental setting and refer to Appendix B for additional details.

Datasets. We opt to use three datasets: ZINC-12k (Dwivedi et al., 2020), ogbg-molpcba (Hu et al., 2020), and peptides-func (Dwivedi et al., 2022b)(resp. `zinc`, `molpcba`, `peptides`). These are from a similar domain, but with different degrees of similarity in structures and features: `peptides`'s graphs are aminoacid-chains, structurally dissimilar from the graphs in `zinc` and `molpcba` (small molecules); nodes in `molpcba` and `peptides` have the same 9 SMILES-derived categories as features, nodes in `zinc` only one category (the atom type).

---

1. We may abbreviate this simply as *"structuralization"* or *"struct"*.
2. The approach can be extended to consider edge features and non-categorical ones.

Extended Abstract Track

Pretraining. All pretrainings use the same base model from GPSE (Cantürk et al., 2024), with either the original graph structures or their feature-structuralized variants. Following Cantürk et al. (2024), pretraining involves jointly predicting predefined P/SEs from the output node representations. For structuralization, the model predicts P/SEs for the original nodes in both the original and transformed graphs. Pretraining is conducted on each dataset and on all pairwise combinations, totaling 12 possibilities. We downsampled `molpcba` to match the training set sizes of `zinc` and `peptides`.

Evaluations. We evaluate the pretrained models in two settings: (i) pretraining target prediction, assessing how well the models predict test targets from the three datasets; and (ii) downstream tasks, examining if and how the pretrained embeddings improve GNN generalization. These analyses are performed with varying downstream training sample sizes.

4. Results and Discussion

Pretraining. Figure 2 visualizes how pretrained GPSEs generalize, showing that pretraining on multiple datasets generally improves performance. Models not pretrained on `peptides` perform near to chance on `peptides`, unlike `zinc` or `molpcba`. While the gap between in- and off-domain pretraining narrows with more data, it never fully closes (see Appendix E). Adding data from other sources can still be beneficial. Pretraining on "structuralized graphs" slightly worsens in-domain and hampers out-of-dataset generalization. More results are in Appendix C.
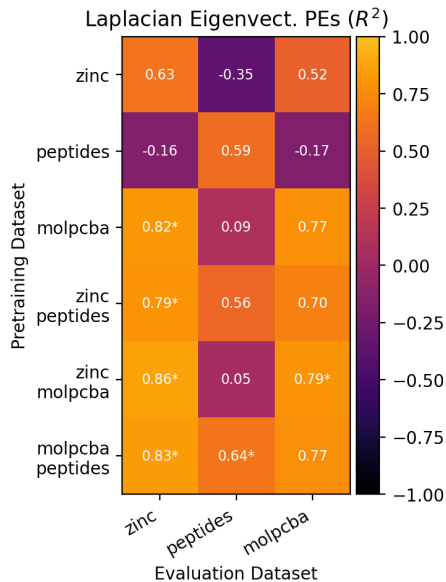

Figure 2: GPSE pretraining eval. on LapPEs (Test R^2).

Downstream applications. We now present observations for downstream evaluations on `zinc` and `peptides`, which we visually summarize in Figure 3. We also ran preliminary experiments on `molpcba`, but noticed an inherent task hardness beyond the aforementioned datasets; our initial results are in Appendix D, where we also report an extended version of Figure 3. Next, when referring to a 'baseline', we intend the same downstream architecture "deprived" of embedding from pretrained models.

(i) Pre-trained embeddings can be — but are not always — beneficial (Q1). Pretrained embeddings improve generalization over the baseline with sufficient data, but can be detrimental in data-scarce settings[3]. Notably, in `zinc`, pretrained models outperform the baseline even with just 10% of the training data.

(ii) When pretraining becomes beneficial, all pretraining mixes seem to provide better generalization than the baseline (Q2). However, the composition of the corpus may have a strong impact, which tend to reflect the similarity between source and target datasets. E.g., pretraining on `peptides` is less beneficial when transferring on `zinc` for both

---

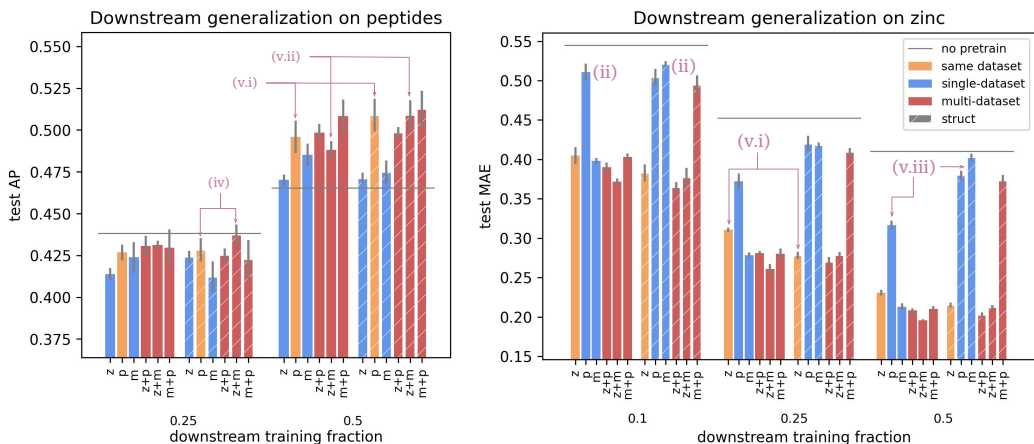3. The following observations focus on cases where pretraining is comparable or better than the baseline.

Figure 3: Downstream test performance for different training fraction and pretraining data ('z': `zinc`, 'p': `peptides`, 'm': `molpcba`). Left: Average Precision on `peptides`, higher is better. Right: Mean Absolute Error on `zinc`, lower is better.

GPSE and Struct, due to differences in both structure and features (see mark ii).

(iii) Including datasets in pretraining other than the target does not generally deteriorate performance (Q2). In general, the performance is at least as good as the one attained by the best of the two pretraining datasets.

(iv) Pretraining mixes that do not include the target dataset can perform as well as pretraining on the target (Q2). In some cases, "off-dataset" pretraining may even generalize slightly better than pretraining on the same dataset (see mark iv). We believe that this may be mostly due to an increase in pretraining data (see Appendix E).

(v) Feature structuralization does not yield a consistent, significant improvement across settings (Q3). Structuralization improves performance when in-dataset pretraining, outperforming vanilla GPSE on `zinc` when pretrained on `zinc` (mark v.i) and on `peptides` when pretrained on `peptides` (mark v.i). It can also help when the pretraining mix is off-distribution in structure but not features, e.g., on `peptides` with `zinc+molpcba` pretraining (mark v.ii). However, structuralization is less effective when features are off-distribution, such as pretraining on `molpcba` and transferring to `zinc` (mark v.iii).

We refer to Appendix E for more considerations on the size of pretraining corpora.

5. Conclusions

We analyzed the extent to which pretrained GNNs can be transferred across datasets by measuring the impact of pretraining datasets on downstream generalization and the inclusion of feature information via structuralization. Our work can be extended in different ways by: studying non-molecular domains, e.g., social and collaboration networks; closely enquiring into the underperformance of structuralization and designing more compatible pretraining strategies and architectures; exploring other feature-agnostic pretraining approaches, e.g., to natively handle continuous features for domains such as geometric graphs.

Extended Abstract Track

References

Giorgos Bouritsas, Fabrizio Frasca, Stefanos P. Zafeiriou, and Michael Bronstein. Improving graph neural network expressivity via subgraph isomorphism counting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

Xavier Bresson and Thomas Laurent. Residual Gated Graph ConvNets. *arXiv:1711.07553*, 2017. URL https://arxiv.org/abs/1711.07553.

Semih Cantürk, Renming Liu, Olivier Lapointe-Gagné, Vincent Létourneau, Guy Wolf, Dominique Beaini, and Ladislav Rampášek. Graph positional and structural encoder. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 5533–5566. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/canturk24a.html.

Vijay Prakash Dwivedi, Chaitanya K. Joshi, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *arXiv:2003.00982*, 2020. URL https://arxiv.org/abs/2003.00982.

Vijay Prakash Dwivedi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Graph neural networks with learnable structural and positional representations. In *International Conference on Learning Representations*, 2022a.

Vijay Prakash Dwivedi, Ladislav Rampášek, Michael Galkin, Ali Parviz, Guy Wolf, Anh Tuan Luu, and Dominique Beaini. Long range graph benchmark. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 22326–22340. Curran Associates, Inc., 2022b.

Mikhail Galkin, Xinyu Yuan, Hesham Mostafa, Jian Tang, and Zhaocheng Zhu. Towards Foundation Models for Knowledge Graph Reasoning. In *International Conference on Learning Representations*, 2024.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 22118–22133. Curran Associates, Inc., 2020.

Qian Huang, Hongyu Ren, Peng Chen, Gregor Kržmanc, Daniel Zeng, Percy Liang, and Jure Leskovec. PRODIGY: Enabling In-context Learning Over Graphs, May 2023. URL http://arxiv.org/abs/2305.12600. arXiv:2305.12600 [cs].

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Hao Liu, Jiarui Feng, Lecheng Kong, Ningyue Liang, Dacheng Tao, Yixin Chen, and Muhan Zhang. One for All: Towards Training One Graph Model for All Classification Tasks, December 2023a. URL http://arxiv.org/abs/2310.00149. arXiv:2310.00149 [cs].

Jiawei Liu, Cheng Yang, Zhiyuan Lu, Junze Chen, Yibo Li, Mengmei Zhang, Ting Bai, Yuan Fang, Lichao Sun, Philip S. Yu, and Chuan Shi. Towards Graph Foundation Models: A Survey and Beyond, December 2023b. URL http://arxiv.org/abs/2310.11829. arXiv:2310.11829 [cs].

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Haitao Mao, Zhikai Chen, Wenzhuo Tang, Jianan Zhao, Yao Ma, Tong Zhao, Neil Shah, Mikhail Galkin, and Jiliang Tang. Position: Graph foundation models are already here. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 34670–34692. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/mao24a.html.

Ladislav Rampášek, Mikhail Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a General, Powerful, Scalable Graph Transformer. *Advances in Neural Information Processing Systems*, 35, 2022.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.

Xingtong Yu, Chang Zhou, Yuan Fang, and Xinming Zhang. Text-Free Multi-domain Graph Pre-training: Toward Graph Foundation Models, May 2024. URL http://arxiv.org/abs/2405.13934. arXiv:2405.13934 [cs].

Haihong Zhao, Aochuan Chen, Xiangguo Sun, Hong Cheng, and Jia Li. All in one and one for all: A simple yet effective method towards cross-domain graph pretraining. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 4443–4454, New York, NY, USA, 2024a. Association for Computing Machinery. ISBN 9798400704901. doi: 10.1145/3637528.3671913. URL https://doi.org/10.1145/3637528.3671913.

Jianan Zhao, Hesham Mostafa, Mikhail Galkin, Michael Bronstein, Zhaocheng Zhu, and Jian Tang. GraphAny: A Foundation Model for Node Classification on Any Graph. June 2024b. URL http://arxiv.org/abs/2405.20445. arXiv:2405.20445 [cs].

Extended Abstract Track

Appendix A. Dealing with different features across datasets

Pioneering approaches for feature-agnostic architectures have either resorted to (i) Large Language Models (LLMs) to embed features in a shared semantic space (Liu et al., 2023a; Huang et al., 2023), (ii) adopted dimensionality reduction techniques (Zhao et al., 2024a; Yu et al., 2024), or (iii) assumed the architecture is parameterized by a 'bank' of linear GNNs whose optimal predictions can be precalculated in closed form (Zhao et al., 2024b).

As for (i), end users have little control over the semantic relations between the LLM-encoded features, which can also significantly depend on the prompting pattern and technique. This is particularly relevant when working with 'abstract' categorical features or continuous ones. Techniques in class (ii) leverage covariance information in the features to derive a certain number of principal components. This can be interpreted as a form of 'canonization', which, however, neglects structural information in the graph and does not necessarily guarantee, in itself, a form of alignment of the semantic spaces across datasets. Importantly, we note that covariance information is trivial for purely categorical feature spaces, e.g., following a one-hot encoding scheme. Approach (iii) is particularly interesting as the learnable components of the approach effectively work in the prediction space, sidestepping the problem of aligning semantic spaces across datasets. However, it poses an architectural constraint, viz., the use of linear GNNs, that may be too limiting in applications such as the ones considered in this study.

Beyond our scope, in the specific context of Knowledge Graphs, Galkin et al. (2024) propose to construct a graph between relation types based on concept of sharing 'head' or 'tail' entities. Message-passing on this graph allows the representation of any relation type and thus enables applications across Knowledge Graphs. Structuralization may share similarities with this approach in the high-level intuition; however, graphs generated by structuralization jointly represent original and feature-induced nodes and edges. The model being pretrained can access both and potentially (learn to) capture patterns in their interaction.

Appendix B. More experimental details

B.1. Pretraining and pretraining target prediction

In all pretrainings, we employ the same backbone architecture proposed by Cantürk et al. (2024): a 20-layer GNN with 512-dimensional Residual Gated GCN message-passing layers Bresson and Laurent (2017) and prediction heads constituted by 2-Layer Perceptrons with 32 hidden-dimensions, totaling slightly more than 2M parameters. The layers of the backbone architecture are regularized with dropout, whose rate is set to 0.2. As in (Cantürk et al., 2024), input graphs are augmented with virtual nodes, and 20-dimensional random node features drawn from a standard Normal distribution.

For optimization, we follow the setting proposed in (Cantürk et al., 2024): we employ the AdamW optimizer (Loshchilov and Hutter, 2019) for 120 epochs with weight decay set to 0.00001, a base learning rate of 0.005 and a Cosine-decay scheduler with 5 warmup epochs. We use the same 'mae + cosine similarity' loss (Cantürk et al., 2024).

The P/SE targets exactly correspond to those chosen in (Cantürk et al., 2024), viz., 4 Laplacian eigenvectors (taken in their absolute value) along with corresponding eigenvalues, 7 Electrostatic PEs, 20 Random Walk SEs, 20 Heat Kernel Diag. SEs. There is only one
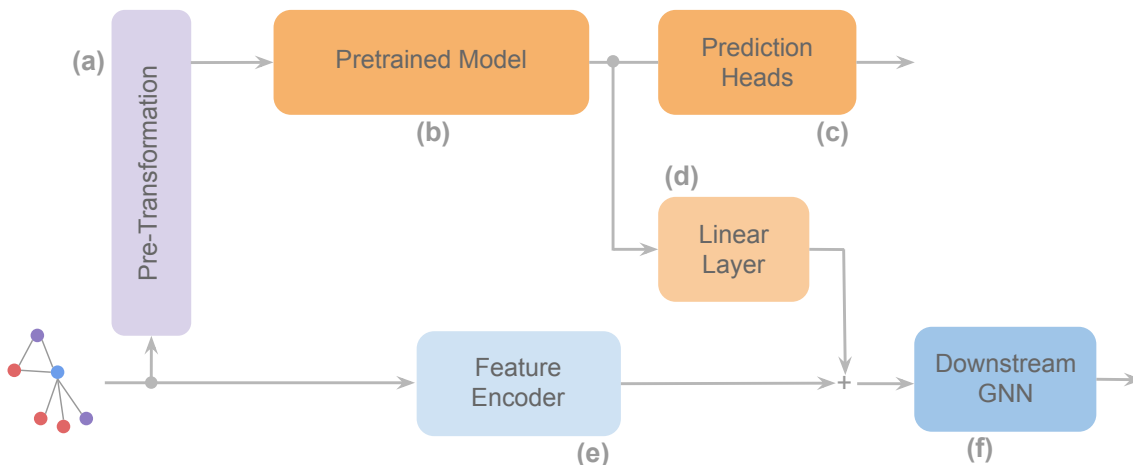
Figure 4: Architecure pipeline for downstream experiments.

exception: we do not consider Cycle Counting Graph Encodings as we observed they would have led to intractable precomputation runtimes in the case of structuralization.

We remark that, when pretraining a model on structuralized graphs, we ask it to predict the above targets on the original nodes calculated for *both* the original and the structuralized form of the input graph. This means having double the pretraining targets than when pretraining a vanilla GPSE model.

B.2. Downstream evaluation

In all downstream evaluation experiments we follow the pipeline depicted in Figure 4. In particular, component (b) is the pretrained model, from which we extract the generated node representations upstream the prediction heads used in pretraining ((c)). These representations are linearly transformed by component (d), and then summed along with the encoding of the explicit, original node features in output from component (e). These constitute the initial node representations updated and later pooled by the downstream GNN depicted in (f). Component (a) is, effectively, a graph transformation which (i) either discard features or perform structuralization as required; (ii) inject random features as prescribed by the GPSE model (Cantürk et al., 2024). Note that, during downstream adaptation, only components (d), (e), (f) are trained.

The GNN in (f) is, in all cases, a 3-layer Graph Isomorphism Network (Xu et al., 2019) employing a single linear pre-message-passing layer and a 2-Layer Perceptron post-message-passing and before graph readout. The hidden size of this architecture is fixed to 128. No dropout or weight-decay regularization is applied.

In accordance with Cantürk et al. (2024), dropout is applied before and after component (d). We do not perform tuning of dropout rates, but rather apply the same exact values chosen by the authors in (Cantürk et al., 2024) for the most prominent models in the respective full-dataset adaptation experiments. In particular, for zinc the dropout has a rate of 0.5 before (d) and is not applied after that component; as for peptides the dropout has a rate of 0.1 before (d) and, again, is not applied thereafter; for molpcba

Extended Abstract Track

the dropout has a rate of 0.3 before (d) and 0.1 after that. We reserve to tune these values for each experiment in future developments of this study; this would allow us to verify whether pretrained representations can be more beneficial in lower data regimes if differently regularized.

As for the optimization approach, we always used Adam (Kingma and Ba, 2015) and set a budget of 200 epochs. As we noticed a potentially significant impact of the optimization scheme on the results, for each experiment we tuned the choice of the base learning rate (0.0005, 0.001, 0.005) and the scheduler: Reduce-on-validation-plateaux vs. Cosine-decay-with-warmup. For the former we set a patience of 10 epochs; for the latter a number of warmup epochs equal to 5.

Appendix C. Results on pretraining target predictions

Figures 5, 6 and 7 show the evaluation of all pretraining targets. In particular, we group P/SEs by category and report the median R^2 scores attained on the test sets of the respective evaluation datasets.

Predicting the P/SE targets of the original graphs with GPSE. We observe that GPSE is able to fit most targets relatively well, although the prediction of Laplacian Eigenvector PEs seems to be a harder task (Figure 5). Either way, we note that achieving satisfactory generalization on `peptides` seems to only be possible if `peptides` was used in the pretraining.

Predicting the P/SE targets of the original graphs after structuralization. For GPSE on structuralized graphs (Figure 6), we observe our model is only able to generalize to other datasets if pretrained on multiple datasets. This indicates that the inclusion of feature information may render transfer more difficult unless the pretraining corpus is larger and/or 'diverse' (and to enquire into the relative impact of size and diversity would be an interesting future endeavor). As for the present results, we find interesting how cross-dataset generalization is relatively better across `molpcba` and `peptides`— which share the same feature encoding — and is, instead, always problematic when transferring from `molpcba+peptides` to `zinc`— which has a different feature encoding. We recall that *targets are purely structural* in this setting and thus hypothesize that the model is harmfully leveraging feature information. One last noteworthy observation is that the prediction of Eigenvalues is particularly hard for this model, even when pretraining 'in-dataset'.

Predicting the P/SE targets of structuralized graphs (after structuralization). Similar conclusions to the above can be drawn for this setting (Figure 7), except for the fact that the prediction of targets seems generally more difficult — and sometimes not possible — even when pretraining 'in-dataset' (see, e.g., the prediction of Random Walk SEs on `molpcba` or Heat Kernel Diag. SEs on all datasets). These generally lower values may contribute to explain the underperformance of structuralization in some downstream evaluation settings, and suggests more focused efforts are required on designing better pretraining strategies and/or constructing better pretraining corpora for our approach.
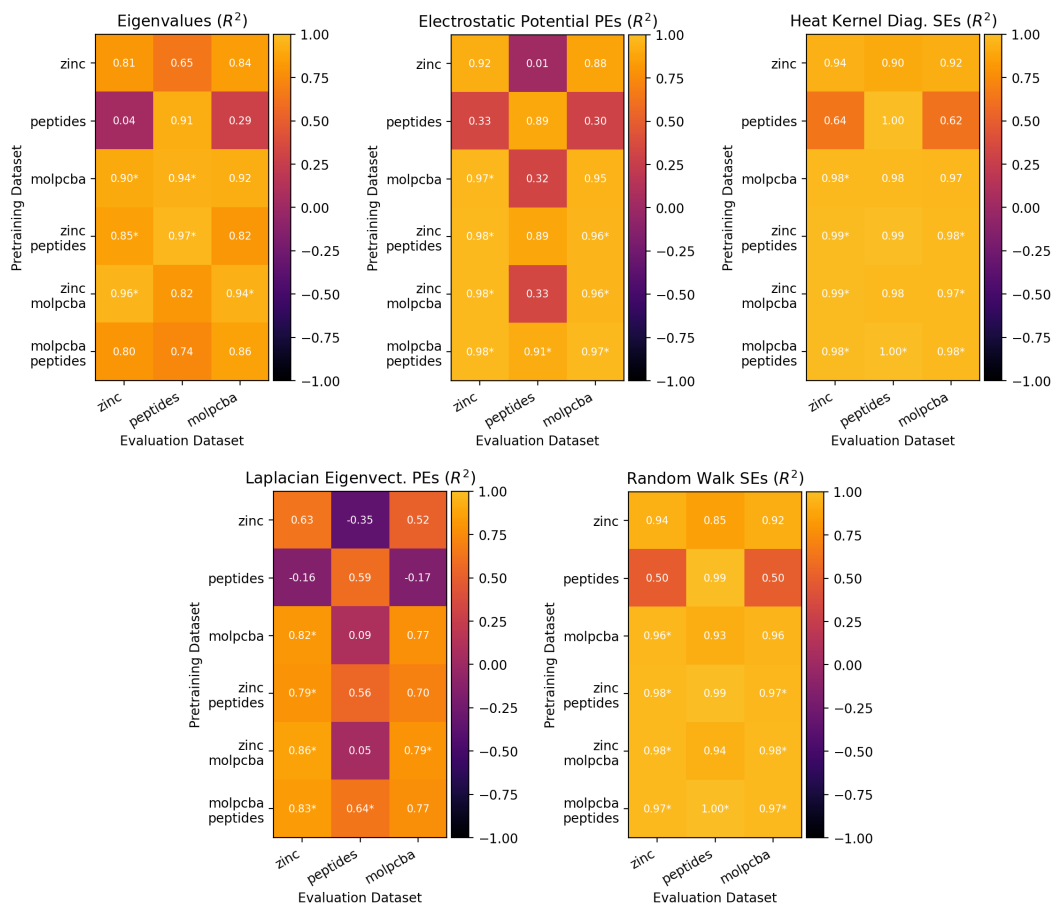
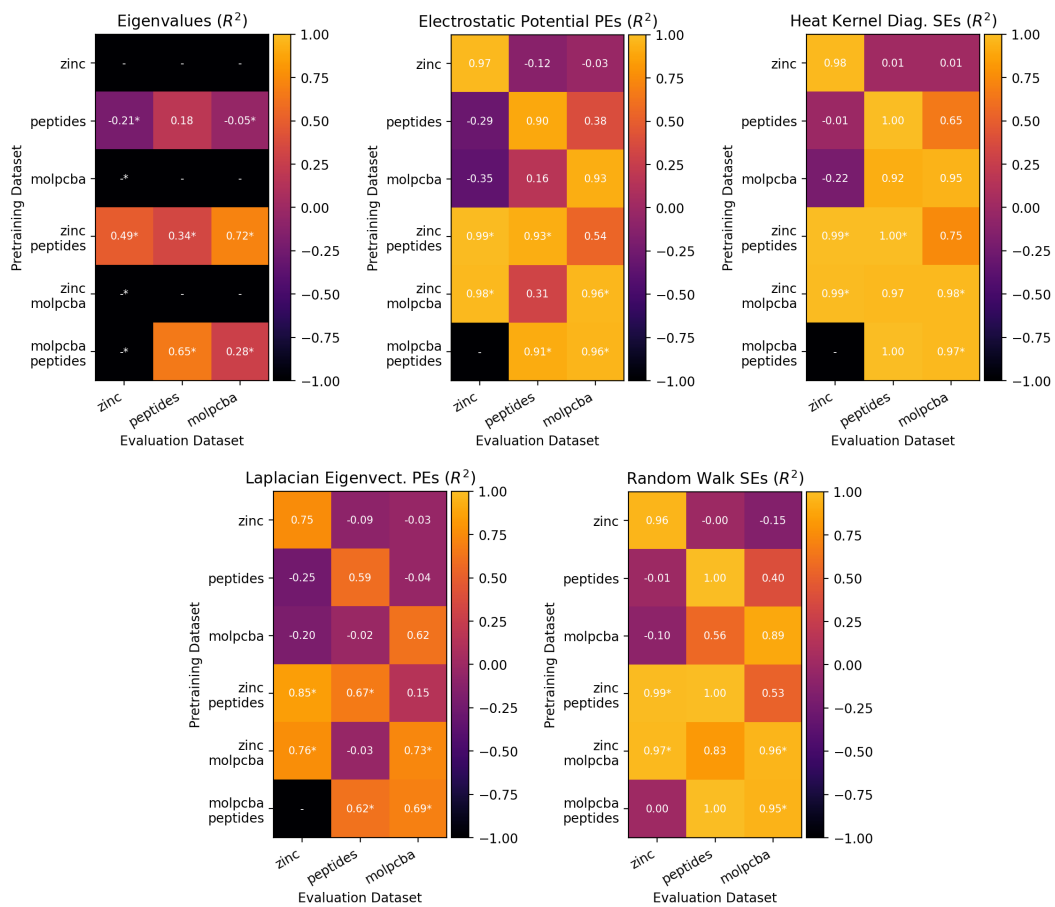Figure 5: Evaluation of predicting the P/SE targets of the original graphs with GPSE.

Extended Abstract Track

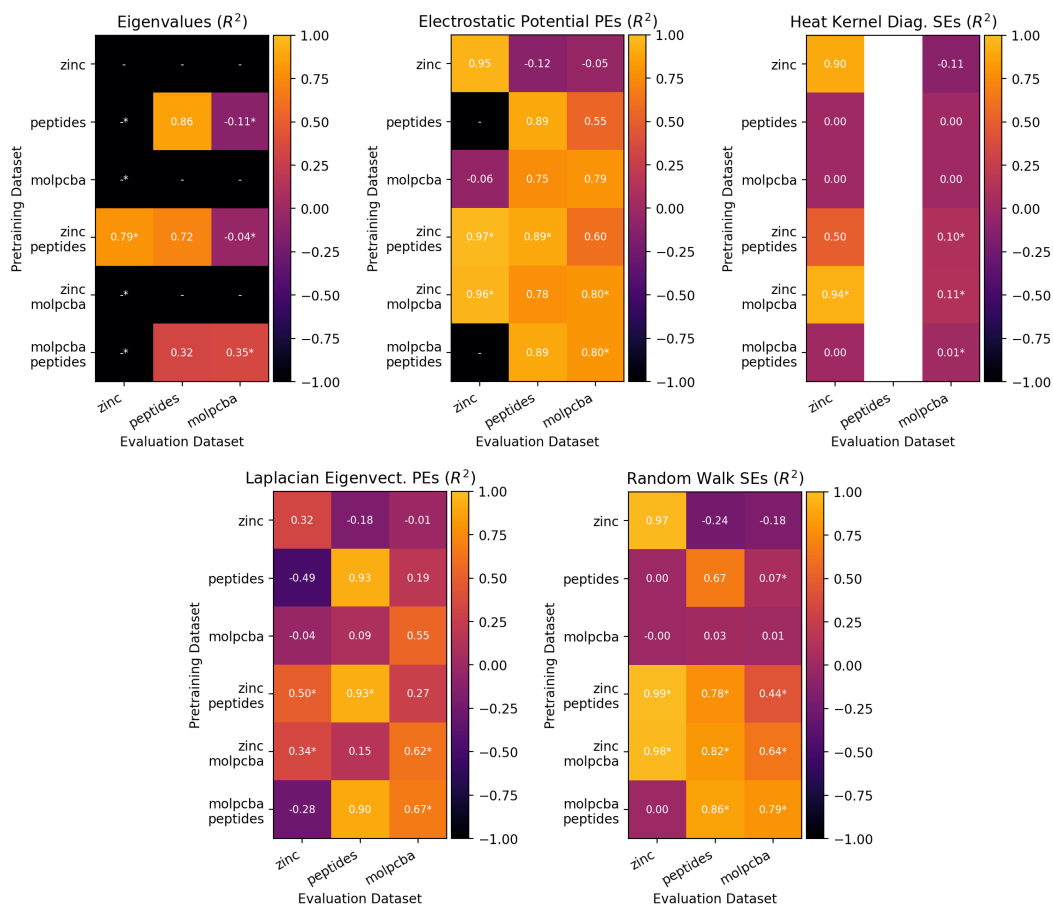Figure 6: Evaluation of predicting the P/SE targets of the original graphs after structuralization.

Figure 7: Evaluation of predicting the P/SE targets of structuralized graphs (after structuralization).
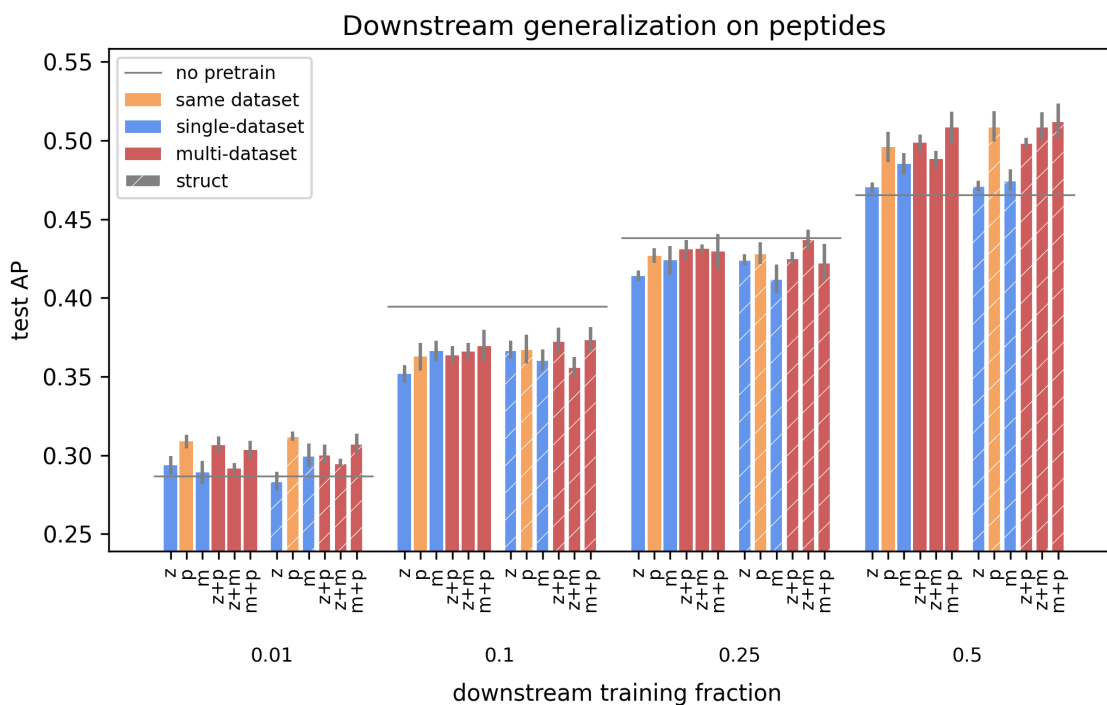
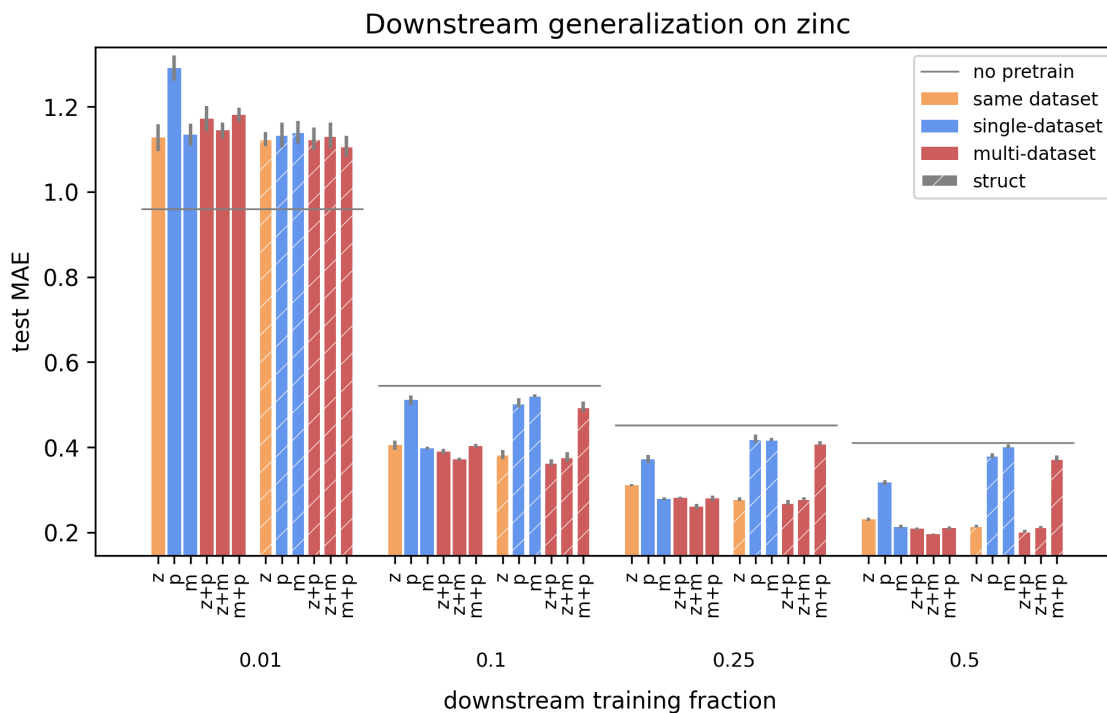Figure 8: Results of the full downstream evaluation on `peptides`.

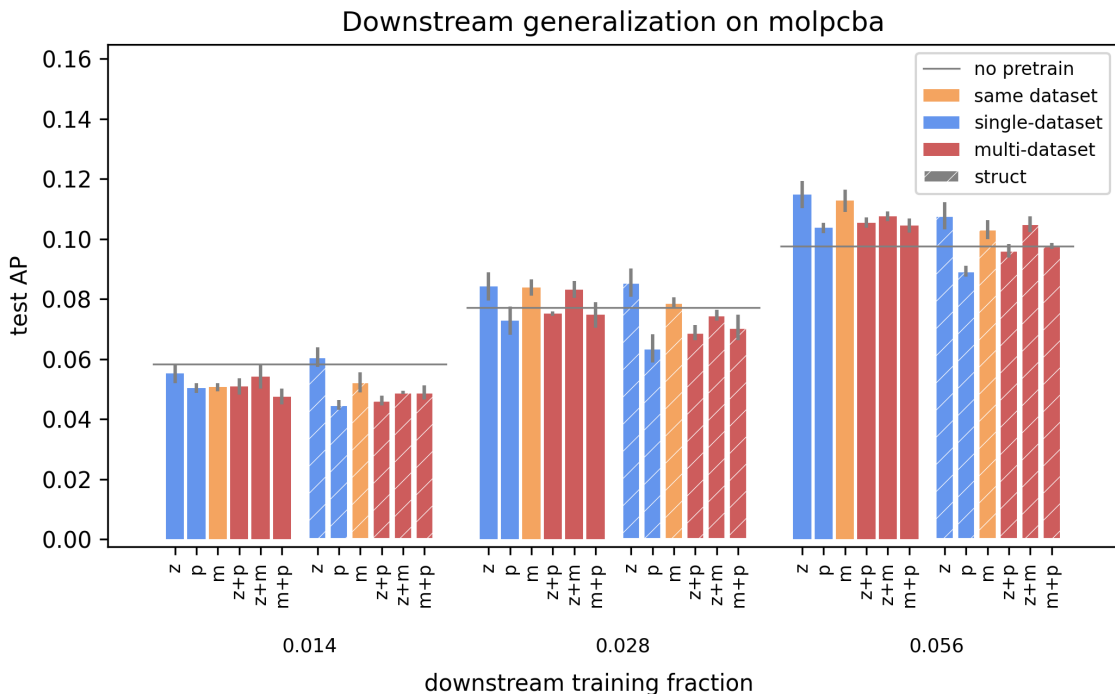Figure 9: Results of the full downstream evaluation on `zinc`.

Figure 10: Results of the full downstream evaluation on `molpcba`.

Appendix D. Full results on downstream applications

We present the full results on all downstream datasets in Figures 8, 9 and 10.

About the lowest data regimes. In the aforementioned figures it is possible appreciate the behavior of the models in lower data regimes beyond what reported in Figure 3. We observe that node representations from pretrained models are particularly harmful on `peptides` in the 0.1 training ratio setting, while, intriguingly, they provide more benefits in the lowest data regime (0.01 ratio, around 100 graphs). We hypothesize that, with such data scarcity, on `peptides` it may even be hard to learn a reasonable message-passing scheme, while pretrained embeddings already provide some ready-to-use representations that can more immediately correlate with the targets being predicted. As the amount of training data is increased up to the 0.25 ratio, the message-passing parameters can more reasonably be fitted (see baseline), but it is likely that the models leverage spurious correlations in the pretrained node embeddings. Finally, for the 0.5 ratio, enough data are available to optimally make use of both the two sources of information. As for `zinc`, the only setting where embeddings from pretrained models are not beneficial is the one of most scarcity, and otherwise provide a significant positive impact. This clearer picture is coherent with the observation that generalizing on `zinc` becomes relatively easier when the model has access to cyclic-like structural information (as the one that is provided by the pretrained model) and beyond what can be captured by message-passing (Bouritsas et al., 2022).

14

Extended Abstract Track

About `molpcba`. Without even training our models, it is clear that `molpcba` is a completely different challenge than `zinc` and `peptides`. This is because (i) the dataset is both significantly larger and (ii) the learning task is seemingly more difficult. For (i), `molpcba` contains roughly $437,929$ graphs and is thus at least 28 times larger than `zinc` ($12,000$ graphs) and `peptides` ($15,535$ graphs). Despite the abundance of data, it seems that it is much more difficult to train a model that generalizes well on `molpcba`. This is witnessed by the fact that "strong" models on `molpcba` achieve an Average Precision (AP) of ≈ 0.3 compared to ≈ 0.7 on `peptides` (Cantürk et al., 2024) (ii). Our preliminary results from downstream evaluation on `molpcba` are reported in Figure 10 and indeed showcase the inherent difficulty of the task. Training ratio 0.014 corresponds to using $\approx 5,000$ training graphs, and is then comparable with the 0.5 ratio in `zinc` and `peptides`. For this amount of data, however, the generalization performance is relatively much worse, as it only hardly reaches 0.06 test AP. The only regime where GPSE pretrained models seem to be significantly beneficial is ratio 0.056, four times the size of the training data for which the same pretrained models significantly outperformed the baseline on both `zinc` and `peptides`. Structuralization does seem to struggle even at this regime, but this may be due to a lower quality of the pretrained models (see Appendix C). In general, we believe that more extensive experimentation is due on `molpcba` in order to draw more solid conclusions. For example, it may be required to explore substantially larger training ratios, an analysis which we could not carry out at this time with the available resources.

Appendix E. Using more pretraining data

What is the impact of using more pretraining data? Let us first reason on the fact that this is not always possible. In some downstream applications one may only have access to a limited amount of samples, of which only a smaller subset is labeled. In our experiments we have observed that, at least on `zinc` and `peptides`, it may be beneficial to augment the available in-domain data with additional data sources, and that this sometimes may even improve generalization (see Figure 3).

On the other hand, it is important to notice that, throughout the experiments discussed so far, we have artificially sampled a small fraction of `molpcba`'s training data to match the size of the training sets in the other datasets. This setting is different that the one considered in (Cantürk et al., 2024), where models were pretrained on the full set of non-isomorphic structures in `molpcba`. A complete analysis on how our full set of results would change when considering the complete `molpcba` dataset is left for future work; nevertheless, we ran some preliminary experiments we will briefly discuss below.

Pretraining target prediction. We pretrained two additional GPSE models: one with twice the number of `molpcba` samples than in the rest of our experiments (around 20k graphs) (1); one on the full `molpcba` training set (2). Then, we moved to evaluate the performance of both (1) and (2) on the prediction of pretraining targets across the three datasets. We mention noteworthy results for Electrostatic Potential PEs and Laplacian Eigenvector PEs (the performance on the other targets is mostly saturated already with 10k pretraining data points). As for the Electrostatic Potential PEs, we report that the performance of GPSE on `peptides` improves from the median R^2 of 0.32 to 0.41 of (1) and 0.70 of (2). Concerning the Laplacian Eigenvector PEs, on `peptides`, the median R^2 of 0.09

improves to 0.15 in the case of (1) and 0.35 in the case of (2). This suggests that, as already mentioned in Section 4, more data can help in off-domain generalization, although — as reasonably expected — much less efficiently than in-domain additional data: ≈ 10k more (unlabeled) samples from the `peptides`' training set enable median R^2's of 0.91 and 0.64 on, resp. Electrostatic Potential PEs and Laplacian Eigenvector PEs over the `peptides`' test set. Last we report that the "in-dataset" performance for these targets also improve with more data. As for the Laplacian Eigenvector PEs, we report that the performance of GPSE on the `molpcba`'s test set improves from the median R^2 of 0.77 to 0.85 of (1) and 0.96 of (2).

Downstream evaluation on `peptides`. We evaluated models (1) and (2) on the `peptides` downstream task, following the same setting and procedure described in Appendix B. As a first important observation, we note that more pretraining data do not allow models (1), (2) to outperform the baseline in the data regimes 0.1, 0.25. We do notice, however, that more pretraining data closes the gap with models pretrained on data mixtures: (1), (2) perform comparably with, e.g., the model pretrained on `zinc+molpcba`. Last, we observe that, in the 0.5 ratio setting, model (2) attains a test AP of 0.513 ± 0.002, slightly outperforming the best pretraining mix `molpcba+peptides` (0.508 ± 0.010).