A Large-Scale Analysis of Multi-Task, Cross-Domain Pre-training for Graph Neural Networks

ALON BEBCHUK - 314023516, BEN COHEN - 323855288, and TIM KUSHMARO - 212052708

TL;DR. We present a large-scale, systematic analysis of multi-task, cross-domain pre-training for Graph Neural Networks (GNNs). We evaluate combinations of node, link, and graph-level objectives on a diverse set of downstream tasks, comparing against from-scratch, single-task, and single-domain baselines to establish best practices.

1 Background & Motivation

While pre-training has revolutionized fields like NLP and Computer Vision, its application to Graph Neural Networks (GNNs) remains underdeveloped. Most GNNs are still trained from scratch for specific tasks, and existing pre-training research is often narrow, focusing on single tasks or domains [Hu et al. 2020b,a; You et al. 2020]. This has left a critical gap: there are no established best practices for creating general-purpose, pre-trained GNNs.

This work provides the first large-scale, systematic study of multi-task, cross-domain pre-training for GNNs. We will investigate how different combinations of pre-training tasks and data from multiple domains affect downstream performance. While recent work has explored multi-task frameworks [Wang et al. 2022; Wu et al. 2022], no comprehensive comparison exists to identify which combinations produce the most generalizable representations. Our goal is to establish robust, transferable, and actionable guidelines for the community.

Our key contributions will be:

(1) **Systematic Evaluation:** A rigorous evaluation of how various combinations of node, link, and graph-level pre-training tasks impact downstream performance.
(2) **Cross-Domain Analysis:** A deep dive into the transferability of learned representations across different graph domains and task types.
(3) **Actionable Guidelines:** A set of evidence-based best practices for GNN pre-training.
(4) **Open Science:** The release of all code, pre-trained models, and experimental logs to facilitate future research.

2 Proposed Approach

Our approach is centered on a modular GNN architecture that can be pre-trained on a diverse set of tasks and then efficiently adapted to downstream applications.

2.1 Core Architecture

The model consists of three main components: a domain-specific input encoder, a shared GNN backbone, and task-specific prediction heads.

- **Input Encoder:** A domain-specific linear encoder, `Linear(D_in, 256) -> LayerNorm -> ReLU -> Dropout`, maps raw node features from a specific domain (with dimension D_{in}) into a shared 256-dimensional hidden space.
- **GNN Backbone:** A stack of 5 Graph Isomorphism Network (GIN) layers serves as the core representation learner. The update rule for a node v from layer l to $l + 1$ is a four-step process:

Authors' Contact Information: Alon Bebchuk - 314023516; Ben Cohen - 323855288; Tim Kushmaro - 212052708.

(1) **Aggregation:** $a_v^{(l)} = (1+\epsilon)h_v^{(l)} + \sum_{u \in \mathcal{N}(v)} h_u^{(l)}$
(2) **GIN Convolution:** $h'_{\text{conv}} = \text{MLP}_{\text{GIN}}(a_v^{(l)})$
(3) **Residual Connection:** $h'_{\text{res}} = h'_{\text{conv}} + h_v^{(l)}$
(4) **Post-Activation:** $h_v^{(l+1)} = \text{Dropout}(\text{ReLU}(\text{LayerNorm}(h'_{\text{res}})))$

- **Standardized Heads:** We use simple, reusable MLP heads or decoders for various prediction tasks.

2.2 Multi-Task Pre-training

We will pre-train the GNN backbone using a combination of self-supervised, supervised, and adversarial objectives on a pool of graph datasets.

- **Self-Supervised (Generative):**
  - *Node Feature Masking*: Reconstructs the initial 256-dim embeddings for 15% of nodes whose features are masked.
  - *Link Prediction*: Classifies edges as real or fake, with one uniformly sampled negative edge for each positive edge (1:1 ratio).
- **Self-Supervised (Contrastive):**
  - *Node-level (GraphCL-style)*: Contrasts augmented views of nodes. Augmentations include attribute masking, edge dropping, and subgraph sampling.
  - *Graph-level (InfoGraph-style)*: Maximizes mutual information between a graph's summary vector and its node embeddings,.
- **Supervised Auxiliary Task:**
  - *Graph Property Prediction*: Regresses on z-score standardized structural properties (node count, edge count, avg. clustering coefficient).
- **Domain-Adversarial Objective:** To promote domain-invariance, a gradient reversal layer (GRL) is used to train the GNN backbone to produce embeddings that confuse a domain classifier.

The tasks are combined into a single objective function where self-supervised and supervised tasks are balanced using uncertainty weighting, and the domain-adversarial loss is incorporated with a scheduled weight, λ:

$$\mathcal{L}_{\text{total}} = \sum_{i \in \text{Tasks}} \left(\frac{1}{2\sigma_i^2} \mathcal{L}_i + \log \sigma_i \right) - \lambda \mathcal{L}_{\text{domain}}$$

The novelty of our approach lies not in any single component, but in the large-scale, systematic combination and evaluation of these methods to produce a unified framework and a set of best practices for GNN pre-training.

3 Experimental Plan

Our experimental design is structured to systematically answer our core research questions regarding the effectiveness of different pre-training strategies.

3.1 Research Questions (RQs)

Our study is guided by the following research questions:

RQ1: When does multi-task pre-training surpass from-scratch training, and how can we mitigate negative transfer?
RQ2: Which combination of pre-training tasks yields the most generalizable representations?

RQ3: How can we most effectively adapt pre-trained models to downstream tasks (e.g., full fine-tuning vs. linear probing)?

RQ4: Is the optimal pre-training strategy dependent on the downstream task type?

3.2 Datasets

- **Pre-training Pool:** A combination of four graph classification datasets from TUDatasets [Morris et al. 2020]: `MUTAG`, `PROTEINS`, `NCI1`, and `ENZYMES`.
- **Downstream Tasks:** A diverse set of tasks to evaluate generalization.
  - *Graph Classification:* `ENZYMES` (in-domain); `FRANKENSTEIN`, `PTC_MR` (out-of-domain).
  - *Node Classification:* `Cora`, `CiteSeer` [Sen et al. 2008].
  - *Link Prediction:* `Cora`, `CiteSeer`.

3.3 Baselines and Comparisons

We will compare a **From-Scratch** baseline against the pre-training schemes detailed in Table 1. These schemes are designed to systematically isolate the impact of different strategies (e.g., single- vs. multi-task, generative vs. contrastive, single- vs. multi-domain) and the effect of auxiliary and adversarial objectives.

Table 1. Experimental Pre-training Schemes

Scheme ID	Name	Tasks Included	Purpose & Research Question Addressed
B1	From-Scratch	None	(Baseline) Establishes baseline performance. Answers **RQ1**.
B2	Single-Task (Generative)	NFM	(Baseline) A representative generative single-task model. For **RQ1 & RQ2**.
B3	Single-Task (Contrastive)	NC	(Baseline) A representative contrastive single-task model. For **RQ1 & RQ2**.
B4	Single-Domain (All Objectives)	NFM + LP + NC + GC + GPP	(Baseline) Isolate benefit of multi-domain data by pre-training on `ENZYMES` only. For **RQ2**.
S1	Multi-Task (Generative)	NFM + LP	Tests synergy of purely generative tasks. Addresses **RQ2**.
S2	Multi-Task (Contrastive)	NC + GC	Tests synergy of purely contrastive tasks. Addresses **RQ2**.
S3	Multi-Task (All Self-Supervised)	NFM + LP + NC + GC	Tests the synergy of all self-supervised paradigms. For **RQ2**.
S4	Multi-Task (All Objectives)	NFM + LP + NC + GC + GPP	Combines all objectives to establish a performance ceiling. For **RQ2**.
S5	Multi-Task (Domain-Invariant)	NFM + LP + NC + GC + GPP + DA	Extends the all-objective model with domain-adversarial training to improve transferability and mitigate negative transfer. For **RQ1 & RQ2**.

3.4 Evaluation Protocols

Success will be measured by performance and efficiency, averaged over 3 random seeds.

- **Finetuning Strategies:** We will use two methods: (1) **Full Fine-tuning**, where the GNN backbone is unfrozen, and (2) **Linear Probing**, where it is frozen. In both cases, a new prediction head is trained from scratch.
- **Performance Metrics:** Accuracy, F1-Score, and AUC-ROC.
- **Efficiency Metrics:** Convergence speed (epochs to best validation score) and wall-clock training time.

3.5 Experimental Scope

The study requires a significant number of runs:

- **Pre-training Runs:** 8 models $\times$ 3 seeds = **24 pre-training runs**.
- **Finetuning Runs:** A from-scratch baseline plus 8 pre-trained models on 7 downstream tasks using 2 strategies and 3 seeds results in $(1 \times 7 \times 1 \times 3) + (8 \times 7 \times 2 \times 3) = 21 + 336 =$ **357 finetuning runs**.

References

Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. 2020b. Strategies for Pre-training Graph Neural Networks. In *International Conference on Learning Representations*.

Ziniu Hu, Yuxiao Dong, Kuansan Wang, Kai-Wei Chang, and Yizhou Sun. 2020a. GPT-GNN: Generative Pre-training of Graph Neural Networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

Christopher Morris, Johannes Klicpera, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. 2020. Tudataset: A collection of benchmark datasets for learning with graphs. In *ICML 2020 Workshop on Graph Representation Learning and Beyond (GRL+)*.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. 2008. Collective classification in network data. In *AI magazine*, Vol. 29. 93–93.

Keqiang Wang, Jing Wang, Lin Li, Wayne Xin Zhao, Quan Gan, Wei Li, Xiangnan He, and Ji-Rong Wen. 2022. Graphmvp: A multi-view prototypical-based pre-training framework for graph neural networks. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1883–1893.

Likang Wu, Zhi Li, Hongke Zhao, Zhefeng Wang, Baoxing Huai, and Nicholas Jing Yuan. 2022. Multi-task self-supervised graph neural networks enable stronger task generalization. *arXiv preprint arXiv:2210.02016* (2022).

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph contrastive learning with augmentations. In *Advances in neural information processing systems*, Vol. 33. 5812–5823.