Pareto-Based Heterogeneous Knowledge Distillation for MLPs on Graphs

Wenrui Zhao[1], Yijun Tian[2,*], Zhichao Xu[3], Yawei Wang[4], Chuxu Zhang[5]

[1]George Mason University
[2]University of Notre Dame
[3]University of Utah
[4]The George Washington University
[5]University of Connecticut

Abstract

Heterogeneous Graph Neural Networks (HGNNs) have demonstrated remarkable capabilities in capturing effective information in heterogeneous graphs, achieving outstanding performance in various learning tasks. However, their heavy dependency on neighbor information may result in high latency, which restricts their practicality in real-world applications. Recent studies have attempted to overcome such latency in Graph Neural Networks (GNNs) by distilling knowledge into student models that do not rely on graph structure. But these approaches primarily focus on replicating teachers' predictive outcomes while neglecting the structural knowledge they encoded. This limitation makes such approaches less effective when graphs become complex, particularly in heterogeneous graphs. Motivated by this challenge, we propose HGKD, a novel hierarchical knowledge distillation framework that transfers both structural knowledge and predictive outcomes from HGNN teachers to a multi-layer perceptron (MLP) student. Additionally, we provide two variants of HGKD that help the student learn from multiple teacher models via Pareto learning, and incorporate low-cost neighbor information. We evaluate HGKD and its variants on a range of heterogeneous graph datasets. The results demonstrate that the student model achieves performance comparable to, or even exceeding, that of HGNN teachers, despite not relying on graph structures during inference.

Introduction

Heterogeneous graphs have become a powerful structure for modeling complex real-world systems. The capability of representing relationships among multiple types of nodes and edges has enabled their impact across various domains (Sun, Yu, and Han 2009; Sun and Han 2013), including academic networks (Wang et al. 2023), social networks (Cao et al. 2021), biological networks (Bai et al. 2021) and film networks (Shi et al. 2019; Hu et al. 2020a).

To leverage the rich structural and semantic information inherent in heterogeneous graphs, HGNNs have been widely adopted for graph learning tasks, achieving impressive performance (Shi et al. 2019; Wang et al. 2021a). However, despite their strong potential, HGNNs remain underutilized in real-world deployments. A key limitation lies in their dependence on graph topology, a challenge shared with traditional GNNs (Zhang et al. 2020). Typical HGNNs follow a neighborhood aggregation paradigm, requiring access to neighboring node features during inference (Schlichtkrull et al. 2017; Wang et al. 2019; Tian et al. 2022a; Fu et al. 2020; Zhang et al. 2019; Chen et al. 2023). In practical scenarios, especially in latency-sensitive applications, retrieving such neighborhood information can introduce unacceptable latency, rendering HGNN-based methods impractical.

To address inference inefficiency in GNNs, prior works (Zhang et al. 2022b; Chen et al. 2022; Wang et al. 2021b; Hu et al. 2021; Wu et al. 2023a; Tian et al. 2022b; Zheng et al. 2021) have explored accelerating inference with GNN-to-MLP knowledge distillation (KD) (Hinton, Vinyals, and Dean 2015; Ba and Caruana 2014). These distillation strategies enable the graph-independent student to replace the graph-dependent teacher during inference without accessing neighborhood information. In node classification, Zhang et al. (2022b) and others have shown that an MLP student can achieve performance comparable to that of the GNN teacher (Liu et al. 2024; Wu et al. 2023b; Lu et al. 2024).

Similarly, HGNNs rely on neighborhood information, and the incorporation of heterogeneous semantics, such as metapath-based aggregation, further increases the complexity and latency of neighbor retrieval. To eliminate this latency, we attempt to extend GNN-to-MLP KD to HGNNs by: (1) applying the GNN-to-MLP KD paradigm with HGNNs as teachers; and (2) applying traditional GNNs to heterogeneous graphs and subsequently performing GNN-to-MLP KD. However, both attempts proved to be ineffective due to the inherent complexity of heterogeneous graphs. The presence of multiple node and edge types encodes rich and complex semantics that traditional GNNs and naive KD strategies fail to model, leading to suboptimal performance. To address this problem, we propose a flexible **H**eterogeneous **G**raph **K**nowledge **D**istillation (**HGKD**) framework that facilitates effective HGNN-to-MLP knowledge distillation. HGKD explicitly transfers heterogeneous graph structural knowledge to the MLP student through a metapath-based method. Specifically, our framework extracts teacher knowledge at three levels, including:

- `Soft Target`: The student learns from the soft targets generated by teachers at this level. This level aligns the

*Corresponding Author (meetyijun@gmail.com).

student's output distribution with the teacher's, allowing the MLP to capture high-level predictive knowledge.

- `Node Embedding`: At this level, node embeddings generated by teachers through iterative aggregation are transferred to the student, enabling it to learn the local graph structural information encoded in the teacher's embeddings and infer structure-aware embeddings aligned with the graph directly from node features.
- `Structural Knowledge`: For each metapath type, a similarity matrix is computed from teacher embeddings over metapath-based neighborhoods, enabling the transfer of global semantic information guided by metapaths.

HGKD also incorporates ground-truth supervision to improve performance and identify potential teacher errors. Notably, our design ensures that HGKD does not depend on the internal architecture of teacher models, enabling the framework to be extended to variants that learn from multiple teachers with different model architectures.

To further improve the performance and flexibility of HGKD across practical scenarios, we propose two enhanced variants: **HGKD+** and **HGKD-N**. HGKD+ mitigates the limitations of single-teacher distillation through a Pareto-based multi-teacher strategy, allowing the student to aggregate complementary knowledge from teachers specializing in different relational or semantic patterns, thereby enhancing robustness and generalizability across tasks. HGKD-N aims to further close the gap between HGNNs and MLPs by incorporating limited neighborhood information. While the core HGKD framework removes structural dependency, HGKD-N introduces a lightweight mechanism that helps the student capture local patterns with minimal extra cost. These variants strengthen HGKD's capacity to balance efficiency, adaptability, and predictive performance.

We evaluate our methods on widely used heterogeneous graph datasets. Experimental results demonstrate that HGKD enables the MLP-based student model to achieve performance comparable to the teacher model and, in most cases, surpass it. In addition, HGKD+ allows the student to outperform the strongest teacher model, while HGKD-N, by incorporating limited neighborhood information, further enhances the performance of both HGKD and HGKD+. To summarize, the contributions of this work are as follows:

- We propose HGKD, a novel knowledge distillation framework that enables MLPs to learn from HGNNs effectively, achieving competitive performance while eliminating the dependency on graph structure during inference.
- We present two advanced variants of HGKD: (1) HGKD+ introduces a Pareto-based multi-teacher distillation strategy, enabling flexible knowledge transfer from diverse HGNN teacher models to a single student. (2) HGKD-N incorporates limited neighborhood information to help the student better capture local structural patterns.
- Our experiments across multiple heterogeneous graph datasets reveal that the MLP is capable of effectively imitating HGNN teachers by learning from them. Through HGKD, the distilled MLP achieves performance that consistently matches or exceeds that of the teacher model.

Related Work

This work is closely related to heterogeneous graph neural networks and knowledge distillation.

Heterogeneous Graph Neural Networks. In existing research, various HGNNs have been designed to address learning tasks on heterogeneous graphs (Zhang et al. 2019; Tian et al. 2023a; Bi et al. 2025). For example, HAN (Wang et al. 2019) introduces hierarchical attention mechanisms into HGNNs, employing both node-level and semantic-level attention to capture the importance of different node types and metapaths. HGT (Hu et al. 2020b) proposes a Transformer-based HGNN, leveraging node- and edge-type dependent attention to model dynamic relationships. MAGNN (Fu et al. 2020) enhances heterogeneous graph learning through metapath instance encoding and hierarchical aggregation, effectively capturing local structure and semantic information. HetGNN (Zhang et al. 2019) introduces an HGNN framework based on random walk and heterogeneous neighbor aggregation, capable of handling multiple node and edge types. However, the heavy reliance of these HGNN models on graph structure during inference results in substantial latency, restricting their deployment in real-world applications requiring low-latency inference.

GNN Knowledge Distillation. Some works aim to distill knowledge from GNNs into lightweight GNN models (Liu, Zhang, and Wang 2020; Tian et al. 2023b; Deng and Zhang 2021; Zhuang et al. 2022; Yang et al. 2020; Zhang et al. 2021). TinyGNN (Yan et al. 2020) introduces an efficient framework that learns local structure knowledge through peer node information and a neighbor distillation strategy. GraphSAIL (Xu et al. 2020) proposes a graph structure-aware incremental learning framework that enables efficient model updates while mitigating forgetting by preserving nodes' long-term preferences and items' properties through explicit local, global, and self-information preservation. CPF (Yang, Liu, and Shi 2021) presents a distillation framework that transfers knowledge from an arbitrary trained GNN teacher to a lightweight student, integrating parameterized label propagation and feature transformation to preserve structural and feature-based knowledge for more interpretable and effective semi-supervised learning.

These approaches enable lightweight GNN students to achieve performance comparable to their teachers but do not address the inference delay caused by structural dependencies. To overcome this limitation, GNN-to-MLP distillation methods are proposed. GLNN (Zhang et al. 2022b) trains an MLP student using logits generated by GNNs(Hamilton, Ying, and Leskovec 2017), removing graph dependency during inference and accelerating prediction. VQGraph (Yang et al. 2024) leverages a VQ-VAE-based structure-aware tokenizer to encode nodes' local substructures as discrete codes, facilitating knowledge transfer from GNNs to MLP students. LightHGNN (Feng et al. 2024) distills knowledge from Hypergraph Neural Networks (Feng et al. 2019) into MLPs through soft labels. However, distilling knowledge learned by HGNNs on heterogeneous graphs into MLP students remains underexplored. Given the substantially higher structural complexity of heterogeneous graphs, existing methods exhibit limited effectiveness when applied to HGNNs.

Notation	Definitions
$\mathcal{V}$	The set of nodes in a graph
v	A node $v \in \mathcal{V}$
$\mathcal{E}$	The set of edges in a graph
$e_{i,j}$	An edge $e_{i,j} \in \mathcal{E}$ connects node i and j
$\mathcal{X}$	The feature matrix of nodes $v \in \mathcal{V}$
Φ	The set of metapath types
p	A type of metapath $p \in \Phi$
$\mathcal{N}_v^p$	A set of metapath p based neighbor of node v
$\mathcal{N}^p$	A set of one-hop metapath p based neighbor
h_v	The embedding of node v
$\mathcal{V}_i$	The node set of node-type of interest
$\|h_v\|$	Euclidean norms of h_v

Table 1: Frequently used notations

Preliminary

This section presents the key terminologies and definitions. The frequently used notations are summarized in Table 1.

Heterogeneous Graph. A heterogeneous graph is defined as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ associated with a node type mapping function $\varphi : \mathcal{V} \to \mathcal{A}$ and an edge type mapping $\psi : \mathcal{E} \to \mathcal{R}$. $\mathcal{A}$ and $\mathcal{R}$ denote the sets of node and edge types, and $|\mathcal{A}| + |\mathcal{R}| > 2$. Attributes of nodes in $\mathcal{V}$ are denoted as $\mathcal{X}$.

Metapath. A metapath (Dong, Chawla, and Swami 2017; Sun et al. 2011) $p \in \Phi$ is defined as a path in the form of $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \cdots \xrightarrow{R_l} A_{l+1}$, abbreviated as $A_1 \cdots A_{l+1}$. It describes a composite relation $R = R_1 \circ R_2 \circ \cdots \circ R_l$ between node types A_1 and A_{l+1}, where $\circ$ denotes the composition operator on relations.

Metapath based Neighbors. The metapath based neighbors $\mathcal{N}_v^p$ are defined as the set of nodes connected to node v through metapath p.

Heterogeneous Graph Embedding. The heterogeneous graph embedding task (Wang et al. 2023; Mei, Pan, and Liu 2022) learns node representations $h_v \in \mathbb{R}^d$ for v, where $d \ll |\mathcal{V}|$, given a heterogeneous graph $\mathcal{G}$ and node attributes $\mathcal{X}$.

Heterogeneous Graph Neural Networks. Typical HGNNs take a heterogeneous graph as input and follow the message-passing paradigm. In this scheme, an aggregation function $A(\cdot)$ is used to aggregate information from neighbors, and the representation of each node is updated iteratively in every layer via an update function $U(\cdot)$.

For layer l, the representation of node u is obtained by:

$$h_u^l = U(A(\{h_u^{(l-1)}, h_v^{(l-1)}, e_{u,v} \mid v \in \mathcal{N}_u\})), \quad (1)$$

where $e_{u,v}$ represents the edge between u and v, $\mathcal{N}_u$ is the neighbor set of node u.

Multi-Objective Optimization. Multi-Objective Optimization (MOO) aims to optimize multiple conflicting objectives simultaneously (Sener and Koltun 2018). Formally, a MOO problem is defined as:

$$\min_{x \in X} \quad F(x) = (f_1(x), f_2(x), ..., f_m(x)), \quad (2)$$

where $x \in X$ represents the decision variable in the feasible solution space X, $F(x)$ consists of m objective functions $f_i(x) : x \to \mathbb{R}$, each mapping a solution x to an objective score. In HGKD+, we formulate the multi-teacher learning problem as a MOO problem.

Dominance. In a MOO problem, a solution x' is said to dominate another solution x (denoted as $F(x') \prec F(x)$) if x' is at least as good as x in all objectives and strictly better in at least one objective:

$$\begin{aligned} \forall i \in \{1, ..., m\}, \quad f_i(x') \le f_i(x) \quad \text{and} \\ \exists j \in \{1, ..., m\}, \quad f_j(x') < f_j(x). \end{aligned} \quad (3)$$

Pareto Optimal. A solution x is considered Pareto-optimal if no other solution $x \in X$ improves one objective without deteriorating at least one other objective. In other words, no feasible solution $x' \in X$ dominates x:

$$\nexists x' \in X,\ F(x') \prec F(x). \quad (4)$$

Such an x is referred to as a Pareto-optimal solution.

Methodology

In this section, we introduce a novel **H**eterogeneous **G**raph **K**nowledge **D**istillation (HGKD) framework, which enables MLP student models to learn from HGNN teacher models hierarchically. To further improve its performance and adaptability, we propose two enhanced variants of HGKD. An overview of HGKD and its variants is shown in Figure 1.

HGKD

HGKD aims to transfer both predictive and structural knowledge from a powerful HGNN teacher to a lightweight MLP student, enabling efficient inference without relying on graph structure. The framework consists of three complementary objectives: soft target distillation; node embedding distillation; and structural knowledge distillation.

Soft Target Distillation. One simple yet efficient way to transfer knowledge from a cumbersome model to a compact model is training the compact student model with soft targets generated by the cumbersome teacher (Hinton, Vinyals, and Dean 2015; Liu, Zheng, and Hao 2022; Heo et al. 2019).

Specifically, for each node v in the target node set $\mathcal{V}_i$, we generate soft target z_v with a HGNN teacher and minimize the Kullback-Leibler (KL) divergence to align the the student model's prediction $\hat{y}_v$ with z_v. The objective is:

$$\mathcal{L}_{ST} = \sum_{v \in \mathcal{V}_i} \mathcal{L}_{KL}(\hat{y}_v, z_v). \quad (5)$$

It facilitates direct supervision from the teacher's predictive distribution and accelerates early-stage convergence.

Graph Embedding Distillation. Node embeddings in heterogeneous graphs capture rich structural and semantic relationships among diverse node and edge types (Dong, Chawla, and Swami 2017; Fu, Lee, and Lei 2017). In HGKD, we distill the embedding h_v^l of $v \in \mathcal{V}_i$ from the last layer l of teachers. We then maximize the similarity between h_v^l and the embedding of v in the student model by:

$$\mathcal{L}_{EMB} = \sum_{v \in \mathcal{V}_i} (1 - \frac{h_v^l \cdot \hat{h}_v^m}{\|h_v^l\| \|\hat{h}_v^m\|}), \quad (6)$$

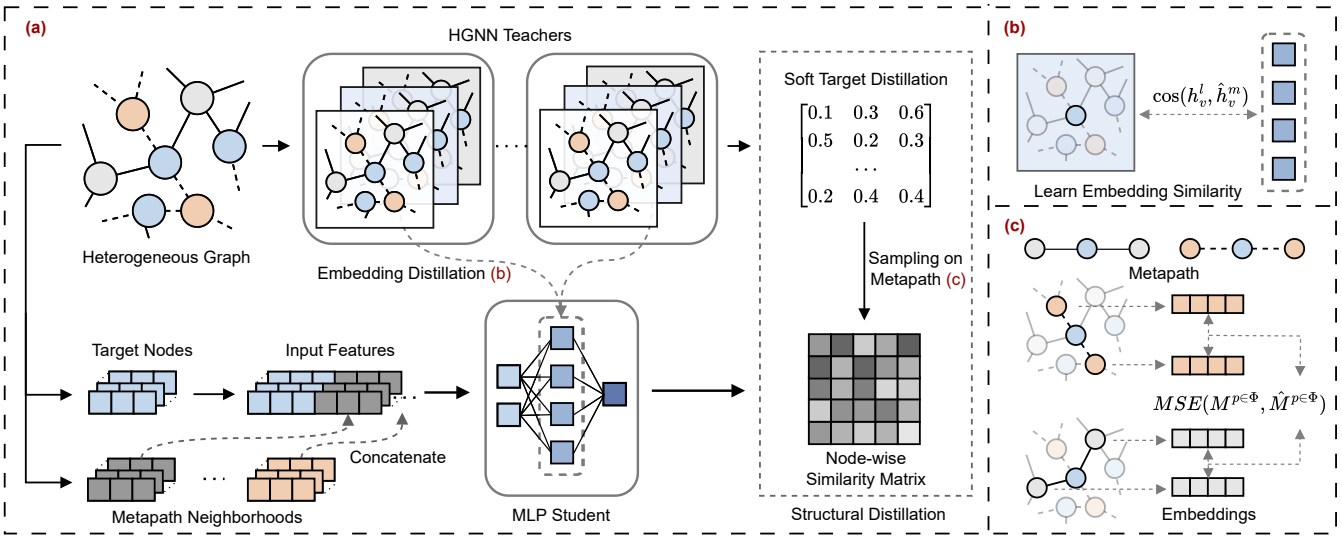

Figure 1: Overall framework of HGKD. Figure (a) illustrates the hierarchical KD process, where the student model learns from the teacher at three levels. In HGKD-N, features are augmented by concatenating neighborhood information. In HGKD+, multiple HGNN teachers are involved to provide complementary guidance. Figure (b) shows the embedding-level distillation process, where the student learns representations from the teacher's final layer. Figure (c) presents the process of sampling metapath-based similarity matrices and aligning the student with the resulting semantic patterns.

where $\hat{h}_v^m$ is the embedding of node v in the last layer m of the student model. This objective allows the student to learn context-aware representations from teacher models.

Structural Knowledge Distillation. This level aims to distill metapath-based node-pair relationships, which collectively encode the global semantic patterns associated with each metapath type in the heterogeneous graph.

Nodes connected through one or more metapaths often exhibit semantic relationships. For example, two movies linked via the Movie-Director-Movie metapath are likely to share genres and other attributes, as they are directed by the same person. We extract such metapath-based knowledge and transfer the encoded structural information from HGNNs to the MLP, effectively compensating for the missing structural information in non-graph-dependent students.

Specifically, we sample a set of one-hop metapath based neighbor pairs $\mathcal{N}^p$ for each metapath type $p \in \Phi$. For each pair of neighbors $i, j \in \mathcal{N}^p$, we obtain their embedding h_i, h_j and $\hat{h}_i, \hat{h}_j$ from teacher and student respectively. We then construct the teacher similarity matrices M^p using cosine similarity to represent the learned structural knowledge:

$$M^p_{i,j} = \frac{h_i \cdot h_j}{\|h_i\|\|h_j\|}, \forall \{i,j\} \in \mathcal{N}^p. \tag{7}$$

The student similarity matrices $\hat{M}^p$ are derived by substituting h with corresponding student embeddings $\hat{h}$. To transfer this structural knowledge, we minimize the mean squared error (MSE) between the similarity matrices M and $\hat{M}$:

$$\mathcal{L}_{SK} = \sum_{p\in\Phi} \mathcal{L}_{MSE}(M^p, \hat{M}^p). \tag{8}$$

Finally, we incorporate ground-truth supervision through a cross-entropy loss, helping the student identify errors made by teachers. Let y_v denote the label of node v. The loss is:

$$\mathcal{L}_{label} = \sum_{v\in\mathcal{V}_i} \mathcal{L}_{CE}(y_v, \hat{y}_v). \tag{9}$$

The final objective function $\mathcal{L}$ is defined as the weighted combination of the four loss funstions described above: the soft target loss $\mathcal{L}_{ST}$, the graph embedding loss $\mathcal{L}_{EMB}$, the structural knowledge loss $\mathcal{L}_{SK}$ and the label loss $\mathcal{L}_{label}$:

$$\mathcal{L} = \lambda\mathcal{L}_{ST} + \gamma\mathcal{L}_{EMB} + \mu\mathcal{L}_{SK} + \eta\mathcal{L}_{label}, \tag{10}$$

where $\lambda, \gamma, \mu, \eta$ are the corresponding weight parameters.

HGKD-N: Incorporating Neighbor Information

In practical scenarios, certain types of subgraphs can be obtained at very low cost. For example, in academic networks, information on authors and conferences is typically associated with each paper, making it inexpensive to retrieve neighbors based on Paper-Author and Paper-Conference relations. To leverage such low-cost information, we propose a variant of HGKD, **HGKD-N**eighborhood.

For each node v, we sample a set of neighbors $\mathcal{N}_v^s$ based on simple relationships s, where each s consists of a single relation $R \in \mathcal{R}$. For each s, we average the feature of nodes in $\mathcal{N}_v^s$ to obtain aggregated feature vectors $\bar{X}_v^s$, and concatenate $\bar{X}_v^s$ with the original node feature x_v to form the enhanced node representation x'_v:

$$\bar{X}_v^s = \frac{1}{|\mathcal{N}_v^s|} \sum_{i\in\mathcal{N}_v^s} x_i, \quad x'_v = Concat(x_v, \bar{X}_v^s), \tag{11}$$

and use x'_v as input to the student model. In our experiments, we find this method simple but highly effective.

Algorithm 1: HGKD+: Multi-Teacher Distillation Strategy

Require: Student parameters θ; Teachers $T_1 \cdots T_n$; Labels Y; KD loss $\mathcal{L}_{KD} = \lambda\mathcal{L}_{ST} + \gamma\mathcal{L}_{EMB} + \mu\mathcal{L}_{SK}$; Supervised loss $\mathcal{L}_{label}$; Learning rate η; Iterations N; Weight parameters α, β.

for $t = 1$ to N **do**
  Compute KD losses $\mathcal{L}_i = \mathcal{L}_{KD}(\theta, T_i)$ for all teachers
  Compute supervised loss $\mathcal{L}_{label} = \mathcal{L}_{label}(\theta, Y)$
  Compute gradients: $g_i = \nabla_\theta \mathcal{L}_i$, $g_{label} = \nabla_\theta \mathcal{L}_{label}$
  Initialize projected gradients: $g_i' \leftarrow g_i$
  for $i = 1$ to n **do**
    for $j = i + 1$ to n **do**
      if $(g_i')^\top g_j' < 0$ **then**
        $g_i' \leftarrow g_i' - \frac{(g_i')^\top g_j'}{\|g_j'\|^2} g_j'$
      end if
    end for
  end for
  Compute final gradient:

$$g_{\text{final}} = \alpha \sum_{i=1}^{n} g_i' + \beta g_{\text{label}}$$

  Update $\theta \leftarrow \text{Optimizer}(\theta, \eta, g_{\text{final}})$
end for
return θ

HGKD+: Distill Knowledge from Multi-Teacher

Different teacher models capture diverse aspects of knowledge due to their distinct parameters and inherent architectures. Learning from multiple teachers allows the student to aggregate complementary knowledge and mitigate the limitations of any single teacher (Liu, Zhang, and Wang 2020; Wu et al. 2022; Zhang et al. 2022a; Tian et al. 2024). Therefore, we introduce a variant designed for scenarios where multiple teachers are available, **HGKD+**. HGKD+ formulates multi-teacher KD as a MOO problem and optimizes the student along directions that improve all objectives simultaneously, thereby approximating a Pareto-optimal solution.

Inspired by PCGrad and other Pareto learning methods (Yu et al. 2020; Hoang et al. 2023; Lin et al. 2019; Bahlous-Boldi et al. 2025), HGKD+ detects potential conflicts among teachers' guidance and trains the student by leveraging their consensus. Specifically, we apply the assembled loss $\mathcal{L}$ to the distillation of each teacher $T_1 \cdots T_n$ and calculate the corresponding gradients $g_1 \cdots g_n$ separately. For each pair (g_i, g_j), we resolve conflicts by subtracting the projection of one gradient onto the other if $g_i^T g_j < 0$. The processed gradients are then aggregated and used to update the student parameters. The Full procedure is presented in Algorithm 1.

HGKD+ integrates knowledge from multiple teacher models during the optimization phase, making it agnostic to any specific teacher architecture and compatible with the enhancements introduced in HGKD-N.

Experiments

In this section, we evaluate HGKD and its variants by comparing them against baselines to validate their effectiveness.

Experimental Setup

Datasets. We evaluate our approach on two widely used real-world datasets: IMDB and DBLP (Fu et al. 2020). IMDB models a movie recommendation scenario, where the heterogeneous graph contains three node types: movies, directors, actors; and their relations. DBLP represents a bibliographic network composed of authors, papers, conferences, and terms, capturing relationships in academic publishing.

Baselines. To evaluate HGKD comprehensively, we implement it with different teacher models and compare its performance against the teachers themselves, a simpler KD method, and an MLP trained directly on node features.

For teacher models, we select HAN, which is based on explicitly defined metapaths, and HGT, which leverages a relational attention. This setup allows us to assess HGKD across teachers based on different modeling paradigms, particularly whether its metapath-based distillation strategy remains effective when the teacher does not explicitly use metapaths.

Experimental settings. We evaluate the performance of HGKD using Micro-F1 and Macro-F1 as metrics. The datasets are split into training and testing with a 20%-80% ratio, and half of the training set is used for validation. For baseline architectures, we follow the parameter settings in the original literature and ensure performance alignment. The student MLP in HGKD is implemented with three layers and uses the same embedding dimension as the HGNN teachers. For HGKD-N, to offset the additional overhead introduced by feature augmentation, the model is adjusted to two layers. The baseline MLP and the MLP used in GLNN adopt the same architecture as the student in HGKD. More experimental details are provided in the Appendix.

Performance Comparison

In this section, we conduct the node classification task on two real-world heterogeneous graph datasets and perform comparative analyses to evaluate the proposed method.

Comparison Between Student and Teacher. We begin by evaluating HGKD using different HGNNs as teacher models. This analysis includes three components. To assess whether knowledge is successfully transferred from teachers to students, we compare each student model with its corresponding teacher. To quantify the contribution of HGKD, we compare the student against a baseline MLP of identical architecture trained solely on raw node features. Finally, to examine the necessity of HGKD's multi-level distillation design, we compare HGKD with a representative GNN-to-MLP KD method, GLNN, which performs only single-level distillation, on the node classification task on DBLP.

As shown in Table 2, our results demonstrate: (1) HGKD effectively distills knowledge from HGNNs, enabling the MLP student to achieve substantial performance improvements (ranging from 8.42% to 21.38%) over the baseline MLP trained solely on raw node features across all datasets. (2) HGKD successfully aligns the student with its teacher,

Datasets	Metrics	Train	HAN	HGKD$_{HAN}$	$\Delta HGNN$	HGT	HGKD$_{HGT}$	$\Delta HGNN$	MLP	ΔMLP
IMDB	Ma-F1	20%	56.38	**57.12**	0.74 (1.31%)	54.88	**55.65**	0.77 (1.40%)	49.63	7.49 (15.09%)
		40%	59.78	**60.52**	0.74 (1.24%)	57.54	**58.46**	0.92 (1.60%)	55.82	4.70 (8.42%)
		60%	60.80	**62.45**	1.65 (2.71%)	58.94	**59.97**	1.03 (1.75%)	56.60	5.85 (10.34%)
		80%	63.66	**65.06**	1.40 (2.20%)	60.66	**62.24**	1.58 (2.60%)	59.46	5.60 (9.42%)
	Mi-F1	20%	56.84	**57.50**	0.66 (1.16%)	55.38	**56.21**	0.83 (1.50%)	49.77	7.73 (15.53%)
		40%	59.93	**60.63**	0.70 (1.17%)	57.63	**58.72**	1.09 (1.89%)	55.53	5.10 (9.18%)
		60%	60.92	**62.62**	1.70 (2.79%)	58.94	**59.99**	1.05 (1.78%)	56.37	6.25 (11.09%)
		80%	63.67	**65.07**	1.40 (2.20%)	60.57	**62.38**	1.75 (2.89%)	59.00	6.07 (10.29%)
DBLP	Ma-F1	20%	**91.89**	91.21	-0.68 (-0.74%)	77.79	**78.16**	0.37 (0.48%)	75.14	16.07 (21.38%)
		40%	**92.08**	91.05	-1.03 (-1.12%)	81.69	**82.20**	0.51 (0.62%)	77.31	13.74 (17.78%)
		60%	**92.60**	91.83	-0.77 (-0.83%)	80.99	**81.86**	0.87 (1.07%)	78.65	13.18 (16.76%)
		80%	**94.31**	94.03	-0.28 (-0.30%)	83.39	**83.51**	0.12 (0.14%)	81.73	12.30 (15.06%)
	Mi-F1	20%	**92.57**	91.86	-0.71 (-0.77%)	78.75	**79.06**	0.31 (0.39%)	76.08	15.78 (20.74%)
		40%	**92.61**	91.50	-1.11 (-1.20%)	82.67	**83.04**	0.37 (0.45%)	77.70	13.80 (17.76%)
		60%	**92.98**	92.17	-0.81 (-0.87%)	81.33	**82.07**	0.74 (0.91%)	79.17	13.00 (16.43%)
		80%	**94.71**	94.31	-0.40 (-0.42%)	84.01	**84.01**	0.00 (0.00%)	82.41	11.90 (14.45%)

Table 2: Performance comparison between HGKD student models, HGNN teachers and baseline MLPs. HGKD$_{teacher_1}$ denotes a student trained by HGKD using $teacher_1$ as the teacher model. $\Delta HGNN$ reports the absolute and percentage performance differences between students model and teachers. ΔMLP reports the absolute and percentage performance differences between the best-performing student model and the baseline MLPs. Bold font indicates the best-performing results.

Metrics	Train	HGKD	GL$_{GCN}$	GL$_{HAN}$	GL$_{HGT}$	ΔGL
Ma-F1	20%	**92.13**	79.54	87.32	77.55	4.81 (5.51%)
	40%	**92.15**	81.84	88.50	80.58	3.65 (4.12%)
	60%	**92.01**	82.05	88.95	79.74	3.06 (3.44%)
	80%	**93.99**	83.01	89.08	83.25	4.91 (5.51%)
Mi-F1	20%	**92.78**	81.12	87.96	78.48	4.82 (5.48%)
	40%	**92.65**	82.71	89.03	80.99	3.62 (4.07%)
	60%	**92.36**	82.75	89.34	80.10	3.02 (3.38%)
	80%	**94.34**	83.89	89.54	83.76	4.80 (5.36%)

Table 3: Performance comparison between HGKD and GLNN trained with different teacher models. HGKD denotes the best-performing student model trained by our methods. GL$_{teacher_1}$ denotes a student trained by GLNN with the corresponding teacher. ΔGL reports the absolute and percentage improvements over the best-performing GLNN student. Best results are highlighted in bold.

while allowing the student to overcome the teacher's limitations when the teacher underperforms. When teachers achieve very high accuracy (e.g., metrics above 90), the HGKD student exhibits only a slight performance drop, with at most 1.2% degradation. In all other cases, the student matches the teacher's performance while avoiding some of its errors, achieving improvements of up to 2.89%.

Since GLNN was originally developed for homogeneous GNN teachers, and HGNNs generally exhibit superior performance on heterogeneous graph learning tasks, we evaluate GLNN in two configurations: one using its original GNN teacher (Kipf and Welling 2017) directly on heterogeneous graphs, and another adapting GLNN to use HGNNs as teachers. This provides a comprehensive assessment of GLNN in heterogeneous graph settings. As shown in Table 3, with the same student architecture, HGKD consistently and significantly outperforms the best-performing GLNN student, with performance improvements ranging from 3.38% to 5.51%. These results demonstrates the necessity of HGKD's multi-level distillation strategy, which enhances the student's ability to learn from the teacher.

Performance Analysis on Neighborhood Information. In this section, we evaluate the effectiveness of incorporating neighborhood information in the HGKD-N variant. To this end, we compare HGKD-N with the original HGKD under various teacher configurations, in order to assess the contribution of neighborhood information to student performance. Specifically, we aim to determine whether explicitly encoding limited local structural context improves the student's ability to absorb and reproduce the knowledge transferred from the teacher. Furthermore, we directly compare HGKD-N students with their respective teachers. This allows us to assess whether such auxiliary information helps students more closely match or surpass the teacher.

As shown in Table 4, HGKD-N outperforms HGKD in all but one case, where a marginal 0.04% drop in macro metrics occurs. On average, HGKD-N yields consistent improvements of 1.83% in macro-F1 and 1.69% in micro-F1, with the most substantial gain reaching 4.33%. These results indicate that neighborhood-aware input features provide additional context that benefits the student's predictive capabilities. In direct comparisons with HGNN teacher models, HGKD-N outperforms the teacher in a greater proportion of cases than the original HGKD, achieving superior results in 87.5% of evaluated instances, while incurring at most a 0.67% performance drop. Specifically, when adopting HAN as the teacher, students achieve an average improvement of 2.06% in macro-F1 and 2.02% in micro-F1 over the teacher. When using HGT as the teacher, the gains

Datasets	Metrics	Train	HAN	HGKD$_{HAN}$	-N$_{HAN}$	$\Delta HGKD$	$\Delta HGNN$	HGT	HGKD$_{HGT}$	-N$_{HGT}$	$\Delta HGKD$	$\Delta HGNN$
IMDB	Ma-F1	20%	56.38	57.12	58.52	1.40 (2.45%)	2.14 (3.80%)	54.88	55.65	56.37	0.72 (1.29%)	1.49 (2.72%)
		40%	59.78	60.52	63.11	2.59 (4.28%)	3.33 (5.57%)	57.54	58.46	60.99	2.53 (4.33%)	3.45 (6.00%)
		60%	60.8	62.45	62.81	0.36 (0.58%)	2.01 (3.31%)	58.94	59.97	61.82	1.85 (3.08%)	2.88 (4.89%)
		80%	63.66	65.06	66.49	1.43 (2.20%)	2.83 (4.45%)	60.66	62.24	62.93	0.69 (1.11%)	2.27 (3.74%)
	Mi-F1	20%	56.84	57.5	59.00	1.50 (2.61%)	2.16 (3.80%)	55.38	56.21	56.81	0.60 (1.07%)	1.43 (2.58%)
		40%	59.93	60.63	63.05	2.24 (3.69%)	3.12 (5.21%)	57.63	58.72	61.02	2.30 (3.92%)	3.39 (5.88%)
		60%	60.92	62.62	62.85	0.23 (0.37%)	1.87 (3.07%)	58.94	59.99	61.86	1.87 (3.12%)	2.92 (4.96%)
		80%	63.67	65.07	66.47	1.40 (2.15%)	2.80 (4.40%)	60.57	62.38	62.97	0.59 (0.94%)	2.40 (3.96%)
DBLP	Ma-F1	20%	91.89	91.21	92.13	0.92 (1.01%)	0.24 (0.26%)	77.79	78.16	80.52	2.36 (3.02%)	2.73 (3.51%)
		40%	92.08	91.05	92.15	1.10 (1.21%)	0.07 (0.08%)	81.69	82.2	82.68	0.48 (0.58%)	0.99 (1.21%)
		60%	92.6	91.83	92.01	0.18 (0.20%)	-0.59 (-0.64%)	80.99	80.99	82.74	1.75 (2.16%)	1.75 (2.16%)
		80%	94.31	94.03	93.99	-0.04 (-0.04%)	-0.32 (-0.34%)	83.39	83.51	85.00	1.49 (1.76%)	1.61 (1.93%)
	Mi-F1	20%	92.57	91.86	92.78	0.92 (1.00%)	0.21 (0.23%)	78.75	79.06	80.46	1.40 (1.77%)	1.71 (2.17%)
		40%	92.61	91.5	92.65	1.15 (1.26%)	0.44 (0.48%)	82.67	82.67	83.24	0.57 (0.69%)	0.94 (1.14%)
		60%	92.98	92.17	92.36	0.19 (0.21%)	-0.62 (-0.67%)	81.33	81.33	83.18	1.85 (2.28%)	2.59 (3.19%)
		80%	94.71	94.31	94.34	0.03 (0.03%)	-0.36 (-0.38%)	84.01	84.01	85.61	1.60 (1.89%)	1.60 (1.90%)

Table 4: Performance comparison between HGKD-N, HGKD and HGNN teachers. -N$_{teacher_1}$ denotes a student trained by HGKD-N using $teacher_1$ as the teacher. $\Delta HGKD$ reports the absolute and percentage performance differences between students trained by HGKD and HGKD-N. $\Delta HGNN$ reports the performance differences between students and their teachers.

Metrics	Train	HGKD$_{HAN}$	HGKD$_{HGT}$	HGKD+	$\Delta HGKD$	HGKD-N+	$\Delta HGKD$
Ma-F1	20%	57.12	55.65	57.76	0.64 (1.12%)	57.60	0.48 (0.84%)
	40%	60.52	58.46	60.50	-0.02 (-0.03%)	63.63	3.11 (5.14%)
	60%	62.45	59.97	62.57	0.12 (0.19%)	64.50	2.05 (3.28%)
	80%	65.06	62.24	65.43	0.37 (0.57%)	67.72	2.66 (4.09%)
Mi-F1	20%	57.50	56.21	57.99	0.49 (0.85%)	58.54	1.04 (1.81%)
	40%	60.63	58.72	60.79	0.16 (0.26%)	63.75	3.12 (5.15%)
	60%	62.62	59.99	62.62	0.00 (0.00%)	64.60	1.98 (3.16%)
	80%	65.07	62.38	65.42	0.35 (0.54%)	67.64	2.57 (3.95%)

Table 5: Performance comparison between HGKD+, HGKD-N+ and single teacher HGKD. HGKD+ denotes students trained by the multi-teacher version of HGKD. HGKD-N+ denotes students trained by HGKD+ with limited neighbor information. $\Delta HGKD$ values show absolute and percentage differences between them and the original HGKD.

are higher, with macro-F1 increasing by 3.27% and micro-F1 by 3.22%. These results further demonstrate that limited neighborhood information can reduce the performance gap between student and teacher, enabling the student to more precisely replicate and enhance the teacher's performance.

Performance Analysis on Multi-teacher Distillation. In this section, we evaluate the multi-teacher learning strategy proposed in HGKD+. We examine whether HGKD+ enables the student model to automatically align with the better-performing teacher among multiple candidates, while mitigating the limitations of individual teachers. Since previous results have demonstrated that the original HGKD allows the student to match the performance of its teacher, we compare HGKD+ against the single-teacher HGKD to assess whether the multi-teacher mechanism further corrects teacher-specific errors and yields additional gains. We also evaluate HGKD-N+, the combination of HGKD+ and HGKD-N, to demonstrate that HGKD+ can be integrated with HGKD-N to form an effective joint variant. Experiments are conducted on the IMDB node-classification task.

In Table 5, incorporating the Pareto-based multi-teacher learning mechanism allows HGKD+ to achieve performance improvements in 6 out of 8 cases, with only a marginal performance degradation of 0.03% observed in one case. The joint variant **HGKD-N+** further enhances the performance of HGKD, achieving an average improvement of 3.34% in macro-F1 and 3.52% in micro-F1 over the original HGKD.

Conclusion

In this paper, we propose a knowledge distillation framework for HGNNs, along with variants that support multi-teacher learning and the incorporation of limited neighbor information. Our approach transfers knowledge from HGNN teacher models at three levels, addressing the limitations of conventional KD methods that focus solely on mimicking teacher predictions. The proposed method enables the student model to achieve performance that matches or surpasses that of its teacher models in heterogeneous graph learning tasks, while eliminating the dependency on graph structure. Extensive experiments and comparisons with baselines across diverse datasets demonstrate the effectiveness and superiority of our framework.

References

Ba, J.; and Caruana, R. 2014. Do deep nets really need to be deep? *Advances in neural information processing systems*, 27.

Bahlous-Boldi, R.; Ding, L.; Spector, L.; and Niekum, S. 2025. Pareto-Optimal Learning from Preferences with Hidden Context. *arXiv*.

Bai, Y.; Ying, R.; Ren, H.; and Leskovec, J. 2021. Modeling heterogeneous hierarchies with relation-specific hyperbolic cones. In *NIPS*.

Bi, J.; Wang, Y.; Yan, D.; Xiao, X.; Hecker, A.; Tresp, V.; and Ma, Y. 2025. Prism: Self-pruning intrinsic selection method for training-free multimodal data selection. *arXiv preprint arXiv:2502.12119*.

Cao, Y.; Peng, H.; Wu, J.; Dou, Y.; Li, J.; and Yu, P. S. 2021. Knowledge-Preserving Incremental Social Event Detection via Heterogeneous GNNs. In *WWW*.

Chen, J.; Chen, S.; Bai, M.; Gao, J.; Zhang, J.; and Pu, J. 2022. SA-MLP: Distilling Graph Knowledge from GNNs into Structure-Aware MLP. *arXiv*.

Chen, M.; Huang, C.; Xia, L.; Wei, W.; Xu, Y.; and Luo, R. 2023. Heterogeneous Graph Contrastive Learning for Recommendation. In *WSDM*.

Deng, X.; and Zhang, Z. 2021. Graph-Free Knowledge Distillation for Graph Neural Networks. In *IJCAI*.

Dong, Y.; Chawla, N. V.; and Swami, A. 2017. metapath2vec: Scalable Representation Learning for Heterogeneous Networks. In *KDD*.

Feng, Y.; Luo, Y.; Ying, S.; and Gao, Y. 2024. LightHGNN: Distilling Hypergraph Neural Networks into MLPs for 100x Faster Inference. In *ICLR*.

Feng, Y.; You, H.; Zhang, Z.; Ji, R.; and Gao, Y. 2019. Hypergraph neural networks. In *AAAI*.

Fu, T.-y.; Lee, W.-C.; and Lei, Z. 2017. HIN2Vec: Explore Meta-paths in Heterogeneous Information Networks for Representation Learning. In *CIKM*.

Fu, X.; Zhang, J.; Meng, Z.; and King, I. 2020. MAGNN: Metapath Aggregated Graph Neural Network for Heterogeneous Graph Embedding. In *WWW*.

Hamilton, W. L.; Ying, R.; and Leskovec, J. 2017. Inductive representation learning on large graphs. In *NIPS*.

Heo, B.; Lee, M.; Yun, S.; and Choi, J. Y. 2019. Knowledge transfer via distillation of activation boundaries formed by hidden neurons. In *AAAI*.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the Knowledge in a Neural Network. *arXiv*.

Hoang, L. P.; Le, D. D.; Tuan, T. A.; and Thang, T. N. 2023. Improving pareto front learning via multi-sample hypernetworks. In *AAAI*.

Hu, W.; Fey, M.; Zitnik, M.; Dong, Y.; Ren, H.; Liu, B.; Catasta, M.; and Leskovec, J. 2020a. Open graph benchmark: datasets for machine learning on graphs. In *NIPS*.

Hu, Y.; You, H.; Wang, Z.; Wang, Z.; Zhou, E.; and Gao, Y. 2021. Graph-MLP: Node Classification without Message Passing in Graph. *arXiv*.

Hu, Z.; Dong, Y.; Wang, K.; and Sun, Y. 2020b. Heterogeneous Graph Transformer. In *WWW*.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

Lin, X.; Zhen, H.-L.; Li, Z.; Zhang, Q.; and Kwong, S. 2019. Pareto multi-task learning. In *NIPS*.

Liu, B.; Tan, X.; Zeng, X.; and Guo, D. 2024. GNN-to-MLP Distillation based on Structural Knowledge for Link Prediction. In *ICAICE*.

Liu, J.; Zheng, T.; and Hao, Q. 2022. HIRE: Distilling High-order Relational Knowledge From Heterogeneous Graph Neural Networks. *Neurocomputing*.

Liu, Y.; Zhang, W.; and Wang, J. 2020. Adaptive multi-teacher multi-level knowledge distillation. *Neurocomputing*.

Lu, W.; Guan, Z.; Zhao, W.; and Yang, Y. 2024. AdaGMLP: AdaBoosting GNN-to-MLP Knowledge Distillation. In *KDD*.

Mei, G.; Pan, L.; and Liu, S. 2022. Heterogeneous graph embedding by aggregating meta-path and meta-structure through attention mechanism. *Neurocomput*.

Schlichtkrull, M.; Kipf, T. N.; Bloem, P.; van den Berg, R.; Titov, I.; and Welling, M. 2017. Modeling Relational Data with Graph Convolutional Networks. *arXiv*.

Sener, O.; and Koltun, V. 2018. Multi-task learning as multi-objective optimization. In *NIPS*.

Shi, C.; Hu, B.; Zhao, W. X.; and Yu, P. S. 2019. Heterogeneous Information Network Embedding for Recommendation. *IEEE Trans. on Knowl. and Data Eng.*

Sun, Y.; and Han, J. 2013. Mining heterogeneous information networks: a structural analysis approach. *SIGKDD Explor. Newsl.*

Sun, Y.; Han, J.; Yan, X.; Yu, P. S.; and Wu, T. 2011. PathSim: meta path-based top-K similarity search in heterogeneous information networks. *Proc. VLDB Endow.*

Sun, Y.; Yu, Y.; and Han, J. 2009. Ranking-based clustering of heterogeneous information networks with star network schema. In *KDD*.

Tian, Y.; Dong, K.; Zhang, C.; Zhang, C.; and Chawla, N. V. 2023a. Heterogeneous Graph Masked Autoencoders. In *AAAI*.

Tian, Y.; Han, Y.; Chen, X.; Wang, W.; and Chawla, N. V. 2024. Beyond Answers: Transferring Reasoning Capabilities to Smaller LLMs Using Multi-Teacher Knowledge Distillation. In *WSDM*.

Tian, Y.; Pei, S.; Zhang, X.; Zhang, C.; and Chawla, N. V. 2023b. Knowledge Distillation on Graphs: A Survey. *ACM Comput. Surv.*

Tian, Y.; Zhang, C.; Guo, Z.; Huang, C.; Metoyer, R.; and Chawla, N. V. 2022a. RecipeRec: A Heterogeneous Graph Learning Model for Recipe Recommendation. In *IJCAI*.

Tian, Y.; Zhang, C.; Guo, Z.; Zhang, X.; and Chawla, N. 2022b. Learning mlps on graphs: A unified view of effectiveness, robustness, and efficiency. In *ICLR*.

Wang, X.; Bo, D.; Shi, C.; Fan, S.; Ye, Y.; and Yu, P. S. 2023. A Survey on Heterogeneous Graph Embedding: Methods, Techniques, Applications and Sources. *IEEE Transactions on Big Data*.

Wang, X.; Ji, H.; Shi, C.; Wang, B.; Ye, Y.; Cui, P.; and Yu, P. S. 2019. Heterogeneous Graph Attention Network. In *WWW*.

Wang, X.; Liu, N.; Han, H.; and Shi, C. 2021a. Self-supervised Heterogeneous Graph Neural Network with Co-contrastive Learning. In *KDD*.

Wang, Y.; Jin, J.; Zhang, W.; Yu, Y.; Zhang, Z.; and Wipf, D. 2021b. Bag of Tricks for Node Classification with Graph Neural Networks. *arXiv*.

Wu, L.; Huang, Y.; Lin, H.; Liu, Z.; Fan, T.; and Li, S. Z. 2022. Automated Graph Self-supervised Learning via Multi-teacher Knowledge Distillation. *arXiv*.

Wu, L.; Lin, H.; Huang, Y.; and Li, S. Z. 2023a. Quantifying the knowledge in GNNs for reliable distillation into MLPs. In *ICML*.

Wu, L.; Lin, H.; Huang, Y.; and Li, S. Z. 2023b. Quantifying the Knowledge in GNNs for Reliable Distillation into MLPs. In *ICML*.

Xu, Y.; Zhang, Y.; Guo, W.; Guo, H.; Tang, R.; and Coates, M. 2020. GraphSAIL: Graph Structure Aware Incremental Learning for Recommender Systems. In *CIKM*.

Yan, B.; Wang, C.; Guo, G.; and Lou, Y. 2020. TinyGNN: Learning Efficient Graph Neural Networks. In *KDD*.

Yang, C.; Liu, J.; and Shi, C. 2021. Extract the Knowledge of Graph Neural Networks and Go Beyond it: An Effective Knowledge Distillation Framework. In *WWW*.

Yang, L.; Tian, Y.; Xu, M.; Liu, Z.; Hong, S.; Qu, W.; Zhang, W.; CUI, B.; Zhang, M.; and Leskovec, J. 2024. VQGraph: Rethinking Graph Representation Space for Bridging GNNs and MLPs. In *ICLR*.

Yang, Y.; Qiu, J.; Song, M.; Tao, D.; and Wang, X. 2020. Distilling Knowledge From Graph Convolutional Networks. In *CVPR*.

Yu, T.; Kumar, S.; Gupta, A.; Levine, S.; Hausman, K.; and Finn, C. 2020. Gradient Surgery for Multi-Task Learning. In *NIPS*.

Zhang, C.; Liu, J.; Dang, K.; and Zhang, W. 2022a. Multi-scale distillation from multiple graph neural networks. In *AAAI*.

Zhang, C.; Song, D.; Huang, C.; Swami, A.; and Chawla, N. V. 2019. Heterogeneous Graph Neural Network. In *KDD*.

Zhang, D.; Huang, X.; Liu, Z.; Zhou, J.; Hu, Z.; Song, X.; Ge, Z.; Wang, L.; Zhang, Z.; and Qi, Y. 2020. AGL: a scalable system for industrial-purpose graph machine learning. *Proc. VLDB Endow.*

Zhang, S.; Liu, Y.; Sun, Y.; and Shah, N. 2022b. Graph-less Neural Networks: Teaching Old MLPs New Tricks via Distillation. In *ICLR*.

Zhang, W.; Jiang, Y.; Li, Y.; Sheng, Z.; Shen, Y.; Miao, X.; Wang, L.; Yang, Z.; and Cui, B. 2021. ROD: Reception-aware Online Distillation for Sparse Graphs. In *KDD*.

Zheng, W.; Huang, E. W.; Rao, N.; Katariya, S.; Wang, Z.; and Subbian, K. 2021. Cold brew: Distilling graph node representations with incomplete or missing neighborhoods. In *ICLR*.

Zhuang, Y.; Lyu, L.; Shi, C.; Yang, C.; and Sun, L. 2022. Data-free adversarial knowledge distillation for graph neural networks. In *IJCAI*.