SALE-MLP: Structure Aware Latent Embeddings for GNN to Graph-free MLP Distillation

Harsh Pal , **Sarthak Malik** , **Rajat Patel** , **Aakarsh Malhotra**
AI Garage, Mastercard, Gurugram, Haryana, India
{harsh.pal, sarthak.malik, aakarsh.malhotra}@mastercard.com, prajat5232@gmail.com

Abstract

Graph Neural Networks (GNNs), with their ability to effectively handle non-Euclidean data structures, have demonstrated state-of-the-art performance in learning node and graph-level representations. However, GNNs face significant computational overhead due to their message-passing mechanisms, making them impractical for real-time large-scale applications. Recently, Graph-to-MLP (G2M) knowledge distillation has emerged as a promising solution, utilizing MLPs to reduce inference latency. However, existing methods often lack structural awareness (SA), limiting their ability to capture essential graph-specific information. Moreover, some methods require access to large-scale graphs, undermining their scalability. To address these issues, we propose SALE-MLP (Structure-Aware Latent Embeddings for GNN-to-Graph-Free MLP Distillation), a novel graph-free and structure-aware approach that leverages unsupervised structural losses to align the MLP feature space with the underlying graph structure. SALE-MLP does not rely on precomputed GNN embeddings nor require graph during inference, making it efficient for real-world applications. Extensive experiments demonstrate that SALE-MLP outperforms existing G2M methods across tasks and datasets, achieving 3–4% improvement in node classification for inductive settings while maintaining strong transductive performance.

1 Introduction

Graphs have emerged as essential data structures for modeling non-Euclidean relations and their interactions. From applications such as protein interactions [Jha *et al.*, 2022], entity alignment [Chaurasiya *et al.*, 2022; Surisetty *et al.*, 2022], and social media [Zhang *et al.*, 2022b], these interconnected networks effectively represent the intricacies of such systems. To learn rich node and graph-level representations from these graph structures, GNNs employ message-passing mechanisms by aggregating information from neighboring nodes [Kipf and Welling, 2022]. As a result, GNNs achieve state-of-the-art performance in tasks such as node classification, link prediction, and graph classification, demonstrating their potential in handling non-Euclidean data [Hamilton *et al.*, 2017; Veličković *et al.*, 2018; Wu *et al.*, 2020].

Despite their success, GNNs face critical challenges such as high latency and computational overhead. Due to the neighborhood aggregation process, the applicability of GNNs in real-world scenarios remains limited [Hamilton *et al.*, 2017; Kipf and Welling, 2022]. To address the latency bottleneck [Liu *et al.*, 2022], techniques such as pruning [Huang *et al.*, 2024], partitioning [Modak *et al.*, 2024], and quantization [Ding *et al.*, 2021; Tailor *et al.*, 2021; Zhao *et al.*, 2020] aim to reduce computational complexity. However, these methods fail to eliminate the message-passing overhead in GNNs. Alternatively, knowledge distillation (KD) [Zhang *et al.*, 2022a; Tian *et al.*, 2022; Lu *et al.*, 2024; Yan *et al.*, 2020; Malik *et al.*, 2024] transfers the knowledge of computationally intensive teacher models to lightweight student models. KD methods replicate the teacher's output through soft labels or latent embedding spaces. MLPs are often selected as student models in such distillation approaches due to their strong memory capabilities [Szegedy, 2013]. By leveraging only node features, MLPs can effectively capture the knowledge transferred from GNNs while maintaining lower inference latency.

Though lightweight, student MLPs struggle to fully capture the crucial graph-specific structural knowledge from GNNs. Traditional MLP-based methods, such as GLNN [Zhang *et al.*, 2022a], rely solely on node features. They lack explicit structural information and often lead to suboptimal performance. Additionally, DeepWalk-based methods like NOSMOG[Tian *et al.*, 2022] are unsuitable for online training or inductive scenarios, where models must generalize to unseen nodes without accessing the entire graph. Maintaining the whole graph in such cases is resource-intensive and undermines the purpose of large-scale graph distillation. While structure-aware methods [Chen *et al.*, 2022; Wang *et al.*, 2024] improve performance, they struggle with heterophilic graphs. Furthermore, methods [Wu *et al.*, 2023c] attempt to make MLPs more structure-aware through knowledge transfer from GNN embeddings. However, it remains unclear whether distilling the GNN latent embedding space is the best approach for student MLPs, particularly for node classification tasks and their effectiveness is not well established. This raises the question: **can we make MLPs structure-aware while maintaining graph-free inference?**

In this work, we propose SALE-MLP, a novel, flexible, and adaptable structure-aware (G2M) KD method. SALE-MLP learns structure-aware latent embeddings from node features by distilling GNNs into MLPs, without relying on GNN embeddings or explicit graphs as input. SALE-MLP enhances the student model's generalization by mimicking GNN-like message passing and improving the embedding space to better represent graph structures. Our approach can integrate with any unsupervised graph-structure loss and adopt diverse GNN teachers, making the framework fundamentally loss, teacher, and task-agnostic (suitable for both node classification and link prediction). It excels in unseen scenarios where structural information is vital. Extensive experiments on node classification and link prediction tasks show that SALE-MLP outperforms existing G2M distillation methods in both transductive and inductive settings. The performance gain is particularly notable (3–4%) in practically relevant inductive settings. Furthermore, ablation studies validate the effectiveness and efficiency of SALE-MLP.

2 Preliminaries

Notation: Consider a graph $G = (A, X)$ with the node set $\mathcal{V} = \{v_1, v_2, \ldots, v_{\mathcal{N}}\}$ and edge set E, where $\mathcal{N}$ is total number of nodes. Let the node features be $X \in \mathbb{R}^{\mathcal{N} \times d}$ with d as the size of node features. A is the adjacency matrix, where $A_{ij} = 1$ if node i and j are connected, else, $A_{ij} = 0$. Furthermore, $Y \in \mathbb{R}^{\mathcal{N} \times C}$ represents the label matrix where C is the number of classes. Finally, labeled nodes are denoted as $(\mathcal{V}^L, X^L, Y^L)$, and unlabeled nodes as $(\mathcal{V}^U, X^U, Y^U)$.

Experimentation Setting: We conduct experiments in two setups: (i) **Transductive setting**: The entire graph G is accessible while training using Y^L and the model is evaluated on the unlabeled nodes X^U to predict Y^U (ii) **Inductive setting:** The unlabeled nodes V^U are partitioned into two disjoint sets of nodes $V^U = V^U_{obs} \cup V^U_{in}$, representing observed nodes and inductive nodes, respectively. This creates two disjoint graphs: G_{obs} based on nodes $V^L \cup V^U_{obs}$ and G_{ind} based on nodes V^U_{in}. While training $Y = Y^L$ and only $G = G_{obs}$ is accessible to the model while evaluation is performed using the entire graph G. The inductive setting closely resembles real-world scenarios as the entirety of large graphs is either unavailable or too compute-intensive to train on.

GNN Distillation: The message-passing mechanism is essential for learning node representations in GNNs by aggregating information from its neighboring nodes. It typically involves two key steps: *message aggregation* and *representation update* due to which the computational complexity of GNNs increases drastically. So, GNN distillation methods aim to train a student model that replicates the performance of a GNN while reducing computation. The student model is trained using a combination of adjacency matrix A, node features X, GNN output logits $\hat{y}$, and GNN embeddings. The student is considered message-passing-free if the neighborhood aggregation step is unnecessary, or graph-free if no structural information is extracted from the graph during inference. The distillation loss, such as KL divergence [Hinton, 2015], minimizes the difference between the student's output and the GNN's output.

3 Related Work

GNNs require substantial time and memory for training and inference over relational data. Graph knowledge distillation addresses this challenge and can be categorized into two types: graph-to-graph distillation and G2M distillation. While graph-to-graph distillation methods [Chen *et al.*, 2021; Joshi *et al.*, 2022; Lassance *et al.*, 2020; Ren *et al.*, 2021; Wu *et al.*, 2022a] successfully train compact student GNNs, methods like LSP [Yang *et al.*, 2020] and TinyGNN [Yan *et al.*, 2020] leverage structural insights from teachers. Additionally, RDD [Zhang *et al.*, 2020] uses the reliability of nodes and edges to improve the performance of student GNNs. Although effective in compression, these student GNNs still inherit the computational overhead of message passing, limiting their practical applicability. In contrast, G2M methods, based on structural awareness, are subdivided into structure-aware (SA-G2M) and non-structure-aware (NSA-G2M) approaches.

3.1 Non-Structure (NSA-G2M) Distillation

To tackle the problem of complex message-passing, MLP-based student models distill knowledge from teacher GNNs and replicate the teacher's performance while reducing memory and computational complexity. GLNN [Zhang *et al.*, 2022a] pioneered G2M distillation by mimicking GNNs using teacher prediction probabilities as soft labels, along with ground-truth labels as hard labels. By simply trying to replicate the GNN output logits, MLPs achieve a significant performance improvement over MLPs trained solely using ground-truth labels, demonstrating the drastic usefulness of distillation methods. Furthermore, GSDN [Wu *et al.*, 2022b] enhanced this approach by incorporating mixup augmentation [Han *et al.*, 2022] on input features and enforcing locality-based consistency. However, these methods rely on logit-based distillation, ignoring neighborhood and graph structure, which leads to information loss and poor generalization on unseen nodes. FF-G2M [Wu *et al.*, 2023a] addresses this by using spectral-graph theory to preserve high-frequency information during the distillation process, ensuring that high pairwise differences in teacher logits between neighboring nodes are maintained.

Additionally, KRD [Wu *et al.*, 2023b] handles this information loss by introducing a novel strategy of training student MLPs with only confident knowledge points using a reliable sampling strategy. Recently, AdaGMLP [Lu *et al.*, 2024] proposed an ensemble G2M distillation approach that improves the generalization capabilities of the student MLP to some extent. They also use feature masking and alignment of AdaBoosted MLPs, which help mitigate incomplete/corrupted node features and graph structures that are prevalent in real-world graphs. While these methods benefit from graph-free MLP inference, not enforcing structural awareness causes them to lose crucial information, leading to suboptimal performance, especially in inductive or few-shot learning. As most real-world graph systems are inductive and face time and memory constraints, it becomes increasingly important for these G2M methods to incorporate structural information.

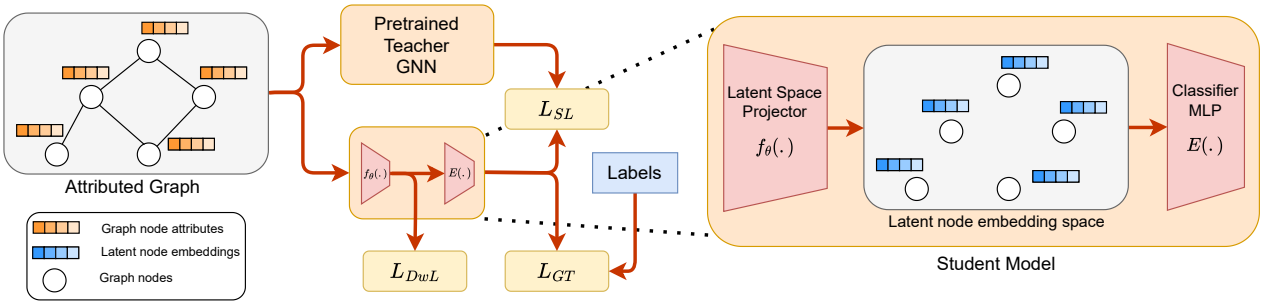


Figure 1: SALE-MLP: Representing Latent-Space Projector $f(.)$ for structural information and MLP classifier $E(.)$ for final classifcation.

3.2 Structure-Aware (SA-G2M) Distillation

Structure-aware MLPs carry more information about graphs and can better generalize on unseen nodes in inductive scenarios. Thus, researchers have proposed various approaches to use structural information into student MLPs. SA-MLP [Chen *et al.*, 2022] explicitly passes structural information along with node features. However, it is impractical for large-scale graphs as it requires access to the complete graph during inference, making it impractical for large-scale graphs. NOSMOG [Tian *et al.*, 2022] utilizes DeepWalk-generated positional features and also aligns GNN-MLP latent spaces using representational similarity loss. However, the use of neighborhood approximations limits its effectiveness with unseen nodes in inductive settings.

[Wu *et al.*, 2023c] used class prototype embeddings to distill structural information, but the extent to which the MLPs are made structure-aware is limited. Similarly, SSL-GM [Wang *et al.*, 2024] proposed a self-supervised framework that aligns embedding spaces without explicit structural information. However, the effectiveness of such GNN latent space alignment remains questionable due to limited expressiveness [Xu *et al.*, 2019] and over-smoothing issues [Li *et al.*, 2018]. Moreover, completely GNN-free methods [Winter *et al.*, 2024] show substantially reduced performance compared to G2M strategies, as GNN knowledge is not utilized. This shows the critical need for G2M distillation that effectively utilizes graph structural information with strong expressiveness and generalization, while maintaining graph-free inference. Our proposed SALE-MLP addresses these limitations by aligning node feature representations with graph topology, optimizing both structural awareness and efficiency for inductive scenarios, while also better approximating GNN output and ground truth.

4 Proposed Approach

This section presents SALE-MLP, which consists of two primary components: the Latent-Space Projector (LSP) and the MLP classifier, as illustrated in Figure 1. These two components together map the node features to a latent space, which is then used to make the final predictions of the student MLP. Additionally, we demonstrate our method using DeepWalk [Perozzi *et al.*, 2014]. For more details on the implementation and additional results (including other unsupervised structural losses), read the supporting material[1].

4.1 Structure-Aware Student MLP

Our approach enhances GNN distillation by aligning the student MLP's representation space with the graph structure. Specifically, nodes with similar structures or that are closely connected in the graph should have similar vector representations. Node representations generated by unsupervised structural losses [Perozzi *et al.*, 2014; Grover and Leskovec, 2016; Postăvaru *et al.*, 2020; Tang *et al.*, 2015] are well-established in generating representations that align with the graph topology and are supported by efficient implementations for large graphs. One such loss, DeepWalk, learns attribute-free graph node embeddings by generating random walks to explore the neighborhood structure and uses the SkipGram model [Mikolov *et al.*, 2013] to align the representations of nearby nodes. Since the student MLP model takes the node features as input, the random walk-based SkipGram model is instead applied to the embedding space of the MLP in our method.

Given a random walk node sequence for a node i, $S_i = \{v_1, v_2, \ldots, v_{|S_i|}\}$ of length $|S_i|$, the embedding $f(v_i)$ for each node v_i (where $1 \leq i \leq \mathcal{N}$) is generated. Representation $f(v_i)$ predicts the context of v_i within a window of size t by modeling probability $\Pr(v_j \mid v_i)$ as mentioned in Eq. 1.

$$\Pr(v_j \mid v_i) = \frac{\exp(f(v_i)^\top \cdot f(v_j))}{\sum_{k=1}^{\mathcal{N}} \exp(f(v_i)^\top \cdot f(v_k))} \quad (1)$$

where $v_j \in S_i$ and $f(.)$ is node-embedding that maps node attributes of v_i to latent-embedding space. To train $f(.)$, the loss function $\mathcal{L}_{DwL}$, an InfoNCE approximation [Church, 2017] is typically used for efficient implementation. The equivalent loss is defined in Eq. 2. The constraints in Eq. 3 encourage neighboring nodes to move closer to v_i by maximizing the probability of the random walk S_i. The constraints in Eq. 4 represent a contrastive loss, pushing the embeddings of the negative nodes in $\bar{S}_i$ farther away. The set $\bar{S}_i$ contains k randomly sampled negative nodes for each node j in S_i.

$$\mathcal{L}_{DwL} = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}_L} \left[\mathcal{L}_{DwL(P)}(V_i) + \mathcal{L}_{DwL(N)}(V_i)\right] \quad (2)$$

$$\mathcal{L}_{DwL(P)}(V_i) = -\frac{1}{|S_i|} \sum_{j \in S_i} \log \sigma(f(\mathrm{v}_i)^\top f(v_j)) \quad (3)$$

[1]https://github.com/ganzagun/SALE-MLP

$$\mathcal{L}_{DwL(N)}(V_i) = \frac{1}{|\bar{S}_i|} \sum_{j=1}^{\bar{S}_i} \log \sigma(f(\mathrm{v}_i)^\top f(v_j)) \quad (4)$$

4.2 Training MLPs by GNNs Distillation

Given a cumbersome pre-trained GNN, the goal of distillation is to train a lightweight MLP using both ground truth labels and soft labels (from the teacher). For any labeled node $v \in \mathcal{V}^L$, the ground truth label is y_v while z_v is the corresponding soft label, as predicted by the teacher GNN. The node classification objective is captured by cross-entropy CE(.) loss between the student prediction $\hat{y}_v$ and the ground truth label y_v:

$$\mathcal{L}_{CE} = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} \mathrm{CE}(\sigma(\hat{y}_i), y_i) \quad (5)$$

GNN distillation is performed using the KL-Divergence $\mathcal{D}_{KL}(.)$ between the MLP prediction $\hat{y}_v$ and the soft labels z_v, as generated by GNN teacher:

$$\mathcal{L}_{SL} = \frac{\tau^2}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} \mathcal{D}_{KL}(\sigma(\hat{y}_i/\tau), \sigma(z_i/\tau)) \quad (6)$$

4.3 Overall Loss

The final objective function $\mathcal{L}$ is defined as the weighted combination of ground truth cross-entropy loss $\mathcal{L}_{GT}$, soft label distillation loss $\mathcal{L}_{SL}$, and the unsupervised structural loss (DeepWalk loss $\mathcal{L}_{DwL}$ in this case):

$$\mathcal{L} = \lambda \mathcal{L}_{GT} + (1 - \lambda)\mathcal{L}_{SL} + \alpha \mathcal{L}_{DwL} \quad (7)$$

where α is a trade-off that balances the impact of $\mathcal{L}_{DwL}$, and λ balances $\mathcal{L}_{GT}$ and $\mathcal{L}_{SL}$. By incorporating $\mathcal{L}_{DwL}$, the distillation process is instead based on structure-aware MLPs. Additionally, the joint end-to-end training of both losses aligns the representation space, ensuring that neighbor embeddings are close even when labels differ (heterophily), thus preserving distinguishably. Furthermore, MLP overfitting, particularly with limited training labels, is mitigated by regularizing the unsupervised structural loss. Hence, it replicates teacher GNN's generalization in the student MLP with structure-aware embeddings via soft labels. Additionally, we perform pretraining of the latent space encoder, aiding faster convergence, especially for larger graphs. Efficient DeepWalk implementations can also be utilized with separate training. Algorithm 1 summarizes the proposed SALE-MLP.

Algorithm 1 Proposed SALE-MLP

Input: An attributed network $G = (V, E, X)$;
window-size = w;
embedding-size = d;
walk-per-node = γ;
walk-length = t;
Graph node features: Z^m;
Num of pretraining steps: q;
Output: Node embeddings $f(v)$ for each $v \in V$ and Classifier $E(.)$ for node classification;
$f(.), E(.) \leftarrow$ initialize parameters;
Pretraining Latent Space Encoder
for $t = 1$ to q **do**
  Sample K negative samples of $v_i \in \mathcal{V}$ for $i \in |\mathcal{V}|$
  Calculate unsupervised loss $\mathcal{L}_{DwL}$ using Eq. 2
  Perform backpropagation on $f(.)$
end for
MLP Distillation
for $t = 1$ to T **do**
  $S \leftarrow$ generate a set of random walks on G;
  Sample $|\bar{S}_i|$ negative samples of $v_i \in \mathcal{V}$ for $i \in |\mathcal{V}|$
  Calculate unsupervised loss $\mathcal{L}_{DwL}$ using Eq. 2
  Calculate final loss $\mathcal{L}$ using Eq. 7
  Perform backpropagation on $f(.)$ and $E(.)$
end for
return $f(.)$ anb $E(.)$;

5 Experimental Details

Datasets: The experimentation utilizes six widely-adopted public benchmark datasets: (Cora, Citeseer, Pubmed) [Sen *et al.*, 2008], (Amazon-Photo, Amazon-Computer) [Feng *et al.*, 2022], and the large-scale graph ogbn-arxiv [Hu *et al.*, 2020]. These datasets are selected based on their prevalent usage across state-of-the-art G2M methods (GLNN, NOSMOG, KRD, AdaGMLP). These span different scales (small: Cora/Citeseer, medium: Pubmed/Amazon, large: ogbn-arxiv) and network types (citation networks and co-purchase networks). Furthermore, for Cora, Citeseer, and Pubmed, we follow splits from [Kipf and Welling, 2022] and for the Amazon dataset, we use splits from [Zhang *et al.*, 2022a] i.e., using 20-shot for training, 30-shot for validation, and remaining for testing. While ogbn-arxiv uses standard splits [Hu *et al.*, 2020]. Detailed dataset statistics are provided in the supporting material.

Implementation details: We take GCN[Kipf and Welling, 2022], SAGE[Hamilton *et al.*, 2017] and GAT[Veličković *et al.*, 2018] as teacher GNN model. The model has dimension 128 and 2 hidden layers for all datasets except ogbn-arxiv for which 256 dimensions and 3 hidden layers are used. The choices are consistent with other G2M methods. For SALE-MLP, we perform a grid search for the hyper-parameters below on the validation data:

# Hidden layers = {2,3}	# Walks = {1,2,5}
Walk len = {3,5,10}	Pre-train Epochs = {1,2,5,10}
λ = {0.0, 0.1, ... , 1.0}	α = {1, 1.5, 2, 2.5, 3, 3.5, 4}
Hidden layer Dimensionality = {64,128,256}	

Baselines: We chose GLNN [Zhang *et al.*, 2022a], NOSMOG [Tian *et al.*, 2022], KRD [Zhang *et al.*, 2020] and AdaGMLP [Lu *et al.*, 2024] for comparison. Their publicly available codebases are replicated along with reported best hyperparameters for evaluation. Additionally, as NOSMOG is not a graph-free method, we also report results for NOSMOG by removing the Deepwalk embeddings component for a fair evaluation of methods with graph-free students. For AdaGMLP and KRD, we note that the teacher GNN used in the original work is larger than our teacher GNN.

Setting	Method	Cora	Citeseer	Pubmed	A-photo	A-computer	ogbn-arxiv
Transductive	SAGE(Teacher)	79.16 ± 1.61	67.79 ± 2.80	74.70 ± 2.33	90.42 ± 0.68	82.70 ± 1.37	70.69 ± 0.39
	MLP	59.12 ± 1.49	58.29 ± 1.94	68.42 ± 3.06	77.25 ± 1.90	67.60 ± 2.23	55.36 ± 0.34
	GLNN	78.97 ± 1.56	69.23 ± 2.39	74.70 ± 2.25	91.8 ± 0.49	82.56 ± 1.34	64.61 ± 0.15
	NOMSOG(w/o deepwalk)	79.93 ± 1.51	69.62 ± 1.45	74.95 ± 3.81	92.24 ± 1.01	82.91 ± 1.21	68.23 ± 0.32
	NOMSOG (not Graph-free)	80.93 ± 1.65	70.67 ± 2.25	75.83 ± 3.06	92.44 ± 0.51	83.72 ± 1.44	71.10 ± 0.34
	KRD	79.08 ± 1.00	71.59 ± 1.19	79.76 ± 0.62	90.99 ± 1.16	82.61 ± 0.96	71.13 ± 0.21
	AdaGMLP	83.20 ± 1.17	71.21 ± 4.17	78.92 ± 0.36	92.12 ± 2.22	81.13 ± 1.81	71.68 ± 0.51
	SALE-MLP **(Ours)**	**84.01 ± 0.46**	**74.01 ± 1.94**	**81.40 ± 0.91**	**94.01 ± 1.01**	**84.29 ± 2.85**	**72.56 ± 0.53**
Inductive	SAGE(Teacher)	81.03 ± 1.71	69.14 ± 2.99	75.07 ± 2.89	90.56 ± 1.47	82.83 ± 1.51	70.69 ± 0.58
	MLP	59.44 ± 3.36	59.31 ± 4.56	68.28 ± 3.25	77.44 ± 1.50	67.69 ± 2.21	55.29 ± 0.67
	GLNN	73.21 ± 1.53	68.48 ± 2.38	74.52 ± 2.95	89.49 ± 1.12	80.27 ± 2.11	59.04 ± 0.46
	NOMSOG(w/o deepwalk)	73.74 ± 1.96	68.78 ± 2.13	74.55 ± 3.41	89.55 ± 1.77	80.29 ± 1.41	60.88 ± 1.21
	NOMSOG (not Graph-free)	81.36 ± 1.53	70.30 ± 2.30	75.87 ± 3.32	92.61 ± 1.09	84.36 ± 1.57	70.09 ± 0.55
	KRD	75.56 ± 1.58	70.66 ± 0.34	78.42 ± 0.54	89.74 ± 1.94	81.22 ± 1.59	62.32 ± 0.57
	AdaGMLP	74.80 ± 2.21	69.30 ± 4.27	78.72 ± 0.20	92.51 ± 1.25	79.56 ± 1.21	65.07 ± 0.24
	SALE-MLP **(Ours)**	**83.73 ± 0.90**	**73.58 ± 1.48**	**81.90 ± 1.12**	**92.74 ± 1.64**	**84.48 ± 2.65**	**71.54 ± 0.65**

Table 1: Node-Classification accuracy in both Settings (over 10 runs), **bold** represents the best while underlined the second-best.

6 Results and Analysis

In this section, we evaluate SALE-MLP across several key aspects: its performance, inference time, impact of various loss component, robustness and hyperparameter sensitivity

6.1 Classification Performance

SALE-MLP is evaluated in both transductive and inductive settings, with results for SAGE shown in Table 1. In both the reported settings, SALE-MLP consistently outperforms existing methods and even surpasses the teacher GNN across all datasets, demonstrating several key advantages:

Transductive Setting: While NSA-G2M methods (KRD, AdaGMLP) benefit from accessing prediction node logits during training, SALE-MLP still achieves superior performance. Notably, SALE-MLP outperforms NOSMOG even with its DeepWalk embeddings, highlighting the effectiveness of our structure-aware latent space alignment.

Inductive Setting: SALE-MLP demonstrates substantial improvements over existing methods, particularly on the large-scale ogbn-arxiv dataset. NSA-G2M methods show significant performance degradation compared with transductive settings due to their lack of structural awareness which impacts most in inductive settings. While NOSMOG's performance heavily relies on DeepWalk embeddings as the performance without DeepWalk drops across the dataset. In contrast, SALE-MLP maintains consistent performance across both settings due to its inherent structure-aware embedding space, achieved without requiring explicit graph access. Furthermore, statistical significance of the experiments is reported in the supporting material.

6.2 Inference Time

The Figure 2 illustrates the accuracy-inference time trade-off on PubMed (transductive setting). SALE-MLP achieves the highest accuracy with only 0.2ms inference time, approaching GLNN's efficiency (0.096ms) while significantly outperforming it in accuracy. The variation in inference times across methods is attributed to the number of layers, hidden dimensions and ensemble of models (for AdaGMLP) associated with the best-performing model. While the higher inference time of NOSMOG can be attributed to the process of finding the neighboring nodes and calculating the mean embedding. Notably, SALE-MLP achieves 150× speedup over SAGE GNN (33.2ms) while delivering superior performance, demonstrating its practical utility for real-world applications.

Dateset	w/o $\mathcal{L}_{GT}$	w/o $\mathcal{L}_{SL}$	w/o $\mathcal{L}_{DwL}$	SALE-MLP
Cora	82.01	80.37	78.97	**83.74**
Citeseer	71.13	69.35	69.23	**72.38**

Table 2: Accuracy of different SALE-MLP loss components.

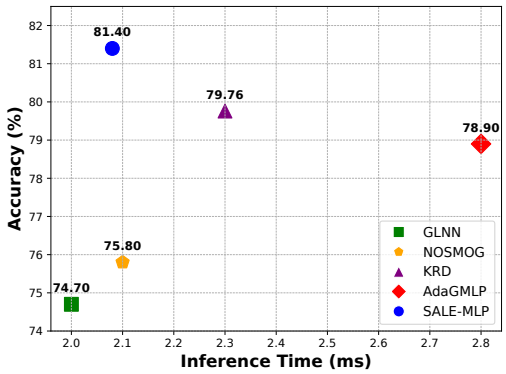


Figure 2: Trade off between time (x) and accuracy (y-axis).

6.3 Loss-Component Ablation

An ablation study by systematically removing individual loss components is also conducted, with results reported in Table 2. The analysis reveals several key insights:

Impact of Structure-Aware Learning ($\mathcal{L}_{DwL}$): Removing $\mathcal{L}_{DwL}$ causes a significant performance degradation, showing

that structure-aware latent space is crucial for both effective knowledge transfer and maintaining better classification.

Role of Soft Labels ($\mathcal{L}_{SL}$): Without GNN distillation via $\mathcal{L}_{SL}$, the model struggles to capture the teacher GNN's inductive bias and generalization capabilities, highlighting the importance of teacher guidance.

Ground Truth Supervision ($\mathcal{L}_{GT}$): The removal of $\mathcal{L}_{GT}$ has minimal impact, as $\mathcal{L}_{SL}$ provides sufficient supervision. Notably, SALE-MLP without $\mathcal{L}_{SL}$ achieves performance comparable to the teacher GNN, and with distillation providing complementary information, explains why SALE-MLP ultimately outperforms the teacher GNN (Table 1).

6.4 Consistency with Graph Topology

To quantify SALE-MLP's effectiveness in capturing graph topology, we evaluate the cut-value ($\mathcal{CV}$) between model predictions and graph structure [Zhang *et al.*, 2022a]. The min-cut problem partitions $\mathcal{N}$ nodes into K disjoint subsets while minimizing edge cuts, formulated as [Dhillon *et al.*, 2004]:

$$max \frac{1}{K} \sum_{k=1}^{K=1} \frac{(C_k^T A C_k)}{(C_k^T D C_k)} \tag{8}$$

Following [Bianchi *et al.*, 2020], replacing C with model predictions $\hat{Y}$ yields:

$$max \frac{tr(\hat{Y}^T A \hat{Y})}{(\hat{Y}^T D \hat{Y})} \tag{9}$$

Here, $\mathcal{CV}$ is proportional to the consistency between the topology of the graph and model predictions. And a Higher $\mathcal{CV}$ indicates better capture of graph-structural information.

Dataset	GNN	MLP	GLNN	NOSMOG	KRD	ADAGMLP	SALE-MLP
Cora	0.938	0.890	0.893	0.936	0.856	0.877	**0.938**
Citeseer	0.972	0.939	0.949	0.967	0.912	0.962	**0.972**
Pubmed	0.984	0.881	0.939	0.959	0.870	**0.982**	0.965
A-Computer	0.941	0.463	0.876	0.918	0.893	0.911	**0.927**
A-Photo	0.943	0.604	0.891	0.923	0.886	0.922	**0.939**

Table 3: Performance comparison of Min-cut across methods.

$\mathcal{CV}$ values across datasets are reported in Table 3. Inductive bias in GNN is best at understanding graph topology leading to the best GNN performance. While GLNN benefits from GNN distillation compared to MLP, the structure awareness introduced in NOSMOG further improves the $\mathcal{CV}$. Finally, SALE-MLP outperforms all G2M methods further proving the advantage of having structure-aware latent space derived directly from the graph topology. Additionally, Table 4 shows the impact of removing the loss components. And as observed removing $\mathcal{L}_{SL}$ minimally impacts $\mathcal{CV}$, indicating SALE-MLP effectively captures structural information through embedding alignment alone.

6.5 Performance on Heterophilic Dataset

Next, we report the performance of SALE-MLP on high-heterophily datasets Actor [Tang *et al.*, 2009] and Wisconsin (https://tinyurl.com/Wiscosn) in inductive settings. High

Dateset	w/o $\mathcal{L}_{GT}$	w/o $\mathcal{L}_{SL}$	w/o $\mathcal{L}_{DwL}$	SALE-MLP
Cora	0.890	0.936	0.893	**0.938**
Citeseer	0.939	0.967	0.949	**0.972**

Table 4: Mincut of different SALE-MLP loss components.

Dateset	SAGE	GLNN	NOSMOG	MLP	SALE-MLP
Actor	26.8%	22.7%	23.1%	**28.67%**	24.5%
Wisconsin	50.7%	45.9%	67.1%	53.2%	**70.0%**

Table 5: Node classification under heterophillic (inductive) setting.

heterophily poses unique challenges for G2M methods, requiring effective integration of both node features and graph structure.

As shown in Table 5, SALE-MLP outperforms other distillation methods in both the datasets, with significant gains in Wisconsin. MLP performs better in Actor, given sufficient training data, due to over-smoothing in GNNs. However, leveraging the neighborhood information, GNNs regain their edge in a few-shot scenario. Further, SALE-MLP outperforms MLP by providing a sweet spot with information from both structure and MLP, if trained using heterophilic specific GNNs. The larger improvement in Wisconsin can be attributed to the fact that the number of nodes in Wisconsin is smaller and it is easier for SALE-MLP to capture the relation between structure and node features. Moreover, while SALE-MLP does not explicitly address heterophily, its structure-aware latent embeddings naturally combine node features and structural information, leading to improved performance in heterophilic conditions. Additional experiments with varying heterophily ratios are presented in the supporting material.

6.6 Generalization Ability of SALE-MLP

Evaluating generalizability is another aspect explored for different approaches in Cora and Citeseer datasets. We trained and plotted the performance of SALE-MLP and other G2M methods with limited labeled samples per class (k-shots). This experimental setup is crucial for understanding how well the model can generalize when only a small amount of labeled data is available, simulating real-world scenarios where labeled data is often scarce.

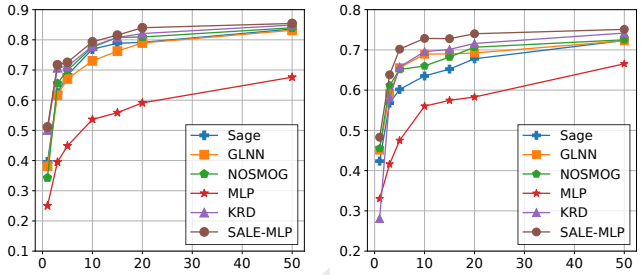


Figure 3: Accuracy (Y) vs k-shot (X) plot for L:Cora & R:Citeseer.

Figure 3 shows SALE-MLP's superior generalization capabilities across different k-shot scenarios. The model exhibits strong performance even with few labeled examples (lower

k values). As more labeled examples are introduced, SALE-MLP shows sharp initial accuracy improvements, indicating effective learning from minimal supervision. The performance gradually saturates at higher k values, suggesting efficient knowledge extraction from the available labeled data. While KRD showing competitive performance to SALE-MLP due to its reliable node selection strategy that helps capture important data patterns. However, throughout the range of k values, SALE-MLP consistently outperforms all G2M methods including KRD.

6.7 Different Training Strategies

An ablation study is conducted to understand the most optimal strategy by examining how the model performs under different strategies. We investigate three distinct training strategies for SALE-MLP: (i) end-to-end training without pre-training $f(.)$, (ii) unsupervised pre-training of $f(.)$ (warm-start) followed by freezing $f(.)$ while training the classifier $E(.)$, and (iii) pre-training $f(.)$ for a few epochs, followed by subsequent end-to-end fine-tuning of both $f(.)$ and $E(.)$.

Dateset	No pretrain	Pretrain+Freeze	Pretrain+Fine-tune	SALE-MLP
Cora	0.890	0.936	0.893	**0.938**
Citeseer	0.939	0.967	0.949	**0.972**

Table 6: Accuracy of SALE-MLP under varied train conditions.

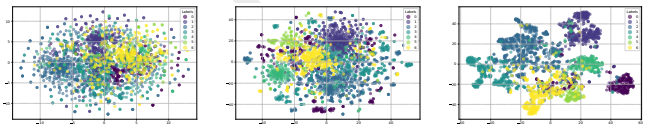
The performance of these approaches on the Cora and Citeseer datasets is reported in Table 6. It can be observed that warm starting $f(.)$ and fine-tuning it end-to-end yields the best results. This superior performance can be attributed to two reasons: this strategy better handles the relatively slower convergence of the unsupervised structural loss (Deepwalk loss). And training in a structure-aware space effectively prevents MLP from overfitting on node features.

6.8 Input Feature-Space Analysis

To understand the effectiveness of the generated latent space, we analyze different feature spaces using t-SNE plots. Figures 4(a) and 5(a) show the node-feature vs. node label plots, representing the input space used by most G2M methods (GLNN, KRD, AdaGMLP). The poorly distinguishable class clusters in this space explain the relatively lower performance of methods like GLNN that directly utilize node features. Figures 4(b) and 5(b) illustrate NOSMOG's enhanced input space, incorporating DeepWalk embeddings. The improved cluster separation explains NOSMOG's superior performance over GLNN. Finally, Figures 4(c) and 5(c) depict SALE-MLP's latent space, which demonstrates the clearest cluster separation, validating its superior performance through better structural representation learning.

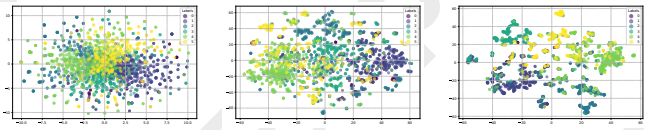
6.9 Hyperparameter Sensitivity Analysis

We examine the impact of λ and α on the performance of SALE-MLP. Figure 6 shows model sensitivity to λ ([0,1] with step of 0.1). $\lambda < 0.6$ is favorable with little impact when varied between [0,0.6]. This translates to the supervision of the teacher being pivotal, but relying more on direct supervision limits the learning of the student ($\lambda > 0.6$). Figure 6 illustrates sensitivity to α ([1,4] with step of 0.5). Performance peaks around 2.5, signifying a balance between $\mathcal{L}_{DwL}$ and other losses, as overall performance drops with lower or higher values. Here, lower values degrade latent embedding quality, while higher values reduce influence of teacher.



(a) Node features (b) NOSMOG (c) SALE-MLP

Figure 4: t-SNE plots of the input feature space for Cora.



(a) Node features (b) NOSMOG (c) SALE-MLP

Figure 5: t-SNE plots of the input feature space for Citeseer.

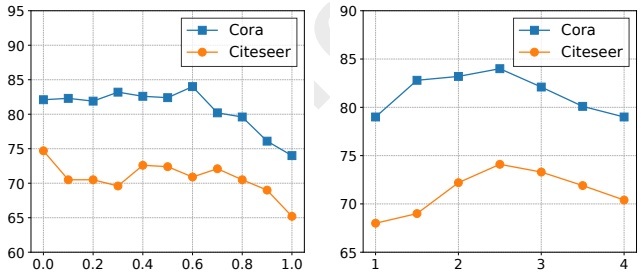


Figure 6: Accuracy over different hyperparameters, left: λ, right: α.

7 Conclusion

This paper addresses the issue of graph-free structural awareness in existing G2M distillation approaches. We propose SALE-MLP, which aims to align the node features to graph topology in a latent space using the unsupervised structural loss. The latent space learns both graph structure and node features, achieving the best performance compared with various SOTA methods. With extensive experiments on six datasets, we demonstrate that SALE outperforms GNN by 3.11%, other SOTA methods by 6.47%, and MLP by 16.75% on average in the inductive setting. In addition, the time was reduced by $150\times$ compared to GNN. Also, we analyze consistency and generalization. Ablation studies on loss components, training strategies, hyperparameter sensitivity, latent representation, and heterophily settings establish the effectiveness of SALE-MLP. Finally, experiments with different teachers and varied structural loss further demonstrate the broad applicability and adaptability of SALE-MLP.

Contribution

Harsh Pal, Sarthak Malik and Rajat Patel contributed equally.

References

[Bianchi *et al.*, 2020] Filippo Maria Bianchi, Daniele Grattarola, and Cesare Alippi. Spectral clustering with graph neural networks for graph pooling. In *International conference on machine learning*, pages 874–883, 2020.

[Chaurasiya *et al.*, 2022] Deepak Chaurasiya, Anil Surisetty, Nitish Kumar, Alok Singh, Vikrant Dey, Aakarsh Malhotra, Gaurav Dhama, and Ankur Arora. Entity alignment for knowledge graphs: progress, challenges, and empirical studies. *arXiv preprint arXiv:2205.08777*, 2022.

[Chen *et al.*, 2021] Yuzhao Chen, Yatao Bian, Xi Xiao, Yu Rong, Tingyang Xu, and Junzhou Huang. On self-distilling graph neural network. In Zhi-Hua Zhou, editor, *International Joint Conference on Artificial Intelligence*, pages 2278–2284, 2021.

[Chen *et al.*, 2022] Jie Chen, Shouzhen Chen, Mingyuan Bai, Junbin Gao, Junping Zhang, and Jian Pu. Sa-mlp: Distilling graph knowledge from gnns into structure-aware mlp. *arXiv preprint arXiv:2210.09609*, 2022.

[Church, 2017] Kenneth Ward Church. Word2vec. *Natural Language Engineering*, 23:155–162, 2017.

[Dhillon *et al.*, 2004] Inderjit S Dhillon, Yuqiang Guan, and Brian Kulis. Kernel k-means: spectral clustering and normalized cuts. In *ACM international conference on Knowledge discovery and data mining*, pages 551–556, 2004.

[Ding *et al.*, 2021] Mucong Ding, Kezhi Kong, Jingling Li, Chen Zhu, John Dickerson, Furong Huang, and Tom Goldstein. Vq-gnn: A universal framework to scale up graph neural networks using vector quantization. *Neural Information Processing Systems*, 34:6733–6746, 2021.

[Feng *et al.*, 2022] Wenzheng Feng, Yuxiao Dong, Tinglin Huang, Ziqi Yin, Xu Cheng, Evgeny Kharlamov, and Jie Tang. Grand+: Scalable graph random neural networks. In *ACM Web Conference*, pages 3248–3258, 2022.

[Grover and Leskovec, 2016] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *ACM International Conference on Knowledge Discovery and Data mining*, pages 855–864, 2016.

[Hamilton *et al.*, 2017] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Neural Information Processing Systems*, 30, 2017.

[Han *et al.*, 2022] Xiaotian Han, Zhimeng Jiang, Ninghao Liu, and Xia Hu. G-mixup: Graph data augmentation for graph classification. In *International Conference on Machine Learning*, pages 8230–8248, 2022.

[Hinton, 2015] Geoffrey Hinton. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[Hu *et al.*, 2020] 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. *Neural Information Processing Systems*, 33:22118–22133, 2020.

[Huang *et al.*, 2024] Yuyang Huang, Wenjing Lu, and Yang Yang. An efficient prototype-based clustering approach for edge pruning in graph neural networks to battle oversmoothing. In *International Joint Conference on Artificial Intelligence*, pages 4201–4209, 2024.

[Jha *et al.*, 2022] Kanchan Jha, Sriparna Saha, and Hiteshi Singh. Prediction of protein–protein interaction using graph neural networks. *Nature Publishing Group UK London*, 12:8360, 2022.

[Joshi *et al.*, 2022] Chaitanya K Joshi, Fayao Liu, Xu Xun, Jie Lin, and Chuan Sheng Foo. On representation knowledge distillation for graph neural networks. *IEEE transactions on neural networks and learning systems*, 2022.

[Kipf and Welling, 2022] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2022.

[Lassance *et al.*, 2020] Carlos Lassance, Myriam Bontonou, Ghouthi Hacene, Vincent Gripon, Jian Tang, and Antonio Ortega. Deep geometric knowledge distillation with graphs. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8484–8488, 2020.

[Li *et al.*, 2018] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *AAAI conference on artificial intelligence*, volume 32, 2018.

[Liu *et al.*, 2022] Xin Liu, Mingyu Yan, Lei Deng, Guoqi Li, Xiaochun Ye, Dongrui Fan, Shirui Pan, and Yuan Xie. Survey on graph neural network acceleration: An algorithmic perspective. In *International Joint Conference on Artificial Intelligence*, pages 5521–5529, 2022.

[Lu *et al.*, 2024] Weigang Lu, Ziyu Guan, Wei Zhao, and Yaming Yang. Adagmlp: Adaboosting gnn-to-mlp knowledge distillation. In *ACM Conference on Knowledge Discovery and Data Mining*, pages 2060–2071, 2024.

[Malik *et al.*, 2024] Sarthak Malik, Aditi Rai, Himank Sehgal, Akshay Sethi, Aakarsh Malhotra, et al. Grated-mlp: Efficient node classification via graph transformer distillation to mlp. In *Learning on Graphs Conference*, 2024.

[Mikolov *et al.*, 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Neural Information Processing Systems*, 26, 2013.

[Modak *et al.*, 2024] Sudipta Modak, Aakarsh Malhotra, Sarthak Malik, Anil Surisetty, and Esam Abdel-Raheem. Cpa-wac: constellation partitioning-based scalable weighted aggregation composition for knowledge graph embedding. In *International Joint Conference on Artificial Intelligence*, pages 3504–3512, 2024.

[Perozzi *et al.*, 2014] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *ACM international conference on Knowledge discovery and data mining*, pages 701–710, 2014.

[Postăvaru *et al.*, 2020] Ştefan Postăvaru, Anton Tsitsulin, Filipe Miguel Gonçalves de Almeida, Yingtao Tian, Silvio Lattanzi, and Bryan Perozzi. Instantembedding:

Efficient local node representations. *arXiv preprint arXiv:2010.06992*, 2020.

[Ren *et al.*, 2021] Yating Ren, Junzhong Ji, Lingfeng Niu, and Minglong Lei. Multi-task self-distillation for graph-based semi-supervised learning. *arXiv preprint arXiv:2112.01174*, 2021.

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

[Surisetty *et al.*, 2022] Anil Surisetty, Deepak Chaurasiya, Nitish Kumar, Alok Singh, Gaurav Dhama, Aakarsh Malhotra, Ankur Arora, and Vikrant Dey. Reps: Relation, position and structure aware entity alignment. In *ACM World Wide Web Conference Workshop on Graph Learning*, pages 1083–1091, 2022.

[Szegedy, 2013] C Szegedy. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[Tailor *et al.*, 2021] Shyam A. Tailor, Javier F.-Marques, and Nicholas Donald Lane. Degree-quant: Quantization-aware training for graph neural networks. In *International Conference on Learning Representations*, 2021.

[Tang *et al.*, 2009] Jie Tang, Jimeng Sun, Chi Wang, and Zi Yang. Social influence analysis in large-scale networks. In *ACM international conference on Knowledge discovery and data mining*, pages 807–816, 2009.

[Tang *et al.*, 2015] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *international conference on world wide web*, pages 1067–1077, 2015.

[Tian *et al.*, 2022] Yijun Tian, Chuxu Zhang, Zhichun Guo, Xiangliang Zhang, and Nitesh Chawla. Learning mlps on graphs: A unified view of effectiveness, robustness, and efficiency. In *International Conference on Learning Representations*, 2022.

[Veličković *et al.*, 2018] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

[Wang *et al.*, 2024] Zehong Wang, Zheyuan Zhang, Chuxu Zhang, and Yanfang Ye. Graph inference acceleration by learning mlps on graphs without supervision. *arXiv preprint arXiv:2402.08918*, 2024.

[Winter *et al.*, 2024] Daniel Winter, Niv Cohen, and Yedid Hoshen. Classifying nodes in graphs without gnns. *arXiv preprint arXiv:2402.05934*, 2024.

[Wu *et al.*, 2020] Zonghan Wu, Shirui Pan, F. Chen, G. Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32:4–24, 2020.

[Wu *et al.*, 2022a] Lirong Wu, Haitao Lin, Yufei Huang, and Stan Z Li. Knowledge distillation improves graph structure augmentation for graph neural networks. *Neural Information Processing Systems*, 35:11815–11827, 2022.

[Wu *et al.*, 2022b] Lirong Wu, Jun Xia, Haitao Lin, Zhangyang Gao, Zicheng Liu, Guojiang Zhao, and Stan Z Li. Teaching yourself: Graph self-distillation on neighborhood for node classification. *arXiv preprint arXiv:2210.02097*, 2022.

[Wu *et al.*, 2023a] Lirong Wu, Haitao Lin, Yufei Huang, Tianyu Fan, and Stan Z Li. Extracting low-/high-frequency knowledge from graph neural networks and injecting it into mlps: An effective gnn-to-mlp distillation framework. In *AAAI Conference on Artificial Intelligence*, volume 37, pages 10351–10360, 2023.

[Wu *et al.*, 2023b] Lirong Wu, Haitao Lin, Yufei Huang, and Stan Z Li. Quantifying the knowledge in gnns for reliable distillation into mlps. In *International Conference on Machine Learning*, pages 37571–37581, 2023.

[Wu *et al.*, 2023c] Taiqiang Wu, Zhe Zhao, Jiahao Wang, Xingyu Bai, Lei Wang, Ngai Wong, and Yujiu Yang. Edge-free but structure-aware: Prototype-guided knowledge distillation from gnns to mlps. *arXiv preprint arXiv:2303.13763*, 2023.

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

[Yan *et al.*, 2020] Bencheng Yan, C. Wang, G. Guo, and Yunkai Lou. Tinygnn: Learning efficient graph neural networks. In *ACM International Conference on Knowledge Discovery & Data Mining*, pages 1848–1856, 2020.

[Yang *et al.*, 2020] Yiding Yang, Jiayan Qiu, Mingli Song, Dacheng Tao, and Xinchao Wang. Distilling knowledge from graph convolutional networks. In *conference on computer vision and pattern recognition*, pages 7074–7083, 2020.

[Zhang *et al.*, 2020] Wentao Zhang, Xupeng Miao, Yingxia Shao, Jiawei Jiang, Lei Chen, Olivier Ruas, and Bin Cui. Reliable data distillation on graph convolutional network. In *ACM SIGMOD international conference on management of data*, pages 1399–1414, 2020.

[Zhang *et al.*, 2022a] Shichang Zhang, Yozen Liu, Yizhou Sun, and Neil Shah. Graph-less neural networks: Teaching old mlps new tricks via distillation. In *International Conference on Learning Representations*, 2022.

[Zhang *et al.*, 2022b] Yanfu Zhang, Shangqian Gao, Jian Pei, and Heng Huang. Improving social network embedding via new second-order continuous graph neural networks. In *ACM conference on knowledge discovery and data mining*, pages 2515–2523, 2022.

[Zhao *et al.*, 2020] Yiren Zhao, Duo Wang, Daniel Bates, Robert Mullins, Mateja Jamnik, and Pietro Lio. Learned low precision graph neural networks. *arXiv preprint arXiv:2009.09232*, 2020.