Extracting Low-/High- Frequency Knowledge from Graph Neural Networks and Injecting It into MLPs: An Effective GNN-to-MLP Distillation Framework

Lirong Wu[1,2,*]**, Haitao Lin**[1,2,*]**, Yufei Huang**[1,2]**, Tianyu Fan**[2]**, Stan Z. Li**[1,†]

[1] AI Division, School of Engineering, Westlake University, Hangzhou, 310030
[2] Zhejiang University, Hangzhou, 310058
{wulirong, linhaitao, huangyufei, fantianyu, stan.zq.li}@westlake.edu.cn

Abstract

Recent years have witnessed the great success of Graph Neural Networks (GNNs) in handling graph-related tasks. However, MLPs remain the primary workhorse for practical industrial applications due to their desirable inference efficiency and scalability. To reduce their gaps, one can directly distill knowledge from a well-designed teacher GNN to a student MLP, which is termed as GNN-to-MLP distillation. However, the process of distillation usually entails a loss of information, and *"which knowledge patterns of GNNs are more likely to be left and distilled into MLPs?"* becomes an important question. In this paper, we first factorize the knowledge learned by GNNs into low- and high-frequency components in the spectral domain and then derive their correspondence in the spatial domain. Furthermore, we identified a potential *information drowning* problem for existing GNN-to-MLP distillation, i.e., the high-frequency knowledge of the pre-trained GNNs may be overwhelmed by the low-frequency knowledge during distillation; we have described in detail what it represents, how it arises, what impact it has, and how to deal with it. In this paper, we propose an efficient *Full-Frequency GNN-to-MLP* (FF-G2M) distillation framework, which extracts both low-frequency and high-frequency knowledge from GNNs and injects it into MLPs. Extensive experiments show that FF-G2M improves over the vanilla MLPs by 12.6% and outperforms its corresponding teacher GNNs by 2.6% averaged over six graph datasets and three common GNN architectures.

Introduction

In many real-world applications, including social networks, chemical molecules, and citation networks, data can be naturally modeled as graphs. Recently, the emerging Graph Neural Networks (GNNs) (Hamilton, Ying, and Leskovec 2017; Kipf and Welling 2016; Veličković et al. 2017; Wu et al. 2021b, 2022b; Liu, Gao, and Ji 2020; Zhou et al. 2020; Liu et al. 2022; Xia et al. 2022) have demonstrated their powerful capability to handle various graph-related tasks (Zhang and Chen 2018; Fan et al. 2019; Errica et al. 2019; Wu et al. 2021a). However, practical deployments of GNNs in the industry are still less popular due to inference efficiency and scalability challenges incurred by data dependency (Jia et al. 2020; Zhang et al. 2021). In other words,

GNNs generally rely on message passing to aggregate features from the neighborhood, but fetching and aggregating these nodes during inference can burden latency-sensitive applications. In contrast, Multi-Layer Perceptrons (MLPs) involve no data dependence between pairs of nodes and infer much faster than GNNs, but often with less competitive performance. Motivated by these complementary strengths and weaknesses, one solution to reduce their gaps is to perform GNN-to-MLP knowledge distillation (Yang, Liu, and Shi 2021; Zhang et al. 2021; Ghorbani et al. 2021; Gou et al. 2021), which extracts knowledge from a well-trained teacher GNN and then distills it into a student MLP with the same network architecture (e.g., layer number and layer size).

Most of the existing GNN-to-MLP distillation methods (Yang, Liu, and Shi 2021; Zhang et al. 2021; Ghorbani et al. 2021) focus on special designs on either student MLPs or teacher GNNs, but default to distill knowledge in a node-to-node fashion. For example, CPF (Yang, Liu, and Shi 2021) combines Label Propagation (LP) (Iscen et al. 2019) into the student MLPs to improve classification performance and thus still suffers from the neighborhood-fetching latency caused by label propagation, which defeats the original intention of MLPs to be inference-efficient. In contrast, GLNN (Zhang et al. 2021) directly distills knowledge from arbitrary GNNs to vanilla MLPs with the same network architecture. While the distilled MLPs of GLNN can be greatly improved by employing more powerful teacher GNNs, the process of distillation usually entails a loss of information (Kim et al. 2021), which may lead to sub-optimal student MLPs. In this paper, we look away from specific instantiations of teacher GNNs and student MLPs, but rather explore two fundamental questions: (1) *Can existing GNN-to-MLP distillation ensure that sufficient knowledge is distilled from teacher GNNs to student MLPs?* If not, (2) *Which knowledge patterns of GNNs are more likely to be distilled into student MLPs?*

Present Work. In this paper, we identify a potential *information drowning* problem for existing GNN-to-MLP distillation, i.e., the high-frequency knowledge of the pre-trained GNNs may be overwhelmed by the low-frequency knowledge during distillation. To illustrate this, we first factorize GNN knowledge into low- and high-frequency components using graph signal processing theory in the spectral domain and then derive their correspondence in the spatial domain. Moreover, we conduct a comprehensive investigation of the

roles played by low- and high-frequency components in the distillation process and describe in detail what *information drowning* represents, how it arises, what impact it has, and how to deal with it. Extensive experiments have shown that high-frequency and low-frequency knowledge are complementary to each other, and they can further improve performance on top of each other. In this paper, we propose a novel *Full-Frequency GNN-to-MLP* (FF-G2M) distillation framework, which extracts both low- and high-frequency knowledge from teacher GNNs and injects it into student MLPs. As a simple and general framework, FF-G2M outperforms other leading methods across various GNN architectures and graph datasets in terms of classification performance. Codes are available at: https://github.com/LirongWu/FF-G2M.

Related Work

Graph Neural Networks (GNNs). The early GNNs define graph convolution kernels in the spectral domain (Bruna et al. 2013; Defferrard, Bresson, and Vandergheynst 2016) based on the graph signal processing theory, known as ChebyNet (Defferrard, Bresson, and Vandergheynst 2016) and Graph Convolutional Networks (GCN) (Kipf and Welling 2016). The later GNNs directly define updating rules in the spatial space and focus on the design of neighborhood aggregation functions. For instance, GraphSAGE (Hamilton, Ying, and Leskovec 2017) employs a generalized induction framework to generate embeddings for previously unseen nodes by aggregating known node features. Moreover, GAT (Veličković et al. 2017) introduces the self-attention mechanism to assign different importance scores to neighbors for better information aggregation. We refer interested readers to the surveys (Liu, Gao, and Ji 2020; Wu et al. 2020; Zhou et al. 2020) for more GNN architectures.

Graph Knowledge Distillation. Despite the great progress, most existing GNNs share the de facto design that relies on message passing to aggregate features from neighborhoods, which may be one major source of latency in GNN inference. To address this problem, there are previous works that attempt to distill knowledge from large teacher GNNs to smaller student GNNs, termed as GNN-to-GNN (Lassance et al. 2020; Wu et al. 2022a; Ren et al. 2021; Joshi et al. 2021). For example, the student model in RDD (Zhang et al. 2020) and TinyGNN (Yan et al. 2020) is a GNN with fewer parameters but not necessarily fewer layers than the teacher GNN, which makes both designs still suffer from the neighborhood-fetching latency caused by data dependency.

To enjoy the low-latency of MLPs and high-accuracy of GNNs, the other branch of graph knowledge distillation is to directly distill from large teacher GNNs to student MLPs, termed as GNN-to-MLP. The existing work on GNN-to-MLP distillation can be mainly divided into two branches: student MLPs-focused and teacher GNNs-focused. The former branch, such as CPF (Yang, Liu, and Shi 2021), *directly* improves student MLPs by adopting deeper and wider network architectures or incorporating label propagation, both of which burden the inference latency. The other branch, such as GLNN (Zhang et al. 2021), distills knowledge from teacher GNNs to vanilla MLPs with the same network architectures but without other computing-consuming operations;

while the performance of their distilled MLPs can be *indirectly* improved by employing more powerful GNNs, they still cannot match their corresponding teacher GNNs. In this paper, we aim to develop a **model-agnostic** GNN-to-MLP distillation that is applicable to various GNN architectures.

Preliminaries

Notions. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ be an attributed graph, where $\mathcal{V}$ is the set of N nodes with features $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N] \in \mathbb{R}^{N \times d}$ and $\mathcal{E}$ denotes the edge set. Each node $v_i \in \mathcal{V}$ is associated with a d-dimensional features vector $\mathbf{x}_i$, and each edge $e_{i,j} \in \mathcal{E}$ denotes a connection between node v_i and v_j. The graph structure is denoted by an adjacency matrix $\mathbf{A} \in [0,1]^{N \times N}$ with $\mathbf{A}_{i,j} = 1$ if $e_{i,j} \in \mathcal{E}$ and $\mathbf{A}_{i,j} = 0$ if $e_{i,j} \notin \mathcal{E}$. Consider a semi-supervised node classification task where only a subset of node $\mathcal{V}_L$ with labels $\mathcal{Y}_L$ are known, we denote the labeled set as $\mathcal{D}_L = (\mathcal{V}_L, \mathcal{Y}_L)$ and unlabeled set as $\mathcal{D}_U = (\mathcal{V}_U, \mathcal{Y}_U)$, where $\mathcal{V}_U = \mathcal{V} \backslash \mathcal{V}_L$. The node classification aims to learn a mapping $\Phi : \mathcal{V} \to \mathcal{Y}$ with labeled data, so that it can be used to infer the labels $\mathcal{Y}_U$.

Graph Neural Networks (GNNs). Most existing GNNs rely on message passing to aggregate features from the neighborhood. A general GNN framework consists of two key computations for each node v_i: (1) AGGREGATE: aggregating messages from neighborhood $\mathcal{N}_i$; (2) UPDATE: updating node representation from its representation in the previous layer and aggregated messages. Considering a L-layer GNN, the formulation of the l-th layer is as follows

$$
\begin{aligned}
\mathbf{m}_i^{(l)} &= \text{AGGREGATE}^{(l)} \left(\{ \mathbf{h}_j^{(l-1)} : v_j \in \mathcal{N}_i \} \right) \\
\mathbf{h}_i^{(l)} &= \text{UPDATE}^{(l)} \left(\mathbf{h}_i^{(l-1)}, \mathbf{m}_i^{(l)} \right)
\end{aligned}
\tag{1}
$$

where $1 \leq l \leq L$, $\mathbf{h}_i^{(0)} = \mathbf{x}_i$ is the input feature, and $\mathbf{h}_i^{(l)}$ is the representation of node v_i in the l-th layer.

Multi-Layer Perceptrons (MLPs). To achieve efficient inference, the vanilla MLPs (with the same network architecture as the teacher GNNs) are used as the student model by default in this paper. For a L-layer MLP, the l-th layer is composed of a linear transformation, an activation function $\sigma = \text{ReLu}(\cdot)$, and a dropout function $\text{Dropout}(\cdot)$, as

$$
\mathbf{z}_i^{(l)} = \text{Dropout} \left(\sigma \left(\mathbf{z}_i^{(l-1)} \mathbf{W}^{(l-1)} \right) \right), \quad \mathbf{z}_i^{(0)} = \mathbf{x}_i
\tag{2}
$$

where $\mathbf{W}^{(0)} \in \mathbb{R}^{d \times F}$ and $\mathbf{W}^{(l)} \in \mathbb{R}^{F \times F}$ $(1 \leq l < L)$ are weight matrices with the hidden dimension F. In this paper, the network architecture of MLPs, such as the layer number L and layer size F, is set the same as that of teacher GNNs.

GNN-to-MLP Knowledge Distillation. The knowledge distillation is first introduced in (Hinton et al. 2015) to handle mainly image data, where knowledge is transferred from a cumbersome teacher model to a simpler student model. The later works on GNN-to-MLP distillation (Yang, Liu, and Shi 2021; Zhang et al. 2021; Ghorbani et al. 2021) extend it to the graph domain by imposing KL-divergence constraint $\mathcal{D}_{KL}(\cdot, \cdot)$ between the softmax label distributions generated by teacher GNNs and student MLPs and directly optimizing the objective function as follows

$$
\mathcal{L}_{\text{KD}} = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} \mathcal{D}_{KL} \left(\text{softmax} \left(\mathbf{z}_i^{(L)} \right), \text{softmax} \left(\mathbf{h}_i^{(L)} \right) \right)
\tag{3}
$$

Knowledge Factorization from the Perspective of Spectral and Spatial Domain

In this section, we first theoretically factorize the knowledge learned by GNNs into low- and high-frequency components in the spectral domain based on graph signal processing theory (Shuman et al. 2013). The normalized graph Laplacian matrix of graph $\mathcal{G}$ is defined as $\mathbf{L} = \mathbf{I}_N - \widetilde{\mathbf{D}}^{-\frac{1}{2}} \widetilde{\mathbf{A}} \widetilde{\mathbf{D}}^{-\frac{1}{2}}$, where $\widetilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N \in \mathbb{R}^{N \times N}$ is an adjacency matrix with self-loop, $\widetilde{\mathbf{D}} \in \mathbb{R}^{N \times N}$ is a diagonal degree matrix with $\widetilde{\mathbf{D}}_{i,i} = \sum_j \widetilde{\mathbf{A}}_{i,j}$, and $\mathbf{I}_N$ denotes the identity matrix. Since $\mathbf{L}$ is a real symmetric matrix, it can be eigendecomposed as $\mathbf{L} = \mathbf{U}\Lambda\mathbf{U}^\top$, where $\Lambda = \mathrm{diag}\left([\lambda_1, \lambda_2, \cdots, \lambda_N]\right)$ with each eigenvalue $\lambda_l \in [0, 2]$ corresponding to an eigenvectors $\mathbf{u}_l$ in $\mathbf{U}$ (Chung and Graham 1997). According to graph signal processing theory, we can directly take the eigenvector $\{\mathbf{u}_l\}_{l=1}^N$ as bases. Given signal $\mathbf{x} \in \mathbb{R}^d$, the graph Fourier transform and inverse Fourier transform (Sandryhaila and Moura 2013; Ricaud et al. 2019) are defined as $\widehat{\mathbf{x}} = \mathbf{U}^\top \mathbf{x}$ and $\mathbf{x} = \mathbf{U}\widehat{\mathbf{x}}$. Thus, the convolutional $*_G$ between the signal $\mathbf{x}$ and convolution kernel $\mathcal{F}$ can be defined as follows

$$\mathcal{F} *_G \mathbf{x} = \mathbf{U}\left((\mathbf{U}^\top \mathcal{F}) \odot (\mathbf{U}^\top \mathbf{x})\right) = \mathbf{U}\mathbf{g}_\theta \mathbf{U}^\top \mathbf{x} \quad (4)$$

where $\odot$ denotes the element-wise product and $\mathbf{g}_\theta$ is a parameterized diagonal matrix. Most of the existing GNNs architectures can be regarded as a special instantiation on the convolutional kernel $\mathcal{F}$ (i.e., the matrix $\mathbf{g}_\theta$). For example, GCN-Cheby parameterizes g_θ with a polynomial expansion $\mathbf{g}_\theta = \sum_{k=0}^{K-1} \alpha_k \Lambda^k$, and GCN defines the convolutional kernel as $\mathbf{g}_\theta = \mathbf{I}_N - \Lambda$. Considering a special convolution kernel $\mathcal{F}_A = \mathbf{I}_N$, we have $\mathcal{F}_A *_G \mathbf{x} = \mathbf{U}\mathbf{I}_N\mathbf{U}^\top \mathbf{x} = \mathbf{U}\widehat{\mathbf{x}} = \mathbf{x}$, i.e., this is an identity mapping, where all information can be preserved. Next, we decompose the graph knowledge into low-frequency and high-frequency components (Bo et al. 2021a; Wu et al. 2019) by factorizing $\mathcal{F}_A = \mathbf{I}_N$ as follows

$$\mathcal{F}_A = \mathbf{I}_N = \frac{1}{2}\left(\underbrace{\left(\mathbf{I}_N + \widetilde{\mathbf{D}}^{-\frac{1}{2}} \widetilde{\mathbf{A}} \widetilde{\mathbf{D}}^{-\frac{1}{2}}\right)}_{\text{Low-Pass Filter } \mathcal{F}_M} + \underbrace{\left(\mathbf{I}_N - \widetilde{\mathbf{D}}^{-\frac{1}{2}} \widetilde{\mathbf{A}} \widetilde{\mathbf{D}}^{-\frac{1}{2}}\right)}_{\text{High-pass Filter } \mathcal{F}_H}\right)$$

For a given signal $\mathbf{x} \in \mathbb{R}^d$, e.g., node feature, we have $\mathcal{F}_A *_G \mathbf{x} = \frac{1}{2}(\mathcal{F}_M + \mathcal{F}_H) *_G \mathbf{x} = \frac{1}{2}(\mathcal{F}_M *_G \mathbf{x} + \mathcal{F}_H *_G \mathbf{x}) = \mathbf{x}$, which means that any signal $\mathbf{x}$ can be decomposed into the average of two components $\mathcal{F}_M *_G \mathbf{x}$ and $\mathcal{F}_H *_G \mathbf{x}$.

Analysis on the Spectral Domain. The Proposition 1 states what the two components $\mathcal{F}_M *_G \mathbf{x}$ and $\mathcal{F}_H *_G \mathbf{x}$ represent.

Proposition 1 *The convolution kernel $\mathcal{F}_M$ works as a low-pass filter, which filters out high-frequency information, and $\mathcal{F}_M *_G \mathbf{x}$ represents low-frequency knowledge; $\mathcal{F}_H$ works as a high-pass filter, which filters out low-frequency information, and $\mathcal{F}_H *_G \mathbf{x}$ represents high-frequency knowledge.*

Proof 1 *For a L-layer GNN, the signal $\mathbf{x}$ is filtered by the L-order convolution kernel $\mathcal{F}_M^L = (\mathbf{I}_N + \widetilde{\mathbf{D}}^{-\frac{1}{2}} \widetilde{\mathbf{A}} \widetilde{\mathbf{D}}^{-\frac{1}{2}})^L = (2\mathbf{I}_N - \mathbf{L})^L$ to output $\mathcal{F}_M^L *_G \mathbf{x} = \mathbf{U}(2\mathbf{I}_N - \Lambda)^L\mathbf{U}^\top \mathbf{x}$ with $g_\theta^L(\lambda_i) = (2 - \lambda_i)^L$. As shown in Fig. 1, $g_\theta^L(\lambda_i)$ decreases monotonically in the range $\lambda_i \in [0, 2]$ and reaches $g_\theta^L(\lambda_i = 2) = 0$ at $\lambda_i = 2$, which mainly amplifies the low-frequency information and filters out the high-frequency information.*
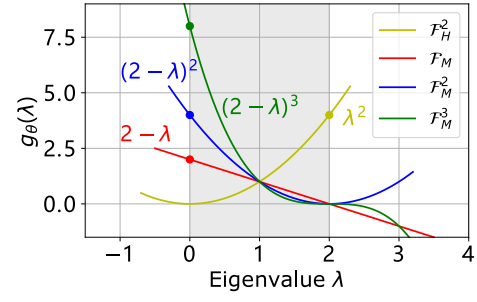

Figure 1: Eigenvalues *vs.* Amplitudes

Similarly, the L-order convolution kernel $\mathcal{F}_H^L$ has $g_\theta^L(\lambda_i) = \lambda_i^L$. As shown in Fig. 1, $g_\theta^L(\lambda_i)$ increases monotonically in the of range $\lambda_i \in [0, 2]$ and reaches $g_\theta^L(\lambda_i = 0) = 0$ at $\lambda_i = 0$, which mainly filters out the low-frequency information but in turn amplifies the high-frequency information.

Correspondence on the Spatial Domain. We have derived that $\mathcal{F}_M *_G \mathbf{x}$ and $\mathcal{F}_H *_G \mathbf{x}$ represent mainly the low- and high-frequency components of signal $\mathbf{x}$, and we next can derived their correspondences in the spatial domain, as follows

$$\begin{aligned} \mathcal{F}_M *_G \mathbf{x}_i &\rightarrow \mathbf{x}_i^{(low)} = \mathbf{x}_i + \sum_{j \in \mathcal{N}_i} \frac{\mathbf{x}_j}{\sqrt{|\mathcal{N}_i||\mathcal{N}_j|}} \\ \mathcal{F}_H *_G \mathbf{x}_i &\rightarrow \mathbf{x}_i^{(high)} = \mathbf{x}_i - \sum_{j \in \mathcal{N}_i} \frac{\mathbf{x}_j}{\sqrt{|\mathcal{N}_i||\mathcal{N}_j|}} \end{aligned} \quad (5)$$

By the derivation in Eq. (5), the low-frequency knowledge $\mathcal{F}_M *_G \mathbf{x}$ is the sum of node feature and its neighborhood features in the spatial domain. On the other hand, the high-frequency knowledge $\mathcal{F}_H *_G \mathbf{x}$ represents the differences between the target node feature with its neighborhood features. There have recently been some novel GNN models (Bo et al. 2021b; Pei et al. 2020; Zhu et al. 2020; Chien et al. 2021) that can capture both low- and high-frequency information simultaneously or adaptively. However, in this paper, we focus on the design of distillation objective functions and do not consider indirect performance improvements by employing these more powerful but complex GNNs. Instead, we consider the most commonly used GNNs, such as GCN (Kipf and Welling 2016), GraphSAGE (Hamilton, Ying, and Leskovec 2017), and GAT (Veličković et al. 2017), all of which rely on multi-layer message passing to aggregate features of neighboring nodes that are multiple hops away, i.e., they essentially work as a low-pass filter $\mathcal{F}_M^L$ or its variants.

Roles Played by Low- and High-Frequency Knowledge during Distillation

Rethinking the Core of Knowledge Distillation

We rethink the core of knowledge distillation from three shallow-to-deep perspectives to highlight our motivations.

- *Firstly*, knowledge distillation enables the representations of MLPs to "mimic" those of GNNs as closely as
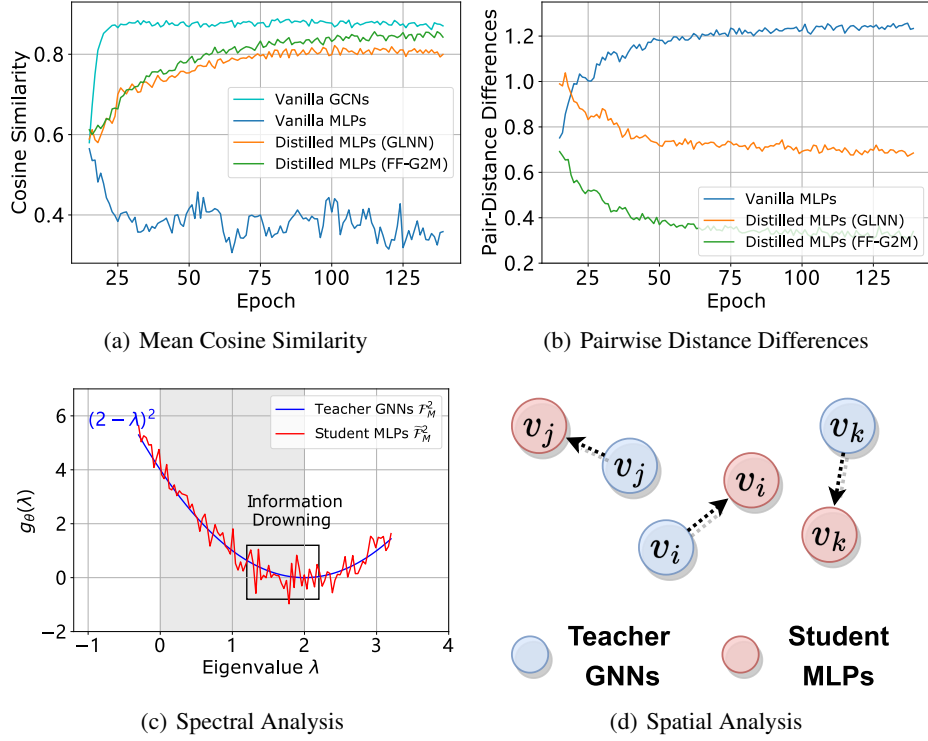
Figure 2: (a) Mean cosine similarity (the higher, the better) between nodes with their first-order neighbors on Cora. (b) Pairwise distance differences (the lower, the better) between teacher GCNs and student MLPs on Cora. (c)(d) Illustrations of how the high-frequency information drowning arises and what potential impact it has in the spectral and spatial domains, respectively.

possible by imposing KL-divergence constraints between their softmax distribution probabilities. However, such a mimicking (or fitting) process is inevitably accompanied by a loss of information, especially high-frequency information, which explains why the performance of student MLPs is always hard to match with that of teacher GNNs.

- **Secondly**, for a neural network framework, any change in the final representations is achieved indirectly by optimizing the mapping function, i.e., the network parameters. In this sense, knowledge distillation essentially optimizes the parameter matrices $\{\mathbf{W}^{(l)}\}_{l=0}^{L-1}$ of the student MLPs to make it functionally approximates the convolution kernel of the teacher GNNs, which makes the student MLPs also serve as a low-pass filter $\widetilde{\mathcal{F}}_M^L$ for graph data.

- **Finally**, the low-pass filter in the spectral domain is equivalent to neighborhood aggregation in the spatial domain as derived in Eq. (5), which in essence can be considered as a special use of the graph topology.

To explore the roles played by graph topology during GNN-to-MLP distillation, we plot the *mean cosine similarity* of nodes with their first-order neighbors for vanilla GCNs, vanilla MLPs, and Distilled MLPs (GLNN) on the Cora dataset in Fig. 2(a), from which we observe that the mean similarity of GCNs and GLNN gradually increases with training, while that of vanilla MLPs gradually decreases, which indicates that knowledge distillation has in-

troduced graph topology as an inductive bias (as GCNs has done), while vanilla MLPs do not. As a result, the distilled MLPs can enjoy the benefits of topology-awareness in training but without neighborhood-fetching latency in inference.

High-Frequency Information Drowning

Next, we discuss a potential high-frequency information drowning problem from both spectral and spatial domains, i.e., the high-frequency information of the pre-trained GNNs may be overwhelmed by the low-frequency knowledge during the process of GNN-to-MLP knowledge distillation.

How information drowning arises? *From the perspective of spectral domain*, the knowledge distillation optimizes the network parameters of the student MLPs to make it functionally approximate the convolution kernel of the teacher GNNs, i.e., $\widetilde{\mathcal{F}}_M^L \approx \mathcal{F}_M^L$. The information loss induced by such approximation may be inconsequential for high-amplitude low-frequency information but can be catastrophic for those high-frequency information with very low amplitude, as shown in Fig. 2(c). As a result, compared to low-frequency information, high-frequency information is more likely to be drowned by these optimization errors.

What impact does information drowning have? *From the perspective of spatial domain*, the information drowning may lead to distilled MLPs that, despite preserving neighborhood smoothing well, can easily neglect differences between nodes, such as pairwisde distances. To illustrate this,
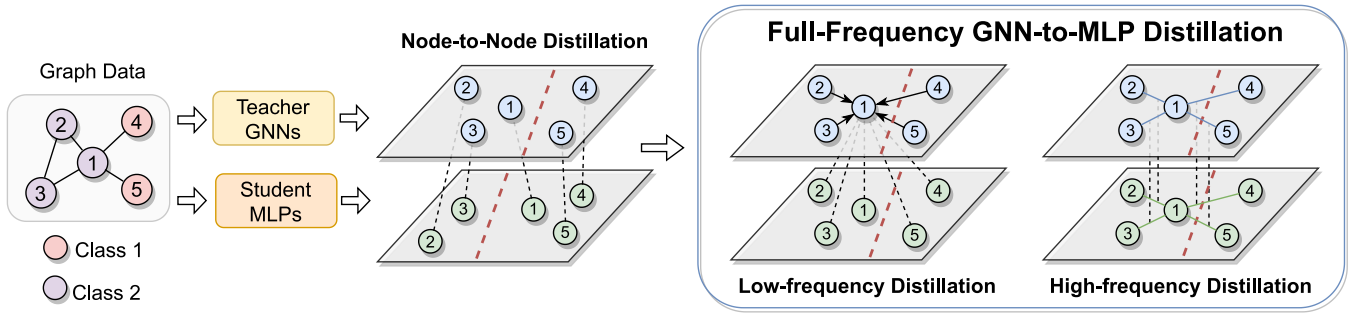
Figure 3: Illustration of the Full-Frequency GNN-to-MLP (FF-G2M) distillation framework, where the dotted red lines denote the predicted class-boundary, the solid black lines denote feature aggregation from the neighborhood, and the dashed black lines denote the distillation of knowledge (neighborhood features and pairwise distances) from teacher GNNs to student MLPs.

we consider a target node v_i and its two neighboring nodes v_j and v_k in Fig. 2(d), where they are mapped closely by GNNs. In the process of knowledge distillation, the representations of these three nodes may be mapped around the representations of teacher GNNs, i.e., they are still mapped closely with most of their low-frequency information preserved; however, the relative distances between nodes, i.e., their high-frequency information, may be drowned dramatically. For example, node v_i is adjacent to node v_j but far from node v_k in the representation space of teacher GNNs. However, in the representation space of student MLPs, node v_i becomes closer to node v_k and farther from node v_j.

The curves of the pairwise distance differences between the teacher GCNs and the student MLP in Fig. 2(b) show that common knowledge distillation (e.g., GLNN) is not good at capturing high frequency information, compared to our proposed FF-G2M. Moreover, extensive qualitative and quantitative experiments have been provided to demonstrate the harmfulness of the identified high-frequency information drowning problem in the experimental section. The detailed experimental settings, including hyperparameters and evaluation metric definitions, are available in **Appendix B&E**.

Full-Frequency GNN-to-MLP (FF-G2M) Knowledge Distillation

The above discussions reached two important insights: (1) the inductive bias of graph topology plays an important role, and (2) it is mainly the low-frequency knowledge of graph data that has been distilled from the teacher GNNs to the student MLPs. Inspired by these two insights, we propose *Low-Frequency Distillation* (LFD) and *High-Frequency Distillation* (HFD) to fully capture the low-frequency and high-frequency knowledge learned by GNNs, respectively. An high-level overview of the proposed *Full-Frequency GNN-to-MLP* (FF-G2M) framework is shown in Fig. 3.

Low-Frequency Distillation (LFD)

The node representations of teacher GNNs are generated by explicit message passing, so it mainly captures the low-frequency information of the graph data as analyzed earlier. Unlike aggregating features from neighborhoods as in

GNNs, we directly distill (diffuse) knowledge from teacher GNNs into the neighborhoods of student MLPs in order to better utilize topological information and low-frequency knowledge captured by GNNs, formulated as follows

$$\mathcal{L}_{\text{LFD}} = \frac{1}{|\mathcal{E}|} \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{N}_i \cup i} \mathcal{D}_{KL}\Big(\sigma(\mathbf{z}_j^{(L)}/\tau_1), \sigma(\mathbf{h}_i^{(L)}/\tau_1)\Big) \quad (6)$$

where τ_1 is the low-frequency distillation temperature, and $\sigma = \text{softmax}(\cdot)$ denotes an activation function.

High-Frequency Distillation (HFD)

As derived in Eq. (5), the high-frequency components in the spectral domain represent the differences between node feature and its neighborhood features in the spatial domain. Inspired by this, we propose High-Frequency Distillation (HFD), a GNN knowledge objective that trains student MLPs to preserve the neighborhood pairwise differences from the representation space of teacher GNNs. The neighborhood pairwise differences around node v_i are defined as the differences between the target node feature $\mathbf{s}_i$ and its neighborhood features $\{\mathbf{s}_j \mid j \in \mathcal{N}_i\}$, which can be computed by the kernel $\mathcal{K}(\mathbf{s}_i, \mathbf{s}_j) = |\mathbf{s}_i - \mathbf{s}_j|$, where $|\cdot|$ denotes the element-wise absolute values. The high-frequency distillation trains the student model to mimic the neighborhood pairwise differences from the teacher GNNs via KL-divergence constraints, which can be defined as follows

$$\mathcal{L}_{\text{HFD}} = \frac{1}{|\mathcal{E}|} \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{N}_i} \mathcal{D}_{\text{KL}}\Big(\sigma\big(\mathcal{K}(\mathbf{z}_i^{(L)}, \mathbf{z}_j^{(L)})/\tau_2\big),$$
$$\sigma\big(\mathcal{K}(\mathbf{h}_i^{(L)}, \mathbf{h}_j^{(L)})/\tau_2\big)\Big) \quad (7)$$

where $\mathcal{K}(\cdot, \cdot)$ denotes the element-wise absolute values, and τ_2 is the high-frequency distillation temperature.

Training Strategy

The pseudo-code of the FF-G2M framework is summarized in **Appendix C**. To achieve GNN-to-MLP knowledge distillation, we first pre-train the teacher GNNs with the classification loss $\mathcal{L}_{\text{label}} = \frac{1}{|\mathcal{V}_L|} \sum_{i \in \mathcal{V}_L} \mathcal{H}\big(y_i, \sigma(\mathbf{h}_i^{(L)})\big)$, where $\mathcal{H}(\cdot)$ denotes the cross-entropy loss and y_i is the ground-truth label of node v_i. Finally, the total objective function

Teacher GNNs	Method	Cora	Citeseer	Pubmed	Amazon-Photo	Coauthor-CS	Coauthor-Phy	*Average*↑
	Vanilla GCN	82.2 ± 0.5	71.6 ± 0.4	79.3 ± 0.3	91.8 ± 0.6	89.9 ± 0.7	91.9 ± 1.2	-
	Vanilla MLP	59.7 ± 1.0	60.7 ± 0.5	71.5 ± 0.5	77.4 ± 1.2	87.5 ± 1.4	89.2 ± 0.9	-
	GLNN	82.8 ± 0.5	72.7 ± 0.4	80.2 ± 0.6	91.4 ± 1.0	92.7 ± 1.0	93.2 ± 0.5	-
GCN	FF-G2M	**84.3 ± 0.4**	**74.0 ± 0.5**	**81.8 ± 0.4**	**94.2 ± 0.4**	**93.8 ± 0.5**	**94.4 ± 0.9**	-
	Δ_{GCN}	2.1	2.4	2.5	2.4	3.9	2.5	2.63
	Δ_{MLP}	24.6	13.3	10.3	16.8	6.3	5.2	12.75
	Δ_{GLNN}	1.5	1.3	1.6	2.8	1.1	1.2	1.58
	Vanilla SAGE	82.5 ± 0.6	70.9 ± 0.6	77.9 ± 0.4	92.0. ± 0.6	89.7 ± 1.0	92.2 ± 0.9	-
	Vanilla MLP	59.7 ± 1.0	60.7 ± 0.5	71.5 ± 0.5	77.4 ± 1.2	87.5 ± 1.4	89.2 ± 0.9	-
	GLNN	82.7 ± 0.8	70.5 ± 0.5	79.7 ± 0.6	91.0 ± 0.9	92.5 ± 1.0	93.0 ± 0.7	-
GraphSAGE	FF-G2M	**84.5 ± 0.8**	**73.1 ± 1.2**	**80.9 ± 0.5**	**94.1 ± 0.5**	**93.6 ± 0.5**	**94.5 ± 1.1**	-
	Δ_{SAGE}	2.0	2.2	3.0	2.1	3.9	2.3	2.58
	Δ_{MLP}	24.8	12.4	9.4	16.7	6.1	5.3	12.45
	Δ_{GLNN}	1.8	2.6	1.2	3.1	1.1	1.5	1.88
	Vanilla GAT	81.9 ± 1.0	71.2 ± 0.5	78.6 ± 0.4	91.4 ± 0.8	90.5 ± 0.8	92.3 ± 1.5	-
	Vanilla MLP	59.7 ± 1.0	60.7 ± 0.5	71.5 ± 0.5	77.4 ± 1.2	87.5 ± 1.4	89.2 ± 0.9	-
	GLNN	82.1 ± 0.7	70.6 ± 0.8	80.4 ± 1.0	91.7 ± 0.7	92.7 ± 0.9	92.7 ± 0.6	-
GAT	FF-G2M	**84.0 ± 0.7**	**73.5 ± 0.8**	**81.1 ± 0.6**	**93.9 ± 0.7**	**94.0 ± 0.7**	**94.7 ± 1.2**	-
	Δ_{GAT}	2.1	2.3	2.5	2.5	3.5	2.4	2.55
	Δ_{MLP}	24.3	12.8	9.6	16.5	6.5	5.5	12.53
	Δ_{GLNN}	1.9	2.9	0.7	2.2	1.3	2.0	1.83

Table 1: Classificatiom accuracy ± std (%) on six real-world datasets, where we consider three different GNN architectures (GCN, GraphSAGE, and GAT) as the teacher and pure MLPs as the student model. The best metrics are marked by bold.

to distill the low- and high-frequency knowledge from the teacher GNNs into the student MLPs is defined as follows

$$\mathcal{L}_{\text{total}} = \frac{\lambda}{|\mathcal{V}_L|} \sum_{i \in \mathcal{V}_L} \mathcal{H}\big(y_i, \sigma(\mathbf{z}_i^{(L)})\big) + \big(1 - \lambda\big)\big(\mathcal{L}_{\text{LFD}} + \mathcal{L}_{\text{HFD}}\big)$$

where λ is the weights to balance the influence of the classification loss and two knowledge distillation losses. The time complexity analysis of FF-G2M is available in **Appendix D**.

Discussion and Comparision

In this subsection, we compare the proposed FF-G2M framework with the commonly used node-to-node distillation (e.g., GLNN) in Fig. 3. While the node-to-node distillation can map neighboring nodes closely in the representation space of MLPs, i.e., preserving low-frequency knowledge, it completely confounds the relative distance between node pairs, i.e., high-frequency knowledge is drowned, leading to a different (incorrect) class-boundary with the teacher GNNs. In terms of the proposed FF-G2M framework, the Low-Frequency Distillation distills (diffuses) the features aggregated from the neighborhood in the teacher GNNs back into their neighborhood of the student MLPs to better utilize the extracted low-frequency knowledge. Besides, High-Frequency Distillation directly distills the neighborhood pairwise differences from teacher GNNs into the student MLPs to better capture the high-frequency knowledge patterns, i.e., the relative positions between pairs of nodes.

Experiments

Datasets. The effectiveness of the FF-G2M framework is evaluated on six public real-world datasets, including Cora (Sen et al. 2008), Citeseer (Giles, Bollacker, and Lawrence 1998), Pubmed (McCallum et al. 2000), Coauthor-CS, Coauthor-Physics, and Amazon-Photo (Shchur et al. 2018). For each dataset, following the data splitting settings of (Kipf and Welling 2016; Liu, Gao, and Ji 2020), we select 20 nodes per class to construct a training set, 500 nodes for validation, and 1000 nodes for testing. A statistical overview of these datasets is placed in **Appendix A**. Besides, we defer the implementation details and hyperparameter settings for each dataset to **Appendix B** and supplementary materials.

Baselines. Three basic components in knowledge distillation are (1) teacher model, (2) student model, and (3) distillation loss. As a model-agnostic general framework, FF-G2M can be combined with any teacher GNN architecture. In this paper, we consider three types of teacher GNNs, including GCN (Kipf and Welling 2016), GraphSAGE (Hamilton, Ying, and Leskovec 2017), and GAT (Veličković et al. 2017). As for the student model, we default to using pure MLPs (with the same network architecture as the teacher GNNs) as the student model for a fair comparison. Finally, the focus of this paper is on designing distillation objectives rather than powerful teacher and student models. Therefore, we only take GLNN (Zhang et al. 2021) as an important baseline to compare FF-G2M with the conventional node-to-node distillation approach. The experiments of all baselines and FF-G2M are implemented based on the standard implementation in the DGL library (Wang et al. 2019) using PyTorch 1.6.0 library on NVIDIA V100 GPU. Each set of experiments is run five times with different random seeds, and the average are reported as metrics.

(a) Pre-trained GCNs

(b) Vanilla MLPs

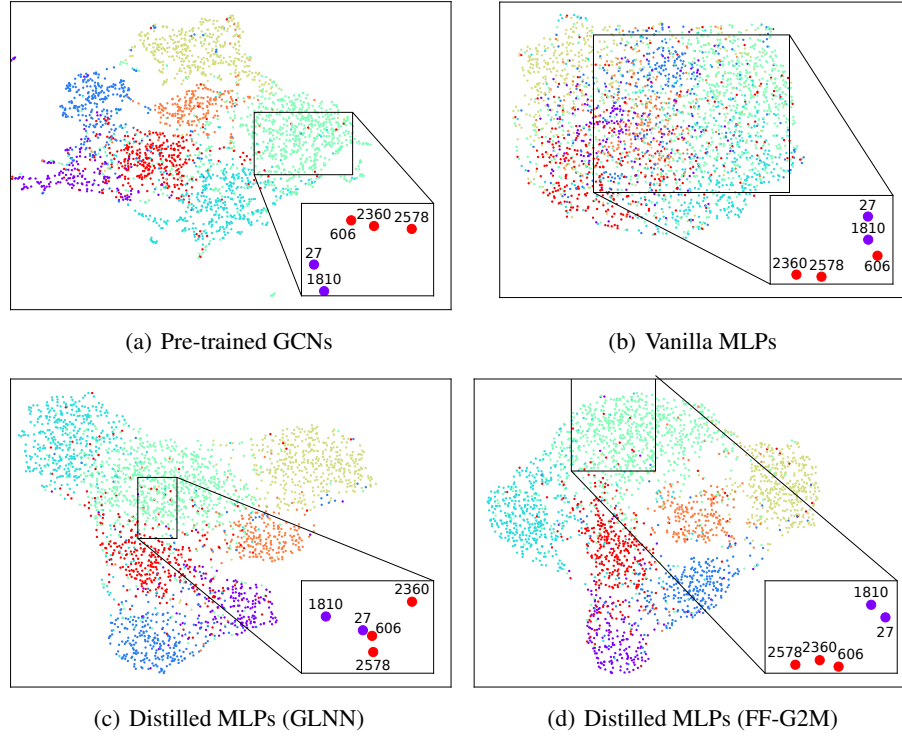(c) Distilled MLPs (GLNN)

(d) Distilled MLPs (FF-G2M)

Figure 4: Representation 2D-Visualizations (by UMAP (McInnes, Healy, and Melville 2018)) of the teacher model and three student models on Cora. Each node is colored by its ground-truth label, and the numbers around the nodes denote the node ids.

Classification Performance Comparison

This paper aims to explore which knowledge patterns *should* and *how to* be distilled into student MLPs, rather than designing more powerful teacher GNNs. Therefore, we consider three classical GNNs, including GCN, GraphSAGE, and GAT, as the teacher models and distill their knowledge into MLPs with the same network architecture. The experimental results on six datasets are reported in Table. 1, from which we can make the following observations: (1) In general, more powerful teacher GNNs can lead to student MLPs with better classification performance. However, such improvements are usually very limited and do not work for all datasets and GNN architectures. For example, on the Citeseer dataset, the performance of GLNN drops over the vanilla implementation of teacher GNNs by 0.4% (Graph-SAGE) and 0.6% (GAT), respectively. (2) The proposed FF-G2M framework can consistently improve the performance of student MLPs across three GNN architectures on all six datasets. For example, FF-G2M can outperform vanilla teacher GNNs by 2.63% (GCN), 2.58% (GraphSAGE), and 2.55% (GAT) averaged over six datasets, respectively.

Qualitative and Quantitative Analysis

Extensive qualitative and quantitative experiments are conducted to explore the existence and harmfulness of the information drowning problem and how to solve it by FF-G2M.

Qualitative Analysis on Visualizations. We consider GCNs as the teacher model and compare its visualization with that of vanilla MLPs, GLNN, and FF-G2M on the Cora dataset (due to space limitations, more results can be found in **Appendix F**). We select a target node (id 27 for Cora) and analyze its relative position relationship with its neighbors in Fig.4, from which we observe that: (1) The vanilla MLPs map neighboring nodes apart, which indicates that it is not even good at capturing low-frequency information. (2) GLNN fails to capture the relative positions between the target node and its neighbors, i.e., high-frequency information. (3) FF-G2M well preserves the relative positions between nodes while mapping neighboring nodes closely, which suggests that it is good at capturing both low- and high-frequency information. For example, on the Cora dataset, the target node (id 27) is the closest to node (id 1810) and the farthest from node (id 2678) in the visualizations of both teacher GCNs and FF-G2M's student MLPs.

Quantitative Analysis on Evaluation Metrics. To study what knowledge patterns of GNNs are actually distilled into MLPs during GNN-to-MLP distillation, we consider both (1) low-frequency knowledge, measured by the *mean cosine similarity* of nodes with their first-order neighbors, and (2) high-frequency knowledge, measured by *KL-divergence* between the pairwise distances of teacher GNNs and student MLPs, respectively. The detailed mathematical definitions of these two evaluation metrics are available in **Appendix**

Method	Cora	Citeseer	Pubmed	Amazon-Photo	Coauthor-CS	Coauthor-Phy
Vanilla GCN	82.2 ± 0.5	71.6 ± 0.4	79.3 ± 0.3	91.8 ± 0.6	89.9 ± 0.7	91.9 ± 1.2
Vanilla MLP	59.7 ± 1.0	60.7 ± 0.5	71.5 ± 0.5	77.4 ± 1.2	87.5 ± 1.4	89.2 ± 0.9
GLNN (Zhang et al. 2021)	82.8 ± 0.5	72.7 ± 0.4	80.2 ± 0.6	91.4 ± 1.0	92.7 ± 1.0	93.2 ± 0.5
Low-Frequency KD w/ $\mathcal{L}_{\mathrm{LFD}}$	83.4 ± 0.9	73.7 ± 0.6	81.0 ± 0.5	92.1 ± 0.8	93.2 ± 0.8	93.7 ± 0.8
High-Frequency KD w/ $\mathcal{L}_{\mathrm{HFD}}$	68.5 ± 0.8	63.2 ± 0.7	74.4 ± 0.4	82.5 ± 1.3	89.3 ± 1.7	91.0 ± 1.6
FF-G2M (full model)	$\mathbf{84.3 \pm 0.4}$	$\mathbf{74.0 \pm 0.5}$	$\mathbf{81.8 \pm 0.4}$	$\mathbf{94.2 \pm 0.4}$	$\mathbf{93.8 \pm 0.5}$	$\mathbf{94.4 \pm 0.9}$

Table 2: Classificatiom accuracy $\pm$ std (%) on six real-world datasets. The best metrics are marked by bold.

(a) Low-frequency Knowledge
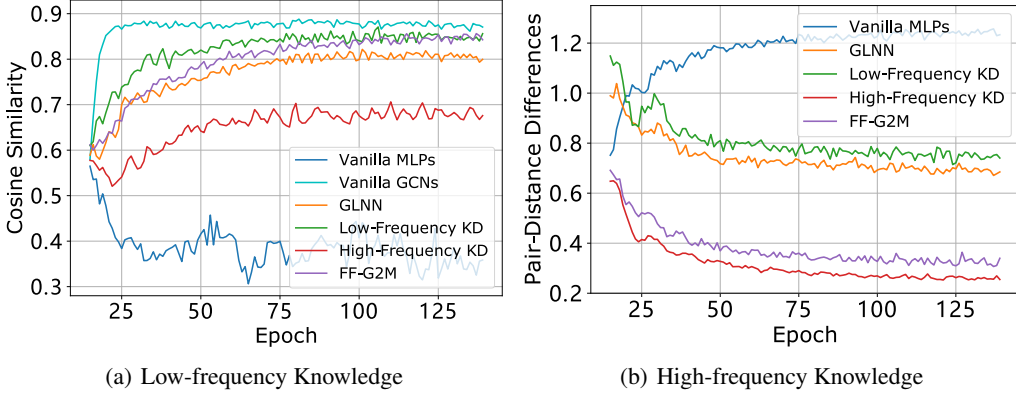
(b) High-frequency Knowledge

Figure 5: (a) Curves of mean cosine similarity (the higher, the better) between nodes with their first-order neighbors. (b) Curves of pairwise distance differences (the lower, the better) between the teacher GNNs and student MLPs.

E. From the experimental results on the Cora dataset reported in Fig. 5, we make four observations: (1) The vanilla MLP does not consider the inductive bias of the graph topology at all and thus fails to capture the low-and high-frequency knowledge in the graph data. (2) GLNN is capable of successfully capturing low-frequency information, i.e., neighborhood smoothing, but is not good at capturing high-frequency knowledge, i.e., difference information between pairs of nodes. (3) The proposed low- and high-frequency distillation has an advantage over GLNN in capturing one type of individual frequency but lags behind in another frequency. (4) The proposed FF-G2M combines both the two distillation and is better at capturing both low- and high-frequency knowledge than GLNN, especially the latter.

Roles of Low- and High-frequency Knowledge

To evaluate the roles played by low- and high-frequency knowledge in GNN-to-MLP distillation, we consider distillation with only $\mathcal{L}_{\mathrm{LFD}}$ and $\mathcal{L}_{\mathrm{HFD}}$, in addition to the full FF-G2M model. The experiments (with GCNs as the teacher model) on six datasets are reported in Table. 2, from which we observe that: (1) The proposed low-frequency distillation $\mathcal{L}_{\mathrm{LFD}}$ makes fuller use of the graph topology and the low-frequency information from GNNs, and in turn outperforms GLNN that adopts node-to-node distillation on all six datasets. (2) While both low- and high-frequency distillation can work alone to improve the performance of vanilla

MLPs, the former plays a *primary* role and the latter a *secondary* (auxiliary) role. More importantly, these two distillations are complementary to each other and can further improve performance on top of each other. (3) The FF-G2M (full model) considers both low- and high-frequency distillation and is capable of capturing full-frequency knowledge, and thus can far outperform GLNN on all six datasets.

Conclusion

In this paper, we factorize the knowledge learned by GNNs into low- and high-frequency components in the spectral and spatial domains and then conduct a comprehensive investigation on their roles played in GNN-to-MLP distillation. Our key finding is existing GNN-to-MLP distillation may suffer from a potential *information drowning* problem, i.e., the high-frequency knowledge of the pre-trained GNNs may be overwhelmed by the low-frequency knowledge during distillation. Therefore, we propose a novel *Full-Frequency GNN-to-MLP* (FF-G2M) knowledge distillation framework, which extracts both low- and high-frequency knowledge from GNNs and injects it into MLPs. As a simple but general framework, FF-G2M outperforms other leading methods across various GNN architectures and graph datasets. Limitations still exist; for example, this paper pays little attention to the special designs on teacher GNNs, and designing more expressive teachers to directly capture full-frequency knowledge may be another promising direction.

Acknowledgments

References

Bo, D.; Wang, X.; Shi, C.; and Shen, H. 2021a. Beyond Low-frequency Information in Graph Convolutional Networks. *arXiv preprint arXiv:2101.00797*.

Bo, D.; Wang, X.; Shi, C.; and Shen, H. 2021b. Beyond Low-frequency Information in Graph Convolutional Networks. *Association for the Advance of Artificial Intelligence (AAAI)*.

Bruna, J.; Zaremba, W.; Szlam, A.; and LeCun, Y. 2013. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*.

Chien, E.; Peng, J.; Li, P.; and Milenkovic, O. 2021. Adaptive Universal Generalized PageRank Graph Neural Network. *International Conference on Learning Representations (ICLR)*.

Chung, F. R.; and Graham, F. C. 1997. *Spectral graph theory*. American Mathematical Soc.

Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. *arXiv preprint arXiv:1606.09375*.

Errica, F.; Podda, M.; Bacciu, D.; and Micheli, A. 2019. A fair comparison of graph neural networks for graph classification. *arXiv preprint arXiv:1912.09893*.

Fan, W.; Ma, Y.; Li, Q.; He, Y.; Zhao, E.; Tang, J.; and Yin, D. 2019. Graph neural networks for social recommendation. In *The world wide web conference*, 417–426.

Ghorbani, M.; Bahrami, M.; Kazi, A.; Soleymani Baghshah, M.; Rabiee, H. R.; and Navab, N. 2021. GKD: Semi-supervised Graph Knowledge Distillation for Graph-Independent Inference. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 709–718. Springer.

Giles, C. L.; Bollacker, K. D.; and Lawrence, S. 1998. CiteSeer: An automatic citation indexing system. In *Proceedings of the third ACM conference on Digital libraries*, 89–98.

Gou, J.; Yu, B.; Maybank, S. J.; and Tao, D. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6): 1789–1819.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. In *Advances in neural information processing systems*, 1024–1034.

Hinton, G.; Vinyals, O.; Dean, J.; et al. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7).

Iscen, A.; Tolias, G.; Avrithis, Y.; and Chum, O. 2019. Label propagation for deep semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5070–5079.

Jia, Z.; Lin, S.; Ying, R.; You, J.; Leskovec, J.; and Aiken, A. 2020. Redundancy-Free Computation for Graph Neural Networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 997–1005.

Joshi, C. K.; Liu, F.; Xun, X.; Lin, J.; and Foo, C.-S. 2021. On Representation Knowledge Distillation for Graph Neural Networks. *arXiv preprint arXiv:2111.04964*.

Kim, T.; Oh, J.; Kim, N.; Cho, S.; and Yun, S.-Y. 2021. Comparing kullback-leibler divergence and mean squared error loss in knowledge distillation. *arXiv preprint arXiv:2105.08919*.

Kipf, T. N.; and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Lassance, C.; Bontonou, M.; Hacene, G. B.; Gripon, V.; Tang, J.; and Ortega, A. 2020. Deep geometric knowledge distillation with graphs. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 8484–8488. IEEE.

Liu, M.; Gao, H.; and Ji, S. 2020. Towards deeper graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 338–348.

Liu, Z.; Luo, Y.; Wu, L.; Li, S.; Liu, Z.; and Li, S. Z. 2022. Are Gradients on Graph Structure Reliable in Gray-box Attacks? *arXiv preprint arXiv:2208.05514*.

McCallum, A. K.; Nigam, K.; Rennie, J.; and Seymore, K. 2000. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3(2): 127–163.

McInnes, L.; Healy, J.; and Melville, J. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.

Pei, H.; Wei, B.; Chang, K. C.-C.; Lei, Y.; and Yang, B. 2020. Geom-gcn: Geometric graph convolutional networks. *arXiv preprint arXiv:2002.05287*.

Ren, Y.; Ji, J.; Niu, L.; and Lei, M. 2021. Multi-task Self-distillation for Graph-based Semi-Supervised Learning. *arXiv preprint arXiv:2112.01174*.

Ricaud, B.; Borgnat, P.; Tremblay, N.; Gonçalves, P.; and Vandergheynst, P. 2019. Fourier could be a data scientist: From graph Fourier transform to signal processing on graphs. *Comptes Rendus Physique*, 20(5): 474–488.

Sandryhaila, A.; and Moura, J. M. 2013. Discrete signal processing on graphs: Graph fourier transform. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 6167–6170. IEEE.

Sen, P.; Namata, G.; Bilgic, M.; Getoor, L.; Galligher, B.; and Eliassi-Rad, T. 2008. Collective classification in network data. *AI magazine*, 29(3): 93–93.

Shchur, O.; Mumme, M.; Bojchevski, A.; and Günnemann, S. 2018. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868*.

Shuman, D. I.; Narang, S. K.; Frossard, P.; Ortega, A.; and Vandergheynst, P. 2013. The emerging field of signal processing on graphs: Extending high-dimensional data analy-

sis to networks and other irregular domains. *IEEE signal processing magazine*, 30(3): 83–98.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Wang, M.; Zheng, D.; Ye, Z.; Gan, Q.; Li, M.; Song, X.; Zhou, J.; Ma, C.; Yu, L.; Gai, Y.; He, T.; Karypis, G.; Li, J.; and Zhang, Z. 2019. Deep Graph Library: A Graph-Centric, Highly-Performant Package for Graph Neural Networks. *arXiv preprint arXiv:1909.01315*.

Wu, F.; Souza, A.; Zhang, T.; Fifty, C.; Yu, T.; and Weinberger, K. 2019. Simplifying graph convolutional networks. In *International conference on machine learning*, 6861–6871. PMLR.

Wu, L.; Lin, H.; Gao, Z.; Tan, C.; Li, S.; et al. 2021a. GraphMixup: Improving Class-Imbalanced Node Classification on Graphs by Self-supervised Context Prediction. *arXiv preprint arXiv:2106.11133*.

Wu, L.; Lin, H.; Gao, Z.; Tan, C.; Li, S.; et al. 2021b. Self-supervised on Graphs: Contrastive, Generative, or Predictive. *arXiv preprint arXiv:2105.07342*.

Wu, L.; Lin, H.; Huang, Y.; and Li, S. Z. 2022a. Knowledge Distillation Improves Graph Structure Augmentation for Graph Neural Networks. In Oh, A. H.; Agarwal, A.; Belgrave, D.; and Cho, K., eds., *Advances in Neural Information Processing Systems*.

Wu, L.; Lin, H.; Xia, J.; Tan, C.; and Li, S. Z. 2022b. Multi-level disentanglement graph neural network. *Neural Computing and Applications*, 1–15.

Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; and Philip, S. Y. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*.

Xia, J.; Wu, L.; Chen, J.; Hu, B.; and Li, S. Z. 2022. SimGRACE: A Simple Framework for Graph Contrastive Learning without Data Augmentation. In *Proceedings of the ACM Web Conference 2022*, 1070–1079.

Yan, B.; Wang, C.; Guo, G.; and Lou, Y. 2020. TinyGNN: Learning Efficient Graph Neural Networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1848–1856.

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 *Proceedings of the Web Conference 2021*, 1227–1237.

Zhang, M.; and Chen, Y. 2018. Link prediction based on graph neural networks. *arXiv preprint arXiv:1802.09691*.

Zhang, S.; Liu, Y.; Sun, Y.; and Shah, N. 2021. Graph-less Neural Networks: Teaching Old MLPs New Tricks via Distillation. *arXiv preprint arXiv:2110.08727*.

Zhang, W.; Miao, X.; Shao, Y.; Jiang, J.; Chen, L.; Ruas, O.; and Cui, B. 2020. Reliable data distillation on graph convolutional network. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, 1399–1414.

Zhou, J.; Cui, G.; Hu, S.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; and Sun, M. 2020. Graph neural networks: A review of methods and applications. *AI Open*, 1: 57–81.

Zhu, J.; Yan, Y.; Zhao, L.; Heimann, M.; Akoglu, L.; and Koutra, D. 2020. Beyond Homophily in Graph Neural Networks: Current Limitations and Effective Designs. *Advances in Neural Information Processing Systems*, 33.