Demystifying GNN-to-MLP Knowledge Transfer: Theoretical Grounding and Dual-Stream Distillation Method

Zhiyuan Yu[1], Mingkai Lin[1*], Wenzhong Li[1*], Zhangyue Yin[2], Shijian Xiao[1], Sanglu Lu[1]

[1]State Key Laboratory for Novel Software Technology, Nanjing University
[2]School of Computer Science, Fudan University
{zhiyuan_yu, mingkai, xiaoshijian}@smail.nju.edu.cn, {zhangyueyin}@m.fudan.edu.cn, {lwz, sanglu}@nju.edu.cn

Abstract

Graph Neural Networks (GNNs) have shown remarkable effectiveness across various applications, but their computational complexity poses significant scalability challenges. To this end, GNN-to-MLP Knowledge Distillation (KD) methods transfer relational inductive biases from GNNs to MLPs, equipping MLPs with graph-aware capabilities that rival or even surpass those of their teacher GNNs. However, a theoretical foundation for understanding GNN-to-MLP KD is still missing. In this paper, we provide a theoretical analysis of how knowledge distillation unlocks the potential of MLPs for graph tasks from the perspective of training dynamics. We demonstrate that label alignment in KD fundamentally reshapes the Neural Tangent Kernel (NTK) matrix of student MLPs, enabling them to learn the teacher model's implicit graph bias. We further investigate finer-grained distillation paradigms and reveal that conventional layer-wise output alignment fails to effectively align the deep-layer graph propagation outcomes. To address this, we propose **D**ual-Stream **A**ligned **MLP** (DA-MLP), which incorporates complementary graph filters in a dual-stream architecture. This approach simultaneously enhances feature space dimensionality for improved representation alignment and preserves graph signals across different frequency bands. Comprehensive experiments on seven benchmark datasets validate that DA-MLP can be seamlessly integrated into existing knowledge distillation frameworks for performance enhancements in both transductive and inductive settings.

Introduction

Graph Neural Networks (GNNs) (Kipf and Welling 2017; Keriven and Peyré 2019; Wu et al. 2021; Lin et al. 2025) excel at capturing relational dependencies in graph-based data through message-passing mechanisms, enabling applications in diverse fields such as chemical compounds (Guo et al. 2022) and social networks (Li et al. 2018). However, the requirement for aggregating information from neighboring nodes during inference often results in high computational costs and limited scalability (Jia et al. 2020; Zhang et al. 2020; Lin et al. 2022), which makes GNNs difficult to deploy for latency-constrained applications that require fast inference.

To address these challenges, *GNN-to-MLP Knowledge Distillation* (KD) has emerged as an effective approach that combines the relational modeling capabilities of GNNs with the computational efficiency of Multi-Layer Perceptrons (MLPs) (Rumelhart, Hinton, and Williams 1988). Considering that MLPs are highly efficient and scalable due to their simpler structure, GNN-to-MLP knowledge distillation aims to empower MLPs to emulate the relational understanding and predictive abilities of GNNs. In this framework, a pre-trained GNN serves as the teacher model, guiding the student MLP to replicate its predictive behavior (Hinton, Vinyals, and Dean 2015b). This process allows MLPs to implicitly learn graph-aware representations without requiring explicit graph structures during inference. While Graph-less Neural Networks (GLNNs) (Zhang et al. 2022) pioneered standard distillation for GNN-to-MLP transfer, they exhibited limitations in capturing structural patterns. Subsequent advances have introduced either enhanced architectures (e.g., VQGraph (Yang et al. 2024), NOSMOG (Tian et al. 2022)) or optimized distillation protocols (e.g., KRD (Wu et al. 2023), AdaGMLP (Lu et al. 2024)) to improve MLPs' structural pattern recognition capabilities.

While existing studies have empirically demonstrated that knowledge distillation enables MLPs to achieve enhanced generalization performance on graph data by transferring knowledge from GNNs, the fundamental mechanisms underlying this improvement remain poorly understood. To fill this theoretical gap, we conduct a systematic investigation into the training dynamics in GNN-to-MLP distillation. Our analysis reveals that: (1) In standard GNN training, the graph structure inherently reshapes the model's Neural Tangent Kernel (NTK) (Jacot, Gabriel, and Hongler 2018; Du et al. 2019a; Kawaguchi and Huang 2019), facilitating more effective function updates; (2) In the distillation setting, student MLPs can dynamically adapt their feature attention patterns through the distillation objectives, which promotes better alignment between their NTK and the underlying graph structure, thus effectively enabling implicit graph propagation capabilities without explicit graph inputs.

Moreover, our theoretical analysis reveals that this feature attention adaptation exhibits limited robustness when handling diverse graph connectivity patterns (Abu-El-Haija et al. 2019; Jin et al. 2021), and does not consistently yield performance improvements. We identify the root cause of the issue as the inherent limitations of coarse-grained label alignment in conventional KD frameworks. We demonstrate that incor-

*corresponding author
Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

porating a fine-grained distillation process can effectively alleviate this problem. To develop a more principled understanding, we conduct a comprehensive theoretical analysis of Feature Transformation (FT) approximation. Our study derives the theoretical bounds on layer-wise approximation errors when feature transformation is used to simulate Graph Propagation (GP) operations. Furthermore, we show that these approximation errors accumulate across layers, leading to a significant mismatch in the deeper layers, which ultimately limits FT's ability to accurately align the outcomes of multi-layer graph propagation.

Based on these findings, we propose a Dual-stream Aligned MLP (**DA-MLP**) framework that simultaneously models and approximates complementary graph filters to facilitate effective fine-grained knowledge distillation. The architecture consists of two branches: the primary branch simulates the teacher's graph signal filtering behavior, while the complementary branch implicitly generates orthogonal filtered outputs through an orthogonality-constrained loss. This dual-stream design theoretically guarantees improved layer-wise alignment, while preserving diverse graph signal characteristics to better accommodate heterogeneous connectivity patterns. We evaluate DA-MLP on seven real-world datasets, demonstrating that it can be seamlessly integrated into existing GNN-to-MLP distillation frameworks and significantly boost their performance on GNN tasks.

The contributions of our work are summarized as follows.

- We establish the first theoretical framework explaining why distillation improves MLPs' graph reasoning capabilities and delve into fine-grained distillation approaches.
- We reveal that GNN-derived supervision dynamically aligns student MLPs' Neural Tangent Kernel (NTK) with graph structures to enable implicit propagation, and fine-grained feature transformation approximation may suffer from error accumulation issues across layers.
- We propose the DA-MLP framework, which employs a dual-stream distillation method for simultaneously reducing approximation error and preserving multi-band graph signals.DA-MLP can be seamlessly integrated into existing GNN-to-MLP frameworks to boost performance.
- Comprehensive experiments across seven benchmark datasets demonstrate DA-MLP's consistent performance gains in both transductive and inductive settings, as well as its robustness to diverse graph connectivity patterns and teacher GNNs.

Preliminary

Graph Neural Networks

We represent the graph-structured data as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, ..., v_n\}$ is the node set and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the edge set. Each node is associated with node attributes $\mathbf{x}_i \in \mathbb{R}^d$ and a class $y_i \in \{1, ..., C\}$, with C being the total number of classes. Additionally, each graph has an adjacency matrix $\mathbf{A} \in \{0, 1\}^{n \times n}$, where $\mathbf{A}_{ij} = 1$ iff $(i, j) \in \mathcal{E}$, otherwise $\mathbf{A}_{ij} = 0$. Generally, GNN models follow a neighborhood aggregation mechanism called message passing, where node representations are updated by aggregating information from their neighboring nodes. Let $\mathbf{z}_i^{(l)}$ denote the output vector of node v_i at the l-th hidden layer and $\mathbf{z}_i^{(0)}$ the original feature. A GNN layer consists of both graph propagation (GP) and feature transformation (FT) operations:

$$\begin{aligned} \mathrm{GP}^{(l)} &: \tilde{\mathbf{z}}_v^{(l)} = \gamma\left(\mathbf{z}_v^{(l-1)}, \Gamma^{(l)}\left(\{\mathbf{z}_u^{(l-1)}, \forall u \in \mathcal{N}(v)\}\right)\right), \\ \mathrm{FT}^{(l)} &: \mathbf{z}_v^{(l)} = \sigma\left(\tilde{\mathbf{z}}_v^{(l)} \cdot \mathbf{W}^{(l)} + \mathbf{b}^{(l)}\right). \end{aligned} \tag{1}$$

where $\mathbb{N}_{v_i}$ is the set of neighbors of v_i. The Γ denotes permutation invariant aggregation operation, e.g. mean, and the γ is a pooling function (Xu et al. 2019) further fuses the information from neighbors and the central node.

Spectral Graph Theory

Given an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with adjacency matrix $\mathbf{A}$ and degree matrix $\mathbf{D} = \mathrm{diag}(d_1, \ldots, d_n)$ where $d_i = \sum_j \mathbf{A}_{ij}$, the *unnormalized Laplacian* is defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}$. Its normalized counterpart, the *symmetric normalized Laplacian*, is given by: $\mathbf{L}_{\text{sym}} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$. As $\mathbf{L}$ is symmetric positive semi-definite, it admits an eigendecomposition: $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$, where $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_n]$ are the eigenvectors and $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ contains the ordered eigenvalues ($0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$). The eigenvalues λ_i correspond to frequencies, with smaller values indicating smoother (lower-frequency) eigenmodes $\mathbf{u}_i$.

A spectral filter g_θ applied to graph signal $\mathbf{x} \in \mathbb{R}^n$ (representing vertex attributes) operates as:

$$g_\theta * \mathbf{x} = \mathbf{U} g_\theta(\mathbf{\Lambda}) \mathbf{U}^\top \mathbf{x}, \tag{2}$$

where $g_\theta(\mathbf{\Lambda})$ is a frequency-response function (e.g., polynomial or exponential) parameterized by θ, which determines how different spectral components are scaled. The widely adopted GCN simplifies this via a first-order approximation with renormalization:

$$\mathbf{Z} = \widetilde{\mathbf{D}}^{-1/2}\widetilde{\mathbf{A}}\widetilde{\mathbf{D}}^{-1/2}\mathbf{X}, \tag{3}$$

where $\widetilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ and $\widetilde{\mathbf{D}}$ is its degree matrix. This formulation bridges spectral and spatial approaches efficiently. The operation in Equation (3) acts as a low-pass filter, reducing the *Dirichlet energy* of node features:

$$E_{\text{Dir}}(\mathbf{x}) = \frac{1}{2}\sum_{i,j} \mathbf{A}_{ij} \|\mathbf{x}_i - \mathbf{x}_j\|^2 = \mathbf{x}^\top \mathbf{L} \mathbf{x}. \tag{4}$$

Minimizing E_{Dir} encourages signal smoothness by attenuating high-frequency components that associated with larger eigenvalues of $\mathbf{\Lambda}$.

Knowledge Distillation

Knowledge distillation (Hinton, Vinyals, and Dean 2015a; Gou et al. 2021) is a model compression paradigm that transfers dark knowledge from a computationally intensive teacher model $f_T(\mathbf{x}; \theta_T)$ to a compact student model $f_S(\mathbf{x}; \theta_S)$. The core mechanism involves matching softened probability distributions through a temperature-scaled softmax operation. The distillation loss is formulated as the Kullback-Leibler divergence (Kullback and Leibler 1951):

$$\mathcal{L}_{KD} = \tau^2 \cdot \mathrm{KL}\left(\sigma(\mathbf{z}_T/\tau) \parallel \sigma(\mathbf{z}_S/\tau)\right). \tag{5}$$

Simultaneously, the student maintains discriminative power through standard cross-entropy loss $\mathcal{L}_{CE}$ with ground truth labels $\mathbf{y}$. The complete optimization objective combines both losses through a balancing coefficient $\alpha \in [0, 1]$:

$$\mathcal{L} = \alpha \mathcal{L}_{KD} + (1 - \alpha)\mathcal{L}_{CE}. \tag{6}$$

This hybrid objective enables the student to simultaneously preserve the teacher's relational knowledge and ground-truth supervision.

Demystifying GNN-MLP Transfer

Towards theoretically answering why GNN-to-MLP knowledge distillation unlocks MLPs' potential for graph machine learning tasks, we exploit the lens of Neural Tangent Kernel (NTK) (Jacot, Gabriel, and Hongler 2018; Du et al. 2019a; Kawaguchi and Huang 2019). Specifically, we theoretically prove that knowledge distillation makes student MLPs exhibit preferential attention to smooth attributes, which further enables them to achieve implicit graph propagation even without explicit access to the graph structure. Note that although we mainly focuses on the properties of MLPs during the training phase, our conclusions can also explain their behavior at test-time, including both inductive and transductive settings.

Training Dynamics Analysis

The Neural Tangent Kernel (NTK) theory provides a rigorous mathematical framework to characterize the learning dynamics of deep neural networks under gradient descent optimization. Consider neural network model $f(\mathbf{X}; \theta) : \mathbb{R}^{n\times m} \to \mathbb{R}$ with parameters $\theta \in \mathbb{R}^d$, where $\mathbf{X} \in \mathbb{R}^{n\times m}$ denotes the input data matrix, the NTK is formally defined as the inner product of parameter gradient vectors:

$$\Theta(\mathbf{X}, \mathbf{X}'; \theta) = \langle \nabla_\theta f(\mathbf{X}; \theta), \nabla_\theta f(\mathbf{X}'; \theta) \rangle_{\mathbb{R}^d} \tag{7}$$

$$= \sum_{k=1}^{d} \frac{\partial f(\mathbf{X}; \theta)}{\partial \theta_k} \frac{\partial f(\mathbf{X}'; \theta)}{\partial \theta_k}, \tag{8}$$

where the kernel function $\Theta : \mathbb{R}^{n\times m} \times \mathbb{R}^{n\times m} \to \mathbb{R}^{n\times n}$ quantifies the sensitivity correlation between input pairs $(\mathbf{X}, \mathbf{X}')$ in function space. Intuitively, when two inputs induce similar gradient descent directions in parameter space, their kernel value is correspondingly amplified.

In the standard supervised learning setup trained with mean squared error (MSE) loss, we optimize the model parameters θ using gradient descent (GD) with learning rate η. With the continuous-time approximation (when the step size becomes infinite small), the gradient descent dynamics can be described by an ordinary differential equation (ODE):

$$\frac{\partial \theta_t}{\partial t} = -\eta \nabla_\theta \mathcal{L}(\theta_t). \tag{9}$$

Here the subscript t denotes the state of parameters θ at (approximate) continuous time t. To study how the network's predictions $f_t(\mathbf{X})$ evolve with time t, one can compute the time derivative of $f_t(\mathbf{X})$, and obtain the following equation:

$$\frac{\partial f_t(\mathbf{X})}{\partial t} = \frac{\eta}{n} \left[\nabla_\theta f_t(\mathbf{X})^T \nabla_\theta f_t(\mathbf{X})\right] (\mathbf{Y} - f_t(\mathbf{X})). \tag{10}$$

Observe that $\nabla_\theta f_t(\mathbf{X})^T \nabla_\theta f_t(\mathbf{X})$ is aligned with the definition in Eq. (11). Using $\Theta_t(\mathbf{X}, \mathbf{X})$ to denote the NTK matrix of the model at time step t, and absorb $1/n$ into the learning rate η, we have:

$$\frac{\partial f_t(\mathbf{X})}{\partial t} \approx -\eta \Theta_t(\mathbf{X}, \mathbf{X})(f_t(\mathbf{X}) - \mathbf{Y}). \tag{11}$$

The equation characterizes the evolution of a general parameterized model. When network width tends to infinity, $\Theta_t(\mathbf{X}, \mathbf{X})$ converges almost surely to a deterministic kernel $\Theta^\infty(\mathbf{X}, \mathbf{X})$, and remains essentially constant throughout training independent of time step t (Lee et al. 2019).

Similar to the training dynamics of general parameterized models, the training dynamics of GNNs in node-level tasks in the overparameterized regime can also be fromulated. For an ℓ-layer GNN with ReLU activation:

$$\mathbf{Z}^{(\ell)} = \mathrm{ReLU}(\mathbf{A}\mathbf{Z}^{(\ell-1)}\mathbf{W}^{(\ell)}), \tag{12}$$

the *node-level Graph Neural Tangent Kernel (GNTK)* (Du et al. 2019b) admits a recursive computation:

$$\overline{\Theta}^{(\ell)} = \Theta^{(\ell-1)} \odot \dot{\Sigma}^{(\ell)} + \overline{\Sigma}^{(\ell)}, \quad \Theta^{(\ell)} = \mathbf{A}\overline{\Theta}^{(\ell)}\mathbf{A}, \tag{13}$$

where $\overline{\Sigma}^{(\ell)}$ and $\dot{\Sigma}^{(\ell)}$ are covariance matrices of layer outputs and derivatives. The adjacency matrix $\mathbf{A}$ directly propagates through the kernel, for example, in a linear GNN with $f(\mathbf{X}; \mathbf{A}) = \mathbf{A}^\ell \mathbf{X}\mathbf{W}$, the NTK simplifies to:

$$\Theta^{(\ell)} = \mathbf{A}^\ell \mathbf{X}(\mathbf{A}^\ell \mathbf{X})^\top = \mathbf{A}^{2\ell} \quad (\text{for } \mathbf{X} = I_n). \tag{14}$$

This confirms that graph structure is hardwired into the NTK. Therefore, from the perspective of training dynamics, we can also observe how GNNs leverage graph implicit bias.

For inductive setting analysis, we can further extend Eq. (11) to accommodate unseen samples, and rewrite it for an arbitrary unseen data point x' as follows:

$$f_{t+1}(\boldsymbol{x}') = f_t(\boldsymbol{x}') + \eta \sum_{\boldsymbol{x}\in\mathbf{X}} \Theta_t(\boldsymbol{x}, \boldsymbol{x}')(y(\boldsymbol{x}) - f_t(\boldsymbol{x})), \tag{15}$$

where $y(\mathbf{x})$ is the ground-truth label for $\mathbf{x}$, and the similarity measure between instances is defined with NTK. Intuitively, for an unseen instance $\mathbf{x}'$ that is more "similar" to $\mathbf{x}$, more ground-truth label information $y(\mathbf{x})$ will propagate to $\mathbf{x}'$, and vice versa. Given the relationship between Θ and $\mathbf{A}$ in Eq. (14), it essentially represents a special form of label propagation.

Distillation Scenario Analysis

The application of Neural Tangent Kernels to GNN-to-MLP knowledge distillation poses unique theoretical challenges. Since structure-agnostic MLPs inherently lack explicit graph-structure modeling capabilities, their NTKs cannot directly incorporate topological information from the input graph. This fundamental discrepancy prevents direct transfer of previous NTK-based analysis frameworks to the distillation setting. To enable further theoretical analysis, we dispense with the infinite-width assumption and introduce parameter-space constraints, yielding an explicitly derivable NTK matrix independent of initialization.

We first demonstrate that through GNN-to-MLP knowledge distillation, the student model exhibits a distinct preference for smooth features, which is formulated as follows:

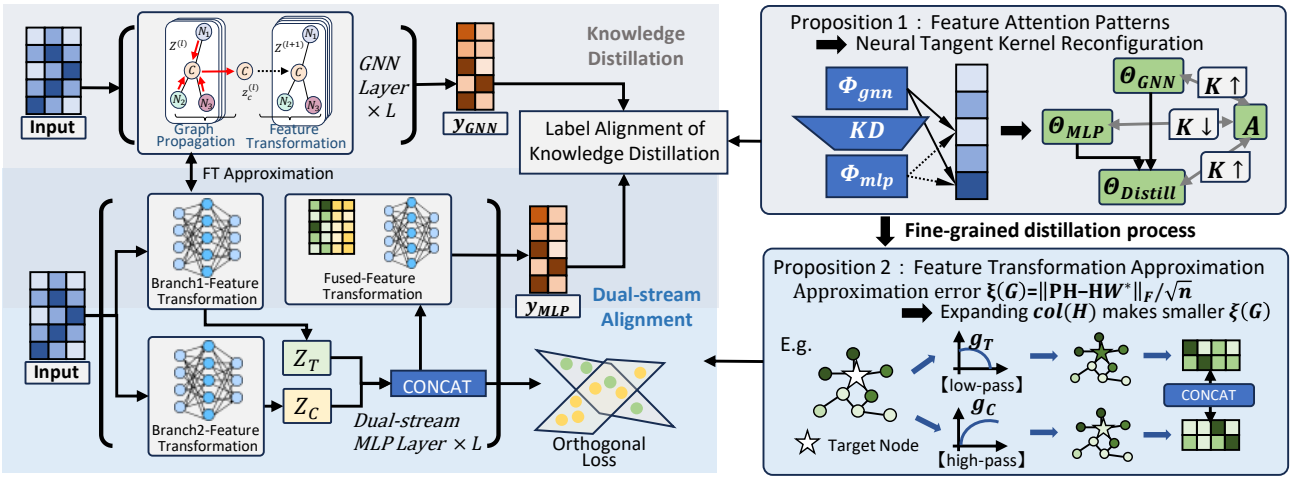

Figure 1: Framework of this paper: Theoretical Grounding (RIGHT) and Dual-Stream Distillation Method (LEFT).

Lemma 1 (Preferential Attention to Smooth Attributes)**.** *Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with input feature matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ and adjacency matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, consider a regression task (i.e., output being scalar) where we have a teacher GNN with mean-pooling aggregation and a student MLP. The student is trained using the distillation loss:*

$$\mathcal{L}_{distill} = \frac{1}{2}\|f_{teacher}(\mathbf{X}, \mathbf{A}) - f_{student}(\mathbf{X})\|^2. \quad (16)$$

Under the assumptions that node features are zero-mean $\mathbb{E}[\mathbf{x}] = 0$ with independent dimensions $Cov(\mathbf{x}_{:i}, \mathbf{x}_{:j}) = 0$ for $i \neq j$, the expected sensitivity of well-trained student's output to feature perturbation satisfies:

$$\mathbb{E}_{v\in\mathcal{V}}\left[\left|\frac{\partial f_{student}(\mathbf{x}_v)}{\partial \mathbf{x}_{v,i}}\right|\right] \propto \frac{1}{E_{Dir}(\mathbf{X}_{[:i]})}, \quad (17)$$

where $E_{Dir}(\mathbf{X}_{[:i]})$ follows the definition in Eq. (4).

Based on the established lemma, we can further derive the following proposition:

Proposition 1. *Let f_{KD} denote the knowledge-distilled MLP from a GNN teacher and f_{MLP} denote the standard MLP without distillation. Let $\Theta^{(f)}$ be the Neural Tangent Kernel matrix of model f. Define the alignment metric between the graph adjacency matrix $\mathbf{A}$ and NTK as:*

$$\kappa(\mathbf{A}, \Theta) \triangleq \frac{\langle \mathbf{A}, \Theta\rangle_F}{\|\mathbf{A}\|_F\|\Theta\|_F}, \quad (18)$$

where $\langle\cdot,\cdot\rangle_F$ denotes the Frobenius inner product and $\|\cdot\|_F$ the Frobenius norm. From Lemma 1, we know that f_{KD} satisfies the property given in Eq. (17). If f_{MLP} fails to satisfy this correlation property, then the distilled model will achieve higher structural alignment with:

$$\kappa(\mathbf{A}, \Theta^{(f_{KD})}) > \kappa(\mathbf{A}, \Theta^{(f_{MLP})}). \quad (19)$$

This proposition suggests that as long as the MLP can adapt its feature preference pattern towards smooth attributes, even without explicitly utilizing graph structural information, its NTK-based propagation in Eq. (15) can consistently exhibit a pronounced propensity to occur preferentially between adjacent nodes, thereby creating an implicit graph propagation effect similar to that in GNNs.

Extensive real-world experiments provide empirical evidence that: (i) After distilling knowledge from GNNs, MLPs exhibit a significantly stronger preference for smooth attributes. (ii) The NTK of distilled MLP demonstrates progressive alignment with the graph adjacency matrix $\mathbf{A}$ during the distillation process.

Dual-Stream Distillation Method

Although label alignment in knowledge distillation helps MLPs learn graph-implicit bias, we identify its inherent limitation of misaligned feature preference where models may prioritize non-discriminative attributes. Our analysis reveals that finer-grained distillation patterns can effectively mitigate this issue. To achieve such refined alignment, we first theoretically establish the feasibility of layer-wise feature transformation for approximating graph propagation. We then propose a novel dual-stream architecture to optimize the distillation process. Both our theoretical analysis and framework design are comprehensively illustrated in Figure 1.

Misaligned Feature Preference

As established in prior work (Wu et al. 2020; Bei et al. 2025), shallow GNN layers with limited receptive fields primarily focus on node-specific attributes, while deeper layers emphasize label propagation. Conventional knowledge distillation only aligns final logits, which may lead to the student MLP's incorrect handling of the original features.

Specifically in empirical studies, we observed the phenomenon of misaligned feature preferences. As illustrated in Figure 2 (LEFT), our experiments reveal that teacher GNNs trained on certain datasets (Chameleon, Squirrel, and Pubmed) exhibit a strong preference for non-smooth features. However, the MLPs distilled knowledges from these teachers (KD-MLP) demonstrate an opposite tendency and

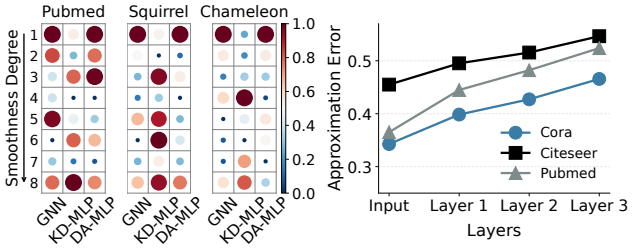

Figure 2: **(LEFT)** We partition feature set into eight bins based on their smoothness levels (with larger number indicates smoother feature) and visualize their attention weights. It shows that coarse-grained distillation (KD-MLP) leads to misaligned feature preference, and finer-grained distillation process (proposed DA-MLP) can mitigate this issue. **(RIGHT)** Normalized approximation error for hidden representations, which grows progressively with deeper layers.

primarily focus on smooth features. This discrepancy can be explained by Proposition 1, which suggests that due to label alignment objective, the student model is compelled to adopt the teacher's graph-implicit bias, even at the cost of abandoning its own potentially superior feature preference pattern. This forced adaptation ultimately leads to student's consistent neglect of non-smooth features, resulting in diminished robustness when handling diverse graph connectivity patterns.

To prevent the student model from over-relying on labels provided by the teacher, we need a more fine-grained distillation process. To develop a more principled understanding for it, we first theoretically investigate whether MLPs can progressively learn GNN knowledge through layer-wise imitation. GNNs inherently integrate both GP (graph propagation) and FT (feature transformation) for information processing, whereas structure-agnostic MLPs rely exclusively on FT. Consequently, fine-grained guidance necessitates approximating GP through FT approximation.

Feature Transformation Approximation

To analyze the approximation effectiveness, we employ $\xi(\mathcal{G})$ to quantify the approximation error as follows:

Definition 1 (*FT Approximation Error*)**.** *Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with its normalized feature $\mathbf{H} \in \mathbb{R}^{n\times d}$, the optimal approximation error for an* FT *operation parameterized by $\mathbf{W}$ to approximate* GP *in Eq. (1) can be formulated as:*

$$\xi(\mathcal{G}) = \inf_{\mathbf{W}\in\mathbb{R}^{d\times d}} \frac{1}{\sqrt{n}}\|\mathbf{PH} - \mathbf{HW}\|_F, \tag{20}$$

where $\|\cdot\|_F^2$ is the Frobenius norm, and $\mathbf{P}$ is the graph propagation matrix that depends on the graph structure $\mathbf{A}$ as well as the choice of GNN.

Assuming that $\mathbf{P}$ is a graph signal filter in the form of Eq. (2), we can derive the approximation bound as follows:

Proposition 2 (Approximation Bounds for Graph Filtering)**.** *Consider a graph $\mathcal{G}$ with Laplacian $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ and graph filter $\mathbf{P} = \mathbf{U}g_\theta(\mathbf{\Lambda})\mathbf{U}^\top$, where $g_\theta : \mathbb{R} \to \mathbb{R}$ is a spectral filter function parameterized by θ, and maps each eigenvalue λ_* in Λ to $g_\theta(\lambda_*)$. For feature matrix $\mathbf{H} \in \mathbb{R}^{n\times d}$ with $\mathrm{rank}(\mathbf{H}) = d$ and $\|\mathbf{h}_i\|_2 = 1,\ \forall i \in \{1, ..., n\}$, the FT approximation error is bounded by:*

$$\xi(\mathcal{G}) \le \sqrt{\sum_{i=d+1}^{n} g_\theta(\lambda_i)^2}, \tag{21}$$

where $g_\theta(\lambda_*)$ is ranked with $|g_\theta(\lambda_1)| \le \cdots \le |g_\theta(\lambda_n)|$.

It can be concluded that the approximation effect gradually improves with the increasement of the rank of $\mathbf{H}$. Furthermore, the exact approximation $\xi(\mathcal{G}) = 0$ holds if and only if the column space of $\mathbf{H}$ is spanned by d eigenvectors of $\mathbf{L}$ corresponding to the non-zero eigenvalues of $g_\theta(\mathbf{\Lambda})$.

As can be concluded from the above proposition, feature transformation approximation is not perfect. This is because graph filters always amplify certain frequency components of graph signals, leading to (21) becoming a relatively loose upper bound for the approximation error. Furthermore, in real-world experiments, we observe that initial features are amenable to FT approximation, but such approximation becomes increasingly difficult with deeper GNN layers, as illustrated in Figure 2 (RIGHT). This implies that more principled designs for intermediate-layer alignment are needed to achieve fine-grained distillation.

Dual-Stream Distillation Method

To address the growth of approximation error, we propose **D**ual-Stream **A**ligned **MLP** (DA-MLP), which uses dual MLP branches to approximate both teacher-related and complementary graph filtering operations, thereby expanding feature spaces while preserving multi-band graph signals.

The core insight stems from Proposition 2: expanding $\mathbf{Z}$'s column space enhances approximation fidelity. DA-MLP achieves this through dual MLP branches, with one simulating the teacher GNN's graph signal filter $\mathbf{g}_T$ and another learning a complementary filter $\mathbf{g}_C$. Consider the case where $\mathbf{g}_T$ operates as a low-pass filter preserving the first k lowest-frequency graph signals. The complementary filter $\mathbf{g}_C$ would complementarily retain the remaining $n - k$ high-frequency components, formally expressed as follows:

$$\mathbf{g}_T : \mathbf{Z} \mapsto \mathrm{span}\{\mathbf{v}_1, \ldots, \mathbf{v}_k\},$$
$$\mathbf{g}_C : \mathbf{Z} \mapsto \mathrm{span}\{\mathbf{v}_{k+1}, \ldots, \mathbf{v}_n\},$$

where $\mathbf{v}_i$ are Laplacian eigenvectors ordered by eigenvalue magnitude. The concatenated outputs $[\mathbf{g}_T * \mathbf{z}^{(l)} \parallel \mathbf{g}_C * \mathbf{z}^{(l)}]$ then theoretically expand $\mathbf{Z}$'s column space as input of next layer. Furthermore, they enable explicit separation of graph signals across spectral bands. This contrasts with traditional graph propagation methods, which typically overlay only partial frequency components.

Note that the filter $\mathbf{g}_T$ corresponds to the teacher GNN's propagation mechanism, which can be obtained whether it is non-parametric (e.g., GCN's adjacency normalization) or parametric (e.g., GAT's attention weights). On the other hand, the complementary filter $\mathbf{g}_C$ is typically non-trivial to derive explicitly. As a result, we employ an implicit alignment

Dataset	Cora	CiteSeer	Pubmed	Photo	CS	Physics	ArXiv	*Improv.*
Models				**Transductive Setting**				
MLP	57.14±2.49	56.78±2.08	68.88±3.71	77.29±1.79	86.44±0.37	88.47±0.11	53.28±1.24	-
GraphSAGE	81.60±0.39	70.56±1.34	78.56±1.29	89.87±0.16	90.74±0.02	91.88±0.22	70.87±0.47	-
GLNN	81.20±0.16	70.78±0.37	78.46±0.25	89.73±2.00	91.49±0.66	92.08±0.24	69.54±0.92	-
+DA-MLP	**82.60±0.41**	**72.39±1.05**	**80.70±1.26**	**91.80±1.31**	**92.82±0.47**	**93.25±0.04**	**71.21±0.66**	**2.04**
KRD	83.75±0.58	72.33±0.58	80.66±0.92	91.01±2.41	92.82±0.37	92.90±0.16	70.57±0.41	-
+DA-MLP	**84.28±1.70**	**73.16±1.33**	**81.94±0.34**	**92.26±2.05**	**93.88±0.14**	**93.53±0.25**	**71.32±0.39**	**1.09**
VQGraph	82.10±0.55	73.50±0.30	81.35±0.87	91.27±1.25	92.98±0.24	93.44±0.07	71.83±0.54	-
+DA-MLP	**83.63±0.26**	**75.02±0.47**	**82.02±0.53**	**92.81±3.58**	**93.29±0.45**	**94.31±0.14**	**72.21±0.29**	**1.18**
AdaGMLP	82.32±1.71	72.64±0.22	81.52±0.85	91.43±0.57	93.57±0.38	93.06±0.04	71.78±0.68	-
+DA-MLP	**82.63±0.40**	**74.74±0.13**	**82.25±0.37**	**93.55±2.24**	**94.26±0.35**	**93.93±0.17**	**72.34±0.25**	**1.27**
Models				**Inductive Setting**				
MLP	60.30±1.40	57.69±3.37	70.08±2.29	76.10±1.66	86.48±2.34	89.62±0.82	56.97±1.15	-
GraphSAGE	78.98±0.74	71.32±1.02	78.11±0.85	88.41±2.84	89.60±0.19	91.85±0.08	70.36±0.57	-
GLNN	70.92±0.96	70.34±0.30	79.12±1.01	89.57±1.47	91.51±0.75	91.93±0.13	62.21±0.49	-
+DA-MLP	**73.32±0.66**	**70.72±0.07**	**81.06±1.13**	**90.84±0.81**	**93.32±0.73**	**93.07±0.17**	**64.19±0.37**	**2.02**
KRD	70.80±1.42	70.67±0.19	81.82±1.26	90.21±0.54	91.52±0.34	92.60±0.76	62.19±0.26	-
+DA-MLP	**73.36±0.88**	**70.83±0.17**	**82.19±2.06**	**91.37±0.21**	**92.81±0.13**	**93.72±0.41**	**63.23±0.14**	**1.41**
VQGraph	72.88±1.70	70.41±0.33	81.04±1.02	89.60±2.28	92.35±0.93	93.16±0.12	68.24±0.64	-
+DA-MLP	**74.70±2.60**	**70.87±0.52**	**83.11±0.90**	**91.13±0.48**	**93.19±1.07**	**93.85±0.24**	**68.66±0.57**	**1.38**
AdaGMLP	73.47±1.67	70.17±0.21	81.34±0.25	90.86±0.37	92.16±1.10	93.07±1.26	65.18±0.17	-
+DA-MLP	**73.64±1.47**	**70.82±0.32**	**83.60±0.78**	**91.75±0.35**	**93.55±2.24**	**93.88±0.29**	**67.32±0.41**	**1.51**

Table 1: Classification Accuracy ± std (%) for both Transductive Setting and Inductive Setting with DA-MLP's Improvements.

strategy via the incorporation of orthogonal loss. Specifically within our dual-stream framework, the filtered outputs $\mathbf{g}_T * \mathbf{z}^{(l)}$ and $\mathbf{g}_C * \mathbf{z}^{(l)}$ are simulated via branch-specific feature transformations of:

$$\tilde{\mathbf{Z}}_T^{(l)} = \mathbf{Z}^{(l)}\mathbf{W}_T^{(l)}, \quad \tilde{\mathbf{Z}}_C^{(l)} = \mathbf{Z}^{(l)}\mathbf{W}_C^{(l)}, \tag{22}$$

which are parameterized by weight matrices $\mathbf{W}_T^{(l)}$ and $\mathbf{W}_C^{(l)}$ respectively. To achieve it, we introduce two loss functions: the **Simulation Loss** ensures accurate approximation of the teacher's graph propagation with:

$$\mathcal{L}_{\text{DIST}} = \sum_{l=1}^{L}\sum_{i=1}^{n} \mathcal{D}(\mathbf{g}_T * \mathbf{z}_i^{(l)}, \mathbf{z}_i^{(l)}\mathbf{W}_T^{(l)}), \tag{23}$$

where $\mathcal{D}(\cdot,\cdot)$ is a distance metric (e.g., MSE), and the **Orthogonal Loss** induces implicit high-pass filtering by enforcing subspace orthogonality with:

$$\mathcal{L}_{\text{ORTH}} = \sum_{l=1}^{L} \|\tilde{\mathbf{Z}}_T^{(l)\top}\tilde{\mathbf{Z}}_C^{(l)}\|_F^2, \tag{24}$$

which guarantees the orthogonality between outputs for the dual feature transformation streams. After that, $\tilde{\mathbf{Z}}_T^{(l)}$ and $\tilde{\mathbf{Z}}_C^{(l)}$ are concatenated and transformed with:

$$\mathbf{Z}^{(l+1)} = \sigma\left(\left[\tilde{\mathbf{Z}}_T^{(l)} \parallel \tilde{\mathbf{Z}}_C^{(l)}\right]\mathbf{\Theta}^{(l)}\right), \tag{25}$$

where $\mathbf{\Theta}^{(l)} \in \mathbb{R}^{d\times d}$ are learnable projection matrices. The final layer output $\mathbf{Z}^{(L)}$ is fed into a task head to produce logits $\hat{\mathbf{Y}}$ for graph learning tasks.

The complete optimization objective integrates knowledge distillation with the dual-stream alignment mechanisms:

$$\mathcal{L}_{\text{KD}} = \lambda \sum_{v\in\mathcal{V}_L} \mathcal{L}(\hat{\mathbf{y}}_v, \mathbf{y}_v) + (1-\lambda)\sum_{v\in\mathcal{V}} \mathcal{L}(\hat{\mathbf{y}}_v, \mathbf{z}_v), \tag{26}$$

$$\mathcal{L} = \mathcal{L}_{\text{KD}} + \alpha\mathcal{L}_{\text{DIST}} + \beta\mathcal{L}_{\text{ORTH}}, \tag{27}$$

where λ, α, β are balancing coefficients.

Experiment

Experiment Setting

Datasets We used seven public benchmarks including Cora, Citeseer, Pubmed (Kipf and Welling 2017), Amazon-Photo, CS, Physics (Zhang et al. 2022) and ogbn-ArXiv (Hu et al. 2020). We follow the data splitting strategy in (Lu et al. 2024).

Baselines We integrate our method into the existing well-performing GNN-to-MLP framework, including GLNN (Zhang et al. 2022), VQGraph (Yang et al. 2024), KRD (Wu et al. 2023), AdaGMLP (Lu et al. 2024). We modify their student model structure and incorporate additional learning objectives of Eq. (23) and Eq. (24).

Implementation For all baselines, we set their depth to 2 layers and use the implementations from the PyTorch Geometric Library. Hyperparameters of the baseline methods are set to the suggested values in their respective papers or carefully tuned for fairness. Accuracy scores are used as the final evaluation metric, and we report the mean and standard deviation over five runs for all classification tasks.

Results and Comparison

The experimental results presented in Table 1 demonstrate the efficacy of the proposed DA-MLP in enhancing the performance of GNN-to-MLP knowledge distillation methods under both transductive and inductive settings, with the teacher model implemented as a two-layer GraphSAGE. In the transductive setting, DA-MLP improves GLNN's accuracy by 2.04% on average, with substantial gains observed on datasets such as Physics (from 92.08% to 93.25%) and Photo (from

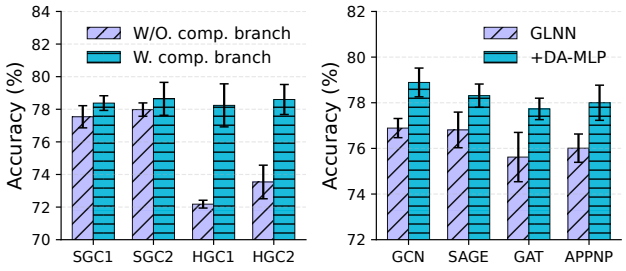

Figure 3: **(LEFT)** Comparison of different GP methods as teacher-related filters. **(RIGHT)** Comparison of different teacher GNN architecture. Accuracies for both are averaged over 3 `Planetoid` datasets.

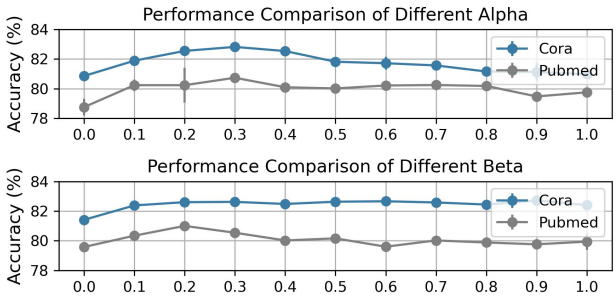

Figure 4: Hyperparameter analysis of α and β.

	Texas	Cornell	Chameleon	Squirrel
ACMGCN	86.49	84.05	49.02	30.24
GLNN	85.63	83.87	49.47	29.45
GLNN + DA-MLP	**86.92**	**84.26**	**50.82**	**30.41**
VQGraph	87.03	84.59	50.76	30.62
VQGraph + DA-MLP	**87.47**	**85.02**	**51.18**	**31.04**
GCNII	76.73	76.49	45.32	24.98
GLNN	77.27	76.37	44.65	25.30
GLNN + DA-MLP	**78.60**	**77.45**	**45.92**	**26.78**
VQGraph	79.04	78.29	45.89	26.02
VQGraph + DA-MLP	**80.48**	**78.91**	**47.14**	**27.16**

Table 2: Results on Heterophilic Graphs.

89.73% to 91.80%). Similarly, DA-MLP improves the performance of KRD, VQGraph, and AdaGMLP by 1.09%, 1.18%, and 1.27%, respectively, showcasing its ability to complement various GNN-to-MLP frameworks. In the inductive setting, DA-MLP also achieves notable improvements, such as increasing the accuracy of GLNN from 79.37% to 80.93% on average. Similar performance improvements have also been observed in other KD frameworks. These experimental results demonstrate that our method can be effectively integrated into existing GNN knowledge distillation frameworks while enhancing their capability.

Ablation Study

To analyze the impact of different teacher filters $\mathbf{g}_T$, we investigated various graph propagation matrices $\mathbf{P}$ derived from $\mathbf{g}_T$ with SAGE as fixed teacher GNN. These include $\mathbf{P} = \overline{\mathbf{A}}\mathbf{X}$ (SGC1), $\mathbf{P} = \overline{\mathbf{A}}^2\mathbf{X}$ (SGC2), $\mathbf{P} = (\mathbf{I} - \overline{\mathbf{A}})\mathbf{X}$ (HGC1), and $\mathbf{P} = (\mathbf{I} - \overline{\mathbf{A}})^2\mathbf{X}$ (HGC2) where $\overline{\mathbf{A}}$ is the row-normalized adjacency matrix, which is aligned with (Zhao et al. 2025). We also explored different teacher architectures (GCN (Kipf and Welling 2017), GraphSAGE (William L. Hamilton 2017), GAT (Veličković et al. 2018), APPNP (Johannes Klicpera 2018)) with their corresponding $\mathbf{F}_G$ in Equation (23). The results in Figure 3 (LEFT) show that the complementary branch makes the model performance robust to variations in $\mathbf{g}_T$, while its removal leads to significant sensitivity. Figure 3 (RIGHT) shows that DA-MLP consistently improves distillation across all teacher models.

Hyper-parameter Analysis

We performed hyperparameter analysis on the Cora and PubMed datasets by varying α and β in [0,1]. As shown in Figure 4, both of the two hyperparameters exhibit optimal performance in the range of 0.2 to 0.4. Additionally, we observed that as these parameters gradually increase, the only noticeable effect is a slower convergence speed. With a sufficient number of training epochs, larger values of α and β may even achieve superior performance. Considering the trade-off between performance and computational efficiency, we opted for smaller values of α and β in our final experiments.

Extending to Heterophilic Graphs

We evaluate DA-MLP on four heterophilic datasets by distilling two state-of-the-art heterophilic GNNs: ACM-GCN (Luan et al. 2022) and GCNII (Chen et al. 2020), with GLNN and VQGraph as baseline KD methods following (Yang et al. 2024). For GCNII, we apply the standard DA-MLP framework. For ACM-GCN, we explicitly model its adaptive channel mixing through dual MLP branches for low/high-pass channels and residual connections for the identity channel. As shown in Table 2, DA-MLP consistently outperforms baseline KD methods on heterophilic graphs, demonstrating its effectiveness in distilling specialized GNNs while preserving model accuracy.

Conclusion

In conclusion, our work demystifies the effectiveness of GNN-to-MLP knowledge distillation by revealing that student MLPs adapt their feature attention to align their Neural Tangent Kernel (NTK) with the graph structure, thereby enabling implicit propagation without explicit edges. Moreover, we theoretically identify limitations in approximating deep graph propagation via feature transformation, exposing significant cumulative approximation errors and robustness issues across diverse connectivity patterns. Based on the theoretical results, we propose the DA-MLP framework, which leverages complementary graph filters to enhance layer-wise approximation and preserve diverse signals during distillation. It consistently boosts the performance of existing distillation methods across diverse datasets.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (Grant Nos. 62572236, 62502201), the Natural Science Foundation of Jiangsu Province (Grant Nos. BK20222003, BK20251198), the Collaborative Innovation Center of Novel Software Technology and Industrialization, and the Sino-German Institutes of Social Computing.

References

Abu-El-Haija, S.; Perozzi, B.; Kapoor, A.; Harutyunyan, H.; Alipourfard, N.; Lerman, K.; Steeg, G. V.; and Galstyan, A. G. 2019. MixHop: Higher-Order Graph Convolutional Architectures via Sparsified Neighborhood Mixing. In *International Conference on Machine Learning*.

Bei, Y.; Chen, W.; Chen, H.; Zhou, S.; Yang, C.; Fan, J.; Huang, L.; and Bu, J. 2025. Correlation-Aware Graph Convolutional Networks for Multi-Label Node Classification. arXiv:2411.17350.

Chen, M.; Wei, Z.; Huang, Z.; Ding, B.; and Li, Y. 2020. Simple and deep graph convolutional networks. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Du, S. S.; Hou, K.; Póczos, B.; Salakhutdinov, R.; Wang, R.; and Xu, K. 2019a. *Graph neural tangent kernel: fusing graph neural networks with graph kernels*. Red Hook, NY, USA: Curran Associates Inc.

Du, S. S.; Hou, K.; Póczos, B.; Salakhutdinov, R.; Wang, R.; and Xu, K. 2019b. Graph Neural Tangent Kernel: Fusing Graph Neural Networks with Graph Kernels. In *Neural Information Processing Systems*.

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

Guo, Z.; Nan, B.; Tian, Y.; Wiest, O.; Zhang, C.; and Chawla, N. 2022. Graph-based Molecular Representation Learning. In *International Joint Conference on Artificial Intelligence*.

Hinton, G.; Vinyals, O.; and Dean, J. 2015a. Distilling the Knowledge in a Neural Network. arXiv:1503.02531.

Hinton, G. E.; Vinyals, O.; and Dean, J. 2015b. Distilling the Knowledge in a Neural Network. *ArXiv*, abs/1503.02531.

Hu, W.; Fey, M.; Zitnik, M.; Dong, Y.; Ren, H.; Liu, B.; Catasta, M.; and Leskovec, J. 2020. Open graph benchmark: datasets for machine learning on graphs. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20.

Jacot, A.; Gabriel, F.; and Hongler, C. 2018. Neural tangent kernel: convergence and generalization in neural networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, 8580–8589. Red Hook, NY, USA: Curran Associates Inc.

Jia, Z.; Lin, S.; Ying, R.; You, J.; Leskovec, J.; and Aiken, A. 2020. Redundancy-Free Computation for Graph Neural Networks. In Gupta, R.; Liu, Y.; Tang, J.; and Prakash, B. A., eds., *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, 997–1005. ACM.

Jin, D.; Yu, Z.; Huo, C.; Wang, R.; Wang, X.; He, D.; and Han, J. 2021. Universal Graph Convolutional Networks. In *Neural Information Processing Systems*.

Johannes Klicpera, S. G., Aleksandar Bojchevski. 2018. Predict then Propagate: Combining neural networks with personalized pagerank for classification on graphs. In *International Conference on Learning Representations*.

Kawaguchi, K.; and Huang, J. 2019. Gradient Descent Finds Global Minima for Generalizable Deep Neural Networks of Practical Sizes. In *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 92–99. IEEE Press.

Keriven, N.; and Peyré, G. 2019. Universal Invariant and Equivariant Graph Neural Networks. In *Neural Information Processing Systems*.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*.

Kullback, S.; and Leibler, R. A. 1951. On Information and Sufficiency. *Annals of Mathematical Statistics*, 22(1): 79–86.

Lee, J.; Xiao, L.; Schoenholz, S. S.; Bahri, Y.; Novak, R.; Sohl-Dickstein, J.; and Pennington, J. 2019. *Wide neural networks of any depth evolve as linear models under gradient descent*. Red Hook, NY, USA: Curran Associates Inc.

Li, Y.; Fan, J.; Wang, Y.; and Tan, K.-L. 2018. Influence Maximization on Social Graphs: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, 30: 1852–1872.

Lin, M.; Hong, X.; Li, W.; and Lu, S. 2025. Unified Graph Neural Networks Pre-training for Multi-domain Graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI 2025)*, volume 39, 12165–12173.

Lin, M.; Li, W.; Li, D.; Chen, Y.; and Lu, S. 2022. Resource-efficient training for large graph convolutional networks with label-centric cumulative sampling. In *Proceedings of the ACM Web Conference (WWW 2022)*, 1170–1180.

Lu, W.; Guan, Z.; Zhao, W.; and Yang, Y. 2024. AdaGMLP: AdaBoosting GNN-to-MLP Knowledge Distillation. In Baeza-Yates, R.; and Bonchi, F., eds., *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024*, 2060–2071. ACM.

Luan, S.; Hua, C.; Lu, Q.; Zhu, J.; Zhao, M.; Zhang, S.; Chang, X.-W.; and Precup, D. 2022. Revisiting Heterophily For Graph Neural Networks. arXiv:2210.07606.

Rumelhart, D. E.; Hinton, G. E.; and Williams, R. J. 1988. *Learning representations by back-propagating errors*, 696–699. Cambridge, MA, USA: MIT Press. ISBN 0262010976.

Tian, Y.; Zhang, C.; Guo, Z.; Zhang, X.; and Chawla, N. V. 2022. NOSMOG: Learning Noise-robust and Structure-aware MLPs on Graphs. *CoRR*, abs/2208.10010.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.

William L. Hamilton, J. L., Zhitao Ying. 2017. Inductive Representation Learning on Large Graphs. In *Neural Information Processing Systems*.

Wu, L.; Lin, H.; Huang, Y.; and Li, S. Z. 2023. Quantifying the Knowledge in GNNs for Reliable Distillation into MLPs. In Krause, A.; Brunskill, E.; Cho, K.; Engelhardt, B.; Sabato, S.; and Scarlett, J., eds., *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, 37571–37581. PMLR.

Wu, T.; Ren, H.; Li, P.; and Leskovec, J. 2020. Graph Information Bottleneck. arXiv:2010.12811.

Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; and Yu, P. S. 2021. A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1): 4–24.

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2019. How Powerful are Graph Neural Networks? In *International Conference on Learning Representations*.

Yang, L.; Tian, Y.; Xu, M.; Liu, Z.; Hong, S.; Qu, W.; Zhang, W.; Cui, B.; Zhang, M.; and Leskovec, J. 2024. VQGraph: Rethinking Graph Representation Space for Bridging GNNs and MLPs. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Zhang, D.; Huang, X.; Liu, Z.; Zhou, J.; Hu, Z.; Song, X.; Ge, Z.; Wang, L.; Zhang, Z.; and Qi, Y. 2020. AGL: A Scalable System for Industrial-purpose Graph Machine Learning. *Proc. VLDB Endow.*, 13(12): 3125–3137.

Zhang, S.; Liu, Y.; Sun, Y.; and Shah, N. 2022. Graphless Neural Networks: Teaching Old MLPs New Tricks via Distillation. arXiv:2110.08727.

Zhao, J.; Zhu, Z.; Galkin, M.; Mostafa, H.; Bronstein, M.; and Tang, J. 2025. Fully-inductive Node Classification on Arbitrary Graphs. arXiv:2405.20445.