Diagnosing Graph Neural Networks: Do They Operate in the Feature Learning Regime and How?

Tao Wen [1] **Gael C. Popescu** [2] **Dawei Zhou** [3] **Yaoqing Yang** [1] **Yujun Yan** [1]

Abstract

The generalization of Graph Neural Networks (GNNs) to other datasets is critical for the development of Graph Foundation Models (GFMs). Yet, unlike many foundation models in vision and language, GNNs can frequently achieve high test accuracy on the same dataset yet generalize poorly to other datasets, with performance brittle to data splits, distribution shifts, and evaluation protocols. These behaviors are consistent with kernel-like dynamics, where training is well-approximated by a linearization around initialization and internal representations change only weakly. By contrast, strong generalization is typically associated with feature learning, where internal representations change substantially to learn generalizable patterns from data. Despite broad recognition of the GNN generalization issues, it remains unclear to what extent GNNs exhibit feature learning and which architectural choices most effectively promote it. To address these gaps, we propose a theoretically grounded and practical diagnostic framework that evaluates GNN generalization across datasets via feature learning and provides actionable guidance for future GNN and GFM design. Using this framework, we study over ten models and datasets and show that data format and model architecture influence how easily GNNs operate in the feature learning regime. In particular, we find that most classic GNNs struggle to work in this regime. On the data side, representing regular data as graphs makes it harder to operate in this regime. On the model side, most message-passing architectures rarely succeed. In contrast, graph transformers work in this regime more effectively.

[1]Dartmouth College [2]Hanover High School [3]Virginia Tech. Correspondence to: Tao Wen <tao.wen.gr@dartmouth.edu>.

The Workshop on Graph Foundation Models at the 43rd International Conference on Machine Learning (ICML 2026), Seoul, South Korea.

1. Introduction

Graph foundation models (GFMs) aim to learn representations that generalize across graph datasets and tasks with large-scale pretraining and minimal task specific retraining (Chen et al., 2024; Frasca et al., 2024; Wang et al., 2025), and have been widely used in diverse graph domains, including molecular graphs (Méndez-Lucio et al., 2024), knowledge graphs (Galkin et al., 2024), and brain connectivity networks (Wang et al., 2026). This emerging paradigm highlights a central requirement for graph neural networks (GNNs) to capture generalizable patterns rather than overfitting to a single dataset, motivating a principled understanding of when and how GNNs generalize to other datasets.

However, a range of GNN generalizability issues have long been observed, but principled understandings of these issues are still unclear, leading to challenging development of GFMs. First, graph learning is inherently difficult due to the irregular structure of graphs, and GNNs can even underperform non-graph baselines in cases where graph structural information harms performance (Zhu et al., 2020; Bechler-Speicher et al., 2025). Second, in contrast to foundation deep vision and language models such as ResNets (He et al., 2016) and Transformers (Vaswani et al., 2017) whose performance typically improves with model depth, GNNs often suffer from performance degradation as depth increases, a phenomenon known as oversmoothing (Li et al., 2018). Finally, GNN performance is often brittle to hyperparameter choices and evaluation protocols, resulting in performance gains that are frequently marginal or dataset-specific (Garg et al., 2020; Bechler-Speicher et al., 2025), with GNNs often showing limited or even negative generalizability to other datasets (Zhu et al., 2021; Ju et al., 2025). All these issues resemble the behaviors of models operating in the *kernel regime*, where internal representations remain largely unchanged, and learning is approximated by linearization around initialization. In contrast, foundation vision and language models typically work in the *feature learning regime*, where they adapt their internal representations to capture generalizable patterns in the data, leading to substantially stronger generalizability both within the same dataset and to other datasets (Ghorbani et al., 2019; Woodworth et al., 2020; Damian et al., 2022; Kornblith et al., 2019b; Brown

et al., 2020). Although feature learning has been widely studied in vision and language tasks (Fort et al., 2020; Wang et al., 2023; Yang et al., 2023b; Radhakrishnan et al., 2024; Beaglehole et al., 2024), it remains under-explored on graph tasks, and few prior works have offered a systematic and practical framework to diagnose the GNNs' generalizability to other datasets via feature learning. To address this gap, in this work, we aim to address the following questions: *(1) Do GNNs operate in the feature learning regime? (2) Which GNN architectures are promising in feature learning? (3) How can their generalizability to other datasets be diagnosed via feature learning?*

To address these questions, we propose a theoretically grounded diagnostic framework that measures the quality of internal representations during the feature learning and kernel regime transition process. Specifically, we vary the model's initial weight scale by a constant multiplier to prompt a regime transition, and evaluate the resulting trend of representation-label alignments to assess how easily the model operates in the feature learning regime. In addition, we adopt a transfer learning setup to assess the model's inherent generalizability to other datasets. Using our framework, We find that: (1) Different from foundation vision and language models, accuracy is not a good generalizability metric for GNNs, as they can achieve high accuracy on the same dataset in the kernel regime, but generalize poorly to other datasets, which requires working in the feature learning regime; (2) Graph-structured inputs may hinder feature learning compared to regular data; (3) While the advantage of graph transformers over message-passing GNNs in same-dataset test accuracy remains under debate (Luo et al., 2024; 2025), they more easily operate in the feature learning regime than most message-passing GNNs, thanks to attention mechanisms and facilitated by positional encodings. While the main body focuses on graph classification, additional results show that our conclusions hold for graph regression and node classification as well. In summary, our key contributions are as follows:

- New framework: We propose the first theoretically grounded, practical framework to systematically diagnose feature learning for GNNs. Our framework reveals discrepancies in generalizability to other datasets when test accuracy on the same dataset is similar, highlighting a new direction to study GNN generalizability.
- New insights: Our framework provides actionable guidance for GNN and GFM design. We find that graph-structured data makes it inherently harder for models to enter the feature learning regime. In addition, message-passing GNNs are less likely to operate in this regime, whereas graph transformers more readily do so, thanks to the attention mechanism, which plays a key role in graph feature learning, and facilitated by positional encodings.

2. Preliminaries

We begin by providing an overview of the notations, followed by an introduction to graph neural networks and graph classification. We then detail the feature learning and kernel regimes, clarify their differences, and explain kernel-target alignment (KTA) as a metric to quantify feature learning.

2.1. Notations

Suppose a set of graphs $\mathcal{G} = \{G_1, G_2, \ldots, G_N\}$ of C classes. Let $G = (\mathbf{A}^G, V^G, E^G, \mathbf{X}^G, y_G)$ be a graph in $\mathcal{G}$, where y_G is the label of G, V^G is the set of nodes with $|V^G| = n_G$, E^G is the set of edges. $\mathbf{X}^G \in \mathbb{R}^{n_G \times F}$ is the node feature matrix, where F denotes the dimensionality of node features. $\mathbf{A}^G \in \mathbb{R}^{n_G \times n_G}$ is the adjacency matrix, where each entry $a^G_{ij} = a^G_{ji} = 1$ if nodes i and j are connected, and $a^G_{ij} = 0$ otherwise. The diagonal degree matrix is $\mathbf{D}^G$, and $D^G_v \triangleq \mathbf{D}^G_{vv}$ is the degree of node $v \in V^G$. We denote the labels for all graphs in the dataset as $\mathbf{y}$, and $\mathbf{Y} \in \mathbb{R}^{N \times C}$ as the one-hot encoded label matrix if $C > 2$. Furthermore, let $\mathbf{Z} \in \mathbb{R}^{N \times d}$ denote the *post-activation graph representations* of the *penultimate* layer, where d is the hidden dimension. The *feature kernel* is then defined as: $\mathbf{K} \triangleq \mathbf{Z}\mathbf{Z}^\top \in \mathbb{R}^{N \times N}$ (He & Ozay, 2022).

2.2. Graph Neural Networks and Graph Classification

GNNs are neural networks that operate on graphs, consisting mainly of two families: (1) Message-Passing Neural Networks (MPNNs) are typically classic GNNs that aggregate information through iterative message passing between local neighbors; (2) Graph Transformers (GTs) are adapted from the Transformer architecture where global attention allows each node to attend to other nodes in the graph (Kim et al., 2022). See Appendix C.2 for their formulations. While GTs are more recent, their advantage over MPNNs in same-dataset test accuracy remains under debate (Luo et al., 2024; 2025). However, as we will show in Section 5, they work in the feature learning regime more effectively than MPNNs and generalize better to other datasets.

The goal of graph classification is to predict the label y_G of a graph G by learning a graph-level representation $\mathbf{h}_G$ that is used as input to a final classification layer L. This is typically achieved using a readout function that aggregates node representations of G from the penultimate layer:

$$\mathbf{h}_G = \text{READOUT}\left(\{\mathbf{h}_v^{(L-1)} \mid v \in V^G\}\right),$$

where $\mathbf{h}_G$ is the graph representation, $\mathbf{h}_v^{(L-1)}$ is node v's representation from the penultimate layer.

2.3. Feature Learning

Feature learning and kernel regimes To understand why neural networks generalize, researchers have identified two regimes: the kernel regime and the feature learning regime. In the kernel regime, internal representations remain largely unchanged during training, and performance depends on the final output layer (Tian, 2025), and thus remains dataset and task specific. In contrast, the feature learning regime involves substantial change of internal representations, enabling the model to capture rich and reusable patterns that are critical for strong generalizability (Ghorbani et al., 2019; Woodworth et al., 2020; Damian et al., 2022; Zhang & Bottou, 2023; Wang et al., 2023; Göring et al., 2025). Such *rich and generalizable patterns* are in general referred to as *features* in feature learning. Transitions between the two regimes can be influenced by setups such as learning rate and initial weight scale (Kunin et al., 2024; Chizat et al., 2019; Simon et al., 2026). In image domain, models typically operate in the feature learning regime, and only transition to the kernel regime under atypical setups such as small learning rates or large initial weight scales (Woodworth et al., 2020; Kunin et al., 2024). However, as we will show in Section 5, many GNNs instead naturally operate in the kernel regime and require special architectural designs to enter the feature learning regime.

To formalize the characteristics of the two regimes, given an input $\mathbf{X}$, neural networks (NNs) can be formulated as:

$$f(\mathbf{X}) = \phi(\mathbf{X})\mathbf{V} \tag{1}$$

where $f(\mathbf{X})$ is the NN, $\mathbf{V}$ is its output layer, and $\phi(\cdot)$ is any NN architecture up to its *penultimate layer*. For example, for a two-layer MLP, $\phi(\mathbf{X}) = \sigma(\mathbf{X}\mathbf{W})$, i.e., first-layer activations. We study representations from the *penultimate layer* as they encode all features learned by the NN, and are commonly used in feature learning (He & Ozay, 2022) and transfer learning (Kornblith et al., 2019b). Specifically, let the *penultimate-layer representations* $\mathbf{Z} \triangleq \phi(\mathbf{X}) \in \mathbb{R}^{N\times d}$ for this two-layer MLP, in the kernel regime, $\mathbf{Z}$ largely stays unchanged, and test accuracy is attributed to the optimization of $\mathbf{V}$. In contrast, in the feature learning regime, $\phi(\mathbf{X})$ is significantly changed, and thus $\mathbf{Z}$ have learned *features* (Tian, 2025).

Quantifying feature learning with KTA Kernel-Target Alignment (KTA) (Cristianini et al., 2001) is a measure of feature learning (Wang et al., 2023; Yang et al., 2024; Kothapalli et al., 2025; Chou et al., 2025; Jiang et al., 2026). Given $\mathbf{K} = \mathbf{Z}\mathbf{Z}^\top \in \mathbb{R}^{N\times N}$ and $\mathbf{y}\mathbf{y}^\top \in \mathbb{R}^{N\times N}$,

$$\text{KTA} = \frac{\langle \mathbf{K}, \mathbf{y}\mathbf{y}^\top \rangle_F}{\|\mathbf{K}\|_F \, \|\mathbf{y}\mathbf{y}^\top\|_F}. \tag{2}$$

It is measured on the *test set* to evaluate the generalizability of the learned representations rather than memorization. For multi-class tasks, we use the one-hot encoded label $\mathbf{Y}$. Furthermore, both $\mathbf{K}$ and $\mathbf{y}\mathbf{y}^\top$ or $\mathbf{Y}\mathbf{Y}^\top$ are mean-centered to ensure the alignment reflects true relational structure rather than trivial offsets in values (Kornblith et al., 2019a). As we will show in Section 5, unlike on image or language data where test accuracy on the same dataset is usually indicative of cross-data generalizability under typical setups (Kornblith et al., 2019c; Brown et al., 2020), on graph data, KTA is more indicative of it than test accuracy on the same dataset.

The complexity of the above KTA computation is $O(N^2 d)$. To make it scalable on large-scale data where $N \gg d$, there exists an equivalent computation as:

$$\text{KTA} = \frac{\|\mathbf{Z}^\top \mathbf{y}\|_F^2}{\|\mathbf{Z}^\top \mathbf{Z}\|_F \, \|\mathbf{y}^\top \mathbf{y}\|_F} \tag{3}$$

whose complexity is $O(Nd^2)$. See Appendix B for its derivation and empirical complexity analysis.

3. Our GNN Diagnostic Framework

Our framework rescales model weights at initialization and measures the KTA values across different scaling factors α to assess whether the models operate in the feature learning regime. Here, α is a constant multiplier applied to the initial weights, not a tunable hyperparameter. By increasing α, we expect a transition from the feature learning regime to the kernel regime (Chizat et al., 2019; Geiger et al., 2020; Woodworth et al., 2020; Kunin et al., 2024; Dominé et al., 2025). Specifically, the case $\alpha = 1$ corresponds to the original initialization scale. Smaller values of α facilitate feature learning by encouraging the models to actively learn features in the data, while larger values promote noise memorization and push the models to the kernel regime, which we theoretically validate in Section 4.

By sweeping α in a wide and uniform range for all models, given the same data, we determine whether models operate in the feature learning regime based on the following criteria: (1) Models that operate in the feature learning regime exhibit higher KTA values, whereas models that do not operate in this regime exhibit lower KTA values. If the KTA of a model remains low across different α, then the model does not operate in the feature learning regime. (2) Models that operate in the feature learning regime should show a generally decreasing KTA as α increases, reflecting a transition into the kernel regime; otherwise, the KTA remains flat. (3) Models operating in the feature learning regime exhibit KTA curves that are clearly separated by different α, with smaller α leading to faster rises and higher peak values during training (Fort et al., 2020). In contrast, models in the kernel regime show uniformly low KTA with insignificant distinction between curves.

4. Theoretical Rationale

In this section, we provide a theoretical rationale for our framework to validate that in a graph classification task, a small initialization scale α makes feature learning easier and leads to higher KTA. We base our theoretical analysis on (Huang et al., 2025), which demonstrates that GCN outperforms MLP in node classification by feature learning. In contrast, we focus on graph classification and analyze the effect of initial weight scale, resulting in a new evaluation framework and leading to new practical insights into GNN architecture design. In addition, while varying initial scale has been primarily established for MLPs and CNNs (Woodworth et al., 2020; Fort et al., 2020), our theoretical analysis shows that it also applies to GNNs.

4.1. Assumptions

Recall the setup in Section 2. We further make the following assumptions:

(1) Node-level: Each node has a *latent type* denoted as $z_i^G \in \{\pm 1\}$. Its feature vector $\mathbf{x}_i^G$ is generated via a signal-noise model, conditioned on this latent type z_i^G and a shared latent *signal vector* $\boldsymbol{\mu} \in \mathbb{R}^d$. The node features are formed by a signal component and a noise component:

$$\mathbf{x}_i^G = [\mathbf{x}_i^{G(1)}, \mathbf{x}_i^{G(2)}] = [z_i^G \boldsymbol{\mu}, \boldsymbol{\xi}_i^G] \tag{4}$$

where $\mathbf{x}_i^{G(1)}, \mathbf{x}_i^{G(2)} \in \mathbb{R}^d$ are signal and noise feature components, respectively, and $\boldsymbol{\xi}_i^G \sim \mathcal{N}\left(\mathbf{0}, \sigma_p^2 \cdot \left(\mathbf{I}_d - \|\boldsymbol{\mu}\|_2^{-2} \cdot \boldsymbol{\mu}\boldsymbol{\mu}^\top\right)\right)$ is the *noise vector* orthogonal to $\boldsymbol{\mu}$. In particular, the signal $\boldsymbol{\mu}$ represents the *generalizable patterns* in the data, i.e., *features*.

(2) Graph-level: Each label $y_G \in \{\pm 1\}$ is a Rademacher variable. We denote the maximum number of nodes in a graph as $\overline{n} = \max\{n_G\}_{G \in \mathcal{G}}$ and the minimum as $\underline{n} = \min\{n_G\}_{G \in \mathcal{G}}$. Graph structure is generated using a stochastic block model with intra-type edge probability p and inter-type edge probability s, where $p > s$. Moreover, we assume the number of graphs N is sufficiently large, and for each G, the majority of nodes have latent types aligned with y_G.

(3) Model-level: Following the assumptions of the two-layer GCN in (Huang et al., 2025), where the first layer performs a graph convolution operation and the second-layer parameters are fixed to either $+1$ or -1, we further adopt a sum readout to aggregate node representations into a graph-level representation for prediction. The resulting graph-level output is given by:

$$f(\mathbf{W}, \mathbf{A}^G, \mathbf{X}^G) = F_{+1}(\mathbf{W}_{+1}, \mathbf{A}^G, \mathbf{X}^G) - F_{-1}(\mathbf{W}_{-1}, \mathbf{A}^G, \mathbf{X}^G) \tag{5}$$

where

$$F_j(\mathbf{W}_j, \mathbf{A}^G, \mathbf{X}^G) = \frac{1}{m} \sum_{r=1}^{m} \sum_{i \in V^G} \left[\sigma(\mathbf{w}_{j,r}^\top \tilde{\mathbf{x}}_i^{G(1)}) + \sigma(\mathbf{w}_{j,r}^\top \tilde{\mathbf{x}}_i^{G(2)})\right] \tag{6}$$

and $\mathbf{W}_{\pm 1}$ refer to the first-layer weights associated with the second-layer fixed parameters. The first-layer weights are initialized by sampling from a Gaussian distribution, i.e., the weight of each hidden neuron $\mathbf{w}_{j,r} \sim \mathcal{N}(0, \sigma_0^2 \cdot \mathbf{I}_d)$ for all $r \in [m]$ and $j \in \{\pm 1\}$, where m is the number of hidden neurons, and σ_0 controls the *scale of the initial weights*, corresponding to α in our framework. After graph convolution, we denote degree-normalized vectors $\tilde{z}_i^G = (D_i^G)^{-1} \sum_{k \in \mathcal{N}(i)} z_k^G$, and $\tilde{\boldsymbol{\xi}}_i^G = (D_i^G)^{-1} \sum_{k \in \mathcal{N}(i)} \boldsymbol{\xi}_k^G$. For other proof-related assumptions, see Assumption A.1.

4.2. Small Initial Weights Facilitate Feature Learning

In this section, we introduce our results. First, we show that the weight update can be written as a linear combination of initial weight $\mathbf{w}_{j,r}^{(0)}$, signal vector $\boldsymbol{\mu}$, and noise vector $\boldsymbol{\xi}_i^G$:

$$\mathbf{w}_{j,r}^{(t)} = \mathbf{w}_{j,r}^{(0)} + j\gamma_{j,r}^{(t)} \|\boldsymbol{\mu}\|_2^{-2} \boldsymbol{\mu} + \sum_{G \in \mathcal{G}} \sum_{i \in V^G} \rho_{j,r,i}^{G(t)} \|\boldsymbol{\xi}_i^G\|_2^{-2} \boldsymbol{\xi}_i^G \tag{7}$$

where $\gamma_{j,r}^{(t)}$ and $\rho_{j,r,i}^{G(t)}$ are the *signal learning* and *noise memorization* coefficients, respectively. Its proof is given in Appendix A.2. If $\gamma_{j,r}^{(t)}$ is significantly large and $\rho_{j,r,i}^{G(t)}$ is relatively small, then the GCN primarily *learns signals rather than noise*, facilitating feature learning. This leads to our theorem that under a small σ_0, at some time $\max_r \gamma_{j,r}^{(t)}$ dominates $\rho_{j,r,i}^{G(t)}$, leading to a higher KTA.

Theorem 4.1. *[informal] (Smaller initialization promotes signal while suppressing noise). Under Assumption A.1, with high probability, there exists a time during training at which at least a hidden neuron becomes strongly aligned with the signal while remaining weakly aligned with the noise. In particular, there exists $0 \le t \le T$ and $r^* \in [m]$ such that for any graph G and node i,*

$$\gamma_{y_G, r^*}^{(t)} \ge 2 \quad \textit{and} \quad |\rho_{y_G, r^*, i}^{G(t)}| \le C \cdot \sigma_0 \sigma_p \sqrt{d/(\underline{n}(p+s))}.$$

Consequently, the corresponding weights satisfy

$$\langle \mathbf{w}_{y_G, r^*}^{(t)}, y_G \boldsymbol{\mu} \rangle \ge 2 - C_1 \sigma_0, \quad |\langle \mathbf{w}_{y_G, r^*}^{(t)}, \tilde{\boldsymbol{\xi}}_i^G \rangle| \le C_2 \sigma_0 + 1,$$

where C_1 and C_2 are positive constants independent of σ_0.

Remark 4.2. The Theorem 4.1 indicates that during a time duration T, the signal coefficient reaches a constant order while that of the noise coefficient remains bounded by a factor of σ_0. In other words, a small σ_0 favors signal learning over noise memorization, enabling the GCN to primarily

learn μ, and thus operate in the feature learning regime. Furthermore, it shows that decreasing the initialization scale σ_0 simultaneously increases the lower bound of $\langle \mathbf{w}_{y_G,r^*}^{(t)}, y_G \boldsymbol{\mu} \rangle$ and decreases the magnitude of $\langle \mathbf{w}_{y_G,r^*}^{(t)}, \tilde{\boldsymbol{\xi}}_i^G \rangle$. Therefore, a small initial weight scale α encourages the learned weights to align with signal rather than noise, which in turn increases KTA, validating our framework.

The formal statements and proofs are in Appendix A. Thus we have theoretically shown that a small α promotes feature learning and increases KTA in graph classification. Therefore, a model that operates in the feature learning regime is expected to exhibit a generally decreasing KTA trend as α increases, which we empirically validate in Section 5.

5. Empirical Results

We aim to address the following questions: **Q1**: Is the feature learning–associated KTA used by our framework a better generalizability metric for GNNs than accuracy? **Q2**: Using our framework, can we assess whether graph data makes feature learning harder for the same model than regular data? **Q3**: Using our framework, can we identify which GNNs operate in the feature learning regime and what design choices enable this behavior?

To address Q1, in Section 5.1 we adopt a transfer learning setup to show that KTA is a better indicator of generalizability to other datasets than accuracy. To address Q2, in Section 5.2 we leverage the MNIST dataset in both image and graph formats (Dwivedi et al., 2023) and compare whether the MLPs work in the feature learning regime for each format. To address Q3, in Section 5.3 we study GNNs including MPNNs and GTs and evaluate how different design choices influence whether GNNs work in the feature learning regime on multiple datasets. The training details are provided in Appendix C.3.

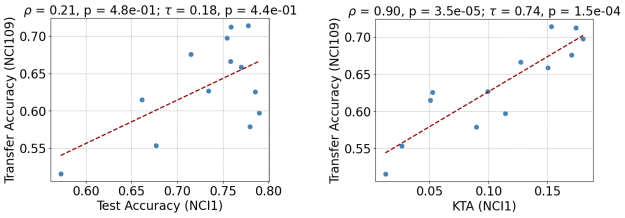
Datasets We use six datasets from diverse domains to assess feature learning on graph level tasks, covering large scale settings, various structural properties and data distributions, and both classification and regression: (1) MNIST (Dwivedi et al., 2023); (2) NCI1 (Morris et al., 2020); (3) PROTEINS (Morris et al., 2020); (4) Peptides-struct (Dwivedi et al., 2022b); (5) OGBG-MOLHIV (Hu et al., 2020); (6) OGBG-PPA (Hu et al., 2020). Their details are provided in Appendix C.1.

Models In addition to positional encodings (PE) including Laplacian PE (LapPE) (Dwivedi et al., 2023) and Random Walk PE (RWPE) (Dwivedi et al., 2022a), we evaluate the following models: (1) MLP, (2) MPNNs: GCN (Kipf & Welling, 2017), GraphSAGE (Hamilton et al., 2017), GAT (Veličković et al., 2018), SGC (Wu et al., 2019), GCNII (Chen et al., 2020), A-DGN (Gravina et al., 2023), GCN$^+$ (Luo et al., 2025); (3) GTs: GT (Dwivedi & Bresson, 2021), TokenGT (Kim et al., 2022), GraphGPS (Rampášek et al., 2022). Their details are in Appendix C.2. In particular, GAT is unique in combining attention mechanisms with message passing.

5.1. Q1: Feature Learning Drives Generalizability

> **Key Takeaway:** While high accuracy on the same dataset can be achieved by GNNs in the kernel regime, generalizability to other datasets is driven by the feature learning regime.

To assess a model's inherent generalizability to other datasets, we adopt a transfer learning setup. Recall that KTA measures the degree of feature learning. Specifically, we pre-train GNNs and evaluate their test accuracy and KTA on a source data, then evaluate the frozen pre-trained GNNs on a target data using a linear probe (Alain & Bengio, 2017). The resulting transfer accuracy directly reflects the generalizability of the GNNs to other datasets. We then compare the correlation between the transfer accuracy and test accuracy or KTA. We study three settings for graph classification: (1) MNIST with digits 0–4 $\rightarrow$ digits 5–9, (2) NCI1$\rightarrow$NCI109, and (3) IMDB-MULTI$\rightarrow$IMDB-BINARY (Morris et al., 2020), covering both graph structure and feature shifts, with (3) being the most challenging.



(a) NCI1 $\rightarrow$ NCI109: test accuracy vs transfer accuracy. *(b)* NCI1 $\rightarrow$ NCI109: KTA vs transfer accuracy.

Figure 1. The correlation between transfer accuracy on NCI109 and test accuracy on NCI1 (left), and the correlation between transfer accuracy on NCI109 and KTA on NCI1 (right). Each point represents a model. p stands for p-value. A good alignment is indicated by points closely following the linear fit (the dashed line) and by high values of Spearman's ρ and Kendall's τ. KTA is a better indicator of generalizability than test accuracy.

As shown in Figure 1 and Table 1, KTA is a better indicator of GNNs' generalizability to other datasets than test accuracy. In Figure 1, we plot for NCI1$\rightarrow$NCI109: (1) the correlation between transfer accuracy and test accuracy, and (2) the correlation between transfer accuracy and KTA. The results for all three experiments are summarized in Table 1. These results show that KTA is consistently more correlated with transfer accuracy than test accuracy. Full figures are in Appendix D.1. Furthermore, in Appendix E.1, we show

on three node classification dataset pairs that KTA remains more indicative of GNNs' generalizability to other datasets. In addition, we show in Appendix F.3 that KTA is also indicative of out-of-distribution generalization performance.

Table 1. The correlation between transfer accuracy and test accuracy (column 2), and the correlation between transfer accuracy and KTA (column 3), measured by Spearman's ρ and Kendall's τ. p stands for p-value. A strong correlation is indicated by high values of ρ and τ. *Test* refers to test accuracy and *Transfer* refers to transfer accuracy. KTA is strongly correlated with transfer accuracy compared with test accuracy.

Task	Test vs Transfer		KTA vs Transfer	
	ρ (p)	τ (p)	ρ (p)	τ (p)
MNIST 0-4→5-9	0.75 (0.0033)	0.56 (0.0067)	0.89 (4.6×10^{-5})	0.72 (2.8×10^{-4})
NCI1→NCI109	0.21 (0.48)	0.18 (0.44)	0.90 (3.5×10^{-5})	0.74 (1.5×10^{-4})
IMDB-MULTI→BINARY	0.59 (0.021)	0.44 (0.023)	0.72 (0.0023)	0.59 (0.0021)

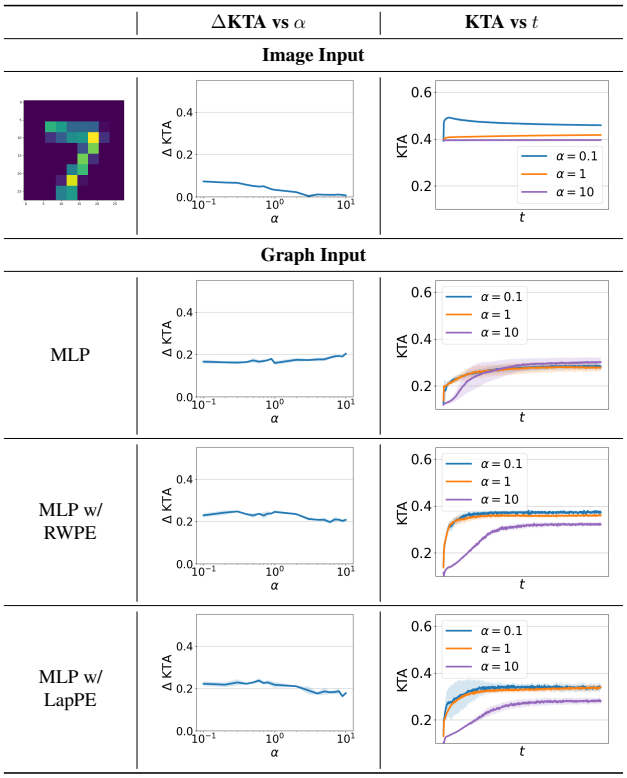
These findings challenge the conventional focus on test accuracy, showing that it alone cannot faithfully assess the generalizability of GNNs to other datasets. Unlike in vision and language where foundation models typically work in the feature learning regime with high test accuracy both on the same dataset and across datasets (Kornblith et al., 2019c; Brown et al., 2020), on graph data, high accuracy on the same dataset can be achieved by GNNs with poorly learned representations and low generalizability in the kernel regime, and assessing GNN generalizability across datasets calls for a feature learning perspective. Next, we examine how data and model designs influence whether GNNs work in the feature learning regime.

5.2. Q2: Impact of Graph Data Format

Key Takeaway: Representing regular data as graphs may hinder feature learning, but positional encodings are helpful.

To evaluate the impact of data formats, we study the feature learning behavior of MLPs on MNIST in both image and graph formats. Prior works have shown that MLPs can easily work in the feature learning regime with MNIST as pixels (Fort et al., 2020; Radhakrishnan et al., 2024; Beaglehole et al., 2024). Therefore, a question naturally arises: when MNIST is represented as graphs, can we observe similar feature learning behavior? To address this, following standard practice, we first convert MNIST into superpixel images and graphs (Dwivedi et al., 2023). Specifically, the original images are downsampled to superpixel images, and pixel values within each superpixel are averaged and unified (first row, left cell of Table 2). In the graph format, each superpixel becomes a node, with node features including its pixel intensity and 2D coordinates in the image (Dwivedi et al., 2023). We demonstrate the feature learning behavior of the models by plotting (1) the KTA change after training (ΔKTA) across α and (2) the KTA curves during training under $\alpha = 0.1, 1, 10$.

Table 2. ΔKTA across different α values (column 2) and KTA curves during training (column 3). Top row: superpixel image MNIST; Other rows: graph MNIST. KTA is evaluated on the test set. Results are averaged over three random seeds; the shaded regions denote the standard deviations. (1) MLP works in the feature learning regime for image MNIST. (2) MLP does not work in the feature learning regime for graph MNSIT. (3) Positional encodings promote feature learning.



As shown in Table 2, representing MNIST images as graphs hinders feature learning. Specifically, the MLP on images shows a decreasing KTA with increasing α and a sharp early rise in the KTA curve at $\alpha = 0.1$, both indicative of feature learning. In contrast, for MLPs on the graph inputs, the KTA does not decrease with α, and the KTA curve at $\alpha = 0.1$ increases slowly, suggesting that MLPs do not work in the feature learning regime here. To better incorporate graph structural information, we use positional encodings as additional inputs. As shown in Table 2, they induce a slightly decreasing KTA trend and an elevated KTA curve at $\alpha = 0.1$, indicating a partial transition from the kernel to the feature learning regime. We note the high initial KTA on images (first row, right cell of Table 2), as raw pixel inputs already contain strong class-relevant patterns in MNIST, enabling the initial MLP to produce moderately label-aligned representations. Therefore, due to the different data formats, here we focus on comparing the KTA trends rather than absolute values.

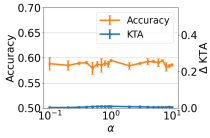
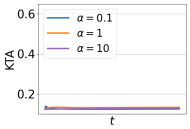
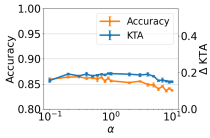
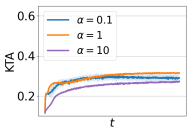
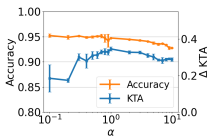
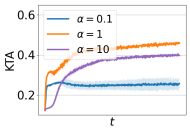
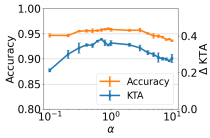
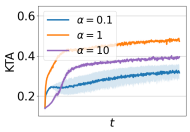
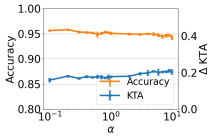
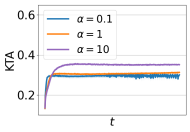
5.3. Q3: Impact of Design Choices

> **Key Takeaway:** Most MPNNs exhibit limited feature learning, but interleaving MPNN layers with MLPs or incorporating attention mechanisms promotes it. Graph transformers are most effective.

We now explore the impact of design choices on feature learning. Specifically, we use SGC as a linear baseline that primarily works in the kernel regime, and compare standard MPNNs with (1) GCN$^+$ to evaluate the effect of interleaving MPNN layers with MLPs, whose architecture alternates between GCN layers and MLP blocks, and (2) GAT and graph transformers to evaluate the effect of attention mechanisms. Based on the results in Table 3 and 4, we observe:

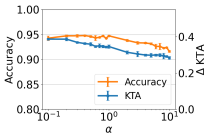
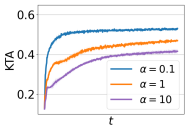
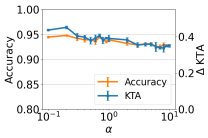
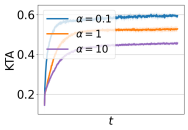
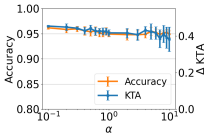
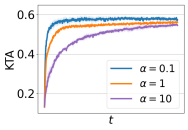
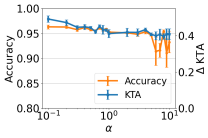
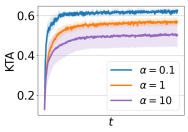
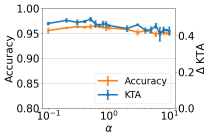
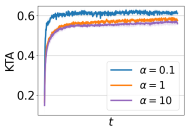
Most MPNNs do not work in the feature learning regime. GCN, GCNII, GraphSAGE, and A-DGN in Table 3 do not exhibit a decreasing KTA with an increasing α. At $\alpha = 0.1$, their KTA curves increase very slowly, and the KTA values can be even lower than those at $\alpha = 1$ or 10. Similar to SGC whose KTA values remain near 0, these models also struggle to work in the feature learning regime.

Table 3. ΔKTA across different α values (column 2) and KTA curves over training epochs (column 3). Accuracy and KTA are evaluated on the test set. Results are averaged over three random seeds; the error bars and shaded regions denote the standard deviations. Most MPNNs do not work in the feature learning regime.

	Accuracy/ΔKTA vs α	KTA vs t
SGC		
GCN		
GCNII		
GraphSAGE		
A-DGN		

Interleaving MPNN layers with MLPs and attention mechanisms promote feature learning, with graph transformers being most effective. In contrast, in Table 4, (1) GCN$^+$, which augments GCN by interleaving GCN layers with MLP blocks, exhibits significantly stronger feature learning than GCN, as indicated by a clearly decreasing KTA as α increases, well-separated KTA curves at different α, and substantially higher KTA values. (2) Attention-based GNNs, including GAT and graph transformers, show a general decrease in KTA with α. Their KTA curves are clearly separated by α, following the order $\alpha = 0.1, 1, 10$ from highest to lowest. Also, their KTA values are substantially higher than those of MPNNs. For example, KTA values for GraphGPS easily exceed 0.5, while the peak KTA for GCN sits around 0.3. (3) Notably, graph transformers, which incorporate MLP interleaving, attention mechanisms, and positional encodings, exhibit the strongest feature learning, as evidenced by their high KTA values.

Table 4. ΔKTA across different α values (column 2) and KTA curves over training epochs (column 3). Accuracy and KTA are evaluated on the test set. Results are averaged over three random seeds; the error bars and shaded regions denote the standard deviations. MPNNs interleaved with MLPs and attention-based architectures exhibit stronger feature learning, with graph transformers being the most effective.

	Accuracy/ΔKTA vs α	KTA vs t
GCN$^+$		
GAT		
GT		
TokenGT		
GraphGPS		

We verify our findings on large scale OGBG-MOLHIV and OGBG-PPA in Table 5. We summarize the peak KTA value for each model at $\alpha = 0.1, 1, 10$, and for simplicity, we

present the results of six representative models to cover the discussed feature learning factors. We can still verify that feature learning is promoted by (1) positional encodings (MLP with LapPE), (2) interleaving with MLP layers (GCN^+), and (3) attention mechanisms (GAT, GraphGPS).

Table 5. KTA values for each model and α on OGBG-MOLHIV and OGBG-PPA. KTA is evaluated on the test set. The results are averaged over three random seeds with mean and standard deviation reported. The highest KTA across α under the same model and dataset is underlined. If the highest KTA of a model on a dataset remains low across different α, then the model does not operate in the feature learning regime. Feature learning is promoted by (1) positional encodings, (2) interleaving with MLP layers, and (3) attention mechanisms.

Model	OGBG-MOLHIV			OGBG-PPA		
	$\alpha = 0.1$	$\alpha = 1$	$\alpha = 10$	$\alpha = 0.1$	$\alpha = 1$	$\alpha = 10$
MLP	$\underline{0.016 \pm 0.003}$	0.014 ± 0.002	0.014 ± 0.000	0.244 ± 0.008	$\underline{0.248 \pm 0.005}$	0.196 ± 0.011
MLP w/ LapPE	0.027 ± 0.000	$\underline{0.028 \pm 0.002}$	0.024 ± 0.002	0.234 ± 0.005	$\underline{0.252 \pm 0.003}$	0.227 ± 0.009
GCN	0.008 ± 0.002	$\underline{0.013 \pm 0.002}$	0.006 ± 0.000	0.201 ± 0.005	$\underline{0.230 \pm 0.004}$	0.180 ± 0.007
GCN^+	0.035 ± 0.003	$\underline{0.039 \pm 0.001}$	0.018 ± 0.001	$\underline{0.301 \pm 0.009}$	0.278 ± 0.006	0.164 ± 0.007
GAT	$\underline{0.049 \pm 0.002}$	0.048 ± 0.006	0.029 ± 0.013	$\underline{0.312 \pm 0.004}$	0.289 ± 0.007	0.235 ± 0.010
GraphGPS	0.053 ± 0.009	$\underline{0.057 \pm 0.007}$	0.014 ± 0.005	$\underline{0.323 \pm 0.003}$	0.300 ± 0.002	0.263 ± 0.008

To show the broad applicability of our findings, in Appendix D.2, we verify them on three more graph classification datasets. Furthermore, in Appendix E.2, we verify them on eight node classification datasets. In addition, our framework is generally applicable. Besides the initial weight scale α, we can also sweep learning rates to prompt a regime transition, leading to the same conclusions, as shown in Appendix F.2. Moreover, in Appendix F.1, we show measuring the alignment between initial and final feature kernels leads to the same conclusions as measuring KTA.

While prior works showed the advantage of positional encodings and MLP interleaving in improving test accuracy on the same dataset (Dwivedi et al., 2022a; Luo et al., 2025), we show their advantage in enhancing generalizability to other datasets from the feature learning perspective. Although the relative advantages of graph transformers and MPNNs remain under debate (Luo et al., 2024; 2025) due to focus on same-dataset test accuracy, based on the insights from our framework, we highlight graph transformers as a promising GNN architecture for feature learning and high generalizability on various data. Moreover, our framework does not depend on source-target dataset pairs and directly evaluates a model's inherent generalizability.

6. Related Works

GNN Feature Learning Prior work on GNN feature learning is largely theoretical and focuses on the Graph Neural Tangent Kernel (GNTK). These works (Du et al., 2019; Huang et al., 2022; Sabanayagam et al., 2023; Yang et al., 2023a) provide theoretical insights into how data characteristics and architectural choices influence GNTK. Specifically, Du et al. (2019) formulated GNTK. Yang et al. (2023a) theoretically attributed the improvement in test accuracy on node classification of GNNs over MLPs to message passing in the kernel regime. Khalafi et al. (2024) theoretically showed a high initial alignment between NTK and target labels benefits optimization and generalization in the same dataset for GNNs. Yang et al. (2024) theoretically studied the NTK dynamics of GNNs in node classification tasks and demonstrated that a parameter-free method can overfit a given homophilic graph. Aminian et al. (2024) derived the error bound of a one-layer GNN in the kernel regime. These works established theoretical conditions and guarantees of GNN feature learning under specific settings and validated their conclusions on synthetic or tiny real-world data. On the contrary, our work goes beyond message passing architectures and same-dataset generalization, and proposes the first theoretically grounded and practical framework to systematically diagnose GNN generalization to other datasets through feature learning across diverse real-world datasets, including large scale node-level and graph-level datasets.

Feature Learning In general feature learning literature, many works studied the theoretical aspect with linear models or MLPs on synthetic or classic real-world data (Geiger et al., 2020; Woodworth et al., 2020; Baratin et al., 2021; Damian et al., 2022; Shan & Bordelon, 2022; Atanasov et al., 2022; Ba et al., 2022; Kothapalli et al., 2025). On the empirical side, the majority focus on vision and language tasks (Fort et al., 2020; Martin & Mahoney, 2021; Yang et al., 2023b; Zhang & Bottou, 2023; Radhakrishnan et al., 2024; Beaglehole et al., 2024), and the study of feature learning in GNNs has been limited. In terms of KTA, Redko & Bennani (2016) treated KTA as a training objective to reduce distribution gaps. Cristianini et al. (2001) showed a good KTA implies good generalization on the same dataset. Yang et al. (2024) used KTA to track training dynamics for node classification. Instead, we study KTA as a feature learning metric and show KTA is indicative of a model's inherent generalizability to other datasets.

7. Conclusion

In this work, we highlight the new problem of GNN feature learning diagnosis. We assess feature learning of various GNNs, emphasizing graph transformers as a promising architecture. Moreover, we show that different from other domains, having GNNs work in the kernel regime is often sufficient to achieve high test accuracy on the same dataset, but generalizability to other datasets requires feature learning. Our findings represent a step toward addressing fundamental challenges in graph learning, especially generalizability across datasets, which is crucial for GFMs.

Impact Statement

This paper presents work whose goal is to advance the research of graph neural network generalizability. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Alain, G. and Bengio, Y. Understanding intermediate layers using linear classifier probes, 2017. URL https://openreview.net/forum?id=ryF7rTqgl.

Aminian, G., He, Y., Reinert, G., Szpruch, L., and Cohen, S. N. Generalization error of graph neural networks in the mean-field regime. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Atanasov, A., Bordelon, B., and Pehlevan, C. Neural networks as kernel learners: The silent alignment effect. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=1NvflqAdoom.

Ba, J., Erdogdu, M. A., Suzuki, T., Wang, Z., Wu, D., and Yang, G. High-dimensional asymptotics of feature learning: how one gradient step improves the representation. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Baratin, A., George, T., Laurent, C., Devon Hjelm, R., Lajoie, G., Vincent, P., and Lacoste-Julien, S. Implicit regularization via neural feature alignment. In Banerjee, A. and Fukumizu, K. (eds.), *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 2269–2277. PMLR, 13–15 Apr 2021. URL https://proceedings.mlr.press/v130/baratin21a.html.

Beaglehole, D., Mitliagkas, I., and Agarwala, A. Feature learning as alignment: a structural property of gradient descent in non-linear neural networks. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=JXCe2ZcUXr.

Bechler-Speicher, M., Finkelshtein, B., Frasca, F., Müller, L., Tönshoff, J., Siraudin, A., Zaverkin, V., Bronstein, M. M., Niepert, M., Perozzi, B., Galkin, M., and Morris, C. Position: Graph learning will lose relevance due to poor benchmarks. In *Forty-second International Conference on Machine Learning Position Paper Track*, 2025. URL https://openreview.net/forum?id=nDFpl2lhoH.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Chen, M., Wei, Z., Huang, Z., Ding, B., and Li, Y. Simple and deep graph convolutional networks. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1725–1735. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/chen20v.html.

Chen, Z., Mao, H., Liu, J., Song, Y., Li, B., Jin, W., Fatemi, B., Tsitsulin, A., Perozzi, B., Liu, H., and Tang, J. Text-space graph foundation models: Comprehensive benchmarks and new insights. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 7464–7492. Curran Associates, Inc., 2024. doi: 10.52202/079017-0239. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/0e0b39c69663e9c073739adf547ed778-Paper-Datasets_and_Benchmarks_Track.pdf.

Chizat, L., Oyallon, E., and Bach, F. On lazy training in differentiable programming. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/ae614c557843b1df326cb29c57225459-Paper.pdf.

Chou, C.-N., Le, H., Wang, Y., and Chung, S. Feature learning beyond the lazy-rich dichotomy: Insights from representational geometry. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=gKdjHLrHDS.

Cristianini, N., Shawe-Taylor, J., Elisseeff, A., and Kandola, J. On kernel-target alignment. In *Proceedings of the 15th International Conference on Neural Information Processing Systems: Natural and Synthetic*, NIPS'01, pp. 367–373, Cambridge, MA, USA, 2001. MIT Press.

Damian, A., Lee, J., and Soltanolkotabi, M. Neural networks can learn representations with gradient descent. In Loh, P.-L. and Raginsky, M. (eds.), *Proceedings of Thirty Fifth Conference on Learning Theory*, volume 178 of *Proceedings of Machine Learning Research*, pp. 5413–5452. PMLR, 02–05 Jul 2022. URL https://proceedings.mlr.press/v178/damian22a.html.

Dominé, C. C. J., Anguita, N., Proca, A. M., Braun, L., Kunin, D., Mediano, P. A. M., and Saxe, A. M. From lazy to rich: Exact learning dynamics in deep linear networks. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=ZXaocmXc6d.

Du, S., Hou, K., Salakhutdinov, R. R., Poczos, B., Wang, R., and Xu, K. Graph neural tangent kernel: Fusing graph neural networks with graph kernels. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/663fd3c5144fd10bd5ca6611a9a5b92d-Paper.pdf.

Dwivedi, V. P. and Bresson, X. A generalization of transformer networks to graphs, 2021. URL https://arxiv.org/abs/2012.09699.

Dwivedi, V. P., Luu, A. T., Laurent, T., Bengio, Y., and Bresson, X. Graph neural networks with learnable structural and positional representations. In *International Conference on Learning Representations*, 2022a. URL https://openreview.net/forum?id=wTTjnvGphYj.

Dwivedi, V. P., Rampášek, L., Galkin, M., Parviz, A., Wolf, G., Luu, A. T., and Beaini, D. Long range graph benchmark. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022b. URL https://openreview.net/forum?id=in7XC5RcjEn.

Dwivedi, V. P., Joshi, C. K., Luu, A. T., Laurent, T., Bengio, Y., and Bresson, X. Benchmarking graph neural networks. *J. Mach. Learn. Res.*, 24(1), January 2023. ISSN 1532-4435.

Fey, M. and Lenssen, J. E. Fast graph representation learning with pytorch geometric, 2019. URL https://arxiv.org/abs/1903.02428.

Fort, S., Dziugaite, G. K., Paul, M., Kharaghani, S., Roy, D. M., and Ganguli, S. Deep learning versus kernel learning: an empirical study of loss landscape geometry and the time evolution of the neural tangent kernel. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Frasca, F., Jogl, F., Eliasof, M., Ostrovsky, M., Schönlieb, C.-B., Gärtner, T., and Maron, H. Towards foundation models on graphs: An analysis on cross-dataset transfer of pretrained gnns, 2024. URL https://arxiv.org/abs/2412.17609.

Galkin, M., Yuan, X., Mostafa, H., Tang, J., and Zhu, Z. Towards foundation models for knowledge graph reasoning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=jVEoydFOl9.

Garg, V. K., Jegelka, S., and Jaakkola, T. Generalization and representational limits of graph neural networks. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Geiger, M., Spigler, S., Jacot, A., and Wyart, M. Disentangling feature and lazy training in deep neural networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(11):113301, nov 2020. doi: 10.1088/1742-5468/abc4de. URL https://dx.doi.org/10.1088/1742-5468/abc4de.

Ghorbani, B., Mei, S., Misiakiewicz, T., and Montanari, A. Limitations of lazy training of two-layers neural network. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/c133fb1bb634af68c5088f3438848bfd-Paper.pdf.

Göring, N. A., London, C., Erturk, A. H., Mingard, C., Nam, Y., and Louis, A. A. Feature learning is decoupled from generalization in high capacity neural networks. In *High-dimensional Learning Dynamics 2025*, 2025. URL https://openreview.net/forum?id=pVJpk4tuVu.

Gravina, A., Bacciu, D., and Gallicchio, C. Anti-symmetric DGN: a stable architecture for deep graph networks. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=J3Y7cgZOOS.

Hamilton, W. L., Ying, R., and Leskovec, J. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 1025–1035, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

He, B. and Ozay, M. Feature kernel distillation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=tBIQEvApZK5.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90.

Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., and Leskovec, J. Open graph benchmark: datasets for machine learning on graphs. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Huang, W., Li, Y., weitao Du, Xu, R., Yin, J., Chen, L., and Zhang, M. Towards deepening graph neural networks: A GNTK-based optimization perspective. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=tT9t_ZctZRL.

Huang, W., Cao, Y., Wang, H., Cao, X., and Suzuki, T. Quantifying the optimization and generalization advantages of graph neural networks over multilayer perceptrons. In Li, Y., Mandt, S., Agrawal, S., and Khan, E. (eds.), *Proceedings of The 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pp. 2854–2862. PMLR, 03–05 May 2025. URL https://proceedings.mlr.press/v258/huang25a.html.

Jiang, K., Cohen, J., and Li, Y. Understanding the evolution of the neural tangent kernel at the edge of stability. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=QKo4c3LAz3.

Ju, L., Yang, X., Li, Q., and Wang, X. Graphbridge: Towards arbitrary transfer learning in GNNs. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=gjRhw5S3A4.

Khalafi, S., Sihag, S., and Ribeiro, A. Neural tangent kernels motivate cross-covariance graphs in neural networks. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Kim, J., Nguyen, T. D., Min, S., Cho, S., Lee, M., Lee, H., and Hong, S. Pure transformers are powerful graph learners. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y. (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=SJU4ayYgl.

Kornblith, S., Norouzi, M., Lee, H., and Hinton, G. Similarity of neural network representations revisited. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3519–3529. PMLR, 09–15 Jun 2019a. URL https://proceedings.mlr.press/v97/kornblith19a.html.

Kornblith, S., Shlens, J., and Le, Q. V. Do better imagenet models transfer better? In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2656–2666, 2019b. doi: 10.1109/CVPR.2019.00277.

Kornblith, S., Shlens, J., and Le, Q. V. Do better imagenet models transfer better? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2661–2671, 2019c.

Kothapalli, V., Pang, T., Deng, S., Liu, Z., and Yang, Y. From spikes to heavy tails: Unveiling the spectral evolution of neural networks. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=DJHB8eBUnt.

Kunin, D., Raventos, A., Dominé, C. C. J., Chen, F., Klindt, D., Saxe, A. M., and Ganguli, S. Get rich quick: exact solutions reveal how unbalanced initializations promote rapid feature learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=eNM94i7R3A.

Li, Q., Han, Z., and Wu, X.-M. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications*

of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence, AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018. ISBN 978-1-57735-800-8.

Luo, Y., Shi, L., and Wu, X.-M. Classic GNNs are strong baselines: Reassessing GNNs for node classification. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=xkljKdGe4E.

Luo, Y., Shi, L., and Wu, X.-M. Can classic GNNs be strong baselines for graph-level tasks? simple architectures meet excellence. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=ZH7YgIZ3DF.

Martin, C. H. and Mahoney, M. W. Implicit self-regularization in deep neural networks: evidence from random matrix theory and implications for learning. *J. Mach. Learn. Res.*, 22(1), January 2021. ISSN 1532-4435.

Méndez-Lucio, O., Nicolaou, C. A., and Earnshaw, B. Mole: a foundation model for molecular graphs using disentangled attention. *Nature communications*, 15(1):9431, 2024.

Morris, C., Kriege, N. M., Bause, F., Kersting, K., Mutzel, P., and Neumann, M. Tudataset: A collection of benchmark datasets for learning with graphs. In *ICML 2020 Workshop on Graph Representation Learning and Beyond (GRL+ 2020)*, 2020. URL www.graphlearning.io.

Radhakrishnan, A., Beaglehole, D., Pandit, P., and Belkin, M. Mechanism for feature learning in neural networks and backpropagation-free machine learning models. *Science*, 383(6690):1461–1467, 2024. doi: 10.1126/science.adi5639. URL https://www.science.org/doi/abs/10.1126/science.adi5639.

Rampášek, L., Galkin, M., Dwivedi, V. P., Luu, A. T., Wolf, G., and Beaini, D. Recipe for a general, powerful, scalable graph transformer. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Redko, I. and Bennani, Y. Kernel alignment for unsupervised transfer learning, 2016. URL https://arxiv.org/abs/1610.06434.

Rozemberczki, B., Allen, C., and Sarkar, R. Multi-scale attributed node embedding. *Journal of Complex Networks*, 9(2):cnab014, 04 2021. ISSN 2051-1329. doi: 10.1093/comnet/cnab014. URL https://doi.org/10.1093/comnet/cnab014.

Sabanayagam, M., Esser, P., and Ghoshdastidar, D. Analysis of convolutions, non-linearity and depth in graph neural networks using neural tangent kernel. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=xgYgDEof29.

Samarin, M., Roth, V., and Belius, D. Feature learning and random features in standard finite-width convolutional neural networks: An empirical study. In *The 38th Conference on Uncertainty in Artificial Intelligence*, 2022. URL https://openreview.net/forum?id=ScIEZdIiqe5.

Shan, H. and Bordelon, B. A theory of neural tangent kernel alignment and its influence on training, 2022. URL https://arxiv.org/abs/2105.14301.

Shchur, O., Mumme, M., Bojchevski, A., and Günnemann, S. Pitfalls of graph neural network evaluation, 2019. URL https://arxiv.org/abs/1811.05868.

Simon, J., Kunin, D., Atanasov, A., Boix-Adserà, E., Bordelon, B., Cohen, J., Ghosh, N., Guth, F., Jacot, A., Kamb, M., Karkada, D., Michaud, E. J., Ottlik, B., and Turnbull, J. There will be a scientific theory of deep learning, 2026. URL https://arxiv.org/abs/2604.21691.

Tian, Y. Provable scaling laws of feature emergence from learning dynamics of grokking, 2025. URL https://arxiv.org/abs/2509.21519.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. Graph attention networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJXMpikCZ.

Wang, Y., Calhoun, V. D., Pearlson, G. D., Kochunov, P., van Erp, T. G., and Du, Y. A graph transformer-based foundation model for brain functional connectivity network. *Pattern Recognition*, 169:111988, 2026. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2025.111988.

URL https://www.sciencedirect.com/science/article/pii/S003132032500648X.

Wang, Z., Engel, A. W., Sarwate, A., Dumitriu, I., and Chiang, T. Spectral evolution and invariance in linear-width neural networks. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=gpqBGyKeKH.

Wang, Z., Liu, Z., Ma, T., Li, J., Zhang, Z., Fu, X., Li, Y., Yuan, Z., Song, W., Ma, Y., Zeng, Q., Chen, X., Zhao, J., Li, J., Jiang, M., Lio, P., Chawla, N., Zhang, C., and Ye, Y. Graph foundation models: A comprehensive survey, 2025. URL https://arxiv.org/abs/2505.15116.

Woodworth, B., Gunasekar, S., Lee, J. D., Moroshko, E., Savarese, P., Golan, I., Soudry, D., and Srebro, N. Kernel and rich regimes in overparametrized models. In Abernethy, J. and Agarwal, S. (eds.), *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pp. 3635–3673. PMLR, 09–12 Jul 2020. URL https://proceedings.mlr.press/v125/woodworth20a.html.

Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., and Weinberger, K. Simplifying graph convolutional networks. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6861–6871. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/wu19e.html.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=ryGs6iA5Km.

Yang, C., Wu, Q., Wang, J., and Yan, J. Graph neural networks are inherently good generalizers: Insights by bridging GNNs and MLPs. In *The Eleventh International Conference on Learning Representations*, 2023a. URL https://openreview.net/forum?id=dqnNW2omZL6.

Yang, C., Wu, Q., Wipf, D., Sun, R., and Yan, J. How graph neural networks learn: Lessons from training dynamics. In Salakhutdinov, R., Kolter, Z., Heller, K., Weller, A., Oliver, N., Scarlett, J., and Berkenkamp, F. (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 56594–56623. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/yang24ae.html.

Yang, Y., Theisen, R., Hodgkinson, L., Gonzalez, J. E., Ramchandran, K., Martin, C. H., and Mahoney, M. W. Test accuracy vs. generalization gap: Model selection in nlp without accessing training or testing data. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, pp. 3011–3021, New York, NY, USA, 2023b. Association for Computing Machinery. ISBN 9798400701030. doi: 10.1145/3580305.3599518. URL https://doi.org/10.1145/3580305.3599518.

Yang, Z., Cohen, W. W., and Salakhutdinov, R. Revisiting semi-supervised learning with graph embeddings. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pp. 40–48. JMLR.org, 2016.

Zhang, J. and Bottou, L. Learning useful representations for shifting tasks and distributions. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Zhu, J., Yan, Y., Zhao, L., Heimann, M., Akoglu, L., and Koutra, D. Beyond homophily in graph neural networks: current limitations and effective designs. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Zhu, Q., Yang, C., Xu, Y., Wang, H., Zhang, C., and Han, J. Transfer learning of graph neural networks with ego-graph information maximization. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=CzVPfeqPOBu.

A. Proofs

We establish in Appendix A.1 high-probability bounds showing that, at initialization under graph convolution, the noise vectors and the inner products between weights and feature components are well controlled. Moreover, the aggregated node type $\tilde{z}_i^G$ can be bounded by the node types z_i^G and homophily level Ξ. Building on these preliminary bounds, for all times $0 \le t \le T^*$, Appendix A.2 derives the exact coefficient iterations. It further establishes $\gamma_{j,r}^{(t)} \ge 0$ which grows monotonically with high probability, and shows that both $\gamma_{j,r}^{(t)}$ and $|\rho_{j,r,i}^{G(t)}|$ admit uniform bounds. Finally, Appendix A.3 proves that, within this controlled regime, $\gamma_{j,r}^{(t)}$ reaches a constant threshold after time T_1, while $|\rho_{j,r,i}^{G(t)}|$ remains small under an signal-noise-ratio (SNR) condition. Below we first formulate the notations and assumptions used throughout our analysis.

Mathematical Notations We use lower bold-faced letters for vectors, upper bold-faced letters for matrices, and non-bold-faced letters for scalars. For a vector $\mathbf{v}$, its ℓ_2-norm is denoted as $\|\mathbf{v}\|_2$. For a matrix $\mathbf{A}$, we use $\|\mathbf{A}\|_2$ to denote its spectral norm and $\|\mathbf{A}\|_{\mathrm{F}}$ for its Frobenius norm. We employ standard asymptotic notations such as $O(\cdot)$, $o(\cdot)$, $\Omega(\cdot)$, and $\Theta(\cdot)$ to describe the limiting behavior. We use $\tilde{O}(\cdot)$, $\tilde{\Omega}(\cdot)$, and $\tilde{\Theta}(\cdot)$ to hide logarithmic factors in these notations respectively. Moreover, we denote $a_n = \mathrm{poly}(b_n)$ if $a_n = O((b_n)^p)$ for some positive constant p and $a_n = \mathrm{polylog}(b_n)$ if $a_n = \mathrm{poly}(\log(b_n))$. Also, we use $a \vee b$ to denote the bigger value of any pairs of a, b. In addition, sequences of integers are denoted as $[m] = \{1, 2, \ldots, m\}$. Lastly, we use $\#$ to denote the number of elements in a set.

Assumptions Our analysis is based on the following assumptions:

Assumption A.1. (Node-level conditions). (1) *Sufficiently large number of nodes:* $\underline{n} = \Omega(\mathrm{polylog}(d))$. (2) *Homophily:* $p, s = \Omega(\sqrt{\frac{\log \overline{n}}{\underline{n}}})$, and $\Xi \triangleq \frac{p-s}{p+s}$ is a positive constant defined as the *homophily level*.

(Graph-level conditions). (3) *Sufficiently large number of graphs:* $N = \Omega(\mathrm{polylog}(d))$. (4) *Label–type alignment:* for each graph G, the majority of nodes have latent types aligned with the graph label: $\#\{i \in V^G : y_G z_i^G > 0\} \ge (1-\epsilon) n_G$, where $0 < \epsilon < \frac{1}{2}$ is a *misalignment rate*. (5) *Dominant aligned contribution:* the aggregated contribution of aligned nodes dominates that of misaligned nodes: $\Delta > 0$ (defined in Lemma A.10).

(Model and optimization conditions). (6) *Sufficiently large hidden dimension:* $d = \tilde{\Omega}((Nm)^{2 \vee \frac{4}{q-2}} \overline{n}^{4 \vee \frac{2q-2}{q-2}})$. (7) *Sufficiently large number of hidden neurons:* $m = \Omega(\mathrm{polylog}(d))$. (8) *Small learning rate and initialization scale:* $\eta \le \tilde{O}(\min\{\overline{n}^{-1}\|\boldsymbol{\mu}\|_2^{-2},\ \sigma_p^{-2} d^{-1}\})$ and $\sigma_0 \le \tilde{O}(m^{-\frac{2}{q-2}} \overline{n}^{-[\frac{1}{q-2}] \vee 1} \min\{\Xi^{-1}\|\boldsymbol{\mu}\|_2^{-1},\ (\sigma_p \sqrt{d/(\underline{n}(p+s))})^{-1}\})$. (9) *Polynomial ReLU:* $\sigma(\cdot)$ is a polynomial ReLU activation function $\sigma(z) \triangleq \max\{0, z\}^q$ for some $q > 2$. (10) *Bounded loss gradient:* For all G, t, the loss gradient is bounded: $0 < \underline{\ell} \le -\ell_G^{'(t)} \le \overline{\ell}$.

A.1. Preliminary Lemmas

In this section, we introduce the preliminary lemmas from the data generation process. Since the data generation per graph is the same as the SNM-SBM framework in (Huang et al., 2025) which established the lemmas on a single graph, by applying a union bound over all graphs, we have the following lemmas:

Lemma A.2. *Suppose* $d = \Omega(\log(4\overline{n}/\delta))$. *Then with probability at least* $1 - \delta$,

$$\sigma_p^2 d/2 \le \|\boldsymbol{\xi}_i^G\|_2^2 \le 3\sigma_p^2 d/2,$$

$$|\langle \boldsymbol{\xi}_i^G, \boldsymbol{\xi}_{i'}^{G'} \rangle| \le 2\sigma_p^2 \cdot \sqrt{d \log(4\overline{n}^2/\delta)},$$

for all $i \in [n_G], G \in \mathcal{G}$ *and* $i' \in [n_{G'}], G' \in \mathcal{G}$.

Lemma A.3. *Suppose that* $d = \Omega(\log(\overline{n}m/\delta))$, $m = \Omega(\log(1/\delta))$. *Then with probability at least* $1 - \delta$,

$$\left|\langle \mathbf{w}_{j,r}^{(0)}, \boldsymbol{\mu} \rangle\right| \le \sqrt{2\log(8m/\delta)} \cdot \sigma_0 \|\boldsymbol{\mu}\|_2,$$

$$\left|\langle \mathbf{w}_{j,r}^{(0)}, \boldsymbol{\xi}_i^G \rangle\right| \le 2\sqrt{\log(8m\overline{n}/\delta)} \cdot \sigma_0 \sigma_p \sqrt{d},$$

for all $r \in [m]$, $j \in \{\pm 1\}$ *and* $i \in [n_G], G \in \mathcal{G}$. *Moreover,*

$$\sigma_0 \|\boldsymbol{\mu}\|_2/2 \le \max_{r \in [m]} j \cdot \langle \mathbf{w}_{j,r}^{(0)}, \boldsymbol{\mu} \rangle \le \sqrt{2\log(8m/\delta)} \cdot \sigma_0 \|\boldsymbol{\mu}\|_2,$$

$$\sigma_0\sigma_p\sqrt{d}/4 \le \max_{r\in[m]} j\cdot\langle \mathbf{w}_{j,r}^{(0)}, \boldsymbol{\xi}_i^G\rangle \le 2\sqrt{\log(8m\overline{n}/\delta)}\cdot\sigma_0\sigma_p\sqrt{d},$$

for all $j\in\{\pm1\}$ *and* $i\in[n_G], G\in\mathcal{G}$.

Lemma A.4. *Let* $p, s = \Omega\Big(\sqrt{\frac{\log(\overline{n}/\delta)}{\underline{n}}}\Big)$*, then with probability at least* $1-\delta$*, we have*

$$\underline{n}(p+s)/4 \le D_i^G \le 3\overline{n}(p+s)/4$$

for all $i\in[n_G], G\in\mathcal{G}$.

Lemma A.5. *Suppose* $\underline{n}\ge 8\,\frac{p+s}{(p-s)^2}\log(4/\delta)$*. Then with probability at least* $1-\delta$*, there exist constants* $0<c<1<C$ *such that*

$$\mathrm{sign}(\tilde{z}_i^G) = \mathrm{sign}(z_i^G) \text{ and } c\Xi \le |\tilde{z}_i^G| \le C\Xi$$

for all $i\in[n_G], G\in\mathcal{G}$.

Remark A.6. In Lemma A.4 in (Huang et al., 2025), they imply $\mathrm{sign}(\tilde{y}_i)=\mathrm{sign}(y_i)$ via $\frac{1}{2}\Xi|y_i|\le|\tilde{y}_i|\le\frac{3}{2}\Xi|y_i|$ where they chose $c=\frac{1}{2}$ and $C=\frac{3}{2}$. We express in a more general form for the benefit of downstream analysis. Note that we use node types z_i^G instead of their node labels y_i.

Lemma A.7. *Suppose* $d=\Omega(\overline{n}^2(p+s)^2\log(4\overline{n}^2/\delta))$*. Then with probability at least* $1-\delta$*,*

$$4\sigma_p^2 d/(3\overline{n}(p+s)) \le \|\tilde{\boldsymbol{\xi}}_i^G\|_2^2 \le 4\sigma_p^2 d/(\underline{n}(p+s)),$$

for all $i\in[n_G], G\in\mathcal{G}$.

Lemma A.8. *Suppose that* $d=\Omega(\overline{n}(p+s)\log(\overline{n}m/\delta))$*,* $m=\Omega(\log(1/\delta))$*. Then with probability at least* $1-\delta$*,*

$$\big|\langle \mathbf{w}_{j,r}^{(0)}, \tilde{\boldsymbol{\xi}}_i^G\rangle\big| \le 2\sqrt{\log(8m\overline{n}/\delta)}\cdot\sigma_0\sigma_p\sqrt{d/(\underline{n}(p+s))},$$

$$\sigma_0\sigma_p\sqrt{d/(\overline{n}(p+s))}/4 \le \max_{r\in[m]} j\cdot\langle \mathbf{w}_{j,r}^{(0)}, \tilde{\boldsymbol{\xi}}_i^G\rangle \le 2\sqrt{\log(8m\overline{n}/\delta)}\cdot\sigma_0\sigma_p\sqrt{d/(\underline{n}(p+s))},$$

for all $j\in\{\pm1\}$ *and* $i\in[n_G], G\in\mathcal{G}$.

A.2. Coefficient Analysis

We first write the gradient update. The goal is to learn $\mathbf{W}$ by minimizing the empirical cross-entropy loss with the graph classification dataset $\mathcal{G}$:

$$L_{\mathcal{G}}^{\mathrm{GCN}}(\mathbf{W}) = \frac{1}{N}\sum_{G\in\mathcal{G}} \ell(y_G\cdot f(\mathbf{W},\mathbf{A}^G,\mathbf{X}^G)) \tag{8}$$

where $\ell(y_G\cdot f(\mathbf{W},\mathbf{A}^G,\mathbf{X}^G)) = \log(1+\exp(-f_G\cdot y_G))$. As a result:

$$\begin{aligned}\mathbf{w}_{j,r}^{(t+1)} &= \mathbf{w}_{j,r}^{(t)} - \eta\nabla_{\mathbf{w}_{j,r}} L_{\mathcal{G}}^{\mathrm{GCN}}(\mathbf{W}) = \mathbf{w}_{j,r}^{(t)} \\ &- \frac{\eta}{Nm}\sum_G \ell_G'^{(t)} y_G \sum_{i\in V^G} \sigma'\left(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{z}_i^G\boldsymbol{\mu}\rangle\right)\tilde{z}_i^G\boldsymbol{\mu}j \\ &- \frac{\eta}{Nm}\sum_G \ell_G'^{(t)} y_G \sum_{i\in V^G} \sigma'\left(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{\boldsymbol{\xi}}_i^G\rangle\right)\tilde{\boldsymbol{\xi}}_i^G j\end{aligned} \tag{9}$$

where $0\le t<T^*$, $\ell_G' \triangleq -\frac{\exp(-y_G\cdot f_G)}{1+\exp(-y_G\cdot f_G)}$, the *aggregated type* $\tilde{z}_i^G = (D_i^G)^{-1}\sum_{k\in\mathcal{N}(i)} z_k^G$, and the *aggregated noise* vector $\tilde{\boldsymbol{\xi}}_i^G = (D_i^G)^{-1}\sum_{k\in\mathcal{N}(i)}\boldsymbol{\xi}_k^G$. Here $\mathcal{N}(i)$ denotes the set of neighbors of node i and η is the learning rate.

Next, we detail the lemmas from the gradient update (Equation (9)) and linear combination (Equation (7)) of $\mathbf{w}_{j,r}^{(t)}$. For the benefit of downstream analysis, we define $\zeta_{j,r,i}^{G(t)} \triangleq \overline{\rho}_{j,r,i}^{G(t)} = \rho_{j,r,i}^{G(t)}\mathbb{1}(\rho_{j,r,i}^{G(t)}\ge 0)$ and $\omega_{j,r,i}^{G(t)} \triangleq \underline{\rho}_{j,r,i}^{G(t)} = \rho_{j,r,i}^{G(t)}\mathbb{1}(\rho_{j,r,i}^{G(t)}<0)$. Clearly $\rho_{j,r,i}^{G(t)} = \zeta_{j,r,i}^{G(t)} + \omega_{j,r,i}^{G(t)}$. Then we give the iterative update of $\gamma_{j,r}^{(t)}$, $\zeta_{j,r,i}^{G(t)}$, and $\omega_{j,r,i}^{G(t)}$, which forms the foundation of the coefficient analysis:

Lemma A.9. *The coefficients $\gamma_{j,r}^{(t)}$, $\zeta_{j,r,i}^{G(t)}$, $\omega_{j,r,i}^{G(t)}$ defined in Equation (7) satisfy the following iterative equations:*

$$\begin{aligned}
&\gamma_{j,r}^{(0)}, \zeta_{j,r,i}^{G(0)}, \omega_{j,r,i}^{G(0)} = 0 \\
&\gamma_{j,r}^{(t+1)} = \gamma_{j,r}^{(t)} - \frac{\eta}{Nm} \sum_{G} \sum_{i \in V^G} \ell_G^{'(t)} \sigma' \left(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{z}_i^G \boldsymbol{\mu} \rangle \right) y_G \tilde{z}_i^G \|\boldsymbol{\mu}\|_2^2 \\
&\zeta_{j,r,i}^{G(t+1)} = \zeta_{j,r,i}^{G(t)} - \frac{\eta}{Nm} \sum_{k \in \mathcal{N}(i)} (D_k^G)^{-1} \ell_G^{'(t)} \sigma' \left(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{\boldsymbol{\xi}}_k^G \rangle \right) \|\boldsymbol{\xi}_i^G\|_2^2 \mathbb{1}(j = y_G) \\
&\omega_{j,r,i}^{G(t+1)} = \omega_{j,r,i}^{G(t)} + \frac{\eta}{Nm} \sum_{k \in \mathcal{N}(i)} (D_k^G)^{-1} \ell_G^{'(t)} \sigma' \left(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{\boldsymbol{\xi}}_k^G \rangle \right) \|\boldsymbol{\xi}_i^G\|_2^2 \mathbb{1}(j = -y_G).
\end{aligned}$$

Proof. Equation (9) already disentangles the gradient update into parts that depend on $\tilde{z}_i^G \boldsymbol{\mu}$ and $\boldsymbol{\xi}_i^G$:

$$\begin{aligned}
\mathbf{w}_{j,r}^{(t+1)} = \mathbf{w}_{j,r}^{(t)} - \eta \nabla_{\mathbf{w}_{j,r}} L_{\mathcal{G}}^{\mathrm{GCN}}(\mathbf{W}) = \mathbf{w}_{j,r}^{(t)} \\
- \frac{\eta}{Nm} \sum_{G} \ell_G^{'(t)} y_G \sum_{i \in V^G} \sigma' \left(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{z}_i^G \boldsymbol{\mu} \rangle \right) \tilde{z}_i^G \boldsymbol{\mu} j \\
- \frac{\eta}{Nm} \sum_{G} \ell_G^{'(t)} y_G \sum_{i \in V^G} \sigma' \left(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{\boldsymbol{\xi}}_i^G \rangle \right) \tilde{\boldsymbol{\xi}}_i^G j
\end{aligned}$$

By comparing it to the weight decomposition Equation (7):

$$\mathbf{w}_{j,r}^{(t)} = \mathbf{w}_{j,r}^{(0)} + j \gamma_{j,r}^{(t)} \|\boldsymbol{\mu}\|_2^{-2} \boldsymbol{\mu} + \sum_{G \in \mathcal{G}} \sum_{i \in V^G} \rho_{j,r,i}^{G(t)} \|\boldsymbol{\xi}_i^G\|_2^{-2} \boldsymbol{\xi}_i^G$$

we immediately have:

$$\gamma_{j,r}^{(t+1)} = \gamma_{j,r}^{(t)} - \frac{\eta}{Nm} \sum_{G} \sum_{i \in V^G} \ell_G^{'(t)} \sigma' \left(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{z}_i^G \boldsymbol{\mu} \rangle \right) y_G \tilde{z}_i^G \|\boldsymbol{\mu}\|_2^2$$

For the noise part, we rewrite the summation as:

$$\begin{aligned}
&\frac{\eta}{Nm} \sum_{G} \ell_G^{'(t)} y_G \sum_{i \in V^G} \sigma' \left(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{\boldsymbol{\xi}}_i^G \rangle \right) \tilde{\boldsymbol{\xi}}_i^G j \\
&= \frac{\eta}{Nm} \sum_{G} \ell_G^{'(t)} y_G \sum_{i \in V^G} \sigma' \left(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{\boldsymbol{\xi}}_i^G \rangle \right) (D_i^G)^{-1} \sum_{k \in \mathcal{N}(i)} \boldsymbol{\xi}_k^G j \\
&= \frac{\eta}{Nm} \sum_{G} \ell_G^{'(t)} y_G \sum_{k \in V^G} \sum_{i : k \in \mathcal{N}(i)} \sigma' \left(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{\boldsymbol{\xi}}_i^G \rangle \right) (D_i^G)^{-1} \boldsymbol{\xi}_k^G j \\
&= \frac{\eta}{Nm} \sum_{G} \ell_G^{'(t)} y_G \sum_{k \in V^G} \sum_{i \in \mathcal{N}(k)} \sigma' \left(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{\boldsymbol{\xi}}_i^G \rangle \right) (D_i^G)^{-1} \boldsymbol{\xi}_k^G j \\
&= \frac{\eta}{Nm} \sum_{G} \ell_G^{'(t)} y_G \sum_{i \in V^G} \sum_{k \in \mathcal{N}(i)} \sigma' \left(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{\boldsymbol{\xi}}_k^G \rangle \right) (D_k^G)^{-1} \boldsymbol{\xi}_i^G j,
\end{aligned}$$

where the third equation is by undirected graph hence $i \in \mathcal{N}(k) \Leftrightarrow k \in \mathcal{N}(i)$, and the fourth is by switching the indices of i and k. Then we immediately have:

$$\begin{aligned}
\rho_{j,r,i}^{G(t+1)} &= \rho_{j,r,i}^{G(t)} - \frac{\eta}{Nm} \sum_{k \in \mathcal{N}(i)} (D_k^G)^{-1} \ell_G^{'(t)} \sigma' \left(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{\boldsymbol{\xi}}_k^G \rangle \right) \|\boldsymbol{\xi}_i^G\|_2^2 j y_G \\
\zeta_{j,r,i}^{G(t+1)} &= \zeta_{j,r,i}^{G(t)} - \frac{\eta}{Nm} \sum_{k \in \mathcal{N}(i)} (D_k^G)^{-1} \ell_G^{'(t)} \sigma' \left(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{\boldsymbol{\xi}}_k^G \rangle \right) \|\boldsymbol{\xi}_i^G\|_2^2 \mathbb{1}(j = y_G) \\
\omega_{j,r,i}^{G(t+1)} &= \omega_{j,r,i}^{G(t)} + \frac{\eta}{Nm} \sum_{k \in \mathcal{N}(i)} (D_k^G)^{-1} \ell_G^{'(t)} \sigma' \left(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{\boldsymbol{\xi}}_k^G \rangle \right) \|\boldsymbol{\xi}_i^G\|_2^2 \mathbb{1}(j = -y_G).
\end{aligned}$$

□

In Lemma A.9, $\rho_{j,r,i}^{G(t)}$ is decomposed into $\zeta_{j,r,i}^{G(t)}$ and $\omega_{j,r,i}^{G(t)}$ which have determined signs. On the contrary, for the iterative of $\gamma_{j,r}^{(t)}$, the sign of $y_G \tilde{z}_i^G$ is unknown, hence the sign of $\gamma_{j,r}^{(t)}$ cannot be determined. In Lemma A.10, we show that under Assumption A.1, it holds with high probability that $\gamma_{j,r}^{(t)} \geq 0$.

Lemma A.10. *Define a misalignment rate $0 < \epsilon < \frac{1}{2}$ that measures the percentage of nodes in a graph whose types are not aligned with the graph label. The difference between the loss-weighted aggregated contribution of aligned nodes and misaligned nodes is defined as*

$$\Delta \triangleq \underline{\ell}(1-\epsilon)c^q \underline{n} - \overline{\ell}\epsilon C^q \overline{n}.$$

Similarly,

$$\Sigma \triangleq \underline{\ell}(1-\epsilon)c^q \underline{n} + \overline{\ell}\epsilon C^q \overline{n}.$$

Also suppose (1) it holds for every G that $\#\{i \in V^G : y_G z_i^G > 0\} \geq (1-\epsilon)n_G$; (2) $\Delta > 0$; (3) $N \geq 8 \log \frac{2}{\delta} (\frac{\Sigma}{\Delta})^2$, then with probability at least $1-\delta$, we have $\gamma_{j,r}^{(t)} \geq 0$ for all $j \in \{\pm 1\}$ and $r \in [m]$.

Proof. By Lemma A.9, we have

$$\gamma_{j,r}^{(t+1)} = \gamma_{j,r}^{(t)} - \frac{\eta}{Nm} \sum_G \sum_{i \in V^G} \ell_G^{'(t)} \sigma' \left(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{z}_i^G \boldsymbol{\mu} \rangle \right) y_G \tilde{z}_i^G \|\boldsymbol{\mu}\|_2^2.$$

Since $\ell_G^{'(t)} \leq 0$ and $\|\boldsymbol{\mu}\|_2^2 \geq 0$, it suffices to lower bound the loss-weighted signed sum. Let $a^{(t)} = \langle \mathbf{w}_{j,r}^{(t)}, \boldsymbol{\mu} \rangle$, and define

$$S_\ell^{(t)} = \sum_G \sum_{i \in V^G} (-\ell_G^{'(t)}) \sigma'(\tilde{z}_i^G a^{(t)}) \tilde{z}_i^G y_G.$$

The sign of $\gamma_{j,r}^{(t)}$ being positive depends on that of $S_\ell^{(t)}$ being positive. Without loss of generality, let $a^{(t)} \geq 0$. Since $\sigma'(x) = q \max\{0, x\}^{q-1}$, $S_\ell^{(t)}$ can be written as

$$S_\ell^{(t)} = \sum_G T_G^{(t)} = \sum_G \sum_{\tilde{z}_i^G > 0} (-\ell_G^{'(t)}) q (a^{(t)})^{q-1} (\tilde{z}_i^G)^q y_G.$$

If $y_G = 1$:

$$\begin{aligned} T_G^{(t)} &= \sum_{\tilde{z}_i^G > 0} (-\ell_G^{'(t)}) q (a^{(t)})^{q-1} (\tilde{z}_i^G)^q \\ &\geq \underline{\ell}\, q (a^{(t)})^{q-1} (\tilde{z}_i^G)^q (1-\epsilon) n_G \\ &\geq \underline{\ell}\, q (a^{(t)})^{q-1} (1-\epsilon) n_G (c\Xi)^q. \end{aligned}$$

Similarly, if $y_G = -1$ we have

$$T_G^{(t)} \geq -\overline{\ell}\, q (a^{(t)})^{q-1} \epsilon n_G (C\Xi)^q.$$

Next, let $N_+ \triangleq \#\{y_G = 1\}$ and $N_- \triangleq \#\{y_G = -1\}$, and apply a union bound over n_G, then

$$S_\ell^{(t)} = \sum_{y_G = 1} T_G^{(t)} + \sum_{y_G = -1} T_G^{(t)} \geq q (a^{(t)})^{q-1} \Xi^q \left[N_+ \underline{\ell}(1-\epsilon) c^q \underline{n} - N_- \overline{\ell} \epsilon C^q \overline{n} \right].$$

Since y_G is a Rademacher variable, by Hoeffding's inequality:

$$\mathrm{P}(|N_+ - \frac{N}{2}| \geq t) \leq 2 \exp\left(-\frac{2t^2}{N}\right).$$

So with probability at least $1-\delta$, $N_+ \geq \frac{N}{2} - t$, $N_- \leq \frac{N}{2} + t$, where $t = \sqrt{\frac{N}{2} \log \frac{2}{\delta}}$. Therefore

$$\begin{aligned} & N_+ \underline{\ell}(1-\epsilon) c^q \underline{n} - N_- \overline{\ell} \epsilon C^q \overline{n} \\ & \geq \frac{N}{2} \left[\underline{\ell}(1-\epsilon) c^q \underline{n} - \overline{\ell} \epsilon C^q \overline{n} \right] - t \left[\underline{\ell}(1-\epsilon) c^q \underline{n} + \overline{\ell} \epsilon C^q \overline{n} \right]. \end{aligned}$$

Thus

$$S_\ell^{(t)} \geq q(a^{(t)})^{q-1}\Xi^q\left(\frac{N}{2}\Delta - \Sigma\sqrt{\frac{N}{2}\log\frac{2}{\delta}}\right).$$

By the condition on N we have $S_\ell^{(t)} \geq 0$. The case where $a^{(t)} < 0$ can be proved following the same logic, where

$$S_\ell^{(t)} \geq q(-a^{(t)})^{q-1}\Xi^q\left(\frac{N}{2}\Delta - \Sigma\sqrt{\frac{N}{2}\log\frac{2}{\delta}}\right).$$

As a result, with high probability $\gamma_{j,r}^{(t)} \geq 0$ and monotonically increases with t. □

The results in Lemma A.10 including the lower bound of $S^{(t)}$ are essential for establishing the constant order lower bound of $\gamma_{j,r}^{(t)}$. Next, let $T^* = \eta^{-1}\text{poly}(\|\boldsymbol{\mu}\|_2^{-1}, d^{-1}\sigma_p^{-2}, \sigma_0^{-1}, N, \overline{n}, m, d)$ be the maximum admissible iterations and denote $\alpha = 4\Big(m(\overline{n}C_F + \log T^*)\Big)^{1/q}$, where C_F is a constant. Below we give the necessary conditions that must be satisfied in order to establish the bounds for the coefficients in the subsequent lemmas. These conditions are also introduced in Assumption A.1:

$$\eta = O\big(\min\big\{Nm/(q\sigma_p^2 d),\ Nm/(q2^{q+2}\alpha^{q-2}\sigma_p^2 d),\ Nm/(\overline{n}q2^{q+2}\alpha^{q-2}\|\boldsymbol{\mu}\|_2^2)\big\}\big),$$

$$\sigma_0 \leq \left[16\sqrt{\log(8m\overline{n}/\delta)}\right]^{-1}\min\left\{\Xi^{-1}\|\boldsymbol{\mu}\|_2^{-1},\ (\sigma_p\sqrt{d/(\underline{n}(p+s))})^{-1}\right\},$$

$$d \geq 1024\log(4\overline{n}^2/\delta)\alpha^2\overline{n}^2N^2.$$

Denote $\beta = 2\max_{G,i,j,r}\left\{|\langle \mathbf{w}_{j,r}^{(0)}, \tilde{z}_i^G\cdot\boldsymbol{\mu}\rangle|,\ |\langle \mathbf{w}_{j,r}^{(0)}, \tilde{\boldsymbol{\xi}}_i^G\rangle|\right\}$, it is straightforward to show the following inequality by Lemma A.3 and Lemma A.8:

$$4\max\left\{\beta,\ 8N\overline{n}\sqrt{\frac{\log(4\overline{n}^2/\delta)}{d}}\alpha\right\} \leq 1.$$

Proposition A.11. *Under Assumption A.1, for $0 \leq t \leq T^*$, we have that*

$$0 \leq \gamma_{j,r}^{(t)},\ \zeta_{j,r,i}^{G(t)} \leq \alpha$$
$$0 \geq \omega_{j,r,i}^{G(t)} \geq -\alpha$$

for all $j \in \{\pm 1\}$, $r \in [m]$, and $i \in [n_G], G \in \mathcal{G}$.

Before proving Proposition A.11, we first provide necessary lemmas. The following lemma for signal learning is immediately known from Equation (7):

Lemma A.12. *For any $t \geq 0$, it holds that $\langle \mathbf{w}_{j,r}^{(t)} - \mathbf{w}_{j,r}^{(0)}, \boldsymbol{\mu}\rangle = j\cdot\gamma_{j,r}^{(t)}$ for all $r \in [m]$, $j \in \{\pm 1\}$.*

Then we characterize noise memorization:

Lemma A.13. *Under Assumption A.1, we have*

$$\hat{\rho}_{j,r,i}^{G(t)} - 8N\overline{n}\sqrt{\log(4\overline{n}^2/\delta)/d}\alpha \leq \langle \mathbf{w}_{j,r}^{(t)} - \mathbf{w}_{j,r}^{(0)}, \tilde{\boldsymbol{\xi}}_i^G\rangle \leq \hat{\rho}_{j,r,i}^{G(t)} + 8N\overline{n}\sqrt{\log(4\overline{n}^2/\delta)/d}\alpha.$$

for all $j \in \{\pm 1\}$, $r \in [m]$, and $i \in [n_G], G \in \mathcal{G}$, where $\hat{\rho}_{j,r,i}^{G(t)} \triangleq (D_i^G)^{-1}\sum_{k\in\mathcal{N}(i)}\rho_{j,r,k}^{G(t)}$.

Proof. Since $\boldsymbol{\mu} \perp \boldsymbol{\xi}_i^G$:

$$\langle \mathbf{w}_{j,r}^{(t)} - \mathbf{w}_{j,r}^{(0)}, \tilde{\boldsymbol{\xi}}_i^G\rangle = \sum_{G'\in\mathcal{G}}\sum_{i'\in V^{G'}}(\zeta_{j,r,i'}^{G'(t)} + \omega_{j,r,i'}^{G'(t)})\|\boldsymbol{\xi}_{i'}^{G'}\|_2^{-2}\langle \boldsymbol{\xi}_{i'}^{G'}, \tilde{\boldsymbol{\xi}}_i^G\rangle$$

For $\zeta_{j,r,i'}^{G'(t)}$:

$$
\begin{aligned}
&\sum_{G'\in\mathcal{G}}\sum_{i'\in V^{G'}}\zeta_{j,r,i'}^{G'(t)}\|\boldsymbol{\xi}_{i'}^{G'}\|_2^{-2}\langle\boldsymbol{\xi}_{i'}^{G'},\tilde{\boldsymbol{\xi}}_i^G\rangle\\
&=\sum_{G'\in\mathcal{G}}\sum_{i'\in V^{G'}}\zeta_{j,r,i'}^{G'(t)}\|\boldsymbol{\xi}_{i'}^{G'}\|_2^{-2}\sum_{k\in\mathcal{N}(i)}(D_i^G)^{-1}\langle\boldsymbol{\xi}_{i'}^{G'},\boldsymbol{\xi}_k^G\rangle\\
&=\sum_{G'\in\mathcal{G}}\sum_{k\in\mathcal{N}(i)}\sum_{i'\neq k}\zeta_{j,r,i'}^{G'(t)}\|\boldsymbol{\xi}_{i'}^{G'}\|_2^{-2}(D_i^G)^{-1}\langle\boldsymbol{\xi}_{i'}^{G'},\boldsymbol{\xi}_k^G\rangle+(D_i^G)^{-1}\sum_{k\in\mathcal{N}(i)}\zeta_{j,r,k}^{G(t)}\\
&\leq\hat{\zeta}_{j,r,i}^{G(t)}+\sum_{G'\in\mathcal{G}}\sum_{k\in\mathcal{N}(i)}\sum_{i'\neq k}(D_i^G)^{-1}|\zeta_{j,r,i'}^{G('t)}|4\sqrt{\log(4\overline{n}^2/\delta)/d}\\
&\leq\hat{\zeta}_{j,r,i}^{G(t)}+4N\overline{n}|\zeta_{j,r,i'}^{G'(t)}|\sqrt{\log(4\overline{n}^2/\delta)/d}\\
&\leq\hat{\zeta}_{j,r,i}^{G(t)}+4N\overline{n}\sqrt{\log(4\overline{n}^2/\delta)/d}\alpha
\end{aligned}
$$

where we define $\hat{\zeta}_{j,r,i}^{G(t)}\triangleq(D_i^G)^{-1}\sum_{k\in\mathcal{N}(i)}\zeta_{j,r,k}^{G(t)}$, the first inequality is by Lemma A.2 and the last is by Proposition A.11. Similarly, we can establish that

$$
\sum_{G'\in\mathcal{G}}\sum_{i'\in V^{G'}}\zeta_{j,r,i'}^{G'(t)}\|\boldsymbol{\xi}_{i'}^{G'}\|_2^{-2}\langle\boldsymbol{\xi}_{i'}^{G'},\tilde{\boldsymbol{\xi}}_i^G\rangle\geq\hat{\zeta}_{j,r,i}^{G(t)}-4N\overline{n}\sqrt{\log(4\overline{n}^2/\delta)/d}\alpha
$$

and the same bounds can be established in the same way for $\omega_{j,r,i'}^{G'(t)}$, thus proving the lemma. □

Using Lemma A.13 we have the following results:

Lemma A.14. *Under Assumption A.1, we have*

$$
\langle\mathbf{w}_{j,r}^{(t)},\tilde{z}_i^G\boldsymbol{\mu}\rangle\leq\langle\mathbf{w}_{j,r}^{(0)},\tilde{z}_i^G\boldsymbol{\mu}\rangle,
$$

$$
\langle\mathbf{w}_{j,r}^{(t)},\tilde{\boldsymbol{\xi}}_i^G\rangle\leq\langle\mathbf{w}_{j,r}^{(0)},\tilde{\boldsymbol{\xi}}_i^G\rangle+\hat{\rho}_{j,r,i}^{G(t)}+8N\overline{n}\sqrt{\log(4\overline{n}^2/\delta)/d}\alpha,
$$

for all $r\in[m]$ and $j\neq z_i^G$.

Proof. For $j\neq z_i^G$, we have that

$$
\langle\mathbf{w}_{j,r}^{(t)},\tilde{z}_i^G\boldsymbol{\mu}\rangle=\langle\mathbf{w}_{j,r}^{(0)},\tilde{z}_i^G\boldsymbol{\mu}\rangle+\tilde{z}_i^G j\gamma_{j,r}^{(t)}\leq\langle\mathbf{w}_{j,r}^{(0)},\tilde{z}_i^G\boldsymbol{\mu}\rangle
$$

where the equality is by Lemma A.12. The inequality is by Lemma A.5 which states $\mathrm{sign}(z_i^G)=\mathrm{sign}(\tilde{z}_i^G)$ with high probability and Lemma A.10 which states $\gamma_{j,r}^{(t)}\geq 0$ with high probability. Also, by Lemma A.13 we have that

$$
\langle\mathbf{w}_{j,r}^{(t)},\tilde{\boldsymbol{\xi}}_i^G\rangle\leq\langle\mathbf{w}_{j,r}^{(0)},\tilde{\boldsymbol{\xi}}_i^G\rangle+\hat{\rho}_{j,r,i}^{G(t)}+8N\overline{n}\sqrt{\log(4\overline{n}^2/\delta)/d}\alpha
$$

□

Lemma A.15. *Under Assumption A.1, we have*

$$
\langle\mathbf{w}_{j,r}^{(t)},\tilde{z}_i^G\boldsymbol{\mu}\rangle\leq\langle\mathbf{w}_{j,r}^{(0)},\tilde{z}_i^G\boldsymbol{\mu}\rangle+C\Xi\gamma_{j,r}^{(t)},
$$

$$
\langle\mathbf{w}_{j,r}^{(t)},\tilde{\boldsymbol{\xi}}_i^G\rangle\leq\langle\mathbf{w}_{j,r}^{(0)},\tilde{\boldsymbol{\xi}}_i^G\rangle+\hat{\rho}_{j,r,i}^{G(t)}+8N\overline{n}\sqrt{\log(4\overline{n}^2/\delta)/d}\alpha,
$$

for all $r\in[m]$ and $j=z_i^G$.

Proof. For $j = z_i^G$, we have that

$$\langle \mathbf{w}_{j,r}^{(t)}, \tilde{z}_i^G \boldsymbol{\mu} \rangle = \langle \mathbf{w}_{j,r}^{(0)}, \tilde{z}_i^G \boldsymbol{\mu} \rangle + \tilde{z}_i^G j \gamma_{j,r}^{(t)} = \langle \mathbf{w}_{j,r}^{(0)}, \tilde{z}_i^G \boldsymbol{\mu} \rangle + |\tilde{z}_i^G| \gamma_{j,r}^{(t)} \leq \langle \mathbf{w}_{j,r}^{(0)}, \tilde{z}_i^G \boldsymbol{\mu} \rangle + C\Xi \gamma_{j,r}^{(t)}$$

where the first equality is by Lemma A.12. The second equality and the inequality is by Lemma A.5 that bounds $|\tilde{z}_i^G|$. Also, by Lemma A.13 we have that

$$\langle \mathbf{w}_{j,r}^{(t)}, \tilde{\boldsymbol{\xi}}_i^G \rangle \leq \langle \mathbf{w}_{j,r}^{(0)}, \tilde{\boldsymbol{\xi}}_i^G \rangle + \hat{\rho}_{j,r,i}^{G(t)} + 8N\overline{n}\sqrt{\log(4\overline{n}^2/\delta)/d\alpha}$$

□

Lemma A.16. *Under Lemma A.14 and Lemma A.15, if* $\max\{\gamma_{j,r}^{(t)}, \rho_{j,r,i}^{G(t)}\} = O(1)$*, we have that* $F_j(\mathbf{W}_j^{(t)}, \mathbf{A}^G, \mathbf{X}^G) = O(\overline{n})$.

Proof. Let

$$V_{\neq}^G(j) := \{i \in V^G : j \neq z_i^G\}, \qquad V_{=}^G(j) := \{i \in V^G : j = z_i^G\},$$

and denote

$$E := 8N\overline{n}\sqrt{\log(4\overline{n}^2/\delta)/d\alpha}.$$

Then

$$\begin{aligned}
F_j(\mathbf{W}_j^{(t)}, \mathbf{A}^G, \mathbf{X}^G) &= \frac{1}{m}\sum_{r=1}^{m}\sum_{i\in V^G}\left[\sigma\left(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{z}_i^G \boldsymbol{\mu}\rangle\right) + \sigma\left(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{\boldsymbol{\xi}}_i^G\rangle\right)\right] \\
&= \frac{1}{m}\sum_{r=1}^{m}\sum_{i\in V_{\neq}^G(j)}\left[\sigma\left(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{z}_i^G \boldsymbol{\mu}\rangle\right) + \sigma\left(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{\boldsymbol{\xi}}_i^G\rangle\right)\right] \\
&\quad + \frac{1}{m}\sum_{r=1}^{m}\sum_{i\in V_{=}^G(j)}\left[\sigma\left(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{z}_i^G \boldsymbol{\mu}\rangle\right) + \sigma\left(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{\boldsymbol{\xi}}_i^G\rangle\right)\right] \\
&\leq \frac{1}{m}\sum_{r=1}^{m}\sum_{i\in V_{\neq}^G(j)} \\
&\left[\sigma\left(\langle \mathbf{w}_{j,r}^{(0)}, \tilde{z}_i^G \boldsymbol{\mu}\rangle\right) + \sigma\left(\langle \mathbf{w}_{j,r}^{(0)}, \tilde{\boldsymbol{\xi}}_i^G\rangle + \hat{\rho}_{j,r,i}^{G(t)} + E\right)\right] \\
&\quad + \frac{1}{m}\sum_{r=1}^{m}\sum_{i\in V_{=}^G(j)} \\
&\left[\sigma\left(\langle \mathbf{w}_{j,r}^{(0)}, \tilde{z}_i^G \boldsymbol{\mu}\rangle + C\Xi\gamma_{j,r}^{(t)}\right) + \sigma\left(\langle \mathbf{w}_{j,r}^{(0)}, \tilde{\boldsymbol{\xi}}_i^G\rangle + \hat{\rho}_{j,r,i}^{G(t)} + E\right)\right] \\
&\leq \overline{n}\, 4^{q+1} \max_{j,r,G,i}\left\{|\langle \mathbf{w}_{j,r}^{(0)}, \tilde{z}_i^G \boldsymbol{\mu}\rangle|, |\langle \mathbf{w}_{j,r}^{(0)}, \tilde{\boldsymbol{\xi}}_i^G\rangle|, |\hat{\rho}_{j,r,i}^{G(t)}|, E, C\Xi\gamma_{j,r}^{(t)}\right\}^q.
\end{aligned}$$

Since $\max\{\gamma_{j,r}^{(t)}, \rho_{j,r,i}^{G(t)}\} = O(1)$, $F_j(\mathbf{W}_j^{(t)}, \mathbf{A}^G, \mathbf{X}^G) = O(\overline{n})$. □

Remark A.17. The order of $O(\overline{n})$ is clearly a result of sum pooling. If we instead use mean pooling, then it becomes $O(1)$.

With all above lemmas we can now prove Proposition A.11 that uniformly bounds the coefficients $\gamma_{j,r}^{(t)}$, $\zeta_{j,r,i}^{G(t)}, \omega_{j,r,i}^{G(t)}$. Note that we already prove $\gamma_{j,r}^{(t)} \geq 0$ in Lemma A.10 with high probability.

Proof. (Proof of Proposition A.11). We employ an induction method as in (Huang et al., 2025).

At the initial step $t = 0$, $\gamma_{j,r}^{(t)}$, $\zeta_{j,r,i}^{G(t)}, \omega_{j,r,i}^{G(t)} = 0$ so Proposition A.11 clearly holds.

Next, we hypothesize that there exists a time $\tilde{T}$ less that T^* during which Proposition A.11 holds true for every moment within the range $0 \leq t \leq \tilde{T} - 1$. Our objective is to show that this proposition remains valid at $t = \tilde{T}$.

We first validate $-\alpha \le \omega_{j,r,i}^{G(t)} \le 0$ holds at $t = \tilde{T}$. Note that $\omega_{j,r,i}^{G(t)} = 0$ when $j = y_G$, so we only have to consider $j = -y_G$. Specifically, we aim to establish

$$\omega_{j,r,i}^{G(\tilde{T})} \ge -\beta - 16N\overline{n}\sqrt{\log(4\overline{n}^2/\delta)/d}\alpha$$

which we discuss in two cases: 1) Under condition

$$\omega_{j,r,i}^{G(\tilde{T}-1)} \le -0.5\beta - 8N\overline{n}\sqrt{\log(4\overline{n}^2/\delta)/d}\alpha$$

by Lemma A.13 we know

$$\langle \mathbf{w}_{j,r}^{(\tilde{T}-1)}, \tilde{\boldsymbol{\xi}}_i^G \rangle \le \langle \mathbf{w}_{j,r}^{(0)}, \tilde{\boldsymbol{\xi}}_i^G \rangle + \hat{\rho}_{j,r,i}^{G(\tilde{T}-1)} + 8N\overline{n}\sqrt{\log(4\overline{n}^2/\delta)/d}\alpha$$

then by the definition of β and the condition on $\omega_{j,r,i}^{G(\tilde{T}-1)}$ we immediately know that $\langle \mathbf{w}_{j,r}^{(\tilde{T}-1)}, \tilde{\boldsymbol{\xi}}_i^G \rangle \le 0$. Therefore:

$$\begin{aligned}
\omega_{j,r,i}^{G(\tilde{T})} &= \omega_{j,r,i}^{G(\tilde{T}-1)} + \frac{\eta}{Nm} \sum_{k \in \mathcal{N}(i)} (D_k^G)^{-1} \ell_G^{'(\tilde{T}-1)} \sigma' \left(\langle \mathbf{w}_{j,r}^{(\tilde{T}-1)}, \tilde{\boldsymbol{\xi}}_k^G \rangle \right) \|\boldsymbol{\xi}_i^G\|_2^2 \mathbb{1}(j = -y_G) \\
&= \omega_{j,r,i}^{G(\tilde{T}-1)} \ge -\beta - 16N\overline{n}\sqrt{\log(4\overline{n}^2/\delta)/d}\alpha.
\end{aligned}$$

2) Under condition

$$\omega_{j,r,i}^{G(\tilde{T}-1)} \ge -0.5\beta - 8N\overline{n}\sqrt{\log(4\overline{n}^2/\delta)/d}\alpha$$

we have that

$$\begin{aligned}
\omega_{j,r,i}^{G(\tilde{T})} &= \omega_{j,r,i}^{G(\tilde{T}-1)} + \frac{\eta}{Nm} \sum_{k \in \mathcal{N}(i)} (D_k^G)^{-1} \ell_G^{'(\tilde{T}-1)} \sigma' \left(\langle \mathbf{w}_{j,r}^{(\tilde{T}-1)}, \tilde{\boldsymbol{\xi}}_k^G \rangle \right) \|\boldsymbol{\xi}_i^G\|_2^2 \mathbb{1}(j = -y_G) \\
&\ge -0.5\beta - 8N\overline{n}\sqrt{\log(4\overline{n}^2/\delta)/d}\alpha - O\left(\frac{\eta\sigma_p^2 d}{Nm}\right) \sigma'(0.5\beta + 8N\overline{n}\sqrt{\log(4\overline{n}^2/\delta)/d}\alpha) \\
&\ge -0.5\beta - 8N\overline{n}\sqrt{\log(4\overline{n}^2/\delta)/d}\alpha - O\left(\frac{\eta q\sigma_p^2 d}{Nm}\right) (0.5\beta + 8N\overline{n}\sqrt{\log(4\overline{n}^2/\delta)/d}\alpha) \\
&\ge -\beta - 16N\overline{n}\sqrt{\log(4\overline{n}^2/\delta)/d}\alpha
\end{aligned}$$

where the first inequality is by the condition on $\omega_{j,r,i}^{G(\tilde{T}-1)}$, $|\ell_G^{'(\tilde{T}-1)}| \le 1$, and Lemma A.2, and the second inequality is by the fact that $0.5\beta + 8N\overline{n}\sqrt{\log(4\overline{n}^2/\delta)/d}\alpha \le 1$, and the third inequality is by the condition that $\eta = O\left(Nm/(q\sigma_p^2 d)\right)$.

Next, we show $0 \le \zeta_{j,r,i}^{G(t)} \le \alpha$ holds at $t = \tilde{T}$. First we can establish another upper bound for $|\ell_G^{'(t)}|$ as follows:

$$\begin{aligned}
|\ell_G^{'(t)}| &= \frac{1}{1 + \exp\left(y_G \cdot [F_{+1}(\mathbf{W}_{+1}^{(t)}, \mathbf{A}^G, \mathbf{X}^G) - F_{-1}(\mathbf{W}_{-1}^{(t)}, \mathbf{A}^G, \mathbf{X}^G)]\right)} \\
&\le \exp\left(-y_G \cdot [F_{+1}(\mathbf{W}_{+1}^{(t)}, \mathbf{A}^G, \mathbf{X}^G) - F_{-1}(\mathbf{W}_{-1}^{(t)}, \mathbf{A}^G, \mathbf{X}^G)]\right) \\
&\le \exp\left(-F_{y_G}(\mathbf{W}_{y_G}^{(t)}, \mathbf{A}^G, \mathbf{X}^G) + \overline{n}C_F\right)
\end{aligned}$$

where the last inequality is by Lemma A.16, and $\overline{n}C_F$ is an upper bound on $F_j(\mathbf{W}_j^{(t)}, \mathbf{A}^G, \mathbf{X}^G)$. Recall the update of $\zeta_{j,r,i}^{G(t)}$:

$$\zeta_{j,r,i}^{G(t+1)} = \zeta_{j,r,i}^{G(t)} - \frac{\eta}{Nm} \sum_{k \in \mathcal{N}(i)} (D_k^G)^{-1} \ell_G^{'(t)} \sigma' \left(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{\boldsymbol{\xi}}_k^G \rangle \right) \|\boldsymbol{\xi}_i^G\|_2^2 \mathbb{1}(j = y_G).$$

Let $t^G_{j,r,i}$ be the final moment before T^* such that $\zeta^{G(t)}_{j,r,i} \leq 0.5\alpha$, then $\zeta^{G(\tilde{T})}_{j,r,i}$ can be expressed as follows:

$$\zeta^{G(\tilde{T})}_{j,r,i} = \zeta^{G(t^G_{j,r,i})}_{j,r,i} \underbrace{- \frac{\eta}{Nm} \sum_{k\in\mathcal{N}(i)} (D^G_k)^{-1} \ell'^{(t^G_{j,r,i})}_G \sigma'\left(\langle \mathbf{w}^{(t^G_{j,r,i})}_{j,r}, \tilde{\boldsymbol{\xi}}^G_k\rangle\right) \|\boldsymbol{\xi}^G_i\|_2^2 \mathbb{1}(j = y_G)}_{I_1}$$
$$\underbrace{- \sum_{t^G_{j,r,i}<t<\tilde{T}} \frac{\eta}{Nm} \sum_{k\in\mathcal{N}(i)} (D^G_k)^{-1} \ell'^{(t)}_G \sigma'\left(\langle \mathbf{w}^{(t)}_{j,r}, \tilde{\boldsymbol{\xi}}^G_k\rangle\right) \|\boldsymbol{\xi}^G_i\|_2^2 \mathbb{1}(j = y_G)}_{I_2}$$

Next we bound $|I_1|$:

$$\begin{aligned}
|I_1| &\leq 2qN^{-1}m^{-1}\eta\sigma_p^2 d \left(\max_k\{\hat{\rho}^{G(t^G_{j,r,i})}_{j,r,k}\} + 0.5\beta + 8N\overline{n}\sqrt{\log(4\overline{n}^2/\delta)/d}\alpha\right)^{q-1} \\
&\leq q2^q N^{-1}m^{-1}\eta\alpha^{q-1}\sigma_p^2 d \\
&\leq 0.25\alpha
\end{aligned}$$

where the first inequality is by Lemma A.2 and Lemma A.13, the second inequality is by the fact that $0.5\beta + 8N\overline{n}\sqrt{\log(4\overline{n}^2/\delta)/d}\alpha \ll \alpha$, and the third inequality is by the condition that $\eta = O\left(\frac{Nm}{q2^{q+2}\alpha^{q-2}\sigma_p^2 d}\right)$.

To bound $|I_2|$, we first bound $\langle \mathbf{w}^{(t)}_{j,r}, \tilde{\boldsymbol{\xi}}^G_k\rangle$:

$$\begin{aligned}
\langle \mathbf{w}^{(t)}_{j,r}, \tilde{\boldsymbol{\xi}}^G_k\rangle &\geq \langle \mathbf{w}^{(0)}_{j,r}, \tilde{\boldsymbol{\xi}}^G_k\rangle + \hat{\rho}^{G(t)}_{j,r,k} - 8N\overline{n}\sqrt{\log(4\overline{n}^2/\delta)/d}\alpha \\
&\geq 0.5\alpha - 0.5\beta - 8N\overline{n}\sqrt{\log(4\overline{n}^2/\delta)/d}\alpha \geq 0.25\alpha.
\end{aligned}$$

Similarly, we can upper bound $\langle \mathbf{w}^{(t)}_{j,r}, \tilde{\boldsymbol{\xi}}^G_k\rangle$ as:

$$\begin{aligned}
\langle \mathbf{w}^{(t)}_{j,r}, \tilde{\boldsymbol{\xi}}^G_k\rangle &\leq \langle \mathbf{w}^{(0)}_{j,r}, \tilde{\boldsymbol{\xi}}^G_k\rangle + \hat{\rho}^{G(t)}_{j,r,k} + 8N\overline{n}\sqrt{\log(4\overline{n}^2/\delta)/d}\alpha \\
&\leq 1.5\alpha + 0.5\beta + 8N\overline{n}\sqrt{\log(4\overline{n}^2/\delta)/d}\alpha \leq 2\alpha.
\end{aligned}$$

Finally, we can bound $|I_2|$ as:

$$\begin{aligned}
|I_2| &\leq \sum_{t^G_{j,r,i}<t<\tilde{T}} \frac{\eta}{Nm} \sum_{k\in\mathcal{N}(i)} (D^G_k)^{-1} \exp\left(-\frac{1}{m}\sigma(\langle \mathbf{w}^{(t)}_{j,r}, \tilde{\boldsymbol{\xi}}^G_k\rangle) + \overline{n}C_F\right) \sigma'\left(\langle \mathbf{w}^{(t)}_{j,r}, \tilde{\boldsymbol{\xi}}^G_k\rangle\right) \|\boldsymbol{\xi}^G_i\|_2^2 \\
&\leq \frac{q2^q\eta\sigma_p^2 dT^*}{Nm}\alpha^{q-1}\exp\left(\overline{n}C_F - \frac{\alpha^q}{4^q m}\right) \\
&\leq 0.25\alpha\exp\left(\overline{n}C_F - \frac{\alpha^q}{4^q m}\right)T^* \\
&\leq 0.25\alpha\exp\left(-\log\left(T^*\right)\right)T^* \\
&\leq 0.25\alpha
\end{aligned}$$

where the first inequality is by the upper bound on $|\ell'^{(t)}_G|$, the second is by the bounds of $\langle \mathbf{w}^{(t)}_{j,r}, \tilde{\boldsymbol{\xi}}^G_k\rangle$, the third is by the condition that $\eta = O\left(\frac{Nm}{q2^{q+2}\alpha^{q-2}\sigma_p^2 d}\right)$, and the fourth is by the definition of α.

Similarly, we can prove that $\gamma^{(t)}_{j,r} \leq \alpha$ using the same approach under the condition that $\eta = O\left(\frac{Nm}{\overline{n}q2^{q+2}\alpha^{q-2}\|\boldsymbol{\mu}\|_2^2}\right)$. There is an extra quantity of $\overline{n}$ in the denominator as the update of $\gamma^{(t)}_{j,r}$ involves a sum pooling operation over nodes in a graph. □

A.3. Feature Learning versus Noise Memorization

We are now ready to establish the main result that the rate of signal learning significantly outpaces that of noise memorization. After a specific number of iterations, the GCN is able to learn the signal from the graph classification data at a constant or higher order, while only memorizing a smaller order of noise.

Lemma A.18. *(Restatement of Theorem 4.1: signal learning dominates noise memorization). Under Assumption A.1 and the SNR condition:*

$$N \cdot \mathrm{SNR}^q \cdot (\underline{n}(p+s))^{q/2-1} \cdot \Delta \geq C_1 \log(6/\sigma_0 \|\boldsymbol{\mu}\|_2) 2^{2q+3} [16 \log(8m\overline{n}/\delta)]^{(q-1)/2}$$

where $C_1 = O(1)$ is a positive constant and $\mathrm{SNR} \triangleq \frac{\|\boldsymbol{\mu}\|_2}{\sigma_p \sqrt{d}}$, there exists a time $T_1 = \frac{\log(6/\sigma_0\|\boldsymbol{\mu}\|_2) 2^{q+1} m}{C_2 q \eta \sigma_0^{q-2} \|\boldsymbol{\mu}\|_2^q \Xi^q \Delta}$ such that

1. $\max_r \gamma_{j,r}^{(T_1)} \geq 2$ *for* $j \in \{\pm 1\}$.

2. $|\rho_{j,r,i}^{G(t)}| \leq \sigma_0 \sigma_p \sqrt{d/(\underline{n}(p+s))}/2$ *for all* $j \in \{\pm 1\}$, $r \in [m]$, $i \in [n_G], G \in \mathcal{G}$, *and* $0 \leq t \leq T_1$.

Proof. We first establish the outcome for $|\rho_{j,r,i}^{G(t)}|$. First define $T_1^+ = \frac{Nm\eta^{-1}\sigma_0^{2-q}\sigma_p^{-q} d^{-q/2} (\underline{n}(p+s))^{(q-2)/2}}{2^{q+1} q [16 \log(8m\overline{n}/\delta)]^{(q-1)/2}}$ and $\Psi^{(t)} = \max_{j,r,G,i}\{|\rho_{j,r,i}^{G(t)}|\}$. We will use induction to demonstrate that

$$\Psi^{(t)} \leq \sigma_0 \sigma_p \sqrt{d/(\underline{n}(p+s))}/2$$

is valid for $0 \leq t \leq T_1^+$. It is known that $\Psi^{(t)} = 0$ by definition. Assuming that there exists a $\tilde{T} \leq T_1^+$ for which it holds for all $0 \leq t \leq \tilde{T} - 1$, from Lemma A.9:

$$\rho_{j,r,i}^{G(t+1)} = \rho_{j,r,i}^{G(t)} - \frac{\eta}{Nm} \sum_{k \in \mathcal{N}(i)} (D_k^G)^{-1} \ell_G^{'(t)} \sigma' \left(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{\boldsymbol{\xi}}_k^G \rangle \right) \|\boldsymbol{\xi}_i^G\|_2^2 j y_G$$

Therefore the induction can proceed as:

$$
\begin{aligned}
\Psi^{(t+1)} &\leq \Psi^{(t)} + \frac{\eta}{Nm} \sum_{k \in \mathcal{N}(i)} (D_k^G)^{-1} |\ell_G^{'(t)}| \sigma' \left(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{\boldsymbol{\xi}}_k^G \rangle \right) \|\boldsymbol{\xi}_i^G\|_2^2 |j y_G| \\
&\leq \Psi^{(t)} + \frac{\eta}{Nm} \sum_{k \in \mathcal{N}(i)} (D_k^G)^{-1} \sigma' \left(\langle \mathbf{w}_{j,r}^{(t)}, \tilde{\boldsymbol{\xi}}_k^G \rangle \right) \|\boldsymbol{\xi}_i^G\|_2^2 \\
&= \Psi^{(t)} + \frac{\eta}{Nm} \sum_{k \in \mathcal{N}(i)} (D_k^G)^{-1} \|\boldsymbol{\xi}_i^G\|_2^2 \\
&\sigma' \left(\langle \mathbf{w}_{j,r}^{(0)}, \tilde{\boldsymbol{\xi}}_k^G \rangle + \sum_{G' \in \mathcal{G}} \sum_{i' \in V^{G'}} (\zeta_{j,r,i'}^{G'(t)} + \omega_{j,r,i'}^{G'(t)}) \|\boldsymbol{\xi}_{i'}^{G'}\|_2^{-2} \langle \boldsymbol{\xi}_{i'}^{G'}, \tilde{\boldsymbol{\xi}}_k^G \rangle \right) \\
&\leq \Psi^{(t)} + \frac{\eta}{Nm} \sum_{k \in \mathcal{N}(i)} (D_k^G)^{-1} \|\boldsymbol{\xi}_i^G\|_2^2 \\
&\sigma' \left(\langle \mathbf{w}_{j,r}^{(0)}, \tilde{\boldsymbol{\xi}}_k^G \rangle + 2\Psi^{(t)} \sum_{G' \in \mathcal{G}} \sum_{i' \in V^{G'}} \|\boldsymbol{\xi}_{i'}^{G'}\|_2^{-2} |\langle \boldsymbol{\xi}_{i'}^{G'}, \tilde{\boldsymbol{\xi}}_k^G \rangle| \right) \\
&= \Psi^{(t)} + \frac{\eta q}{Nm} \sum_{k \in \mathcal{N}(i)} (D_k^G)^{-1} \|\boldsymbol{\xi}_i^G\|_2^2 \\
&\left(\langle \mathbf{w}_{j,r}^{(0)}, \tilde{\boldsymbol{\xi}}_k^G \rangle + 2\Psi^{(t)} \sum_{G' \in \mathcal{G}} \sum_{i' \in V^{G'}} \sum_{k' \in \mathcal{N}(k)} (D_k^G)^{-1} \|\boldsymbol{\xi}_{i'}^{G'}\|_2^{-2} |\langle \boldsymbol{\xi}_{i'}^{G'}, \boldsymbol{\xi}_{k'}^G \rangle| \right)^{q-1} \\
&\leq \Psi^{(t)} + \frac{\eta q}{Nm} \sum_{k \in \mathcal{N}(i)} (D_k^G)^{-1} \|\boldsymbol{\xi}_i^G\|_2^2 \\
&\left(\langle \mathbf{w}_{j,r}^{(0)}, \tilde{\boldsymbol{\xi}}_k^G \rangle + (2 + \frac{4N\overline{n}\sigma_p^2 \sqrt{d \log(4\overline{n}^2/\delta)}}{\sigma_p^2 d}) \Psi^{(t)} \right)^{q-1} \\
&\leq \Psi^{(t)} + \frac{\eta q}{Nm} \sum_{k \in \mathcal{N}(i)} (D_k^G)^{-1} \|\boldsymbol{\xi}_i^G\|_2^2 \left(\langle \mathbf{w}_{j,r}^{(0)}, \tilde{\boldsymbol{\xi}}_k^G \rangle + 4\Psi^{(t)} \right)^{q-1} \\
&\leq \Psi^{(t)} + \frac{\eta q}{Nm} \sum_{k \in \mathcal{N}(i)} (D_k^G)^{-1} 2\sigma_p^2 d \left(4\sqrt{\log(8m\overline{n}/\delta)} \sigma_0 \sigma_p \sqrt{d/(\underline{n}(p+s))} \right)^{q-1}
\end{aligned}
$$

where the second inequality is by $|\ell_G^{'(t)}| \leq 1$, the fourth inequality is by Lemma A.2 and decomposing the summation term into $G' = G, i' = k'$ and others, the fifth inequality is by the condition on d, and the final inequality is by Lemma A.2 and Lemma A.8. By summing over up to $\tilde{T} - 1$, we have:

$$
\begin{aligned}
\Psi^{(\tilde{T})} &\leq \tilde{T} \frac{\eta q}{Nm} 2\sigma_p^2 d \left(4\sqrt{\log(8m\overline{n}/\delta)} \sigma_0 \sigma_p \sqrt{d/(\underline{n}(p+s))} \right)^{q-1} \\
&\leq T_1^+ \frac{\eta q}{Nm} 2\sigma_p^2 d \left(4\sqrt{\log(8m\overline{n}/\delta)} \sigma_0 \sigma_p \sqrt{d/(\underline{n}(p+s))} \right)^{q-1} \\
&\leq \sigma_0 \sigma_p \sqrt{d/(\underline{n}(p+s))}/2^q
\end{aligned}
$$

which completes the proof for noise memorization.

Next, we assume $j = 1$ without loss of generality. Then for $\gamma_{1,r}^{(t)}$:

$$\begin{aligned}
\gamma_{1,r}^{(t+1)} &= \gamma_{1,r}^{(t)} - \frac{\eta}{Nm} \sum_{G} \sum_{i \in V^G} \ell_G^{'(t)} \sigma' \left(\langle \mathbf{w}_{1,r}^{(t)}, \tilde{z}_i^G \boldsymbol{\mu} \rangle \right) y_G \tilde{z}_i^G \|\boldsymbol{\mu}\|_2^2 \\
&= \gamma_{1,r}^{(t)} + \frac{\eta}{Nm} \sum_{G} \sum_{i \in V^G} (-\ell_G^{'(t)}) \sigma' \left(\tilde{z}_i^G \langle \mathbf{w}_{1,r}^{(0)}, \boldsymbol{\mu} \rangle + \tilde{z}_i^G \gamma_{1,r}^{(t)} \right) y_G \tilde{z}_i^G \|\boldsymbol{\mu}\|_2^2 \\
&\geq \gamma_{1,r}^{(t)} + \frac{\eta q}{Nm} \|\boldsymbol{\mu}\|_2^2 \Xi^q \left(\langle \mathbf{w}_{1,r}^{(0)}, \boldsymbol{\mu} \rangle + \gamma_{1,r}^{(t)} \right)^{q-1} \left(\frac{N}{2} \Delta - \Sigma \sqrt{\frac{N}{2} \log \frac{2}{\delta}} \right) \\
&\geq \gamma_{1,r}^{(t)} + \frac{\eta q}{4m} \|\boldsymbol{\mu}\|_2^2 \Xi^q \Delta \left(\langle \mathbf{w}_{1,r}^{(0)}, \boldsymbol{\mu} \rangle + \gamma_{1,r}^{(t)} \right)^{q-1}.
\end{aligned}$$

where the first and second inequalities are by Lemma A.10, i.e. the lower bound on $S^{(t)}$ and the condition on N. Next, we let $\hat{\gamma}_{1,r}^{(t)} = \gamma_{1,r}^{(t)} + \langle \mathbf{w}_{1,r}^{(0)}, \boldsymbol{\mu} \rangle$ and $A^{(t)} = \max_r \hat{\gamma}_{1,r}^{(t)}$. Clearly $A^{(0)} = \max_r \langle \mathbf{w}_{1,r}^{(0)}, \boldsymbol{\mu} \rangle$. Then we have

$$\begin{aligned}
A^{(t+1)} &\geq A^{(t)} + \frac{\eta C_2 q}{4m} \|\boldsymbol{\mu}\|_2^2 \Xi^q \Delta (A^{(t)})^{q-1} \\
&\geq A^{(t)} (1 + \frac{\eta C_2 q}{4m} \|\boldsymbol{\mu}\|_2^2 \Xi^q \Delta (A^{(0)})^{q-2}) \\
&\geq A^{(t)} (1 + \frac{\eta C_2 q}{2^q m} \|\boldsymbol{\mu}\|_2^q \Xi^q \Delta \sigma_0^{q-2})
\end{aligned}$$

where the second inequality is by Lemma A.10 which states that $\gamma_{j,r}^{(t)}$ monotonically increases with t, and the third inequality is by Lemma A.3. It is known that $1 + z \geq \exp(z/2)$ for $z \leq 2$. Therefore, by multiplying over t, we have

$$\begin{aligned}
A^{(t)} &\geq A^{(0)} \left(1 + \frac{\eta C_2 q}{2^q m} \|\boldsymbol{\mu}\|_2^q \Xi^q \Delta \sigma_0^{q-2} \right)^t \\
&\geq A^{(0)} \exp \left(\frac{\eta C_2 q}{2^{q+1} m} \|\boldsymbol{\mu}\|_2^q \Xi^q \Delta \sigma_0^{q-2} t \right) \\
&\geq \frac{\sigma_0 \|\boldsymbol{\mu}\|_2}{2} \exp \left(\frac{\eta C_2 q}{2^{q+1} m} \|\boldsymbol{\mu}\|_2^q \Xi^q \Delta \sigma_0^{q-2} t \right)
\end{aligned}$$

Finally we let $A^{(t)} \geq 3$, then

$$t \geq \frac{\log(6/\sigma_0 \|\boldsymbol{\mu}\|_2) 2^{q+1} m}{C_2 q \eta \sigma_0^{q-2} \|\boldsymbol{\mu}\|_2^q \Xi^q \Delta} = T_1$$

therefore after T_1 iterations, $\max_r \gamma_{1,r}^{(t)} \geq 2$. The case where $j = -1$ can be proved following the same logic. Finally, we need signal learning during the duration of negligible noise memorization, i.e. $T_1 \leq \frac{1}{2} T_1^+$, giving the SNR condition. □

As a result, the learned weights are more aligned with signals than with noises if we only vary σ_0 and keep it small:

Corollary A.19. *(Restatement of Theorem 4.1: weight alignment). Under Assumption A.1, there exist $0 \leq t \leq T$ and $r^* \in [m]$ such that, for $y_G \in \{\pm 1\}$, $\gamma_{y_G, r^*}^{(t)} \geq 2$ and $|\rho_{y_G, r^*, i}^{G(t)}| \leq \sigma_0 \sigma_p \sqrt{d/(\underline{n}(p+s))}/2$. Consequently, the corresponding weights satisfy $\langle \mathbf{w}_{y_G, r^*}^{(t)}, y_G \boldsymbol{\mu} \rangle \geq 2 - C_1 \sigma_0$ and $|\langle \mathbf{w}_{y_G, r^*}^{(t)}, \tilde{\boldsymbol{\xi}}_i^G \rangle| \leq C_2 \sigma_0 + 1$, where C_1 and C_2 are positive constants independent of σ_0.*

Proof. (1) For signal learning:

By Lemma A.12

$$\langle \mathbf{w}_{y_G, r^*}^{(t)}, y_G \boldsymbol{\mu} \rangle = \langle \mathbf{w}_{y_G, r^*}^{(0)}, y_G \boldsymbol{\mu} \rangle + \gamma_{y_G, r^*}^{(t)}$$

By Lemma A.3:

$$|\langle \mathbf{w}_{y_G, r^*}^{(0)}, y_G \boldsymbol{\mu} \rangle| \leq \sqrt{2 \log(8m/\delta)} \sigma_0 \|\boldsymbol{\mu}\|_2$$

Therefore:

$$\langle \mathbf{w}_{y_G,r^*}^{(t)}, y_G \boldsymbol{\mu} \rangle \geq 2 - \sqrt{2\log(8m/\delta)}\sigma_0 \|\boldsymbol{\mu}\|_2 = 2 - C_1\sigma_0$$

where C_1 is independent of σ_0.

(2) For noise memorization:

$$\begin{aligned}
\langle \mathbf{w}_{y_G,r^*}^{(t)}, \tilde{\boldsymbol{\xi}}_i^G \rangle &\leq \langle \mathbf{w}_{y_G,r^*}^{(0)}, \tilde{\boldsymbol{\xi}}_i^G \rangle + \hat{\rho}_{y_G,r^*,i}^{G(t)} + 8N\overline{n}\sqrt{\log(4\overline{n}^2/\delta)/d}\alpha \\
&\leq 2\sqrt{\log(8m\overline{n}/\delta)}\sigma_0\sigma_p\sqrt{d/(\underline{n}(p+s))} + \\
&\sigma_0\sigma_p\sqrt{d/(\underline{n}(p+s))}/2 + 8N\overline{n}\sqrt{\log(4\overline{n}^2/\delta)/d}\alpha \\
&\leq \sigma_0\sigma_p\sqrt{d/(\underline{n}(p+s))}(2\sqrt{\log(8m\overline{n}/\delta)} + \frac{1}{2}) + 1 \\
&= C_2\sigma_0 + 1
\end{aligned}$$

where the first inequality is by Lemma A.13, the second inequality is by Lemma A.8, the third inequality is by the condition on d. Using the same logic we can prove that:

$$\langle \mathbf{w}_{y_G,r^*}^{(t)}, \tilde{\boldsymbol{\xi}}_i^G \rangle \geq \sigma_0\sigma_p\sqrt{d/(\underline{n}(p+s))}(-2\sqrt{\log(8m\overline{n}/\delta)} - \frac{1}{2}) - 1 = -C_2\sigma_0 - 1.$$

□

B. KTA Complexity

Writing Frobenius norm as trace and using cyclicity of trace, there exists an equivalent computation as KTA $= \frac{\|\boldsymbol{\Phi}^\top \mathbf{y}\|_F^2}{\|\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\|_F \|\mathbf{y}^\top\mathbf{y}\|_F}$, where $\boldsymbol{\Phi} \in \mathbb{R}^{N\times d}$ is the internal representation and $\mathbf{y}$ is the label. Note $\boldsymbol{\Phi}$ and $\mathbf{y}$ are mean-centered. In this way, we avoid constructing the full feature and label kernels $\mathbf{K} = \boldsymbol{\Phi}\boldsymbol{\Phi}^\top, \mathbf{L} = \mathbf{y}\mathbf{y}^\top \in \mathbb{R}^{N\times N}$. Since the dominant term is now $\boldsymbol{\Phi}^\top\boldsymbol{\Phi}$ instead of $\mathbf{K}$, the complexity is reduced from $O(N^2 d)$ to $O(Nd^2)$. This is significantly more efficient for large-scale data, where $d \ll N$. Specifically:

$$\begin{aligned}
\mathrm{KTA}(\mathbf{K},\mathbf{L}) &= \frac{\langle \mathbf{K},\mathbf{L}\rangle_F}{\|\mathbf{K}\|_F\|\mathbf{L}\|_F} = \frac{\mathrm{tr}(\mathbf{K}^\top\mathbf{L})}{\sqrt{\mathrm{tr}(\mathbf{K}^\top\mathbf{K})}\sqrt{\mathrm{tr}(\mathbf{L}^\top\mathbf{L})}} \\
&= \frac{\mathrm{tr}(\boldsymbol{\Phi}\boldsymbol{\Phi}^\top\mathbf{y}\mathbf{y}^\top)}{\sqrt{\mathrm{tr}(\boldsymbol{\Phi}\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\boldsymbol{\Phi}^\top)}\sqrt{\mathrm{tr}(\mathbf{y}\mathbf{y}^\top\mathbf{y}\mathbf{y}^\top)}} \\
&= \frac{\mathrm{tr}(\mathbf{y}^\top\boldsymbol{\Phi}\boldsymbol{\Phi}^\top\mathbf{y})}{\sqrt{\mathrm{tr}(\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\boldsymbol{\Phi}^\top\boldsymbol{\Phi})}\sqrt{\mathrm{tr}(\mathbf{y}^\top\mathbf{y}\mathbf{y}^\top\mathbf{y})}} \\
&= \frac{\|\boldsymbol{\Phi}^\top\mathbf{y}\|_F^2}{\|\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\|_F\,\|\mathbf{y}^\top\mathbf{y}\|_F}.
\end{aligned}$$

To assess its complexity on very large-scale graph data, we measured the time and memory per epoch and KTA computation on OGBG-PPA, and Table 6 shows the extra time and memory added by KTA computation relative to each training epoch is negligible.

Table 6. Additional computational overhead of each KTA computation relative to each training epoch in terms of time and memory on OGBG-PPA.

Model	Extra time (%)	Extra memory (%)
GCN	0.00017%	5.82%
GCN+	0.00036%	7.08%
GAT	0.00013%	0.93%
GraphGPS	0.00013%	1.35%

C. Datasets and Experimental Details

Our implementation is based on PyG (Fey & Lenssen, 2019). The experiments are conducted on a single workstation with 8 NVIDIA L40 GPUs.

C.1. Datasets

We used the following datasets for graph-level tasks: (1) **MNIST** is a dataset from (Dwivedi et al., 2023). It downsamples each handwritten digit image into a graph of 75 superpixels linked by adjacency, and each node (superpixel) has exactly eight neighbors, constituting a regular graph structure. The node features are local pixel intensities and 2D coordinates in the image, and the edge attributes are distances between pixels. (2) **NCI** (Morris et al., 2020) contains small-molecule graphs labeled by anti-cancer activity, with atoms as nodes and bond types as edges. (3) **PROTEINS** (Morris et al., 2020) contains protein graphs where nodes represent amino acids. (4) **IMDB** (Morris et al., 2020) contains movie graphs where the nodes are actors connected by co-appearance. Since these datasets have no node features, we use one-hot node degrees as synthetic features following common practice (Xu et al., 2019). The maximum degree is capped at 10. (5) **Peptides-struct** (Dwivedi et al., 2022b) consists of peptide molecules represented as graphs, where nodes correspond to amino acids and edges capture chemical bonds and spatial proximity. (6) **OGBG-MOLHIV** (Hu et al., 2020) contains molecular graphs where nodes represent atoms and edges represent chemical bonds between them, where positive samples represent only 3.5%. (7) **OGBG-PPA** (Hu et al., 2020) contains protein-protein association graphs extracted from biological interaction networks, where nodes represent proteins and edges indicate associations between them. As OGBG-PPA has edge attributes but no node features, we use the averaged edge attributes of a node's adjacent edges as the feature for that node. We use 10-fold random split on the datasets from (Morris et al., 2020) and the given train-test split on the others.

The task is to predict whether a given molecular graph exhibits anti-cancer activity on NCI, determine whether a protein graph represents an enzyme or not on PROTEINS, and classify handwritten digit images represented as graphs into digit classes on MNIST. For IMDB, the task is to classify movie actor–coappearance graphs as Action versus Romance in the BINARY setting, and as Comedy, Sci-Fi, or Drama in the MULTI setting. Peptides-struct requires predicting multiple continuous structural properties of peptide graphs for a graph regression task. On OGBG-MOLHIV, the task is to predict whether a molecular graph is associated with HIV inhibition. Finally, on OGBG-PPA, the task is to classify protein–protein association graphs into one of 37 classes corresponding to the biological species from which the graph is derived.

In Table 7, we give the basic statistics of the datasets, including the number of graphs and classes, the average number of nodes and edges in a graph, as well as the number of node and edge features.

Table 7. Statistics of graph-level datasets used in this study.

Dataset	# Graphs	# Classes	Avg. Nodes	Avg. Edges	# node/edge feats
MNIST	70,000	10	70.6	564.5	3/1
PROTEINS	1,113	2	39.06	72.82	3/-
NCI1	4,110	2	29.87	32.30	37/-
NCI109	4,127	2	29.68	32.13	38/-
IMDB-BINARY	1,000	2	19.77	96.53	-/-
IMDB-MULTI	1,500	3	13.00	65.94	-/-
Peptides-struct	15,535	11	150.94	307.30	9/3
OGBG-MOLHIV	41,127	2	25.5	27.5	9/3
OGBG-PPA	158,100	37	243.4	2,266.1	-/7

C.2. Models

Message-passing neural networks (MPNNs) Classic GNNs are typically MPNNs that aggregate information through iterative message passing between local neighbors:

$$\mathbf{h}_v^{(l+1)} = \text{UPDATE}^{(l)}\left(\mathbf{h}_v^{(l)}, \text{AGG}^{(l)}\left(\left\{\mathbf{h}_u^{(l)} : u \in \mathcal{N}(v)\right\}\right)\right)$$

where $\mathbf{h}_v^{(l)}$ is node v's representation at layer l, $\mathcal{N}(v)$ is the local neighborhood of v, AGG is a permutation-invariant function such as mean, and UPDATE is a differentiable function such as MLP.

In our study, **SGC** (Wu et al., 2019) is a linear baseline that uses multi-hop neighborhood aggregation as input to a linear classifier. **GCN** (Kipf & Welling, 2017) performs localized graph convolution using the normalized adjacency matrix, while **GCNII** (Chen et al., 2020) improves deep GCNs through residual connections. **GraphSAGE** (Hamilton et al., 2017) inductively aggregates local neighbor features. **GAT** (Veličković et al., 2018) applies self-attention to assign learnable weights to neighboring nodes. **GCN**$^+$ from the **GNN**$^+$ family (Luo et al., 2025) augments GCN by interleaving GCN layers with two-layer MLPs, whose architecture takes the form: GCN layer $\rightarrow$ MLP $\rightarrow \cdots \rightarrow$ GCN layer $\rightarrow$ MLP with residual connections. **A-DGN** (Gravina et al., 2023) uses residual Euler discretization of an anti-symmetric continuous ODE to improve stable and long-range message passing.

Graph transformers (GTs) GTs are adapted from the Transformer architecture where global attention allows each node to attend to other nodes in the graph (Kim et al., 2022):

$$\mathbf{h}_v^{(l+1)} = \text{FFN}^{(l)}\left(\sum_{u \in V^G} \alpha_{vu}^{(l)} \cdot \mathbf{W}_V^{(l)} \mathbf{h}_u^{(l)}\right)$$

where $\alpha_{vu}^{(l)}$ is an attention coefficient, $\mathbf{W}_V^{(l)}$ is a value projection, and FFN is a feedforward network.

In our study, **GT** (Dwivedi & Bresson, 2021) extends the Transformer architecture to graphs by incorporating local neighborhood attention and positional encodings. **TokenGT** (Kim et al., 2022) treats all nodes and edges as independent tokens and uses a pure Transformer to achieve expressiveness beyond MPNNs. **GraphGPS** (Rampášek et al., 2022) integrates local message passing and global attention modules in a hybrid Transformer-style framework. Typically, graph transformers neglect structural information due to global attention. Therefore, a critical component of graph transformers is **positional encodings (PEs)**, which act as soft "hints" of graph structure (Dwivedi & Bresson, 2021; Rampášek et al., 2022). Laplacian PE (Dwivedi et al., 2023) (LapPE) and random walk PE (Dwivedi et al., 2022a) (RWPE) are both commonly used positional encodings.

To accurately assess feature learning, following (Woodworth et al., 2020), we rescale the models at initialization by α and and tune optimization hyperparameters so that models achieve near-perfect training performance (close to 100% training accuracy or minimal training loss) in order to avoid confounding effects from underfitting or poor optimization (Samarin et al., 2022). Therefore, the hyperparameters reported for prior state-of-the-art results should not be directly adopted. To address this, we select the smallest value of hidden dimensions and number of layers that achieves near-perfect training performance, while keeping the model sizes approximately within the same magnitude. Note that generally graph models should not be too deep due to issues such as oversmoothing.

Tables 8 9 provide the model configurations used for the feature learning experiments, including the hidden dimension, the dimension of positional encodings if applicable, and the number of layers. We use four attention heads for all attention modules, and mean graph pooling for all settings. Please note that for SGC, the number of layers in fact refers to the number of convolution iterations. Moreover, for the image MNIST dataset, we use a two-layer 5000 hidden unit MLP following (Woodworth et al., 2020).

Table 8. The model hyperparameter settings on MNIST, NCI1, and PROTEINS.

Model	MNIST				NCI1				PROTEINS			
	Hidden Dim	PE Dim	# Layers	# Parameters	Hidden Dim	PE Dim	# Layers	# Parameters	Hidden Dim	PE Dim	# Layers	# Parameters
SGC	8192	-	5	114,698	8192	-	4	327,682	8192	-	4	49,154
MLP	512	-	4	1,055,232	256	-	3	206,592	256	-	3	197,888
MLP w/ PE	256	32	4	266,088	256	32	3	205,508	256	32	3	197,926
GCN	512	-	4	1,055,232	256	-	3	206,592	256	-	3	197,888
GCNII	512	-	4	1,055,744	256	-	3	206,848	256	-	3	198,144
GraphSAGE	256	-	4	528,384	256	-	3	412,672	256	-	3	395,264
GAT	128	-	4	270,080	128	-	3	219,904	128	-	3	202,496
A-DGN	256	-	4	528,896	256	-	4	535,552	256	-	4	526,848
GCN$^+$	128	16	2	170,514	128	16	2	173,998	128	16	2	170,208
GT	128	16	2	269,842	128	16	2	273,190	128	16	2	269,400
TokenGT	128	16	2	269,842	128	16	2	273,190	128	16	2	269,400
GraphGPS	128	16	2	403,218	128	16	2	406,566	128	16	2	402,776

Table 9. The model hyperparameter settings on Peptides-struct, OGBG-MOLHIV, and OGBG-PPA.

Model	Peptides-struct				OGBG-MOLHIV				OGBG-PPA			
	Hidden Dim	PE Dim	# Layers	# Parameters	Hidden Dim	PE Dim	# Layers	# Parameters	Hidden Dim	PE Dim	# Layers	# Parameters
MLP	512	-	4	1,143,296	512	-	4	1,138,176	512	-	4	1,075,200
MLP w/ PE	512	32	4	1,137,930	512	32	4	1,132,674	512	32	4	1,074,922
GCN	512	-	4	1,143,296	512	-	4	1,138,176	512	-	4	1,075,200
GAT	256	-	4	1,107,456	256	-	4	1,104,896	128	-	4	272,896
GCN^+	256	32	2	715,477	256	32	2	715,011	256	32	2	680,719
GraphGPS	128	16	2	423,413	128	16	2	423,171	128	16	2	404,383

C.3. Training

Typically smaller values of α (which correspond to optimization saddle points) require larger learning rates, while large α values destabilize training under high learning rates. Thus, we tune learning rates to decrease monotonically with increasing α. All models on the same dataset are trained for a fixed number of epochs, after which training stabilizes. To reach near-perfect training performance, regularization such as weight decay, early stopping, and dropout are removed. There remain a few cases where near-perfect training performance cannot be achieved due to inherent model limitations, which we attempt to attain the best training performance possible.

Table 10 gives the training hyperparameters of the feature learning experiments, including the learning rate range, number of epochs, and batch size of each dataset. All models on the same dataset share the same training hyperparameter settings, and the optimal learning rate for each model and α is obtained using grid search. We use Adam optimizer (Kingma & Ba, 2015) in all graph experiments.

Table 10. The training hyperparameter settings.

Dataset	Learning Rate Range	# Epochs	Batch Size
MNIST	$[5 \times 10^{-5}, 1 \times 10^{-3}]$	500	64
NCI1	$[5 \times 10^{-4}, 7 \times 10^{-3}]$	500	256
PROTEINS	$[1 \times 10^{-4}, 6 \times 10^{-3}]$	500	256
Peptides-struct	$[1 \times 10^{-4}, 5 \times 10^{-3}]$	500	256
OGBG-MOLHIV	$[1 \times 10^{-4}, 5 \times 10^{-3}]$	500	256
OGBG-PPA	$[0.3 \times 10^{-3}, 3 \times 10^{-3}]$	300	128

For transfer learning, we adopt the following procedure:

MNIST We train the models on the training set of 0-4, compute the test accuracy and KTA on the test set of 0-4, then tune the linear classifiers on the training set of 5-9, and finally evaluate the transfer accuracy on the test set of 5-9. For the models, we use a hidden dimension of 256 for all models so that the linear classifiers have the same size, i.e. 256×5. We train all models for 500 epochs before freezing their weights and probing for 20 epochs with a learning rate of 1×10^{-3}.

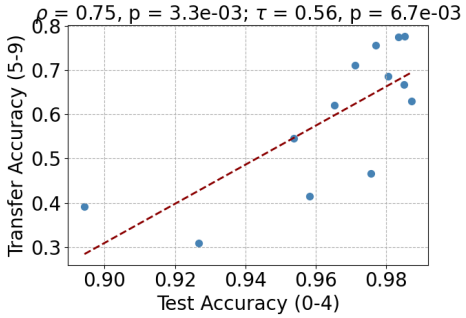
NCI and IMDB We use a hidden dimension of 64 and 200 epochs on IMDB, and a hidden dimension of 128 and 500 epochs on NCI1. We then probe for 20 epochs with a learning rate of 1×10^{-3}. The results are averaged across 10 random splits. Since NCI109 has one more node feature than NCI1 (Table 7), we drop the last feature of NCI109 to align their number of node features.

Node-level Depending on task difficulty level, we use a hidden size of 32 for Planetoid, 64 for Coauthor, and 128 for WikipediaNetwork. We pre-train for 100 epochs and probe for 10 epochs with a learning rate of 1×10^{-3} on all datasets. The results are averaged across 10 random splits. For dataset pairs with different node feature dimensions, we append a linear feature dimension alignment head prior to the frozen trained model, and the alignment head is optimized during probe.

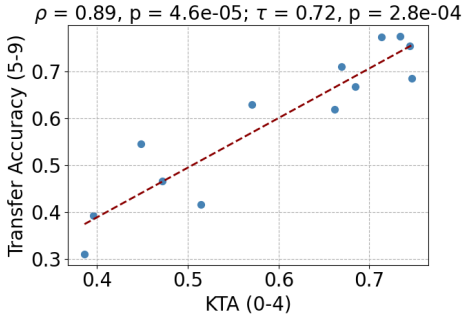
D. Graph-level Results

D.1. Transfer Learning

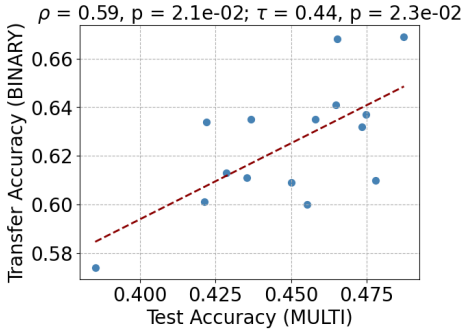
Figure 2 shows the transfer learning results on MNIST and IMDB. The results for NCI are in Figure 1. A stronger correlation is observed between KTA and transfer accuracy in all tasks, and KTA correlates more consistently with transfer accuracy than test accuracy.



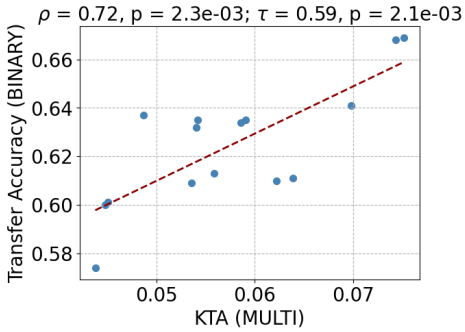
(a) MNIST 0-4 → 5-9: test accuracy vs transfer accuracy.



(b) MNIST 0-4 → 5-9: KTA vs transfer accuracy .



(c) IMDB-MULTI → -BINARY: test accuracy vs transfer accuracy.



(d) IMDB-MULTI → -BINARY: KTA vs transfer accuracy.

Figure 2. Each row represents a transfer setting (MNIST or IMDB), and the columns show test accuracy or KTA, respectively. Each point represents a model. KTA is a better indicator of generalizability than test accuracy.

D.2. Feature Learning

We summarize the peak KTA value for each model at $\alpha = 0.1, 1, 10$ in Table 5 11 12. The results are averaged over three random seeds with mean and standard deviation reported. For simplicity, we present the results of six representative models in Table 5 12, i.e. MLP, MLP with LapPE, GCN, GCN$^+$, GAT, GraphGPS to cover discussed feature learning factors. In Table 5 11, we again verify that feature learning is promoted by (1) positional encodings (e.g., MLP with LapPE), (2) interleaving with MLP layers (e.g., GCN$^+$), and (3) attention mechanisms (e.g., GAT and GraphGPS).

In Table 12, we observe that feature learning depends on receptive fields, whereas on other datasets, models that work in local neighborhoods or on whole graphs both can operate in the feature learning regime. Specifically, since Peptides-struct requires models to capture dependencies between distant nodes, models that work in local neighborhoods no longer work in the feature learning regime even with MLP interleaving (GCN$^+$) and attention mechanisms (GAT). In contrast, graph transformers that operate on whole graphs exhibit strong feature learning. Positional encodings are also very helpful, as indicated by the high KTA values of MLP w/ LapPE.

Table 11. KTA values for each model and α on NCI1 and PROTEINS. KTA is evaluated on the test set. The highest KTA across α under the same model and dataset is underlined. If the highest KTA of a model on a dataset remains low across different α, then the model does not operate in the feature learning regime. The table is thereby organized from top to bottom into three groups: (1) MLP, (2) models that do not work in the feature learning regime, and (3) models that work in the feature learning regime. Feature learning is promoted by (1) positional encodings, (2) interleaving with MLPs, and (3) attention mechanisms.

Model	NCI1			PROTEINS		
	$\alpha = 0.1$	$\alpha = 1$	$\alpha = 10$	$\alpha = 0.1$	$\alpha = 1$	$\alpha = 10$
MLP	0.027 ± 0.002	$\underline{0.028 \pm 0.001}$	0.026 ± 0.001	0.015 ± 0.000	0.015 ± 0.000	$\underline{0.016 \pm 0.000}$
MLP w/ RWPE	0.060 ± 0.003	$\underline{0.065 \pm 0.001}$	0.046 ± 0.005	0.125 ± 0.002	$\underline{0.125 \pm 0.004}$	0.080 ± 0.006
MLP w/ LapPE	0.110 ± 0.008	$\underline{0.128 \pm 0.002}$	0.090 ± 0.002	$\underline{0.233 \pm 0.001}$	0.231 ± 0.001	0.195 ± 0.007
SGC	$\underline{0.048 \pm 0.001}$	0.044 ± 0.001	0.024 ± 0.000	$\underline{0.057 \pm 0.000}$	0.033 ± 0.000	0.036 ± 0.000
GCN	0.110 ± 0.004	$\underline{0.127 \pm 0.003}$	0.052 ± 0.001	$\underline{0.111 \pm 0.001}$	0.088 ± 0.002	0.069 ± 0.007
GCNII	0.065 ± 0.003	$\underline{0.079 \pm 0.003}$	0.043 ± 0.001	$\underline{0.117 \pm 0.003}$	0.049 ± 0.003	0.061 ± 0.009
GraphSAGE	0.105 ± 0.002	$\underline{0.115 \pm 0.007}$	0.062 ± 0.004	0.064 ± 0.004	0.086 ± 0.003	$\underline{0.095 \pm 0.009}$
A-DGN	0.066 ± 0.007	$\underline{0.069 \pm 0.005}$	0.042 ± 0.000	0.021 ± 0.000	0.021 ± 0.000	$\underline{0.022 \pm 0.001}$
GAT	0.153 ± 0.008	$\underline{0.164 \pm 0.002}$	0.072 ± 0.004	$\underline{0.163 \pm 0.002}$	0.159 ± 0.005	0.092 ± 0.007
GCN$^+$	$\underline{0.166 \pm 0.011}$	0.161 ± 0.020	0.119 ± 0.009	$\underline{0.169 \pm 0.047}$	0.145 ± 0.075	0.059 ± 0.003
GT	0.205 ± 0.004	$\underline{0.222 \pm 0.007}$	0.078 ± 0.001	$\underline{0.232 \pm 0.006}$	0.219 ± 0.010	0.160 ± 0.015
TokenGT	0.146 ± 0.005	$\underline{0.165 \pm 0.003}$	0.092 ± 0.026	$\underline{0.215 \pm 0.035}$	0.213 ± 0.013	0.139 ± 0.013
GraphGPS	$\underline{0.198 \pm 0.004}$	0.184 ± 0.008	0.099 ± 0.003	0.193 ± 0.015	$\underline{0.203 \pm 0.023}$	0.187 ± 0.002

Table 12. KTA values for each model and α on Peptides-struct. KTA is evaluated on the test set. The highest KTA across α under the same model is underlined. If the highest KTA of a model on a dataset remains low across different α, then the model does not operate in the feature learning regime. Feature learning is promoted by (1) global attention, and (2) positional encodings.

Model	Peptides-struct		
	$\alpha = 0.1$	$\alpha = 1$	$\alpha = 10$
MLP	0.322 ± 0.003	$\underline{0.334 \pm 0.008}$	0.260 ± 0.027
MLP w/ LapPE	0.597 ± 0.003	$\underline{0.599 \pm 0.001}$	0.402 ± 0.089
GCN	$\underline{0.434 \pm 0.008}$	0.434 ± 0.004	0.369 ± 0.006
GCN$^+$	0.422 ± 0.002	$\underline{0.424 \pm 0.018}$	0.419 ± 0.007
GAT	0.413 ± 0.010	$\underline{0.437 \pm 0.010}$	0.339 ± 0.046
GraphGPS	0.586 ± 0.027	$\underline{0.589 \pm 0.021}$	0.511 ± 0.010

E. Node-level Results

E.1. Transfer Learning

Table 13 shows the transfer learning results on Cora → CiteSeer from Planetoid (Yang et al., 2016), CS → Physics from Coauthor (Shchur et al., 2019), and Squirrel → Chameleon from WikipediaNetwork (Rozemberczki et al., 2021). A strong correlation is observed between KTA and transfer accuracy in all tasks, and KTA correlates more consistently with transfer accuracy than test accuracy.

Table 13. The correlation between transfer accuracy and test accuracy (column 2), and the correlation between transfer accuracy and KTA (column 3), measured by Spearman's ρ and Kendall's τ. p stands for p-value. A strong correlation is indicated by high values of ρ and τ. *Test* refers to test accuracy and *Transfer* refers to transfer accuracy. KTA is more strongly correlated with transfer accuracy than test accuracy.

Task	Test vs Transfer		KTA vs Transfer	
	ρ (p)	τ (p)	ρ (p)	τ (p)
Cora→CiteSeer	$0.46\ (1.3 \times 10^{-1})$	$0.26\ (2.4 \times 10^{-1})$	$0.89\ (1.1 \times 10^{-4})$	$0.70\ (9.7 \times 10^{-4})$
CS→Physics	$-0.22\ (5.3 \times 10^{-1})$	$-0.20\ (4.8 \times 10^{-1})$	$0.68\ (2.9 \times 10^{-2})$	$0.51\ (4.7 \times 10^{-2})$
Squirrel→Chameleon	$0.17\ (6.7 \times 10^{-1})$	$0.11\ (7.6 \times 10^{-1})$	$0.68\ (4.2 \times 10^{-2})$	$0.56\ (4.5 \times 10^{-2})$

E.2. Feature Learning

In Table 14, we study the feature learning of six representative models on eight homophilic node classification datasets of varying scales, where we verify our conclusions on graph-level tasks generally hold even on homophilic node classification tasks. Specifically: (1) graph transformers are the most effective, with GraphGPS achieving the highest KTA on six datasets; (2) MLP interleaving, as implemented in GCN+, improves KTA over GCN on five datasets; and (3) positional encodings improve KTA over MLP on seven datasets.

Table 14. KTA of six representative models on eight homophilic node classification datasets of different scales: small (Cora, CiteSeer, PubMed from Planetoid (Yang et al., 2016)), moderate (Physics and CS from Coauthor (Shchur et al., 2019); Computers and Photo from Amazon (Shchur et al., 2019)), and large (ArXiv from OGBN (Hu et al., 2020)). KTA is evaluated on the test set. The highest KTA across α under the same model and dataset is underlined. If the highest KTA of a model on a dataset remains low across different α, then the model does not operate in the feature learning regime. Feature learning is promoted by (1) positional encodings, (2) interleaving with MLP layers, and (3) attention mechanisms. Due to table size limit, we only report mean here.

Model	Cora			CiteSeer			PubMed			Physics			CS			Computers			Photo			ArXiv		
	$\alpha=0.1$	$\alpha=1$	$\alpha=10$	$\alpha=0.1$	$\alpha=1$	$\alpha=10$	$\alpha=0.1$	$\alpha=1$	$\alpha=10$	$\alpha=0.1$	$\alpha=1$	$\alpha=10$	$\alpha=0.1$	$\alpha=1$	$\alpha=10$	$\alpha=0.1$	$\alpha=1$	$\alpha=10$	$\alpha=0.1$	$\alpha=1$	$\alpha=10$	$\alpha=0.1$	$\alpha=1$	$\alpha=10$
MLP	0.1883	0.1955	0.1758	0.1878	0.2041	0.1855	0.3703	0.4215	0.3947	0.3489	0.3346	0.3234	0.1435	0.1178	0.1297	0.2449	0.2052	0.1662	0.2134	0.1494	0.1562	0.361	0.361	0.272
MLP w/ PE	0.1915	0.1958	0.1850	0.1911	0.2060	0.1749	0.3737	0.4151	0.3493	0.3602	0.3331	0.3284	0.1459	0.1222	0.1330	0.2507	0.2034	0.1761	0.2284	0.1496	0.1516	0.370	0.375	0.259
GCN	0.2196	0.2012	0.1876	0.2092	0.1963	0.1886	0.3749	0.3570	0.3447	0.4110	0.3805	0.3563	0.1594	0.1398	0.1489	0.1695	0.1726	0.1440	0.1499	0.1486	0.1360	0.225	0.210	0.129
GCN+	0.1867	0.1976	0.1993	0.1865	0.1939	0.1864	0.4007	0.3711	0.3161	0.3834	0.3927	0.3616	0.1727	0.1579	0.1626	0.2813	0.2391	0.1478	0.2436	0.2099	0.1430	0.379	0.374	0.162
GAT	0.2339	0.2237	0.2080	0.2189	0.2120	0.2001	0.3902	0.4148	0.3986	0.4460	0.4128	0.3967	0.1750	0.1525	0.1548	0.2315	0.2236	0.2065	0.1940	0.1781	0.1718	0.382	0.376	0.359
GraphGPS	0.2345	0.2443	0.1878	0.2193	0.2245	0.1830	0.4141	0.3963	0.3569	0.4354	0.4525	0.3590	0.2020	0.2011	0.1398	0.3134	0.3076	0.1995	0.2810	0.2776	0.1696	0.378	0.378	0.182

F. Corroborating Results in Other Settings

F.1. Measuring Alignment between Initial and Final Feature Kernels

We measure the degree to which the internal representations have changed during training by the centered alignment between the initial and final feature kernels. In Table 15, we plot the alignment across different α values for different models on MNIST. A low alignment means the model's internal representations have changed significantly during training, indicating feature learning. If a model's alignment generally increases with α or even stays low across α, then it works in the feature learning regime. From the results in Table 15, we can clearly observe the same conclusions as in Table 2, 3, 4: (1) PE helps MLP partially transition into the feature learning regime; (2) most MPNNs do not work in the feature learning regime; (3) MLP-interleaved and attention-based GNNs effectively work in the feature learning regime, with graph transformers being the most effective.

F.2. Sweeping Learning Rates

We fix $\alpha = 1$ and evaluate the effect of varying learning rates alone. Specifically, we evaluate six representative models with learning rates from 10^{-7} to 10^{-2} on MNIST. Learning rates higher than 10^{-2} would eventually lead to training divergence even with aggressive warmup. Table 16 shows KTA for each model across learning rates (LR). The results are averaged over three random seeds with mean and standard deviation reported. As learning rates increase, the KTA values of models that work in the feature learning regime should increase faster and reach higher values. The findings are consistent with insights from main body: attention-based (GAT, GraphGPS) and MLP-interleaved (GCN+) GNNs work in the feature learning regime more effectively, as evidenced by their substantially faster KTA growth and higher average KTA than MLP and GCN.

F.3. OOD Generalization to Different Scaffold Splits

Here we examine if a good KTA also indicates good out-of-distribution (OOD) generalization across scaffold splits on OGBG-MOLHIV. Specifically, in the OGB implementation of this dataset, the train/validation/test subsets are obtained using the scaffold splitting procedure that splits the molecules based on their two-dimensional structural frameworks. As a result, the subsets contain separate structurally different molecules (Hu et al., 2020). In Table 17, we report for $\alpha = 1$ and the six representative models, their (1) peak KTA values and (2) ROC-AUC score at the peak KTA epoch, where we can observe that MLP w/ LapPE, GCN$^+$, GAT, and GraphGPS achieve higher KTA and ROC-AUC values at the same time.

F.4. Uniform Model Width

We apply a uniform model width of 256 on Peptides-struct. Table 18 is constructed in the same way as Table 12, where we clearly observe the same conclusions as in Table 12. In particular, the peak KTA values of the models remain largely the same as in Table 12.

Table 15. Initial and final feature kernel alignment across different α values. Kernels are constructed on the test set. Results are averaged over three random seeds; the shaded regions denote the standard deviations. Left: MLP and most MPNNs do not work in the feature learning regime. Right: MLP with PE, MLP-interleaved MPNN, and attention-based architectures exhibit stronger feature learning, with graph transformers being the most effective.

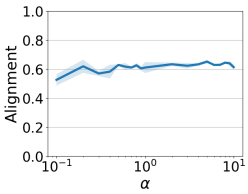
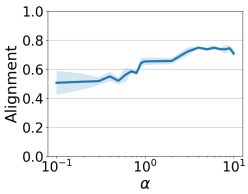
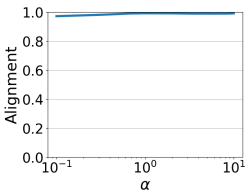
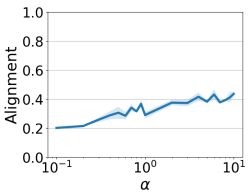
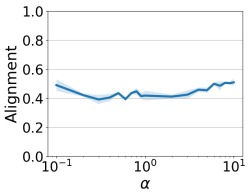
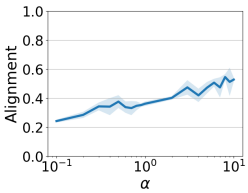
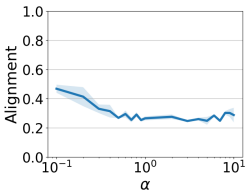
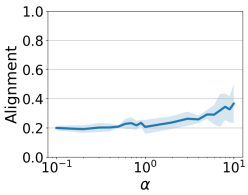
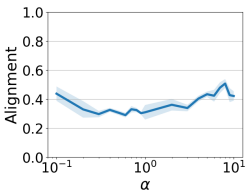
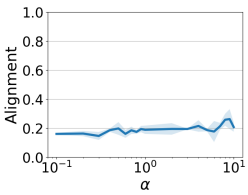
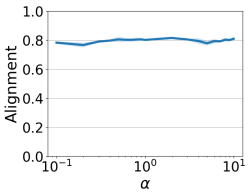
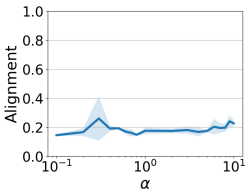
	Alignment vs α		Alignment vs α
MLP		MLP w/ PE	
SGC		GCN$^+$	
GCN		GAT	
GCNII		GT	
GraphSAGE		TokenGT	
A-DGN		GraphGPS	

Table 16. KTA values across learning rates with average KTA per model on MNIST. Feature learning is promoted by (1) positional encodings, (2) interleaving with MLP layers, and (3) attention mechanisms.

Model/LR	10^{-7}	10^{-6}	10^{-5}	10^{-4}	10^{-3}	10^{-2}	**Avg KTA**
MLP	0.139 ± 0.001	0.193 ± 0.004	0.184 ± 0.004	0.242 ± 0.007	0.307 ± 0.003	0.321 ± 0.005	0.231 ± 0.001
MLP w/ PE	0.138 ± 0.005	0.187 ± 0.008	0.206 ± 0.009	0.268 ± 0.006	0.386 ± 0.008	0.436 ± 0.001	0.270 ± 0.001
GCN	0.152 ± 0.001	0.212 ± 0.002	0.217 ± 0.007	0.278 ± 0.009	0.400 ± 0.003	0.424 ± 0.003	0.281 ± 0.002
GCN+	0.155 ± 0.002	0.193 ± 0.002	0.291 ± 0.035	0.379 ± 0.010	0.540 ± 0.004	0.558 ± 0.002	0.353 ± 0.005
GAT	0.140 ± 0.003	0.188 ± 0.007	0.246 ± 0.027	0.442 ± 0.016	0.610 ± 0.002	0.631 ± 0.008	0.376 ± 0.007
GraphGPS	0.163 ± 0.004	0.325 ± 0.012	0.416 ± 0.017	0.568 ± 0.017	0.634 ± 0.013	0.606 ± 0.017	0.452 ± 0.005

Table 17. KTA values for each model and α on OGBG-MOLHIV. KTA is evaluated on the test set. Models with higher KTA values have higher ROC-AUC values.

Model	KTA	ROC-AUC
MLP	0.014 ± 0.002	0.684 ± 0.022
MLP w/ LapPE	0.028 ± 0.002	0.733 ± 0.012
GCN	0.013 ± 0.002	0.724 ± 0.032
GCN$^+$	0.039 ± 0.001	0.742 ± 0.013
GAT	0.048 ± 0.006	0.758 ± 0.014
GraphGPS	0.057 ± 0.007	0.777 ± 0.017

Table 18. KTA values for each model and α on Peptides-struct with uniform model width of 256. Feature learning is promoted by (1) global attention, and (2) positional encodings.

Model	Peptides-struct (uniform model width)		
	$\alpha = 0.1$	$\alpha = 1$	$\alpha = 10$
MLP	0.333 ± 0.008	$\underline{0.339 \pm 0.001}$	0.336 ± 0.003
MLP w/ LapPE	0.599 ± 0.003	0.598 ± 0.001	$\underline{0.602 \pm 0.002}$
GCN	0.427 ± 0.013	$\underline{0.433 \pm 0.005}$	0.413 ± 0.006
GCN$^+$	0.413 ± 0.005	0.430 ± 0.017	$\underline{0.464 \pm 0.022}$
GAT	0.439 ± 0.026	$\underline{0.444 \pm 0.004}$	0.407 ± 0.005
GraphGPS	0.573 ± 0.030	$\underline{0.583 \pm 0.001}$	0.533 ± 0.033