Rethinking Smoothness in Node Features Learned by Graph Convolutional Networks

Anonymous authors
Paper under double-blind review

Abstract

The pioneering works of Oono and Suzuki (ICLR 2020) and Cai and Wang (arXiv:2006.13318) initiated the analysis of feature smoothness in graph convolutional networks (GCNs), uncovering a strong empirical connection between node classification accuracy and the ratio of smooth to non-smooth feature components. However, it remains unclear how to effectively control this ratio in learned node features to enhance classification performance. Furthermore, deep GCNs with ReLU or leaky ReLU activations tend to suppress non-smooth feature components. In this paper, we introduce a novel strategy to enable GCNs to learn node features with **controllable smoothness**, thereby improving node classification accuracy. Our method comprises three core components: (1) deriving a geometric relationship between the inputs and outputs of ReLU and leaky ReLU activations; (2) augmenting the standard message-passing mechanism in graph convolutional layers with a learnable term for efficient smoothness modulation; and (3) theoretically analyzing the attainable smooth-to-non-smooth ratios under the proposed augmented propagation. Extensive experiments demonstrate that our approach substantially enhances node classification performance across GCNs and related architectures.

1 Introduction

Let $G = (V, E)$ be an undirected graph with node set $V = \{v_i\}_{i=1}^n$ and edge set E. Let $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ be the adjacency matrix of the graph G, where $A_{ij} = \mathbf{1}_{(i,j)\in E}$ with $\mathbf{1}$ being the indicator function. Moreover, let $\boldsymbol{G}$ be the augmented normalized adjacency matrix:

$$\boldsymbol{G} \coloneqq (\boldsymbol{D}+\boldsymbol{I})^{-\frac{1}{2}}(\boldsymbol{I}+\boldsymbol{A})(\boldsymbol{D}+\boldsymbol{I})^{-\frac{1}{2}} = \tilde{\boldsymbol{D}}^{-\frac{1}{2}}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{-\frac{1}{2}}, \tag{1}$$

where $\boldsymbol{I} \in \mathbb{R}^{n\times n}$ is the identity matrix, $\boldsymbol{D} \in \mathbb{R}^{n\times n}$ is a diagonal matrix with $D_{ii} = \sum_{j=1}^n A_{ij}$. Starting from the initial node features $\boldsymbol{H}^0 \coloneqq [(\boldsymbol{h}_1^0)^\top, \ldots, (\boldsymbol{h}_n^0)^\top]^\top \in \mathbb{R}^{d\times n}$, the graph convolutional network (GCN) (Kipf & Welling, 2017) updates node features via a cascade of the following graph convolutional layer (GCL):

$$\boldsymbol{H}^l = \sigma(\boldsymbol{W}^l\boldsymbol{H}^{l-1}\boldsymbol{G}), \tag{2}$$

where σ is the activation function (typicaly ReLU), and $\boldsymbol{W}^l \in \mathbb{R}^{d\times d}$ is a learnable weight matrix. GCL smooths node features by aggregating information from neighbors, a property shown to benefit node classification (cf. (Li et al., 2018; Wu et al., 2019; Chen et al., 2020a)), echoing the idea of energy-based semi-supervised learning (cf. (Zhu et al., 2003; Zhou et al., 2003)). However, accurate node classification requires a balance between smooth and non-smooth feature components (Oono & Suzuki, 2020). Beyond GCNs, many other graph neural networks (GNNs) have been proposed using diverse mechanisms, e.g., spectral methods (Defferrard et al., 2016), spatial methods (Gilmer et al., 2017; Veličković et al., 2018), sampling methods (Hamilton et al., 2017a; Ying et al., 2018), and the attention mechanism (Veličković et al., 2018). Many other GNNs can be found in e.g., (Hamilton et al., 2017b; Battaglia et al., 2018; Wu et al., 2020; Zhou et al., 2020; Hamilton, 2020).

It has been observed that deep GCNs perform significantly worse than shallow models (Chen et al., 2020a). The node features of deep GCNs tend to be identical over each connected component of the graph; this

phenomenon is referred to as **over-smoothing** (Li et al., 2018; Nt & Maehara, 2019; Oono & Suzuki, 2020; Cai & Wang, 2020; Chen et al., 2020a; Wu et al., 2023b), which not only occurs for GCN but also for many other GNNs (Hamilton et al., 2017a; Gilmer et al., 2017; Wu et al., 2023a). Intuitively, each GCL smooths neighboring node features; stacking many smoothing layers will inevitably homogenize node features. Algorithms have been developed to alleviate over-smoothing, including decoupling prediction and message passing (Gasteiger et al., 2019), skip connection and batch normalization (Kawamoto et al., 2018; Chen et al., 2019; 2020b; Scholkemper et al., 2024), graph sparsification (Rong et al., 2020), jumping knowledge (Xu et al., 2018), PairNorm (Zhao & Akoglu, 2020), implicit layers (Chamberlain et al., 2021; Thorpe et al., 2022; Gu et al., 2020; Baker et al., 2023; 2024), and controlling the Dirichlet energy of node features (Zhou et al., 2021).

From a theoretical standpoint, Oono & Suzuki (2020) show that, as the depth of a GCN with ReLU activation increases, node features collapse to the eigenspace $\mathcal{M}$ associated with the largest eigenvalue 1 of the matrix $\boldsymbol{G}$ in equation 1. They also empirically demonstrate that node classification accuracy is intricately linked to the ratio between smooth and non-smooth components of node features, i.e., their projections onto $\mathcal{M}$ and its orthogonal complement $\mathcal{M}^{\perp}$, respectively. Their findings suggest that both components are essential for accurate node classification, though the optimal ratio remains unknown and task-dependent. Complementing this, Cai & Wang (2020) prove that the Dirichlet energy—a measure of feature smoothness—also vanishes as GCN depth increases.

A key insight from the analyses in (Oono & Suzuki, 2020; Cai & Wang, 2020) is that both ReLU and leaky ReLU activations reduce the distance of node features to the eigenspace $\mathcal{M}$ and lower their Dirichlet energy. However, Cai & Wang (2020) point out that this notion of over-smoothing—based solely on proximity to $\mathcal{M}$ or low Dirichlet energy—is inappropriate. Cai & Wang (2020) propose that smoothness should instead be assessed via a **normalized smoothness**, such as Dirichlet energy normalized by the feature magnitude. This perspective aligns with the ratio of smooth to non-smooth components studied by Oono & Suzuki (2020), which is closely related to normalized smoothness. Despite its importance, a theoretical understanding of normalized smoothness in GCNs with ReLU or leaky ReLU activation remains an open problem (Cai & Wang, 2020). Moreover, it is intriguing to ask whether such analysis could inspire new efficient algorithms to improve GCN performance.

1.1 Our Contribution

We aim to (1) understand how GCL smooths node features and (2) develop an efficient algorithm to let GCN and related models learn features with task-adaptive normalized smoothness (Cai & Wang, 2020) to improve node classification. Our main contributions are:

- We prove that there is a high-dimensional sphere underlying the input and output vectors of ReLU or leaky ReLU activation. This geometric characterization not only implies theories in (Oono & Suzuki, 2020; Cai & Wang, 2020), but also informs that adjusting the projection of input vectors onto the eigenspace $\mathcal{M}$ can alter the smoothness of the output vectors (cf. Section 3).
- We show that both ReLU and leaky ReLU activation reduce the distance of features to the eigenspace $\mathcal{M}$, effectively smoothing node features without accounting for magnitude. However, when normalized by the feature magnitude, these activations can increase, decrease, or preserve smoothness—highlighting the nuanced behavior of normalized smoothness (cf. Sections 3 and 4).
- Inspired by our established geometric relationship between the input and output of ReLU or leaky ReLU activation, we study how modifying the projection of input features onto the eigenspace $\mathcal{M}$ affects both normalized and unnormalized smoothness of the output vectors. We show that the output's distance to $\mathcal{M}$ never exceeds that of the input, regardless of how the projection is adjusted. In contrast, normalized smoothness can be effectively tuned to any desired level by controlling this projection (cf. Section 4).
- Based on our theoretical insights, we introduce a new smoothness control term (SCT) that enables GCNs and related models to learn node features with task-adaptive normalized smoothness, improving node classification. We validate SCT's effectiveness across both homophilic and heterophilic graphs using several representative GCN-style models (cf. Sections 5 and 6).

To the best of our knowledge, this is the first comprehensive study of how ReLU and leaky ReLU activations affect the smoothness of node features—both in normalized and unnormalized forms.

1.2 Additional Related Works

Another line of related work focuses on controlling node feature smoothness to improve GCN performance. For example, Zhao & Akoglu (2020) introduce a normalization layer to prevent feature homogenization, while Zhou et al. (2021) constrain the Dirichlet energy without accounting for nonlinear activations. Although prior efforts have addressed over-smoothing and smoothness control, without considering activations, there remains a lack of theoretical analysis on how activation functions—particularly when accounting for feature magnitude—affect the smoothness of node features. A recent work (Di Giovanni et al., 2022) studies the normalized Dirichlet energy in GCNs with residual connections, demonstrating how the spectrum of the weight matrix affects smoothness. However, it does not offer an efficient method for controlling normalized smoothness.

1.3 Notation and Organization

Notation. We denote the ℓ_2-norm of a vector $\boldsymbol{u}$ as $\|\boldsymbol{u}\|$. For vectors $\boldsymbol{u}$ and $\boldsymbol{v}$, we denote $\langle \boldsymbol{u}, \boldsymbol{v} \rangle$, $\boldsymbol{u} \odot \boldsymbol{v}$, and $\boldsymbol{u} \otimes \boldsymbol{v}$ as their inner, Hadamard, and Kronecker product, respectively. For a matrix $\boldsymbol{A}$, we denote its $(i,j)^{th}$ entry, transpose, and inverse as A_{ij}, $\boldsymbol{A}^\top$, and $\boldsymbol{A}^{-1}$, respectively. We denote the trace of matrix $\boldsymbol{A}$ as $\text{Trace}(\boldsymbol{A}) = \sum_{i=1}^n A_{ii}$. For two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, we denote their Frobenius inner product as $\langle \boldsymbol{A}, \boldsymbol{B} \rangle_F := \text{Trace}(\boldsymbol{A}\boldsymbol{B}^\top)$ and the Frobenius norm of matrix $\boldsymbol{A}$ as $\|\boldsymbol{A}\|_F := \sqrt{\langle \boldsymbol{A}, \boldsymbol{A} \rangle}$.

Organization. We provide preliminaries and a review of some related results in Section 2. In Section 3, we establish a geometric characterization of how ReLU and leaky ReLU activations affect the smoothness of their input vectors. We study the smoothness of each dimension of node features and take their magnitude into account in Section 4. Our proposed SCT is presented in Section 5. We verify the efficacy of the proposed SCT for improving node classification in Section 6. Technical proofs and additional experimental details are provided in the appendix.

2 Preliminaries and Existing Results

By spectral graph theory (Chung, 1997), the eigenvalues of the normalized adjacency matrix $\boldsymbol{G}$ in equation 1 can be ordered as $1 = \lambda_1 = \ldots = \lambda_m > \lambda_{m+1} \geq \ldots \geq \lambda_n > -1$, where m is the number of connected components in the graph G. Accordingly, the vertex set $V = \{v_k\}_{k=1}^n$ can be partitioned into m connected components $V_1, \ldots, V_m$. Let $\boldsymbol{u}_i = (\mathbf{1}_{\{v_k \in V_i\}})_{1 \leq k \leq n}$ denote the indicator vector for component V_i, where the k-th entry is 1 if $v_k \in V_i$, and 0 otherwise. Let $\boldsymbol{e}_i$ be the eigenvector corresponding to λ_i; then $\{\boldsymbol{e}_i\}_{i=1}^n$ forms an orthonormal basis of $\mathbb{R}^n$. The eigenspace $\mathcal{M}$ is spanned by $\{\boldsymbol{e}_i\}_{i=1}^m$, and its orthogonal complement $\mathcal{M}^\perp$ by $\{\boldsymbol{e}_i\}_{i=m+1}^n$. Oono & Suzuki (2020) relate the indicator vectors $\boldsymbol{u}_i$ to the eigenspace $\mathcal{M}$, specifically showing that:

Proposition 2.1 (Oono & Suzuki (2020))**.** *All eigenvalues of matrix $\boldsymbol{G}$ lie in the interval $(-1, 1]$. Moreover, the nonnegative vectors $\{\tilde{\boldsymbol{D}}^{\frac{1}{2}}\boldsymbol{u}_i / \|\tilde{\boldsymbol{D}}^{\frac{1}{2}}\boldsymbol{u}_i\|\}_{1 \leq i \leq m}$ form an orthonormal basis of $\mathcal{M}$.*

For any matrix $\boldsymbol{H} := [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_n] \in \mathbb{R}^{d \times n}$, we have the decomposition

$$\boldsymbol{H} = \boldsymbol{H}_{\mathcal{M}} + \boldsymbol{H}_{\mathcal{M}^\perp},$$

where $\boldsymbol{H}_{\mathcal{M}} = \sum_{i=1}^m \boldsymbol{H}\boldsymbol{e}_i\boldsymbol{e}_i^\top$ and $\boldsymbol{H}_{\mathcal{M}^\perp} = \sum_{i=m+1}^n \boldsymbol{H}\boldsymbol{e}_i\boldsymbol{e}_i^\top$, satisfying

$$\langle \boldsymbol{H}_{\mathcal{M}}, \boldsymbol{H}_{\mathcal{M}^\perp} \rangle_F = \text{Trace}\left(\sum_{i=1}^m \boldsymbol{H}\boldsymbol{e}_i\boldsymbol{e}_i^\top \Big(\sum_{j=m+1}^n \boldsymbol{H}\boldsymbol{e}_j\boldsymbol{e}_j^\top \Big)^\top \right) = 0.$$

This implies that $\|\boldsymbol{H}\|_F^2 = \|\boldsymbol{H}_{\mathcal{M}}\|_F^2 + \|\boldsymbol{H}_{\mathcal{M}^\perp}\|_F^2$.

2.1 Existing Smoothness Notions of Node Features

Distance to the eigenspace $\mathcal{M}$. Oono & Suzuki (2020) study the smoothness of $\boldsymbol{H}$ using their distance to the eigenspace $\mathcal{M}$ as an unnormalized smoothness notion, i.e.,

Definition 2.2 (Oono & Suzuki (2020))**.** Let $\mathbb{R}^d \otimes \mathcal{M}$ be the subspace of $\mathbb{R}^{d\times n}$ consisting of the sum $\sum_{i=1}^m \boldsymbol{w}_i \otimes \boldsymbol{e}_i$, where $\boldsymbol{w}_i \in \mathbb{R}^d$ and $\{\boldsymbol{e}_i\}_{i=1}^m$ is an orthonormal basis of the eigenspace $\mathcal{M}$. Then we define $\|\boldsymbol{H}\|_{\mathcal{M}^\perp}$—the distance of $\boldsymbol{H}$ to $\mathcal{M}$—as follows:

$$\|\boldsymbol{H}\|_{\mathcal{M}^\perp} \coloneqq \inf_{\boldsymbol{Y}\in\mathbb{R}^d\otimes\mathcal{M}} \|\boldsymbol{H}-\boldsymbol{Y}\|_F = \big\|\boldsymbol{H} - \sum_{i=1}^m \boldsymbol{H}\boldsymbol{e}_i\boldsymbol{e}_i^\top\big\|_F.$$

Since $\boldsymbol{H} = \boldsymbol{H}_{\mathcal{M}} + \boldsymbol{H}_{\mathcal{M}^\perp}$, then we have

$$\|\boldsymbol{H}\|_{\mathcal{M}^\perp} = \inf_{\boldsymbol{Y}\in\mathbb{R}^d\otimes\mathcal{M}} \|\boldsymbol{H}-\boldsymbol{Y}\|_F = \|\boldsymbol{H}-\boldsymbol{H}_{\mathcal{M}}\|_F = \|\boldsymbol{H}_{\mathcal{M}^\perp}\|_F. \tag{3}$$

Dirichlet energy. Cai & Wang (2020) study the unnormalized smoothness of node features using the following Dirichlet energy:

Definition 2.3 (Cai & Wang (2020))**.** Let $\tilde{\Delta} = \boldsymbol{I} - \boldsymbol{G}$ be the (augmented) normalized Laplacian, then the Dirichlet energy $\|\boldsymbol{H}\|_E$ of node features $\boldsymbol{H}$ is defined by $\|\boldsymbol{H}\|_E^2 \coloneqq \mathrm{Trace}(\boldsymbol{H}\tilde{\Delta}\boldsymbol{H}^\top)$.

Normalized Dirichlet energy. Cai & Wang (2020) point out that the smoothness of $\boldsymbol{H}$ should be measured using the normalized Dirichlet energy, defined as $\mathrm{Trace}(\boldsymbol{H}\tilde{\Delta}\boldsymbol{H}^\top)/\|\boldsymbol{H}\|_F^2$. Normalizing the measurement can effectively mitigate biases result from these different scales.

2.2 Two Existing Theories of Over-smoothing

Let $\{\lambda_i\}_{i=1}^n$ be the eigenvalues of $\boldsymbol{G}$, and define $\lambda = \max\{|\lambda_i||\lambda_i < 1\}$ as the second-largest magnitude. Let s_l denote the largest singular value of the weight matrix $\boldsymbol{W}^l$. Oono & Suzuki (2020) show that under the ReLU activation, GCL satisfies $\|\boldsymbol{H}^l\|_{\mathcal{M}^\perp} \le s_l\lambda\|\boldsymbol{H}^{l-1}\|_{\mathcal{M}^\perp}$, implying $\|\boldsymbol{H}^l\|_{\mathcal{M}^\perp} \to 0$ as $l\to\infty$ if $s_l\lambda < 1$. This convergence to the eigenspace $\mathcal{M}$ leads to over-smoothing. A key step in their analysis is the inequality $\|\sigma(\boldsymbol{Z})\|_{\mathcal{M}^\perp} \le \|\boldsymbol{Z}\|_{\mathcal{M}^\perp}$ for any matrix $\boldsymbol{Z}$ when σ is the ReLU activation, showing that the ReLU activation reduces the distance to $\mathcal{M}$. However, extending this result to other activations, including leaky ReLU, remains challenging (Oono & Suzuki, 2020).

Instead of analyzing $\|\boldsymbol{H}\|_{\mathcal{M}^\perp}$, Cai & Wang (2020) prove that under the GCL with ReLU or leaky ReLU activation, the normalized Dirichlet energy satisfies $\|\boldsymbol{H}^l\|_E \le s_l\lambda\|\boldsymbol{H}^{l-1}\|_E$, implying $\|\boldsymbol{H}^l\|_E \to 0$ as $l\to\infty$ and thus over-smoothing. Their proof applies to both activation functions by establishing the inequality $\|\sigma(\boldsymbol{Z})\|_E \le \|\boldsymbol{Z}\|_E$ for any matrix $\boldsymbol{Z}$.

3 Effects of Activation Functions: A Geometric Characterization

In this section, we present the geometric relationship between the input and output vectors of ReLU or leaky ReLU activation. For all subsequent analyses, we adopt $\|\boldsymbol{H}\|_{\mathcal{M}^\perp}$ as our unnormalized smoothness measure, noting its equivalence to $\|\boldsymbol{H}\|_E$ as a seminorm. Specifically, we have

Proposition 3.1. *$\|\boldsymbol{H}\|_{\mathcal{M}^\perp}$ and $\|\boldsymbol{H}\|_E$ are two equivalent seminorms, i.e., there exist two constants $\alpha,\beta>0$ s.t. $\alpha\|\boldsymbol{H}\|_{\mathcal{M}^\perp} \le \|\boldsymbol{H}\|_E \le \beta\|\boldsymbol{H}\|_{\mathcal{M}^\perp}$, for any $\boldsymbol{H}\in\mathbb{R}^{d\times n}$.*

ReLU. Let $\sigma(x) = \max\{x, 0\}$ be the ReLU activation. Our first main result is that there is a high-dimensional sphere underlying the input and output vectors of the ReLU activation. More precisely, we have the following result:

Proposition 3.2 (ReLU)**.** *For any $\boldsymbol{Z} = \boldsymbol{Z}_{\mathcal{M}} + \boldsymbol{Z}_{\mathcal{M}^\perp} \in \mathbb{R}^{d\times n}$, let $\boldsymbol{H} = \sigma(\boldsymbol{Z}) = \boldsymbol{H}_{\mathcal{M}} + \boldsymbol{H}_{\mathcal{M}^\perp}$. Then $\boldsymbol{H}_{\mathcal{M}^\perp}$ lies on the high-dimensional sphere centered at $\boldsymbol{Z}_{\mathcal{M}^\perp}/2$ with radius*

$$r \coloneqq \big(\|\boldsymbol{Z}_{\mathcal{M}^\perp}/2\|_F^2 - \langle \boldsymbol{H}_{\mathcal{M}}, \boldsymbol{H}_{\mathcal{M}} - \boldsymbol{Z}_{\mathcal{M}}\rangle_F\big)^{1/2}.$$

In particular, $\boldsymbol{H}_{\mathcal{M}^\perp}$ lies inside the ball centered at $\boldsymbol{Z}_{\mathcal{M}^\perp}/2$ with radius $\|\boldsymbol{Z}_{\mathcal{M}^\perp}/2\|_F$ and hence

$$\|\boldsymbol{H}\|_{\mathcal{M}^\perp} \leq \|\boldsymbol{Z}\|_{\mathcal{M}^\perp}.$$

Leaky ReLU. For leaky ReLU activation $\sigma_a(x) = \max\{x, ax\}$ for $0 < a < 1$, we have

Proposition 3.3 (Leaky ReLU)**.** *For any $\boldsymbol{Z} = \boldsymbol{Z}_{\mathcal{M}} + \boldsymbol{Z}_{\mathcal{M}^\perp} \in \mathbb{R}^{d\times n}$, let $\boldsymbol{H} = \sigma_a(\boldsymbol{Z}) = \boldsymbol{H}_{\mathcal{M}} + \boldsymbol{H}_{\mathcal{M}^\perp}$. Then $\boldsymbol{H}_{\mathcal{M}^\perp}$ lies on the sphere centered at $(1+a)\boldsymbol{Z}_{\mathcal{M}^\perp}/2$ with radius*

$$r_a \coloneqq \left(\|(1-a)\boldsymbol{Z}_{\mathcal{M}^\perp}/2\|_F^2 - \langle \boldsymbol{H}_{\mathcal{M}} - \boldsymbol{Z}_{\mathcal{M}}, \boldsymbol{H}_{\mathcal{M}} - a\boldsymbol{Z}_{\mathcal{M}}\rangle_F\right)^{1/2}.$$

In particular, $\boldsymbol{H}_{\mathcal{M}^\perp}$ lies inside the ball centered at $(1+a)\boldsymbol{Z}_{\mathcal{M}^\perp}/2$ with radius $\|(1-a)\boldsymbol{Z}_{\mathcal{M}^\perp}/2\|_F$ and hence we see that

$$a\|\boldsymbol{Z}\|_{\mathcal{M}^\perp} \leq \|\boldsymbol{H}\|_{\mathcal{M}^\perp} \leq \|\boldsymbol{Z}\|_{\mathcal{M}^\perp}.$$

3.1 Implications of the Above Geometric Characterizations

Propositions 3.2 and 3.3 show that the unnormalized smoothness $\|\boldsymbol{H}_{\mathcal{M}^\perp}\|_F = \|\boldsymbol{H}\|_{\mathcal{M}^\perp}$ depends on the center and radii r or r_a. Given $\boldsymbol{Z}_{\mathcal{M}^\perp}$, the center of the spheres remains unchanged, and their radii r and r_a are only affected by changes in $\boldsymbol{Z}_{\mathcal{M}}$. This motivates our study of **how variations in $\boldsymbol{Z}_{\mathcal{M}}$ influence the unnormalized smoothness of node features**.

Furthermore, Propositions 3.2 and 3.3 imply both ReLU and leaky ReLU activation reduce the distance of node features to $\mathcal{M}$, i.e., $\|\boldsymbol{H}\|_{\mathcal{M}^\perp} \leq \|\boldsymbol{Z}\|_{\mathcal{M}^\perp}$. Moreover, this inequality is independent of $\boldsymbol{Z}_{\mathcal{M}}$; consider two node features $\boldsymbol{Z}, \boldsymbol{Z}' \in \mathbb{R}^{d\times n}$ s.t. $\boldsymbol{Z}_{\mathcal{M}^\perp} = \boldsymbol{Z}'_{\mathcal{M}^\perp}$ but $\boldsymbol{Z}_{\mathcal{M}} \neq \boldsymbol{Z}'_{\mathcal{M}}$. Let $\boldsymbol{H}$ and $\boldsymbol{H}'$ be the output of $\boldsymbol{Z}$ and $\boldsymbol{Z}'$ via ReLU or leaky ReLU activation, respectively. Then we have $\|\boldsymbol{H}\|_{\mathcal{M}^\perp} \leq \|\boldsymbol{Z}\|_{\mathcal{M}^\perp}$ and $\|\boldsymbol{H}'\|_{\mathcal{M}^\perp} \leq \|\boldsymbol{Z}'\|_{\mathcal{M}^\perp}$. Since $\boldsymbol{Z}_{\mathcal{M}^\perp} = \boldsymbol{Z}'_{\mathcal{M}^\perp}$, we deduce that $\|\boldsymbol{H}'\|_{\mathcal{M}^\perp} \leq \|\boldsymbol{Z}\|_{\mathcal{M}^\perp}$. In other words, when $\boldsymbol{Z}_{\mathcal{M}^\perp} = \boldsymbol{Z}'_{\mathcal{M}^\perp}$, changing $\boldsymbol{Z}_{\mathcal{M}}$ to $\boldsymbol{Z}'_{\mathcal{M}}$ can change the unnormalized smoothness of the output features but cannot change the fact that both ReLU and leaky ReLU activations smooth node features; we demonstrate this result in Fig. 1 a). Notice that without considering the nonlinear activation function, changing $\boldsymbol{Z}_{\mathcal{M}}$ does not affect the unnormalized smoothness of node features measured by $\|\boldsymbol{H}\|_{\mathcal{M}^\perp}$.

In contrast, **normalized smoothness can decrease when $\boldsymbol{Z}_{\mathcal{M}}$ is adjusted, potentially leading to a less smooth output vector**; see the next section for details.

4 Impact of Adjusting $Z_{\mathcal{M}}$ on Output Smoothness

Throughout this section, let $\boldsymbol{Z}$ and $\boldsymbol{H}$ denote the input and output of ReLU or leaky ReLU activation. Traditional smoothness measures—based on distance to the eigenspace $\mathcal{M}$ and Dirichlet energy—do not account for the relative magnitudes of feature dimensions. As noted by Cai & Wang (2020), analyzing the normalized smoothness $\|\boldsymbol{Z}\|_E/\|\boldsymbol{Z}\|_F$ remains an open challenging. These measures aggregate smoothness across all dimensions, so when certain dimensions dominate in magnitude, they disproportionately influence the overall smoothness.

Building on the discussion in Section 3.1, we examine how adjusting $\boldsymbol{Z}_{\mathcal{M}}$ differently affects normalized and unnormalized smoothness. For simplicity, we assume the graph is connected ($m = 1$); the results extend naturally to graphs with multiple components. Given the equivalence between the seminorms $\|\cdot\|_{\mathcal{M}}$ and $\|\cdot\|_E$, we introduce the following definition of dimension-wise normalized smoothness for node features:

Definition 4.1. Let $\boldsymbol{Z} \in \mathbb{R}^{d\times n}$ be the feature matrix over n graph nodes, where $\boldsymbol{z}^{(i)} \in \mathbb{R}^n$ denotes the i^{th} row of $\boldsymbol{Z}$, i.e., the i^{th} feature dimension across all nodes. We define the normalized smoothness of $\boldsymbol{z}^{(i)}$ as:

$$s(\boldsymbol{z}^{(i)}) \coloneqq \|\boldsymbol{z}^{(i)}_{\mathcal{M}}\|/\|\boldsymbol{z}^{(i)}\|,$$

where we set $s(\boldsymbol{z}^{(i)}) = 1$ when $\boldsymbol{z}^{(i)} = \boldsymbol{0}$.

Remark 4.2. Notice that the normalized smoothness $s(\boldsymbol{z}^{(i)}) = \|\boldsymbol{z}^{(i)}_{\mathcal{M}}\|/\|\boldsymbol{z}^{(i)}\|$ is related to the ratio between the smooth and non-smooth components of node features $\|\boldsymbol{z}^{(i)}_{\mathcal{M}}\|/\|\boldsymbol{z}^{(i)}_{\mathcal{M}^\perp}\|$.

The graph is connected implies that

$$\boldsymbol{z}_{\mathcal{M}}^{(i)} = \langle \boldsymbol{z}^{(i)}, \boldsymbol{e}_1 \rangle \boldsymbol{e}_1 \implies \|\boldsymbol{z}_{\mathcal{M}}^{(i)}\| = |\langle \boldsymbol{z}^{(i)}, \boldsymbol{e}_1 \rangle|.$$

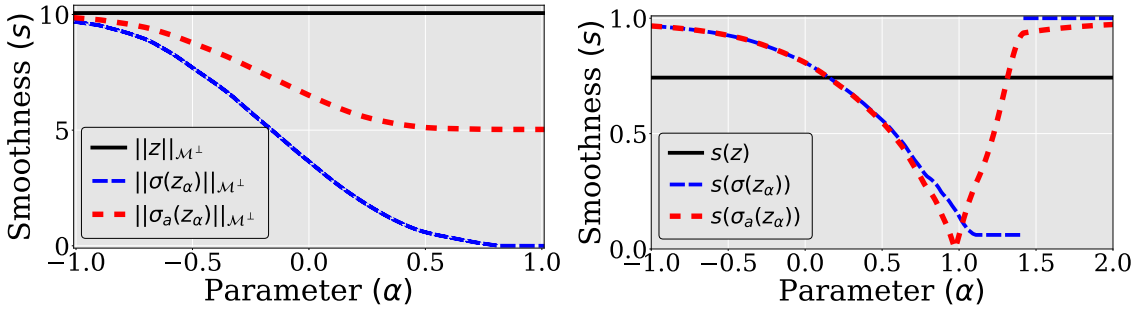
For clarity, we drop the index and write $\boldsymbol{z}$ for $\boldsymbol{z}^{(i)}$ and $\boldsymbol{e}$ for $\boldsymbol{e}_1$, the unique eigenvector of $\boldsymbol{G}$ associated with eigenvalue 1. Moreover, we have

$$s(\boldsymbol{z}) = \frac{\|\boldsymbol{z}_{\mathcal{M}}\|}{\|\boldsymbol{z}\|} = \frac{|\langle \boldsymbol{z}, \boldsymbol{e} \rangle|}{\|\boldsymbol{z}\|} = \frac{|\langle \boldsymbol{z}, \boldsymbol{e} \rangle|}{\|\boldsymbol{z}\| \cdot \|\boldsymbol{e}\|} \implies 0 \leq s(\boldsymbol{z}) \leq 1. \tag{4}$$

It is evident that the larger $s(\boldsymbol{z})$ is, the smoother the node feature $\boldsymbol{z}$ is. In fact, we have

$$s(\boldsymbol{z})^2 + \left(\frac{\|\boldsymbol{z}\|_{\mathcal{M}^\perp}}{\|\boldsymbol{z}\|}\right)^2 = \frac{\|\boldsymbol{z}_{\mathcal{M}}\|^2}{\|\boldsymbol{z}\|^2} + \frac{\|\boldsymbol{z}_{\mathcal{M}^\perp}\|^2}{\|\boldsymbol{z}\|^2} = 1,$$

where $\|\boldsymbol{z}\|_{\mathcal{M}^\perp}/\|\boldsymbol{z}\|$ decreases as $s(\boldsymbol{z})$ increases.



a) Smoothness b) Normalized smoothness

Figure 1: Contrasting the effects of varying the parameter α on the smoothness and normalized smoothness of the output features $\sigma(\boldsymbol{z}_\alpha)$ and $\sigma_a(\boldsymbol{z}_\alpha)$. The discontinuity in $s(\sigma(\boldsymbol{z}_\alpha))$ shown in panel b) arises from the definition of normalized smoothness. Note that $s(\boldsymbol{z}) = 1$ when $\boldsymbol{z} = \boldsymbol{0}$, and $\sigma(\boldsymbol{z}_\alpha)$ can become $\boldsymbol{0}$ for sufficiently large α.

To discuss how the smoothness $s(\boldsymbol{h}) = s(\sigma(\boldsymbol{z}))$ or $s(\sigma_a(\boldsymbol{z}))$ can be adjusted by changing $\boldsymbol{z}_{\mathcal{M}}$, we define

$$\boldsymbol{z}(\alpha) := \boldsymbol{z} - \alpha \boldsymbol{e}.$$

It is clear that $\boldsymbol{z}(\alpha)_{\mathcal{M}^\perp} = \boldsymbol{z}_{\mathcal{M}^\perp}$ and $\boldsymbol{z}(\alpha)_{\mathcal{M}} = \boldsymbol{z}_{\mathcal{M}} - \alpha \boldsymbol{e}$. We see that α only alters $\boldsymbol{z}_{\mathcal{M}}$ while preserves $\boldsymbol{z}_{\mathcal{M}^\perp}$. Moreover, we have

$$s(\boldsymbol{z}(\alpha)) = \sqrt{1 - \frac{\|\boldsymbol{z}(\alpha)_{\mathcal{M}^\perp}\|^2}{\|\boldsymbol{z}(\alpha)\|^2}} = \sqrt{1 - \frac{\|\boldsymbol{z}_{\mathcal{M}^\perp}\|^2}{\|\boldsymbol{z}(\alpha)\|^2}}.$$

It follows that $s(\boldsymbol{z}(\alpha)) = 1$ if and only if $\boldsymbol{z}_{\mathcal{M}^\perp} = \boldsymbol{0}$, showing that when $\boldsymbol{z}_{\mathcal{M}^\perp} = \boldsymbol{0}$, the vector $\boldsymbol{z}$ is the smoothest one.

4.1 The Disparate Effects of α on $\|\cdot\|_{\mathcal{M}^\perp}$ and $s(\cdot)$: Numerical Studies

To explore how the smoothness measures vary with α, we perform some empirical studies in this subsection.

Let $\boldsymbol{z}_\alpha := \boldsymbol{z}(\alpha) = \boldsymbol{z} - \alpha \boldsymbol{e}$, where $\boldsymbol{z} \in \mathbb{R}^{100}$ is a randomly initialized node feature vector (uniformly sampled from $[-1.5, 1.5]$), and $\boldsymbol{e}$ is computed as $\boldsymbol{e} = \tilde{\boldsymbol{D}}^{1/2}\boldsymbol{u}/\|\tilde{\boldsymbol{D}}^{1/2}\boldsymbol{u}\|$ per Proposition 2.1, with $\boldsymbol{u}$ being the all-ones vector and $\tilde{\boldsymbol{D}}$ the augmented degree matrix. We use a connected synthetic graph with 100 nodes, each assigned a random degree between 2 and 10.

We evaluate both unnormalized smoothness $\|\sigma(\boldsymbol{z}_\alpha)\|_{\mathcal{M}^\perp}$, $\|\sigma_a(\boldsymbol{z}_\alpha)\|_{\mathcal{M}^\perp}$ and normalized smoothness $s(\sigma(\boldsymbol{z}_\alpha))$, $s(\sigma_a(\boldsymbol{z}_\alpha))$ across $\alpha \in [-1.5, 1.5]$. As shown in Fig. 1 a), the output smoothness is consistently no greater

than the input smoothness $\|\boldsymbol{z}\|_{\mathcal{M}^\perp}$, confirming the result from Section 3.1: modifying $\boldsymbol{z}_{\mathcal{M}}$ does not affect this inequality. However, an important observation is that **adjusting the eigenspace projection can still alter the output smoothness**, even though the input smoothness remains unchanged.

In contrast, Fig. 1 b) shows that the normalized smoothness $s(\sigma(\boldsymbol{z}(\alpha)))$ and $s(\sigma_a(\boldsymbol{z}(\alpha)))$ can be reduced below $s(\boldsymbol{z})$ by adjusting α. This highlights how modifying the projection onto $\mathcal{M}$ can influence normalized smoothness, even when unnormalized smoothness remains bounded.

4.2 The Smooth Effects of ReLU and Leaky ReLU: Theoretical Analysis

In this subsection, we provide theoretical insights into the empirical findings shown in Fig. 1, focusing on how the smoothness of $\sigma(\boldsymbol{z}(\alpha))$ and $\sigma_a(\boldsymbol{z}(\alpha))$ varies with α. If $\boldsymbol{z}_{\mathcal{M}^\perp} = \mathbf{0}$, Propositions 3.2 and 3.3 imply that $\|\sigma(\boldsymbol{z}(\alpha))\|_{\mathcal{M}^\perp}$ and $\|\sigma_a(\boldsymbol{z}(\alpha))\|_{\mathcal{M}^\perp}$ vanish, leading to $s(\sigma(\boldsymbol{z}(\alpha))) = 1$ for all α. Therefore, we assume $\boldsymbol{z}_{\mathcal{M}^\perp} \neq \mathbf{0}$ in the following analysis.

Proposition 4.3 (ReLU)**.** *Suppose $\boldsymbol{z}_{\mathcal{M}^\perp} \neq \mathbf{0}$. Let $\boldsymbol{h}(\alpha) = \sigma(\boldsymbol{z}(\alpha))$ with σ being the ReLU activation, then we have*

$$\min_\alpha s(\boldsymbol{h}(\alpha)) = \sqrt{\frac{\sum_{x_i = \max \boldsymbol{x}} d_i}{\sum_{j=1}^n d_j}}, \text{ and } \max_\alpha s(\boldsymbol{h}(\alpha)) = 1,$$

where $\boldsymbol{x} \coloneqq \tilde{\boldsymbol{D}}^{-\frac{1}{2}}\boldsymbol{z}$, $\max \boldsymbol{x} = \max_{1\le i\le n} x_i$, and $\tilde{\boldsymbol{D}}$ is the augmented degree matrix, whose diagonal entries are $d_1, d_2, \ldots, d_n$. In particular, the normalized smoothness $s(\boldsymbol{h}(\alpha))$ is monotone increasing as α decreases whenever $\alpha < \|\tilde{\boldsymbol{D}}^{\frac{1}{2}}\boldsymbol{u}_n\| \max \boldsymbol{x}$ and it has range $[\min_\alpha s(\boldsymbol{h}(\alpha)), 1]$.

Proposition 4.4 (Leaky ReLU)**.** *Assume $\boldsymbol{z}_{\mathcal{M}^\perp} \neq \mathbf{0}$ and let $\boldsymbol{h}(\alpha) = \sigma_a(\boldsymbol{z}(\alpha))$, where σ_a is the leaky ReLU activation. Then (1) $\min_\alpha s(\boldsymbol{h}(\alpha)) = 0$, and (2) $\sup_\alpha s(\boldsymbol{h}(\alpha)) = 1$ and $s(\boldsymbol{h}(\alpha))$ has range $[0, 1)$.*

Proposition 4.4 also holds for other variants of ReLU, e.g., ELU[1] and SELU[2].; see Appendix B. We summarize Propositions 3.2, 3.3, 4.3, and 4.4 in the following corollary, which qualitatively explains the empirical results in Fig. 1.

Corollary 4.5. *Suppose $\boldsymbol{z}_{\mathcal{M}^\perp} \neq \mathbf{0}$. Let $\boldsymbol{h}(\alpha) = \sigma(\boldsymbol{z}(\alpha))$ or $\sigma_a(\boldsymbol{z}(\alpha))$ with σ being the ReLU activation and σ_a being the leaky ReLU activation. Then $\|\boldsymbol{z}\|_{\mathcal{M}^\perp} \geq \|\boldsymbol{h}(\alpha)\|_{\mathcal{M}^\perp}, \forall \alpha \in \mathbb{R}$; however, $s(\boldsymbol{h}(\alpha))$ can be smaller than, larger than, or equal to $s(\boldsymbol{z})$ for different α.*

Propositions 4.3, 4.4, and Corollary 4.5, provide theoretical support for the empirical results in Fig. 1. These results show that modifying the projection $\boldsymbol{z}_{\mathcal{M}}$ can influence both the unnormalized and normalized smoothness of the output $\boldsymbol{h} = \sigma(\boldsymbol{z})$ or $\sigma_a(\boldsymbol{z})$. In particular, the normalized smoothness can be tuned to any value within the ranges established in the propositions. This insight opens the door to designing algorithms that control feature smoothness to enhance GCN performance—a direction we explore in the next section.

5 Controlling Smoothness of Node Features

While the ideal smoothness level for node features in a given classification task is unknown, our theory shows that both normalized and unnormalized smoothness can be modulated by adjusting the input's projection onto $\mathcal{M}$. Motivated by this, we propose the following learnable smoothness control term to dynamically regulate the smoothness of node features:

$$\boldsymbol{B}_{\boldsymbol{\alpha}}^l = \sum_{i=1}^m \boldsymbol{\alpha}_i^l \boldsymbol{e}_i^\top, \tag{5}$$

where l is the layer index, $\{\boldsymbol{e}_i\}_{i=1}^m$ is the orthonormal basis of the eigenspace $\mathcal{M}$, and $\boldsymbol{\alpha}^l := \{\boldsymbol{\alpha}_i^l\}_{i=1}^m$ is a collection of learnable vectors with $\boldsymbol{\alpha}_i^l \in \mathbb{R}^d$ being approximated by a multi-layer perceptron (MLP). The

[1]ELU: $f(x) = \max(x, 0) + \min(0, a \cdot (e^x - 1))$ where $a > 0$.
[2]SELU: $f(x) = c(\max(x, 0) + \min(0, a \cdot (e^x - 1)))$ where $a, c > 0$.

detailed configuration of $\boldsymbol{\alpha}_i^l$ will be specified in each experiment later. One can see that $\boldsymbol{B}_{\boldsymbol{\alpha}}^l$ always lies in $\mathbb{R}^d \otimes \mathcal{M}$. We integrate SCT into GCL, resulting in the following update equation:

$$\boldsymbol{H}^l = \sigma(\boldsymbol{W}^l \boldsymbol{H}^{l-1} \boldsymbol{G} + \boldsymbol{B}_{\boldsymbol{\alpha}}^l). \tag{6}$$

We call the corresponding model GCN-SCT.

We now detail the proposed SCT. For each layer l, we introduce a learnable matrix $\boldsymbol{A}^l \in \mathbb{R}^{d \times m}$, where each column $\boldsymbol{\alpha}_i^l$ corresponds to a basis vector in the eigenspace $\mathcal{M}$. Here, d is the feature dimension and m is the dimension of $\mathcal{M}$. We observe that SCT performs best when guided by degree pooling over graph subcomponents. Let $\boldsymbol{Q} \coloneqq [\boldsymbol{e}_1, \ldots, \boldsymbol{e}_m] \in \mathbb{R}^{n \times m}$ be the matrix of orthogonal basis vectors. Pooling is performed via $\boldsymbol{H}^l \boldsymbol{Q}$. For the first architecture, we let

$$\boldsymbol{A}^l = \boldsymbol{W} \odot (\boldsymbol{H}^l \boldsymbol{Q}),$$

where $\boldsymbol{W} \in \mathbb{R}^{d \times m}$ is learnable and performs pooling over $\boldsymbol{H}^l$ using the eigenvectors $\boldsymbol{Q}$. The second architecture uses a residual connection with hyperparameter $\beta_l = \log(\theta/l + 1)$ and learnable matrices $\boldsymbol{W}_0, \boldsymbol{W}_1 \in \mathbb{R}^{d \times d}$ and the softmax function ϕ. Resulting in

$$\boldsymbol{A}^l = \phi(\boldsymbol{H}^l \boldsymbol{Q}) \odot (\beta_l \boldsymbol{W}_0 \boldsymbol{H}^0 \boldsymbol{Q} + (1 - \beta_l) \boldsymbol{W}_1 \boldsymbol{H}^l \boldsymbol{Q}).$$

In Section 6, we use the first architecture for GCN-SCT as GCN uses only $\boldsymbol{H}^l$ information at each layer. We use the second architecture for GCNII-SCT and EGNN-SCT, which use both $\boldsymbol{H}^0$ and $\boldsymbol{H}^l$ information at each layer. This design offers two key advantages: (1) it enables effective control of normalized smoothness, and (2) it is computationally efficient, relying only on eigenvectors associated with eigenvalue 1 of matrix $\boldsymbol{G}$.

Remark 5.1. Computing the basis of eigenspace $\mathcal{M}$ (eigenvectors corresponding to eigenvalue 1 of $\boldsymbol{G}$) does not introduce substantial computational overhead. In particular, the basis of $\mathcal{M}$ is given by the indicator functions of each connected component of the graph, and we can identify connected components for undirected graphs using disjoint set union (DSU) Galil & Italiano (1991) with linear time complexity with respect to the number of nodes.

5.1 Integrating SCT into Other GCN-style Models

In this subsection, we present two other representative applications of SCT. First, we apply SCT to GCNII (Chen et al., 2020b), a state-of-the-art (SOTA) GCN-style model (Chen et al., 2020b; Luan et al., 2022), to demonstrate that SCT can enhance node classification by improving feature smoothness. Second, we apply SCT to EGNN (Zhou et al., 2021), which originally controls feature smoothness via Dirichlet energy bounds under linear activation. We show that our new theoretical insights into activation functions, combined with SCT, can improve EGNN performance when nonlinear activations are considered.

GCNII. Each GCNII layer uses a skip connection to the initial layer $\boldsymbol{H}^0$ and is given as follows:

$$\boldsymbol{H}^l = \sigma\big(((1 - \alpha_l)\boldsymbol{H}^{l-1}\boldsymbol{G} + \alpha_l \boldsymbol{H}^0)((1 - \beta_l)\boldsymbol{I} + \beta_l \boldsymbol{W}^l)\big),$$

where $\alpha_l, \beta_l \in (0, 1)$ are learnable. We integrate SCT $\boldsymbol{B}_{\boldsymbol{\alpha}}^l$ into GCNII, resulting in the following GCNII-SCT layers:

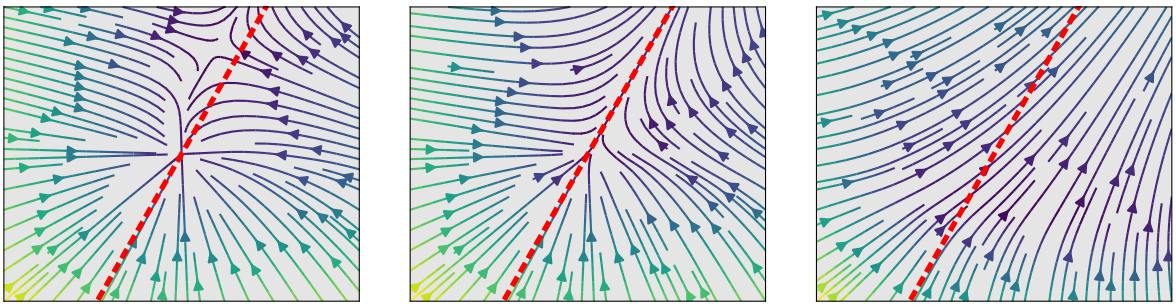
$$\boldsymbol{H}^l = \sigma\big(((1 - \alpha_l)\boldsymbol{H}^{l-1}\boldsymbol{G} + \alpha_l \boldsymbol{H}^0)((1 - \beta_l)\boldsymbol{I} + \beta_l \boldsymbol{W}^l) + \boldsymbol{B}_{\boldsymbol{\alpha}}^l\big).$$

We call the resulting model GCNII-SCT.

EGNN. EGNN (Zhou et al., 2021) controls the smoothness of node features by constraining the lower and upper bounds of the Dirichlet energy of node features without considering the nonlinear activation function. Each EGNN layer can be written as follows:

$$\boldsymbol{H}^l = \sigma\big(\boldsymbol{W}^l(c_1 \boldsymbol{H}^0 + c_2 \boldsymbol{H}^{l-1} + (1 - c_{\min})\boldsymbol{H}^{l-1}\boldsymbol{G})\big), \tag{7}$$

where c_1, c_2 are learnable weights that satisfy $c_1 + c_2 = c_{\min}$ with $c_{\min}$ being a hyperparameter. To constrain the Dirichlet energy, EGNN initializes trainable weights $\boldsymbol{W}^l$ as diagonal matrices with prescribed singular



a) $\alpha = -0.25$ b) $\alpha = 0.0$ c) $\alpha = 1.0$

Figure 2: Node feature trajectories, with colorized magnitude, for varying smoothness control parameter α. For classical GCN b), the node features converge to the eigenspace $\mathcal{M}$ (red dashed line).

values and regularizes them to maintain orthogonality during training. Ignoring the activation function σ, the node features $\boldsymbol{H}^l$ at layer l satisfy:

$$c_{\min}\|\boldsymbol{H}^0\|_E \leq \|\boldsymbol{H}^l\|_E \leq c_{\max}\|\boldsymbol{H}^0\|_E,$$

where $c_{\max}$ is the square of the maximal singular value of the initialization of $\boldsymbol{W}^1$. Similarly, we modify EGNN to result in the following EGNN-SCT layer:

$$\boldsymbol{H}^l = \sigma\big(\boldsymbol{W}^l((1-c_{\min})\boldsymbol{H}^{l-1}\boldsymbol{G} + c_1\boldsymbol{H}^0 + c_2\boldsymbol{H}^{l-1}) + \boldsymbol{B}^l_{\boldsymbol{\alpha}}\big),$$

where everything remains the same as the EGNN layer, except that we include our proposed SCT $\boldsymbol{B}^l_{\boldsymbol{\alpha}}$.

6 Experiments

We evaluate SCT on a range of benchmark node classification tasks to demonstrate its effectiveness in GCN-style models. Datasets include citation networks (Cora, Citeseer, PubMed, Coauthor-Physics, Ogbn-arxiv), web knowledge bases (Cornell, Texas, Wisconsin), and Wikipedia networks (Chameleon, Squirrel); see Appendix C.1 for details. We implement GCN (Kipf & Welling, 2017) and GCNII (Chen et al., 2020b) (without weight sharing) using PyTorch Geometric (PyG) (Fey & Lenssen, 2019), and use the official codebase for EGNN (Zhou et al., 2021)[3].

6.1 Node Feature Trajectory

Following Oono & Suzuki (2020), we visualize the trajectory of node features for a simple graph with two connected nodes and 1D features. In this case, Eq. equation 6 simplifies to $\boldsymbol{h}^1 = \sigma(w\boldsymbol{h}^0\boldsymbol{G} + \boldsymbol{b}_\alpha)$, where $w = 1.2$, $\boldsymbol{h}^0, \boldsymbol{h}^1, \boldsymbol{b}_\alpha \in \mathbb{R}^2$, and $\boldsymbol{G} \in \mathbb{R}^{2\times 2}$ with $\boldsymbol{G} = [0.592, 0.194; 0.194, 0.908]$, a positive definite matrix with largest eigenvalue 1. We sample 20 initial vectors $\boldsymbol{h}^0$ uniformly from $[-1, 1] \times [-1, 1]$. Figure 2 shows the resulting trajectories relative to the eigenspace $\mathcal{M}$ (red dashed line). In panel a), some trajectories do not converge directly to $\mathcal{M}$; in b), setting $\alpha = 0.0$ recovers GCL, and all trajectories converge to $\mathcal{M}$; in c), large α (e.g., 1.0) causes significant initial deviation. These results show that α effectively controls feature trajectories.

6.2 Baseline Comparisons for Node Classification

6.2.1 Citation Networks

We compare the three GCN-style models from Section 5, with and without our proposed SCT, across varying depths in Table 1. For Cora, Citeseer, and PubMed, we use fixed splits from (Yang et al., 2016);

[3]https://github.com/Kaixiong-Zhou/EGNN

Layers	2	4	16	32
Cora				
GCN/GCN-SCT	81.1/**82.9**	80.4/**82.8**	64.9/**71.4**	60.3/**67.2**
GCNII/GCNII-SCT	82.2/**83.8**	82.6/**84.3**	84.6/**84.8**	85.4/**85.5**
EGNN/EGNN-SCT	83.2/**84.1**	84.2/**84.5**	**85.4**/83.3	**85.3**/82.0
Citeseer				
GCN/GCN-SCT	**70.3**/69.9	67.6/**67.7**	18.3/**55.4**	25.0/**51.0**
GCNII/GCNII-SCT	68.2/**72.8**	68.9/**72.8**	72.9/**73.8**	**73.4**/**73.4**
EGNN/EGNN-SCT	72.0/**73.1**	71.9/**72.0**	72.4/**72.6**	72.3/**72.9**
PubMed				
GCN/GCN-SCT	79.0/**79.8**	76.5/**78.4**	40.9/**76.1**	22.4/**77.0**
GCNII/GCNII-SCT	78.2/**79.7**	78.8/**80.1**	80.2/**80.7**	79.8/**80.7**
EGNN/EGNN-SCT	79.2/**79.8**	79.5/**80.4**	80.1/**80.3**	80.0/**80.4**
Coauthor-Physics				
GCN/GCN-SCT	92.4/$\mathbf{92.6 \pm 1.6}$	92.1/$\mathbf{92.5 \pm 5.9}$	13.5/$\mathbf{50.9 \pm 15.0}$	13.1/$\mathbf{43.6 \pm 16.0}$
GCNII/GCNII-SCT	92.5/$\mathbf{94.4 \pm 0.4}$	92.9/$\mathbf{94.2 \pm 0.3}$	92.9/$\mathbf{93.7 \pm 0.7}$	92.9/$\mathbf{94.1 \pm 0.3}$
EGNN/EGNN-SCT	92.6/$\mathbf{93.9 \pm 0.7}$	92.9/$\mathbf{94.1 \pm 0.4}$	93.1/$\mathbf{94.0 \pm 0.7}$	93.3/$\mathbf{93.8 \pm 1.3}$
Ogbn-arxiv				
GCN/GCN-SCT	70.4/$\mathbf{72.1 \pm 0.3}$	71.7/$\mathbf{72.7 \pm 0.3}$	70.6/$\mathbf{72.3 \pm 0.2}$	68.5/$\mathbf{72.3 \pm 0.3}$
GCNII/GCNII-SCT	70.1/$\mathbf{72.0 \pm 0.3}$	71.4/$\mathbf{72.2 \pm 0.2}$	71.5/$\mathbf{72.4 \pm 0.3}$	70.5/$\mathbf{72.1 \pm 0.3}$
EGNN/EGNN-SCT	68.4/$\mathbf{68.5 \pm 0.6}$	71.1/$\mathbf{71.3 \pm 0.5}$	72.7/$\mathbf{72.8 \pm 0.5}$	**72.7**/72.3 ± 0.5

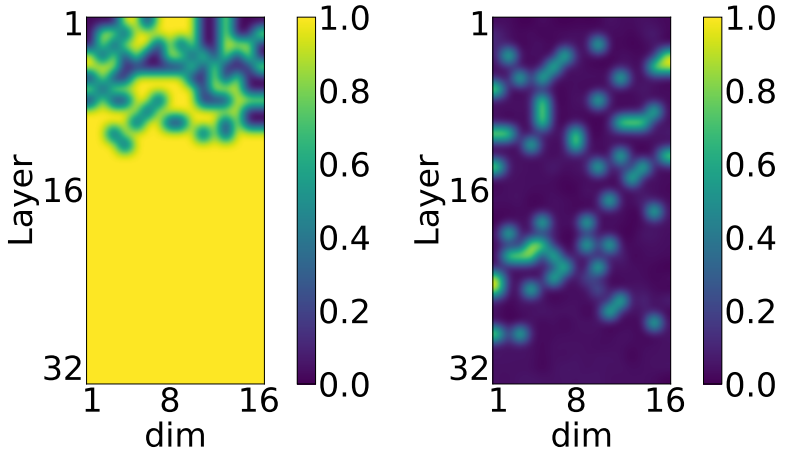
Table 1: Model accuracies across varying depths on citation datasets. While GCN-SCT with 16 or 32 layers shows accuracy drops, this is due to vanishing gradients—not over-smoothing. For Cora, Citeseer, and PubMed, we follow Chen et al. (2020b) using fixed splits and a single forward pass, reporting only test accuracy. For Coauthor-Physics and Ogbn-arxiv, we use the splits from Zhou et al. (2021) and report both test accuracy and standard deviation. Baseline results are taken from Chen et al. (2020b); Zhou et al. (2021); standard deviations were not reported. (Unit: %)

for Coauthor-Physics and Ogbn-arxiv, we follow (Zhou et al., 2021). Dataset details are in Appendix C. Following (Chen et al., 2020b), we train using Adam (Kingma & Ba, 2014) with a single pass, 1500 max epochs, and 100-epoch patience. Hyperparameter grids are listed in Table 6, and we accelerate tuning via Bayesian meta-learning (Biewald, 2020), running 200 iterations per model.

Table 1 reports the best test accuracy (ReLU vs. leaky ReLU) for GCN, GCNII, and their SCT variants[4]. EGNN uses the SReLU activation as in (Zhou et al., 2021). Results show SCT consistently improves GCN and GCNII. EGNN-SCT (with ReLU or leaky ReLU activation) may underperform EGNN (with SReLU activation), due to activation choice. However, Appendix C.3 shows that EGNN-SCT with SReLU activation outperforms EGNN across all tasks. Since SReLU is a shifted ReLU, our theory still applies. Model sizes and runtimes are reported in Table 5 in the appendix.

Table 1 shows that GCN accuracy drops at depths 16 and 32, even with SCT. To investigate, we analyze the normalized smoothness of node features in GCN and GCN-SCT. Figure 3 visualizes heatmaps for Citeseer with 32 layers, where rows represent layers and columns represent feature dimensions. In GCN (Fig. 3 a), features from layers 14–32 exhibit smoothness near 1, indicating homogeneity. In contrast, GCN-SCT (Fig. 3 b) produces more diverse features. This suggests the performance drop in deep GCN-SCT is not due to over-smoothing. Unlike GCNII and EGNN, GCN-SCT lacks skip connections, which help mitigate vanishing gradients (He et al., 2016a;b). As shown in Appendix C.3, GCN and GCN-SCT suffer from vanishing gradients, while the other models do not.

[4]See Appendix C for full comparisons.



a) GCN b) GCN-SCT

Figure 3: Normalized smoothness heatmaps for each feature dimension across layers in a) GCN and b) GCN-SCT on Citeseer (32 layers, 16 hidden dimensions). GCN features become fully smooth from layer 14 onward, while GCN-SCT maintains control over smoothness at all depths. Horizontal and vertical axes denote feature dimension and layer index, respectively.

6.2.2 Other Datasets

We further evaluate model performance using 10-fold cross-validation and fixed 48/32/20% splits, following (Pei et al., 2020). Tables 2 and 3 report accuracy and per-epoch runtime for GCN and GCNII (with and without SCT, using the leaky ReLU activation) on five heterophilic datasets: Cornell, Texas, Wisconsin, Chameleon, and Squirrel. EGNN is excluded, as these datasets were not considered in (Zhou et al., 2021). Baseline GCN and GCNII results are from (Chen et al., 2020b); all other models are tuned via Bayesian meta-learning to maximize validation accuracy. We report the best test accuracy over depths $\{2, 4, 8, 16, 32\}$.

SCT significantly improves accuracy and, in some cases, reduces runtime by achieving optimal performance at moderate depths. Table 9 (Section C.4) reports mean and standard deviation of test accuracy, while Table 10 shows that SCT adds minimal overhead at depth 8.

Cornell	Texas	Wisconsin	Chameleon	Squirrel
52.70/**55.95**	52.16/**62.16**	45.88/**54.71**	28.18/**38.44**	23.96/**35.31**
74.86/**75.41**	69.46/**83.34**	74.12/**86.08**	60.61/**64.52**	38.47/**47.51**

Table 2: Mean test accuracy on WebKB and WikipediaNetwork with fixed 48/32/20% splits. Rows: (1) GCN/GCN-SCT, (2) GCNII/GCNII-SCT. (Unit: %)

Cornell	Texas	Wisconsin	Chameleon	Squirrel
0.7/1.8	0.7/0.8	0.7/0.8	0.6/0.7	1.6/4.0
2.0/2.0	3.1/2.0	2.0/1.5	1.5/1.3	5.5/3.7

Table 3: Average epoch time on WebKB and WikipediaNetwork with fixed 48/32/20% splits. Rows: (1) GCN/GCN-SCT, (2) GCNII/GCNII-SCT. (Unit: $\times 10^{-2}$ second)

7 Concluding Remarks

We established a geometric understanding of how ReLU and leaky ReLU activation functions influence GCN feature smoothness. Our analysis of dimension-wise normalized smoothness shows that activation functions

not only smooth features but can also reduce or preserve their normalized smoothness. These insights guided the design of SCT, a new simple yet effective mechanism for controlling both normalized and unnormalized smoothness in GCN features. SCT offers theoretical guarantees for smoothness control in GCN-style models.

References

Justin Baker, Qingsong Wang, Cory D Hauck, and Bao Wang. Implicit graph neural networks: A monotone operator viewpoint. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 1521–1548. PMLR, 23–29 Jul 2023. URL `https://proceedings.mlr.press/v202/baker23a.html`.

Justin M Baker, Qingsong Wang, Martin Berzins, Thomas Strohmer, and Bao Wang. Monotone operator theory-inspired message passing for learning long-range interaction on graphs. In *International Conference on Artificial Intelligence and Statistics*, pp. 2233–2241. PMLR, 2024.

Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

Lukas Biewald. Experiment tracking with weights and biases, 2020. URL `https://www.wandb.com/`. Software available from wandb.com.

Chen Cai and Yusu Wang. A note on over-smoothing for graph neural networks. *arXiv preprint arXiv:2006.13318*, 2020.

Ben Chamberlain, James Rowbottom, Maria I Gorinova, Michael Bronstein, Stefan Webb, and Emanuele Rossi. Grand: Graph neural diffusion. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 1407–1418. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/chamberlain21a.html`.

Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 3438–3445, 2020a.

Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. Simple and deep graph convolutional networks. In Hal Daumé III and Aarti Singh (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 2020b.

Zhengdao Chen, Lisha Li, and Joan Bruna. Supervised community detection with line graph neural networks. In *International Conference on Learning Representations*, 2019.

Fan RK Chung. *Spectral graph theory*, volume 92. American Mathematical Soc., 1997.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29, 2016.

Francesco Di Giovanni, James Rowbottom, Benjamin P Chamberlain, Thomas Markovich, and Michael M Bronstein. Understanding convolution on graphs via energies. *arXiv preprint arXiv:2206.10991*, 2022.

Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

Zvi Galil and Giuseppe F Italiano. Data structures and algorithms for disjoint set union problems. *ACM Computing Surveys (CSUR)*, 23(3):319–344, 1991.

Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. Combining neural networks with personalized pagerank for classification on graphs. In *International Conference on Learning Representations*, 2019.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 1263–1272. JMLR.org, 2017.

Fangda Gu, Heng Chang, Wenwu Zhu, Somayeh Sojoudi, and Laurent El Ghaoui. Implicit graph neural networks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 11984–11995. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/file/8b5c8441a8ff8e151b191c53c1842a38-Paper.pdf`.

William Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017a.

William L Hamilton. *Graph representation learning*. Morgan & Claypool Publishers, 2020.

William L Hamilton, Rex Ying, and Jure Leskovec. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584*, 2017b.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016a.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pp. 630–645. Springer, 2016b.

Tatsuro Kawamoto, Masashi Tsubaki, and Tomoyuki Obuchi. Mean-field theory of graph neural networks in graph partitioning. *Advances in Neural Information Processing Systems*, 31, 2018.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Thirty-Second AAAI conference on artificial intelligence*, 2018.

Sitao Luan, Chenqing Hua, Qincheng Lu, Jiaqi Zhu, Mingde Zhao, Shuyuan Zhang, Xiao-Wen Chang, and Doina Precup. Revisiting heterophily for graph neural networks. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022.

Hoang Nt and Takanori Maehara. Revisiting graph neural networks: All we have is low-pass filters. *arXiv preprint arXiv:1905.09550*, 2019.

Kenta Oono and Taiji Suzuki. Graph neural networks exponentially lose expressive power for node classification. In *International Conference on Learning Representations*, 2020.

Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. In *International Conference on Learning Representations*, 2020.

Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. Dropedge: Towards deep graph convolutional networks on node classification. In *International Conference on Learning Representations*, 2020.

Michael Scholkemper, Xinyi Wu, Ali Jadbabaie, and Michael T Schaub. Residual connections and normalization can provably prevent oversmoothing in gnns. *arXiv preprint arXiv:2406.02997*, 2024.

Matthew Thorpe, Tan Minh Nguyen, Hedi Xia, Thomas Strohmer, Andrea Bertozzi, Stanley Osher, and Bao Wang. GRAND++: Graph neural diffusion with a source term. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=EMxu-dzvJk`.

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

Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (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.

Xinyi Wu, Amir Ajorlou, Zihui Wu, and Ali Jadbabaie. Demystifying oversmoothing in attention-based graph neural networks. *Advances in Neural Information Processing Systems*, 36:35084–35106, 2023a.

Xinyi Wu, Zhengdao Chen, William Wei Wang, and Ali Jadbabaie. A non-asymptotic analysis of oversmoothing in graph neural networks. In *The Eleventh International Conference on Learning Representations*, 2023b.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. Representation learning on graphs with jumping knowledge networks. In *International conference on machine learning*, pp. 5453–5462. PMLR, 2018.

Zhilin Yang, William Cohen, and Ruslan Salakhudinov. Revisiting semi-supervised learning with graph embeddings. In *International conference on machine learning*, pp. 40–48. PMLR, 2016.

Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 974–983, 2018.

Lingxiao Zhao and Leman Akoglu. Pairnorm: Tackling oversmoothing in gnns. In *International Conference on Learning Representations*, 2020.

Dengyong Zhou, Olivier Bousquet, Thomas Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. *Advances in neural information processing systems*, 16, 2003.

Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020.

Kaixiong Zhou, Xiao Huang, Daochen Zha, Rui Chen, Li Li, Soo-Hyun Choi, and Xia Hu. Dirichlet energy constrained learning for deep graph neural networks. *Advances in Neural Information Processing Systems*, 34:21834–21846, 2021.

Xiaojin Zhu, Zoubin Ghahramani, and John D Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International conference on Machine learning (ICML-03)*, pp. 912–919, 2003.

A Missing Proofs in Section 3

Proof of Proposition 3.1. We can write matrix $\boldsymbol{H}$ as:

$$\boldsymbol{H} = \sum_{i=1}^{n} \boldsymbol{H}\boldsymbol{e}_i\boldsymbol{e}_i^\top,$$

where each $\boldsymbol{e}_i$ is the eigenvector of $\boldsymbol{G}$ associated with eigenvalue λ_i. This indicates that

$$\begin{aligned}
\boldsymbol{H}\tilde{\Delta} &= \boldsymbol{H}(\boldsymbol{I} - \boldsymbol{G}) \\
&= \sum_{i=1}^{n} (\boldsymbol{H}\boldsymbol{e}_i\boldsymbol{e}_i^\top - \boldsymbol{H}\boldsymbol{e}_i\boldsymbol{e}_i^\top\boldsymbol{G}) \\
&= \sum_{i=1}^{n} (1-\lambda_i)\boldsymbol{H}\boldsymbol{e}_i\boldsymbol{e}_i^\top \\
&= \sum_{i=m+1}^{n} (1-\lambda_i)\boldsymbol{H}\boldsymbol{e}_i\boldsymbol{e}_i^\top.
\end{aligned}$$

Then using the fact that $1 - \lambda_i \geq 0$ for each i, we obtain

$$\begin{aligned}
\|\boldsymbol{H}\|_E^2 &= \mathrm{Trace}(\boldsymbol{H}\tilde{\Delta}\boldsymbol{H}^\top) \\
&= \mathrm{Trace}\Big(\sum_{i=m+1}^{n} (1-\lambda_i)\boldsymbol{H}\boldsymbol{e}_i\boldsymbol{e}_i^\top (\sum_{j=1}^{n} \boldsymbol{H}\boldsymbol{e}_j\boldsymbol{e}_j^\top)^\top \Big) \\
&= \mathrm{Trace}\Big(\sum_{i=m+1}^{n} \sum_{j=1}^{n} (1-\lambda_i)\boldsymbol{H}\boldsymbol{e}_i\boldsymbol{e}_i^\top\boldsymbol{e}_j\boldsymbol{e}_j^\top\boldsymbol{H}^\top \Big) \\
&= \mathrm{Trace}\Big(\sum_{i=m+1}^{n} (1-\lambda_i)\boldsymbol{H}\boldsymbol{e}_i\boldsymbol{e}_i^\top\boldsymbol{e}_i\boldsymbol{e}_i^\top\boldsymbol{H}^\top \Big) \\
&= \mathrm{Trace}\Big(\sum_{i=m+1}^{n} \sqrt{1-\lambda_i}\boldsymbol{H}\boldsymbol{e}_i\boldsymbol{e}_i^\top (\sum_{j=m+1}^{n} \sqrt{1-\lambda_j}\boldsymbol{H}\boldsymbol{e}_j\boldsymbol{e}_j^\top)^\top \Big) \\
&= \Big\| \sum_{i=m+1}^{n} \sqrt{1-\lambda_i}\boldsymbol{H}\boldsymbol{e}_i\boldsymbol{e}_i^\top \Big\|_F^2.
\end{aligned}$$

That is,

$$\|\boldsymbol{H}\|_E = \Big\| \sum_{i=m+1}^{n} \sqrt{1-\lambda_i}\boldsymbol{H}\boldsymbol{e}_i\boldsymbol{e}_i^\top \Big\|_F.$$

On the other hand, equation 3 implies

$$\|\boldsymbol{H}\|_{\mathcal{M}^\perp} = \|\boldsymbol{H}_{\mathcal{M}^\perp}\|_F = \Big\| \sum_{i=m+1}^{n} \boldsymbol{H}\boldsymbol{e}_i\boldsymbol{e}_i^\top \Big\|_F.$$

We first show that both $\|\boldsymbol{H}\|_{\mathcal{M}^\perp}$ and $\|\boldsymbol{H}\|_E$ are seminorms. Since $\|c\boldsymbol{H}\|_F = |c| \cdot \|\boldsymbol{H}\|_F, \forall c \in \mathbb{R}$, we have $\|c\boldsymbol{H}\|_{\mathcal{M}^\perp} = |c| \cdot \|\boldsymbol{H}\|_{\mathcal{M}^\perp}$ and $\|c\boldsymbol{H}\|_E = |c| \cdot \|\boldsymbol{H}\|_E$. Moreover, for any two matrices $\boldsymbol{H}^1$ and $\boldsymbol{H}^2$ s.t.

$\boldsymbol{H} = \boldsymbol{H}^1 + \boldsymbol{H}^2$, we have

$$\sum_{i=m+1}^{n} \boldsymbol{H}^1 \boldsymbol{e}_i \boldsymbol{e}_i^\top + \sum_{i=m+1}^{n} \boldsymbol{H}^2 \boldsymbol{e}_i \boldsymbol{e}_i^\top = \sum_{i=m+1}^{n} \boldsymbol{H} \boldsymbol{e}_i \boldsymbol{e}_i^\top,$$
$$\sum_{i=m+1}^{n} \sqrt{1-\lambda_i} \boldsymbol{H}^1 \boldsymbol{e}_i \boldsymbol{e}_i^\top + \sum_{i=m+1}^{n} \sqrt{1-\lambda_i} \boldsymbol{H}^2 \boldsymbol{e}_i \boldsymbol{e}_i^\top = \sum_{i=m+1}^{n} \sqrt{1-\lambda_i} \boldsymbol{H} \boldsymbol{e}_i \boldsymbol{e}_i^\top.$$

Then the triangle inequality of $\|\cdot\|_F$ implies that of $\|\boldsymbol{H}\|_{\mathcal{M}^\perp}$ and $\|\boldsymbol{H}\|_E$, respectively.

Now since $0 < 1 - \lambda_{m+1} \leq 1 - \lambda_i \leq 2$ for any $i = m+1, \ldots, n$, we may take $\alpha = \sqrt{1-\lambda_{m+1}}$ and $\beta = \sqrt{2}$. Then

$$\begin{aligned}
\alpha \|\boldsymbol{H}\|_{\mathcal{M}^\perp} &= \Big\| \alpha \sum_{i=m+1}^{n} \boldsymbol{H} \boldsymbol{e}_i \boldsymbol{e}_i^\top \Big\|_F \\
&\leq \Big\| \sum_{i=m+1}^{n} \sqrt{1-\lambda_i} \boldsymbol{H} \boldsymbol{e}_i \boldsymbol{e}_i^\top \Big\|_F \\
&\leq \Big\| \beta \sum_{i=m+1}^{n} \boldsymbol{H} \boldsymbol{e}_i \boldsymbol{e}_i^\top \Big\|_F \\
&= \beta \|\boldsymbol{H}\|_{\mathcal{M}^\perp}.
\end{aligned}$$

The result thus follows from $\|\boldsymbol{H}\|_E = \Big\| \sum_{i=m+1}^{n} \sqrt{1-\lambda_i} \boldsymbol{H} \boldsymbol{e}_i \boldsymbol{e}_i^\top \Big\|_F$. □

A.1 ReLU

Lemma A.1. *Let $\boldsymbol{Z} \in \mathbb{R}^{d \times n}$, and let $\boldsymbol{Z}^+ = \max(\boldsymbol{Z}, 0)$ and $\boldsymbol{Z}^- = \max(-\boldsymbol{Z}, 0)$ be the positive and negative parts of $\boldsymbol{Z}$. Then (1) $\boldsymbol{Z}^+, \boldsymbol{Z}^-$ are (component-wise) nonnegative and $\boldsymbol{Z} = \boldsymbol{Z}^+ - \boldsymbol{Z}^-$ and (2) $\langle \boldsymbol{Z}^+, \boldsymbol{Z}^- \rangle_F = 0$.*

Proof of Lemma A.1. Notice that for any $a \in \mathbb{R}$, we have

$$\max(a, 0) = \begin{cases} a & \text{if } a \geq 0 \\ 0 & \text{otherwise} \end{cases}.$$

and

$$\max(-a, 0) = \begin{cases} 0 & \text{if } a \geq 0 \\ -a & \text{otherwise} \end{cases}.$$

This implies that $a = \max(a, 0) - \max(-a, 0)$ and $\max(a, 0) \cdot \max(-a, 0) = 0$.

Let Z_{ij} be the $(i, j)^{th}$ entry of $\boldsymbol{Z}$. Then $\boldsymbol{Z} = \boldsymbol{Z}^+ - \boldsymbol{Z}^-$ follows from $Z_{ij} = \max(Z_{ij}, 0) - \max(-Z_{ij}, 0)$. Also,

$$\begin{aligned}
\langle \boldsymbol{Z}^+, \boldsymbol{Z}^- \rangle_F &= \mathrm{Trace}((\boldsymbol{Z}^+)^\top \boldsymbol{Z}^-) \\
&= \sum_{i=1}^{d} \sum_{j=1}^{j} \max(Z_{ij}, 0) \max(-Z_{ij}, 0) = 0.
\end{aligned}$$

□

Before proving Proposition 3.2, we notice the following relation between $\boldsymbol{Z}$ and $\boldsymbol{H}$.

Lemma A.2. *Given $\boldsymbol{Z} \in \mathbb{R}^{d\times n}$, let $\boldsymbol{H} = \sigma(\boldsymbol{Z})$ with σ being ReLU, then $\boldsymbol{H}$ lies on the high-dimensional sphere, in $\|\cdot\|_F$ norm, that is centered at $\boldsymbol{Z}/2$ and with radius $\|\boldsymbol{Z}/2\|_F$. That is, $\boldsymbol{H}$ and $\boldsymbol{Z}$ satisfy the following equation:*

$$\left\|\boldsymbol{H} - \frac{\boldsymbol{Z}}{2}\right\|_F^2 = \left\|\frac{\boldsymbol{Z}}{2}\right\|_F^2. \tag{8}$$

Proof of Lemma A.2. We observe that $\boldsymbol{H} = \sigma(\boldsymbol{Z}) = \max(\boldsymbol{Z}, 0) = \boldsymbol{Z}^+$ is the positive part of $\boldsymbol{Z}$. Then we have

$$\langle \boldsymbol{H}, \boldsymbol{Z}\rangle_F = \langle \boldsymbol{H}, \boldsymbol{Z}^+ - \boldsymbol{Z}^-\rangle_F = \langle \boldsymbol{H}, \boldsymbol{Z}^+\rangle_F - \langle \boldsymbol{H}, \boldsymbol{Z}^-\rangle_F = \langle \boldsymbol{H}, \boldsymbol{H}\rangle_F,$$

where we have used $\boldsymbol{Z} = \boldsymbol{Z}^+ - \boldsymbol{Z}^-$ and $\langle \boldsymbol{H}, \boldsymbol{Z}^-\rangle_F = \langle \boldsymbol{Z}^+, \boldsymbol{Z}^-\rangle_F = 0$ from Lemma A.1.

Therefore, one can deduce the desired result as follows:

$$\begin{aligned}\langle \boldsymbol{H}, \boldsymbol{H}\rangle_F - \langle \boldsymbol{H}, \boldsymbol{Z}\rangle_F = 0 \Rightarrow &\|\boldsymbol{H}\|_F^2 - 2\left\langle \boldsymbol{H}, \frac{\boldsymbol{Z}}{2}\right\rangle_F + \left\|\frac{\boldsymbol{Z}}{2}\right\|_F^2 = \left\|\frac{\boldsymbol{Z}}{2}\right\|_F^2 \\ \Rightarrow &\left\|\boldsymbol{H} - \frac{\boldsymbol{Z}}{2}\right\|_F^2 = \left\|\frac{\boldsymbol{Z}}{2}\right\|_F^2.\end{aligned}$$

□

Applying $\|\boldsymbol{H}\|_F^2 = \|\boldsymbol{H}_{\mathcal{M}} + \boldsymbol{H}_{\mathcal{M}^\perp}\|_F^2 = \|\boldsymbol{H}_{\mathcal{M}}\|_F^2 + \|\boldsymbol{H}_{\mathcal{M}^\perp}\|_F^2$, to both $\frac{\boldsymbol{Z}}{2}$ and $\boldsymbol{H} - \frac{\boldsymbol{Z}}{2}$, we obtain

$$\left\|\frac{\boldsymbol{Z}}{2}\right\|_F^2 = \left\|\frac{\boldsymbol{Z}_{\mathcal{M}^\perp}}{2}\right\|_F^2 + \left\|\frac{\boldsymbol{Z}_{\mathcal{M}}}{2}\right\|_F^2,$$

and

$$\left\|\boldsymbol{H} - \frac{\boldsymbol{Z}}{2}\right\|_F^2 = \left\|\boldsymbol{H}_{\mathcal{M}^\perp} - \frac{\boldsymbol{Z}_{\mathcal{M}^\perp}}{2}\right\|_F^2 + \left\|\boldsymbol{H}_{\mathcal{M}} - \frac{\boldsymbol{Z}_{\mathcal{M}}}{2}\right\|_F^2.$$

Then equation 8 becomes

$$\left\|\frac{\boldsymbol{Z}_{\mathcal{M}^\perp}}{2}\right\|_F^2 - \left\|\boldsymbol{H}_{\mathcal{M}^\perp} - \frac{\boldsymbol{Z}_{\mathcal{M}^\perp}}{2}\right\|_F^2 = \left\|\boldsymbol{H}_{\mathcal{M}} - \frac{\boldsymbol{Z}_{\mathcal{M}}}{2}\right\|_F^2 - \left\|\frac{\boldsymbol{Z}_{\mathcal{M}}}{2}\right\|_F^2 \tag{9}$$

By direct calculation, we have

$$\begin{aligned}\left\|\boldsymbol{H}_{\mathcal{M}} - \frac{\boldsymbol{Z}_{\mathcal{M}}}{2}\right\|_F^2 - \left\|\frac{\boldsymbol{Z}_{\mathcal{M}}}{2}\right\|_F^2 &= \langle \boldsymbol{H}_{\mathcal{M}}, \boldsymbol{H}_{\mathcal{M}}\rangle_F - 2\left\langle \boldsymbol{H}_{\mathcal{M}}, \frac{\boldsymbol{Z}_{\mathcal{M}}}{2}\right\rangle_F \\ &= \langle \boldsymbol{H}_{\mathcal{M}}, \boldsymbol{H}_{\mathcal{M}} - \boldsymbol{Z}_{\mathcal{M}}\rangle_F.\end{aligned} \tag{10}$$

Combining equation 9 and equation 10, we obtain the following result:

Lemma A.3. *For any $\boldsymbol{Z} = \boldsymbol{Z}_{\mathcal{M}} + \boldsymbol{Z}_{\mathcal{M}^\perp}$, let*

$$\boldsymbol{H} = \sigma(\boldsymbol{Z}) = \boldsymbol{H}_{\mathcal{M}} + \boldsymbol{H}_{\mathcal{M}^\perp},$$

then we have

$$\left\|\frac{\boldsymbol{Z}_{\mathcal{M}^\perp}}{2}\right\|_F^2 - \left\|\boldsymbol{H}_{\mathcal{M}^\perp} - \frac{\boldsymbol{Z}_{\mathcal{M}^\perp}}{2}\right\|_F^2 = \langle \boldsymbol{Z}_{\mathcal{M}}^+, \boldsymbol{Z}_{\mathcal{M}}^-\rangle_F.$$

where $\boldsymbol{Z}_{\mathcal{M}}^+ = \sum_{i=1}^m \boldsymbol{Z}^+\boldsymbol{e}_i\boldsymbol{e}_i^\top$, $\boldsymbol{Z}_{\mathcal{M}}^- = \sum_{i=1}^m \boldsymbol{Z}^-\boldsymbol{e}_i\boldsymbol{e}_i^\top$.

Proof of Lemma A.3. Recall that $\boldsymbol{H} = \sigma(\boldsymbol{Z}) = \max(\boldsymbol{Z}, 0) = \boldsymbol{Z}^+$. Also, $\boldsymbol{Z} = \boldsymbol{Z}^+ - \boldsymbol{Z}^-$ implies $\boldsymbol{Z}_{\mathcal{M}} = \boldsymbol{Z}_{\mathcal{M}}^+ - \boldsymbol{Z}_{\mathcal{M}}^- = \boldsymbol{H}_{\mathcal{M}}^+ - \boldsymbol{Z}_{\mathcal{M}}^-$. Therefore, we see that

$$\langle \boldsymbol{H}_{\mathcal{M}}, \boldsymbol{H}_{\mathcal{M}} - \boldsymbol{Z}_{\mathcal{M}}\rangle_F = \langle \boldsymbol{Z}_{\mathcal{M}}^+, \boldsymbol{Z}_{\mathcal{M}}^-\rangle_F.$$

□

By using the fact that $\langle \boldsymbol{Z}_{\mathcal{M}}^+, \boldsymbol{Z}_{\mathcal{M}}^-\rangle_F \geq 0$ in Lemma A.3, we reveal a geometric relation between $\boldsymbol{Z}$ and $\boldsymbol{H}$ mentioned in Proposition 3.2.

Proof of Proposition 3.2. Since $\boldsymbol{Z}^+, \boldsymbol{Z}^- \geq 0$ are nonnegative and all the eigenvectors $\boldsymbol{e}_i$ are also nonnegative, we see that $\boldsymbol{Z}^+_{\mathcal{M}} = \sum_{i=1}^m \boldsymbol{Z}^+ \boldsymbol{e}_i \boldsymbol{e}_i^\top$ and $\boldsymbol{Z}^-_{\mathcal{M}} = \sum_{i=1}^m \boldsymbol{Z}^- \boldsymbol{e}_i \boldsymbol{e}_i^\top$ are nonnegative. This indicates that

$$\langle \boldsymbol{Z}^+_{\mathcal{M}}, \boldsymbol{Z}^-_{\mathcal{M}} \rangle_F = \mathrm{Trace}\Big(\boldsymbol{Z}^+_{\mathcal{M}} (\boldsymbol{Z}^-_{\mathcal{M}})^\top \Big) \geq 0.$$

Then according to Lemma A.3, we obtain

$$\Big\| \frac{\boldsymbol{Z}_{\mathcal{M}^\perp}}{2} \Big\|_F^2 - \Big\| \boldsymbol{H}_{\mathcal{M}^\perp} - \frac{\boldsymbol{Z}_{\mathcal{M}^\perp}}{2} \Big\|_F^2 = \langle \boldsymbol{Z}^+_{\mathcal{M}}, \boldsymbol{Z}^-_{\mathcal{M}} \rangle_F \geq 0.$$

So we have

$$\begin{aligned} \Big\| \boldsymbol{H}_{\mathcal{M}^\perp} - \frac{\boldsymbol{Z}_{\mathcal{M}^\perp}}{2} \Big\|_F &= \sqrt{\Big\| \frac{\boldsymbol{Z}_{\mathcal{M}^\perp}}{2} \Big\|_F^2 - \langle \boldsymbol{Z}^+_{\mathcal{M}}, \boldsymbol{Z}^-_{\mathcal{M}} \rangle_F} \\ &= \sqrt{\Big\| \frac{\boldsymbol{Z}_{\mathcal{M}^\perp}}{2} \Big\|_F^2 - \langle \boldsymbol{H}_{\mathcal{M}}, \boldsymbol{H}_{\mathcal{M}} - \boldsymbol{Z}_{\mathcal{M}} \rangle_F}, \end{aligned}$$

which shows that $\boldsymbol{H}_{\mathcal{M}^\perp}$ lies on the high-dimensional sphere that we have claimed. Furthermore, we conclude that

$$0 \leq \Big\| \boldsymbol{H}_{\mathcal{M}^\perp} - \frac{\boldsymbol{Z}_{\mathcal{M}^\perp}}{2} \Big\|_F \leq \Big\| \frac{\boldsymbol{Z}_{\mathcal{M}^\perp}}{2} \Big\|_F. \tag{11}$$

This demonstrates that $\boldsymbol{H}_{\mathcal{M}^\perp}$ lies on the high-dimensional sphere we have stated.

Since the sphere $\Big\| \boldsymbol{H}_{\mathcal{M}^\perp} - \frac{\boldsymbol{Z}_{\mathcal{M}^\perp}}{2} \Big\|_F^2 = \Big\| \frac{\boldsymbol{Z}_{\mathcal{M}^\perp}}{2} \Big\|_F^2$ passes through the origin, the distance of any $\boldsymbol{H}_{\mathcal{M}^\perp}$ to the origin must be no greater than the diameter of this sphere, i.e., $\|\boldsymbol{H}_{\mathcal{M}^\perp}\|_F \leq \|\boldsymbol{Z}_{\mathcal{M}^\perp}\|_F$. Also, this can be derived from the following inequality:

$$\|\boldsymbol{H}_{\mathcal{M}^\perp}\|_F - \Big\| \frac{\boldsymbol{Z}_{\mathcal{M}^\perp}}{2} \Big\|_F \leq \Big\| \boldsymbol{H}_{\mathcal{M}^\perp} - \frac{\boldsymbol{Z}_{\mathcal{M}^\perp}}{2} \Big\|_F \leq \Big\| \frac{\boldsymbol{Z}_{\mathcal{M}^\perp}}{2} \Big\|_F.$$

We see that the maximal smoothness $\|\boldsymbol{H}_{\mathcal{M}^\perp}\|_F = \|\boldsymbol{Z}_{\mathcal{M}^\perp}\|_F$ is attained when $\boldsymbol{H}_{\mathcal{M}^\perp} = \boldsymbol{Z}_{\mathcal{M}^\perp}$, the intersection of the surface and the line passing through the center and the origin.

After all, we complete the proof by using the fact that $\|\boldsymbol{Z}_{\mathcal{M}^\perp}\|_F = \|\boldsymbol{Z}\|_{\mathcal{M}^\perp}$ for any matrix $\boldsymbol{Z}$, which implies $\|\boldsymbol{H}\|_{\mathcal{M}^\perp} = \|\boldsymbol{H}_{\mathcal{M}^\perp}\|_F \leq \|\boldsymbol{Z}_{\mathcal{M}^\perp}\|_F = \|\boldsymbol{Z}\|_{\mathcal{M}^\perp}$. □

A.2 Leaky ReLU

For the leaky ReLU activation, we have the following Lemma:

Lemma A.4. *If $\boldsymbol{H} = \sigma_a(\boldsymbol{Z})$ with σ_a being the leaky ReLU activation function, then $\boldsymbol{H}$ lies on the high-dimensional sphere centered at $(1+a)\boldsymbol{Z}/2$ with radius $\|(1-a)\boldsymbol{Z}/2\|_F$.*

Proof of Lemma A.4. Notice that

$$\boldsymbol{H} = \sigma_a(\boldsymbol{Z}) = \boldsymbol{Z}^+ - a\boldsymbol{Z}^-.$$

Then $\boldsymbol{H} - \boldsymbol{Z} = (1-a)\boldsymbol{Z}^-$ and $\boldsymbol{H} - a\boldsymbol{Z} = (1-a)\boldsymbol{Z}^+$. Using $\langle \boldsymbol{Z}^-, \boldsymbol{Z}^+ \rangle_F = 0$, we have

$$\begin{aligned} &\langle \boldsymbol{H} - \boldsymbol{Z}, \boldsymbol{H} - a\boldsymbol{Z} \rangle_F = 0 \\ \Rightarrow &\|\boldsymbol{H}\|_F^2 - 2\Big\langle \boldsymbol{H}, \frac{(1+a)\boldsymbol{Z}}{2} \Big\rangle_F + a\|\boldsymbol{Z}\|_F^2 = 0 \\ \Rightarrow &\|\boldsymbol{H}\|_F^2 - 2\Big\langle \boldsymbol{H}, \frac{(1+a)\boldsymbol{Z}}{2} \Big\rangle_F = -a\|\boldsymbol{Z}\|_F^2 \\ \Rightarrow &\Big\| \boldsymbol{H} - \frac{(1+a)}{2}\boldsymbol{Z} \Big\|_F^2 = \Big\| \frac{(1+a)}{2}\boldsymbol{Z} \Big\|_F^2 - a\|\boldsymbol{Z}\|_F^2 = \Big\| \frac{(1-a)}{2}\boldsymbol{Z} \Big\|_F^2. \end{aligned}$$

□

Moreover, we notice that

Lemma A.5. *For any $\boldsymbol{Z} = \boldsymbol{Z}_{\mathcal{M}} + \boldsymbol{Z}_{\mathcal{M}^\perp}$, let $\boldsymbol{H} = \sigma_a(\boldsymbol{Z}) = \boldsymbol{H}_{\mathcal{M}} + \boldsymbol{H}_{\mathcal{M}^\perp}$, then*

$$\begin{aligned}&\left\|\frac{(1-a)}{2}\boldsymbol{Z}_{\mathcal{M}^\perp}\right\|_F^2 - \left\|\boldsymbol{H}_{\mathcal{M}^\perp} - \frac{(1+a)}{2}\boldsymbol{Z}_{\mathcal{M}^\perp}\right\|_F^2\\ =&(1-a)^2\langle \boldsymbol{Z}_{\mathcal{M}}^+, \boldsymbol{Z}_{\mathcal{M}}^-\rangle_F.\end{aligned}$$

Proof of Lemma A.5. Similar to the proof of Lemma A.3, the orthogonal decomposition implies that

$$\begin{aligned}&\left\|\frac{(1-a)}{2}\boldsymbol{Z}_{\mathcal{M}^\perp}\right\|_F^2 - \left\|\boldsymbol{H}_{\mathcal{M}^\perp} - \frac{(1+a)}{2}\boldsymbol{Z}_{\mathcal{M}^\perp}\right\|_F^2\\ =&\left\|\boldsymbol{H}_{\mathcal{M}} - \frac{(1+a)}{2}\boldsymbol{Z}_{\mathcal{M}}\right\|_F^2 - \left\|\frac{(1-a)}{2}\boldsymbol{Z}_{\mathcal{M}}\right\|_F^2\\ =&\langle \boldsymbol{H}_{\mathcal{M}} - \boldsymbol{Z}_{\mathcal{M}}, \boldsymbol{H}_{\mathcal{M}} - a\boldsymbol{Z}_{\mathcal{M}}\rangle_F\\ =&\langle (1-a)\boldsymbol{Z}_{\mathcal{M}}^-, (1-a)\boldsymbol{Z}_{\mathcal{M}}^+\rangle_F\\ =&(1-a)^2\langle \boldsymbol{Z}_{\mathcal{M}}^-, \boldsymbol{Z}_{\mathcal{M}}^+\rangle_F.\end{aligned}$$

□

Proof of Proposition 3.3. Similar to the proof of Proposition 3.2, we apply $\langle \boldsymbol{Z}_{\mathcal{M}}^-, \boldsymbol{Z}_{\mathcal{M}}^+\rangle_F \geq 0$ to Lemma A.5 and hence obtain the geometric condition as follows:

$$\left\|\boldsymbol{H}_{\mathcal{M}^\perp} - \frac{(1+a)}{2}\boldsymbol{Z}_{\mathcal{M}^\perp}\right\|_F = \sqrt{\left\|\frac{(1-a)}{2}\boldsymbol{Z}_{\mathcal{M}^\perp}\right\|_F^2 - \langle \boldsymbol{H}_{\mathcal{M}} - \boldsymbol{Z}_{\mathcal{M}}, \boldsymbol{H}_{\mathcal{M}} - a\boldsymbol{Z}_{\mathcal{M}}\rangle_F}.$$

Then we have the following inequality:

$$0 \leq \left\|\boldsymbol{H}_{\mathcal{M}^\perp} - \frac{(1+a)}{2}\boldsymbol{Z}_{\mathcal{M}^\perp}\right\|_F \leq \left\|\frac{(1-a)}{2}\boldsymbol{Z}_{\mathcal{M}^\perp}\right\|_F.$$

Moreover, we deduce that

$$\begin{aligned}\left|\|\boldsymbol{H}_{\mathcal{M}^\perp}\|_F - \left\|\frac{(1+a)}{2}\boldsymbol{Z}_{\mathcal{M}^\perp}\right\|_F\right| &\leq \left\|\boldsymbol{H}_{\mathcal{M}^\perp} - \frac{(1+a)}{2}\boldsymbol{Z}_{\mathcal{M}^\perp}\right\|_F\\ &\leq \left\|\frac{(1-a)}{2}\boldsymbol{Z}_{\mathcal{M}^\perp}\right\|_F.\end{aligned}$$

and hence

$$\begin{aligned}-\left\|\frac{(1-a)}{2}\boldsymbol{Z}_{\mathcal{M}^\perp}\right\|_F \leq \|\boldsymbol{H}_{\mathcal{M}^\perp}\|_F - \left\|\frac{(1+a)}{2}\boldsymbol{Z}_{\mathcal{M}^\perp}\right\|_F\\ \leq \left\|\frac{(1-a)}{2}\boldsymbol{Z}_{\mathcal{M}^\perp}\right\|_F.\end{aligned}$$

Therefore, we obtain $a\|\boldsymbol{Z}_{\mathcal{M}^\perp}\|_F \leq \|\boldsymbol{H}_{\mathcal{M}^\perp}\|_F \leq \|\boldsymbol{Z}_{\mathcal{M}^\perp}\|_F$. (Remark that $\boldsymbol{H}_{\mathcal{M}^\perp}$ achieves its maximal norm when it is equal to $\boldsymbol{Z}_{\mathcal{M}^\perp}$, the intersection of the surface and the line passing through the center and the origin.)

By using the fact that $\|\boldsymbol{Z}_{\mathcal{M}^\perp}\|_F = \|\boldsymbol{Z}\|_{\mathcal{M}^\perp}$ for any matrix $\boldsymbol{Z}$, we conclude that $a\|\boldsymbol{Z}\|_{\mathcal{M}^\perp} \leq \|\boldsymbol{H}\|_{\mathcal{M}^\perp} \leq \|\boldsymbol{Z}\|_{\mathcal{M}^\perp}$. □

B Proofs in Section 4

Throughout this section, we assume that $\boldsymbol{z}_{\mathcal{M}^\perp} \neq \boldsymbol{0}$.

Proof of Proposition 4.3. Recall that $\boldsymbol{e} = \tilde{\boldsymbol{D}}^{\frac{1}{2}}\boldsymbol{u}_n/c$ has only positive entries where $\tilde{\boldsymbol{D}}$ is the augmented degree matrix and $\boldsymbol{u}_n = [1, \ldots, 1]^\top \in \mathbb{R}^n$ and $c = \|\tilde{\boldsymbol{D}}^{\frac{1}{2}}\boldsymbol{u}_n\|$. Let d_i be the i^{th} diagonal entry of $\tilde{\boldsymbol{D}}$. Then we have $\boldsymbol{e} = [\sqrt{d_1}/c, \sqrt{d_2}/c, \ldots, \sqrt{d_n}/c]^\top$ and $c = \sqrt{\sum_{i=1}^n d_i}$.

Note that $\boldsymbol{z}(\alpha) = \boldsymbol{z} - \alpha \boldsymbol{e} = \boldsymbol{z} - \frac{\alpha}{c}\tilde{\boldsymbol{D}}^{\frac{1}{2}}\boldsymbol{u}_n = \tilde{\boldsymbol{D}}^{\frac{1}{2}}(\tilde{\boldsymbol{D}}^{-\frac{1}{2}}\boldsymbol{z} - \frac{\alpha}{c}\boldsymbol{u}_n) = \tilde{\boldsymbol{D}}^{\frac{1}{2}}(\boldsymbol{x} - \frac{\alpha}{c}\boldsymbol{u}_n)$, where we assume $\boldsymbol{x} \coloneqq \tilde{\boldsymbol{D}}^{-\frac{1}{2}}\boldsymbol{z}$. Then we observe that when σ is the ReLU activation function,

$$\boldsymbol{h}(\alpha) = \sigma(\boldsymbol{z}(\alpha)) = \sigma\Big(\tilde{\boldsymbol{D}}^{\frac{1}{2}}(\boldsymbol{x} - \frac{\alpha}{c}\boldsymbol{u}_n)\Big) = \tilde{\boldsymbol{D}}^{\frac{1}{2}}\sigma\Big(\boldsymbol{x} - \frac{\alpha}{c}\boldsymbol{u}_n\Big),$$

and hence

$$\langle \boldsymbol{h}(\alpha), \boldsymbol{e}\rangle = \Big\langle \tilde{\boldsymbol{D}}^{\frac{1}{2}}\sigma\Big(\boldsymbol{x} - \frac{\alpha}{c}\boldsymbol{u}_n\Big), \boldsymbol{e}\Big\rangle = \Big\langle \sigma\Big(\boldsymbol{x} - \frac{\alpha}{c}\boldsymbol{u}_n\Big), \tilde{\boldsymbol{D}}\boldsymbol{u}_n\Big\rangle.$$

We may now assume $\boldsymbol{x} = [x_1, \ldots, x_n]^\top$ is well-ordered s.t. $x_1 \geq x_2 \geq \ldots \geq x_n$. Indeed, there is a collection of indices $\{k_1, ..., k_l\}$ such that

$$\begin{aligned}
&x_1 = \ldots, x_{k_1} \text{ and } x_{k_1} > x_{k_1+1},\\
&x_{k_{j-1}+1} = \ldots = x_{k_j} \text{ and } x_{k_j} > x_{k_j+1} \text{ for any } j = 2, \ldots, l-1,\\
&x_{k_{l-1}+1} = \ldots = x_{k_l} \text{ and } k_l = n.
\end{aligned}$$

That is, $x_1 = x_2 = \ldots = x_{k_1} > x_{k_1+1} = \ldots = x_{k_2} > x_{k_2+1} = \ldots = x_{k_3} > x_{k_3+1} \ldots$.

We first restrict the domain of α s.t. $\boldsymbol{h}(\alpha) \neq 0$. Note that we have

$$\begin{aligned}
\boldsymbol{h}(\alpha) = 0 \Leftrightarrow &\sigma\Big(\boldsymbol{x} - \frac{\alpha}{c}\boldsymbol{u}_n\Big) = 0\\
\Leftrightarrow &x_i - \frac{\alpha}{c} \leq 0 \text{ for } i = 1, \ldots, n\\
\Leftrightarrow &x_1 - \frac{\alpha}{c} \leq 0\\
\Leftrightarrow &\alpha \geq c x_1.
\end{aligned}$$

So we will study the smoothness $s(\boldsymbol{h}(\alpha))$ when $\alpha < cx_1$.

Let $\epsilon > 0$ and consider $\alpha = c(x_1 - \epsilon)$. When $\epsilon \leq x_1 - x_{k_1+1} = x_1 - x_{k_2}$, we see that

$$\boldsymbol{x} - \frac{\alpha}{c}\boldsymbol{u}_n = [\epsilon, \ldots, \epsilon, \epsilon - (x_1 - x_{k_1+1}), \ldots, \epsilon - (x_1 - x_n)]^\top,$$

where only the first k_1 entries are positive since $x_1 - x_i \geq \epsilon$ for any $i \geq k_1 + 1$. Therefore,

$$\begin{aligned}
\boldsymbol{h}(\alpha) &= \tilde{\boldsymbol{D}}^{\frac{1}{2}}\sigma\Big(\boldsymbol{x} - \frac{\alpha}{c}\boldsymbol{u}_n\Big)\\
&= \tilde{\boldsymbol{D}}^{\frac{1}{2}}[\epsilon, \ldots, \epsilon, 0, \ldots, 0]^\top\\
&= [\epsilon\sqrt{d_1}, \ldots, \epsilon\sqrt{d_{k_1}}, 0, \ldots, 0]^\top.
\end{aligned}$$

Hence we have $\|\boldsymbol{h}(\alpha)\| = \epsilon\sqrt{\sum_{i=1}^{k_1} d_i}$. Also, we have

$$\begin{aligned}
\|\boldsymbol{h}(\alpha)\|_{\mathcal{M}} &= |\langle \boldsymbol{h}(\alpha), \boldsymbol{e}\rangle|\\
&= [\epsilon\sqrt{d_1}, \ldots, \epsilon\sqrt{d_{k_1}}, 0, \ldots, 0]^\top [\sqrt{d_1}/c, \sqrt{d_2}/c, \ldots, \sqrt{d_n}/c]\\
&= \frac{\epsilon}{c}\sum_{i=1}^{k_1} d_i.
\end{aligned}$$

Then we obtain the smoothness $s(\boldsymbol{h}(\alpha))$ as follows

$$s(\boldsymbol{h}(\alpha)) = \frac{\|\boldsymbol{h}(\alpha)\|_{\mathcal{M}}}{\|\boldsymbol{h}(\alpha)\|} = \frac{\frac{\epsilon}{c}\sum_{i=1}^{k_1} d_i}{\epsilon\sqrt{\sum_{i=1}^{k_1} d_i}} = \frac{\sqrt{\sum_{i=1}^{k_1} d_i}}{c} = \frac{K_1}{c} < 1,$$

where $K_1 \coloneqq \sqrt{\sum_{i=1}^{k_1} d_i}$. Similarly, we may denote $\sqrt{\sum_{i=k_{j-1}+1}^{k_j} d_i}$ by K_j for $j = 2, \ldots, l$.

Now we are going to show that the smoothness $s(\boldsymbol{h}(\alpha))$ is increasing as α gets smaller whenever $\alpha < cx_1$, implying $\frac{K_1}{c}$ is the minimum of the smoothness $s(\boldsymbol{h}(\alpha))$. Remember that we are considering $\alpha = c(x_1 - \epsilon)$ and we have studied the case when $0 < \epsilon \leq x_1 - x_{k_1+1} = x_1 - x_{k_2}$.

Let $\delta_j \coloneqq x_1 - x_{k_j}$ for $1 \leq j \leq l$. Clearly, we have $\delta_1 = 0$ and $\delta_j < \delta_{j+1}$ for $1 \leq j \leq l-1$. Fix a $j' \in \{2, \ldots, l-1\}$, we see that when $\delta_{j'} < \epsilon \leq x_1 - x_{k_{j'}+1}$,

$$\begin{aligned}
&\boldsymbol{x} - \frac{\alpha}{c}\boldsymbol{u}_n \\
&= \Big[\epsilon - \delta_1, \ldots, \epsilon - \delta_1, \epsilon - \delta_2, \ldots, \epsilon - \delta_2, \epsilon - \delta_3, \\
&\qquad \ldots, \epsilon - \delta_{j'}, \epsilon - (x_1 - x_{k_{j'}+1}), \ldots, \epsilon - (x_1 - x_n)\Big]^\top,
\end{aligned}$$

where we have $\epsilon - \delta_j > 0$ for $2 \leq j \leq j'$ and $\epsilon - (x_1 - x_i) \leq 0$ for any $i \geq k_{j'} + 1$. Consequently,

$$\begin{aligned}
\boldsymbol{h}(\alpha) &= \tilde{\boldsymbol{D}}^{\frac{1}{2}} \sigma(\boldsymbol{x} - \frac{\alpha}{c}\boldsymbol{u}_n) \\
&= [(\epsilon - \delta_1)\sqrt{d_1}, \ldots, (\epsilon - \delta_1)\sqrt{d_{k_1}}, (\epsilon - \delta_2)\sqrt{d_{k_1+1}}, \ldots, \\
&\qquad (\epsilon - \delta_2)\sqrt{d_{k_2}}, (\epsilon - \delta_3)\sqrt{d_{k_2+1}}, \ldots, (\epsilon - \delta_{j'})\sqrt{d_{k_{j'}}}, 0, \ldots, 0]^\top.
\end{aligned}$$

Then we can compute

$$\|\boldsymbol{h}(\alpha)\| = \sqrt{\sum_{j=1}^{j'} \sum_{i=k_{j-1}+1}^{k_j} d_i(\epsilon - \delta_j)^2} = \sqrt{\sum_{j=1}^{j'} K_j^2(\epsilon - \delta_j)^2},$$

where we set $k_0 \coloneqq 0$ for simplicity and $K_j = \sqrt{\sum_{i=k_{j-1}+1}^{k_j} d_i}$ for $j = 1, \ldots, j'$. Also, we have

$$\begin{aligned}
\|\boldsymbol{h}(\alpha)\|_{\mathcal{M}} = |\langle \boldsymbol{h}(\alpha), \boldsymbol{e}\rangle| &= \sum_{j=1}^{j'} \sum_{i=k_{j-1}+1}^{k_j} \frac{d_i(\epsilon - \delta_j)}{c} \\
&= \frac{1}{c}\sum_{j=1}^{j'} K_j^2(\epsilon - \delta_j).
\end{aligned}$$

A careful calculation shows that $\frac{\partial}{\partial \epsilon} s(\boldsymbol{h}(\alpha)) > 0$ whenever $\delta_{j'} < \epsilon \leq x_1 - x_{k_{j'}+1}$ which implies that $s(\boldsymbol{h}(\alpha))$ is increasing as ϵ increases. Indeed, we have

$$\begin{aligned}
&\frac{\partial}{\partial \epsilon} s(\boldsymbol{h}(\alpha)) \\
=&\frac{\partial}{\partial \epsilon}\left(\frac{\sum_{j=1}^{j'} K_j^2(\epsilon - \delta_j)}{c\sqrt{\sum_{j=1}^{j'} K_j^2(\epsilon - \delta_j)^2}}\right) \\
=&\frac{\Big(\frac{\partial}{\partial \epsilon}\sum_{j=1}^{j'} K_j^2(\epsilon - \delta_j)\Big)\sqrt{\sum_{j=1}^{j'} K_j^2(\epsilon - \delta_j)^2}}{c\sum_{j=1}^{j'} K_j^2(\epsilon - \delta_j)^2} - \\
&- \frac{\sum_{j=1}^{j'} K_j^2(\epsilon - \delta_j)\Big(\frac{\partial}{\partial \epsilon}\sqrt{\sum_{j=1}^{j'} K_j^2(\epsilon - \delta_j)^2}\Big)}{c\sum_{j=1}^{j'} K_j^2(\epsilon - \delta_j)^2}
\end{aligned}$$

$$=\frac{\Big(\sum_{j=1}^{j'} K_j^2\Big)\sum_{j=1}^{j'} K_j^2(\epsilon - \delta_j)^2 - \sum_{j=1}^{j'} K_j^2(\epsilon - \delta_j)\Big(\sum_{j=1}^{j'} K_j^2(\epsilon - \delta_j)\Big)}{c\sum_{j=1}^{j'} K_j^2(\epsilon - \delta_j)^2\sqrt{\sum_{j=1}^{j'} K_j^2(\epsilon - \delta_j)^2}}.$$

Then to show that $\frac{\partial}{\partial \epsilon} s(\boldsymbol{h}(\alpha)) > 0$, it suffices to show that the numerator is positive, i.e.

$$\Big(\sum_{j=1}^{j'} K_j^2\Big) \sum_{j=1}^{j'} K_j^2 (\epsilon - \delta_j)^2 - \Big(\sum_{j=1}^{j'} K_j^2 (\epsilon - \delta_j)\Big)^2 > 0,$$

since $c \sum_{j=1}^{j'} K_j^2 (\epsilon - \delta_j)^2 \sqrt{\sum_{j=1}^{j'} K_j^2 (\epsilon - \delta_j)^2} > 0$ is always positive. In fact, this follows from the Cauchy inequality $\|\boldsymbol{v}\|\|\boldsymbol{u}\| \geq \langle \boldsymbol{v}, \boldsymbol{u} \rangle$, where we set

$$\begin{aligned} \boldsymbol{v} &\coloneqq [K_1, K_2, \ldots, K_{J'}]^\top, \\ \boldsymbol{u} &\coloneqq [K_1(\epsilon - \delta_1), K_2(\epsilon - \delta_2), \ldots, K_{j'}(\epsilon - \delta_{j'})]^\top. \end{aligned}$$

Moreover, equality happens only when $\boldsymbol{v}$ is parallel to $\boldsymbol{u}$. This is, however, impossible since $\epsilon - \delta_j > \epsilon - \delta_{j+1}$ for any $j = 1, \ldots, j' - 1$ and each K_j is positive.

So we see that $s(\boldsymbol{h}(\alpha))$ is increasing as ϵ increases whenever $0 < \epsilon$, and hence the smoothness $s(\boldsymbol{h}(\alpha))$ is increasing as α decreases whenever $cx_n \leq \alpha < cx_1$.

For the case $j' = l$ where $\delta_l = x_1 - x_n < \epsilon$, we have $x_n - \alpha/c = x_n - (x_1 - \epsilon) = \epsilon - (x_1 - x_n) > 0$, implying $\alpha < cx_n$ and $\boldsymbol{h}(\alpha) = \boldsymbol{z}(\alpha)$. We have shown that the smoothness is increasing as α is going far from $\langle \boldsymbol{z}, \boldsymbol{e} \rangle$; in particular, when $\alpha < \langle \boldsymbol{z}, \boldsymbol{e} \rangle$ and α is decreasing. One can check that

$$\begin{aligned} cx_n = \frac{\sum_{i=1}^n d_i x_n}{c} &= \left\langle x_n \boldsymbol{u}_n, \frac{\tilde{\boldsymbol{D}} \boldsymbol{u}_n}{c} \right\rangle \\ &\leq \left\langle \boldsymbol{x}, \frac{\tilde{\boldsymbol{D}} \boldsymbol{u}_n}{c} \right\rangle \\ &= \left\langle \tilde{\boldsymbol{D}}^{\frac{1}{2}} \boldsymbol{x}, \frac{\tilde{\boldsymbol{D}}^{\frac{1}{2}} \boldsymbol{u}_n}{c} \right\rangle \\ &= \langle \boldsymbol{z}, \boldsymbol{e} \rangle, \end{aligned}$$

which means the smoothness is increasing as α decreases whenever $\alpha < cx_n$.

We conclude that the smoothness increases as α decreases, provided $\alpha < cx_1$. Also, we have $\sup_{\alpha < cx_1} s(\boldsymbol{h}(\alpha)) = 1$ as the case in the proof of Proposition B.1. One can check that $s(\boldsymbol{h}(\alpha))$ is a continuous function for $\alpha < cx_1$ and thus it has range $[K_1/c, 1)$ by the mean value theorem.

Finally, we can establish the result: $K_1/c = \sqrt{\frac{\sum_{x_i = \max \boldsymbol{x}} d_i}{\sum_{j=1}^n d_j}}$ is the minimum of $s(\boldsymbol{h}(\alpha))$ and 1 is the maximum of $s(\boldsymbol{h}(\alpha))$ occurring whenever $\alpha \geq cx_1 = \sqrt{\sum_{j=1}^n d_j} \max_i x_i$. Moreover, $s(\boldsymbol{h}(\alpha))$ has a monotone property when $\alpha < \sqrt{\sum_{j=1}^n d_j} \max_i x_i$ and has range $\Big[\sqrt{\frac{\sum_{x_i = \max \boldsymbol{x}} d_i}{\sum_{j=1}^n d_j}}, 1\Big]$.

It is clear that the assumption on the ordering of the entries of $\boldsymbol{x}$ will not affect this result. □

To prove Proposition 4.4, we first prove an analogous result for the identity function, that is, $\boldsymbol{h} = \sigma(\boldsymbol{z}) = \boldsymbol{z}$.

Proposition B.1. *Suppose $\boldsymbol{z}_{\mathcal{M}^\perp} \neq \boldsymbol{0}$, then $s(\boldsymbol{z}(\alpha))$ achieves its minimum 0 if $\alpha = \langle \boldsymbol{z}, \boldsymbol{e} \rangle$. Moreover, $\sup_\alpha s(\boldsymbol{z}(\alpha)) = 1$ where $s(\boldsymbol{z}(\alpha))$ is close to 1 when α is far away from $\langle \boldsymbol{z}, \boldsymbol{e} \rangle$.*

Proof of Proposition B.1. We know that $0 \leq s(\boldsymbol{z}(\alpha)) \leq 1$ and

$$\begin{aligned} s(\boldsymbol{z}(\alpha)) &= \sqrt{1 - \frac{\|\boldsymbol{z}_{\mathcal{M}^\perp}\|^2}{\|\boldsymbol{z}(\alpha)\|^2}} \\ &= \sqrt{1 - \frac{\|\boldsymbol{z}_{\mathcal{M}^\perp}\|^2}{\|\boldsymbol{z}_{\mathcal{M}^\perp}\|^2 + \|\boldsymbol{z}(\alpha)_{\mathcal{M}}\|^2}} \\ &= \sqrt{1 - \frac{\|\boldsymbol{z}_{\mathcal{M}^\perp}\|^2}{\|\boldsymbol{z}_{\mathcal{M}^\perp}\|^2 + \|\boldsymbol{z}_{\mathcal{M}} - \alpha\boldsymbol{e}\|^2}}. \end{aligned}$$

Suppose $s(\boldsymbol{z}(\alpha)) = 1$. Then we have $\frac{\|\boldsymbol{z}_{\mathcal{M}^\perp}\|^2}{\|\boldsymbol{z}_{\mathcal{M}^\perp}\|^2 + \|\boldsymbol{z}_{\mathcal{M}} - \alpha\boldsymbol{e}\|^2} = 0$ which forces $\|\boldsymbol{z}_{\mathcal{M}^\perp}\| = 0$. However, this contradicts the hypothesis $\boldsymbol{z}_{\mathcal{M}^\perp} \neq 0$. So $s(\boldsymbol{z}(\alpha))$ cannot attain its maximum.

But for any $0 \leq t < 1$, $s(\boldsymbol{z}(\alpha)) = t$ if and only if

$$\begin{aligned} &\sqrt{1 - \frac{\|\boldsymbol{z}_{\mathcal{M}^\perp}\|^2}{\|\boldsymbol{z}_{\mathcal{M}^\perp}\|^2 + \|\boldsymbol{z}_{\mathcal{M}} - \alpha\boldsymbol{e}\|^2}} = t \\ \Leftrightarrow &\frac{\|\boldsymbol{z}_{\mathcal{M}^\perp}\|^2}{\|\boldsymbol{z}_{\mathcal{M}^\perp}\|^2 + \|\boldsymbol{z}_{\mathcal{M}} - \alpha\boldsymbol{e}\|^2} = 1 - t^2 \\ \Leftrightarrow &\|\boldsymbol{z}_{\mathcal{M}^\perp}\|^2 = (1-t^2)\big(\|\boldsymbol{z}_{\mathcal{M}^\perp}\|^2 + \|\boldsymbol{z}_{\mathcal{M}} - \alpha\boldsymbol{e}\|^2\big) \\ \Leftrightarrow &t^2\|\boldsymbol{z}_{\mathcal{M}^\perp}\|^2 = (1-t^2)\|\boldsymbol{z}_{\mathcal{M}} - \alpha\boldsymbol{e}\|^2 \\ \Leftrightarrow &\|\boldsymbol{z}_{\mathcal{M}} - \alpha\boldsymbol{e}\| = \sqrt{\frac{t^2}{1-t^2}} \cdot \|\boldsymbol{z}_{\mathcal{M}^\perp}\| \end{aligned}$$

This implies that $\sup_\alpha s(\boldsymbol{z}(\alpha)) = 1$ and $s(\boldsymbol{z}(\alpha))$ achieves its minimum 0 if and only if $\alpha = \langle \boldsymbol{z}, \boldsymbol{e}\rangle$. It is clear that $s(\boldsymbol{z}(\alpha))$ get closer to 1 when α is going far away from $\langle \boldsymbol{z}, \boldsymbol{e}\rangle$. i.e., $|\alpha - \langle \boldsymbol{z}, \boldsymbol{e}\rangle| = \|\boldsymbol{z}_{\mathcal{M}} - \alpha\boldsymbol{e}\|$ is increasing. □

Proof of Proposition 4.4. First, we notice that leaky ReLU has the following two properties

1. $\sigma_a(x) > 0$ for $x \gg 0$ and $\sigma_a(x) < 0$ for $x \ll 0$.
2. σ_a is a non-trivial linear map for $x \gg 0$.

We will use Property 1 to show that $\min_\alpha s(\boldsymbol{h}(\alpha)) = 0$ and Property 2 to show that $\sup_\alpha s(\boldsymbol{h}(\alpha)) = 1$. Notice that $\sigma_a(x) < 0$ for $x \ll 0$ implies that there exists a sufficient small $\alpha_2 < 0$ s.t. all of the entries of $\boldsymbol{h}(\alpha_2)$ are negative and hence $|\langle \boldsymbol{h}(\alpha_2), \boldsymbol{e}\rangle| < 0$. Similarly, $\sigma_a(x) > 0$ for $x \gg 0$ implies that there exists a sufficient large $\alpha_1 > 0$ s.t. all of the entries of $\boldsymbol{h}(\alpha_1)$ are positive and hence $|\langle \boldsymbol{h}(\alpha_1), \boldsymbol{e}\rangle| > 0$. Since $|\langle \boldsymbol{h}(\alpha), \boldsymbol{e}\rangle|$ is a continuous function of α on $[\alpha_1, \alpha_2]$, the Intermediate Value Theorem follows that there exists an $\alpha \in (\alpha_1, \alpha_2)$ s.t. $|\langle \boldsymbol{h}(\alpha), \boldsymbol{e}\rangle| = 0$. Thus by definition $s(\boldsymbol{h}(\alpha)) = |\langle \boldsymbol{h}(\alpha), \boldsymbol{e}\rangle|/\|\boldsymbol{h}(\alpha)\|$, we see that $\min_\alpha s(\boldsymbol{h}(\alpha)) = 0$.

On the other hand, since σ_a is a non-trivial linear map for $x \gg 0$, we may assume $\sigma_a(x) = cx$ for $x > x_0$ where $c \neq 0$ is some non-zero constant and $x_0 > 0$ is some positive constant. Then we can choose an $\alpha_0 > \langle \boldsymbol{z}, \boldsymbol{e}\rangle$ s.t. for any $\alpha \geq \alpha_0$, all of the entries of $\boldsymbol{z}(\alpha)$ are greater than x_0. Then whenever $\alpha \geq \alpha_0$, we have $\boldsymbol{h}(\alpha) = \sigma_a(\boldsymbol{z}(\alpha)) = c\boldsymbol{z}(\alpha)$. This implies

$$s(\boldsymbol{h}(\alpha)) = \frac{|\langle \boldsymbol{h}(\alpha), \boldsymbol{e}\rangle|}{\|\boldsymbol{h}(\alpha)\|} = \frac{|\langle c\boldsymbol{z}(\alpha), \boldsymbol{e}\rangle|}{\|c\boldsymbol{z}(\alpha)\|} = \frac{|\langle \boldsymbol{z}(\alpha), \boldsymbol{e}\rangle|}{\|\boldsymbol{z}(\alpha)\|} = s(\boldsymbol{z}(\alpha)).$$

Thus $\sup_\alpha s(\boldsymbol{h}(\alpha)) = 1$ follows from the Proof of Proposition B.1 where we see that $\sup_\alpha s(\boldsymbol{z}(\alpha)) = 1$ since $s(\boldsymbol{z}(\alpha))$ gets closer to 1 as α increases.

□

Remark B.2. Indeed, it holds for any continuous function $f : \mathbb{R} \to \mathbb{R}$ satisfying the following

1. $f(x) > 0$ for $x \gg 0$, $f(x) < 0$ for $x \ll 0$ or $f(x) < 0$ for $x \gg 0$, $f(x) > 0$ for $x \ll 0$,
2. f is a non-trivial linear map for $x \gg 0$ or $x \ll 0$.

One can check that the proof above only depends on these two properties. It is worth mentioning that most activation functions, e.g., leaky LU, SiLU, tanh, satisfy condition 1.

Proof of Corollary 4.5. For any α, we notice that $\|\boldsymbol{z}\|_{\mathcal{M}^\perp} = \|\boldsymbol{z}_{\mathcal{M}^\perp}\|_F = \|\boldsymbol{z}(\alpha)\|_{\mathcal{M}^\perp}$ since α only changes the component of $\boldsymbol{z}$ in the eigenspace $\mathcal{M}$. Also, Propositions 3.2 and 3.3 show that $\|\boldsymbol{z}(\alpha)\|_{\mathcal{M}^\perp} \geq \|\boldsymbol{h}(\alpha)\|_{\mathcal{M}^\perp}$ whenever $\boldsymbol{h}(\alpha) = \sigma(\boldsymbol{z}(\alpha))$ or $\sigma_a(\boldsymbol{z}(\alpha))$. Therefore, $\|\boldsymbol{z}\|_{\mathcal{M}^\perp} \geq \|\boldsymbol{h}(\alpha)\|_{\mathcal{M}^\perp}$ holds for any α. Since $\boldsymbol{z}_{\mathcal{M}^\perp} \neq 0$, $s(\boldsymbol{z})$ must lie in $[0, 1)$. □

C Additional Experimental Details

This section provides additional experimental details and results for Section 6. All tasks were run on Nvidia RTX 3090, GV100, and Tesla T4 GPUs. Computational performance metrics, including timing, were measured using Tesla T4 GPUs on Colab.

	# Nodes	**# Edges**	**# Features**	**# Classes**	Splits (Train/Val/Test)
Cornell	183	295	1,703	5	48/32/20%
Texas	181	309	1,703	5	48/32/20%
Wisconsin	251	499	1,703	5	48/32/20%
Chameleon	2,277	36,101	2,325	5	48/32/20%
Squirrel	5,201	217,073	2,089	5	48/32/20%
Citeseer	3,727	4,732	3,703	6	120/500/1000
Cora	2,708	5,429	1,433	7	140/500/1000
PubMed	19,717	44,338	500	3	60/500/1000
Coauthor-Physics	34,493	247,962	8415	5	100/150/34,243
Ogbn-arxiv	169,343	1,166,243	128	40	90,941/29,799/48,603

Table 4: Graph statistics.

C.1 Dataset details

Table 4 presents additional graph statistics.

Citation Datasets: We use Cora, Citeseer, PubMed, Coauthor-Physics, and Ogbn-arxiv, where each dataset is a graph with nodes representing academic publications, features as bag-of-words vectors, labels indicating publication types, and edges denoting citations.

Web Knowledge-Base Datasets: We use the Cornell, Texas, and Wisconsin datasets, each represented as a graph where nodes are CS department webpages, features are bag-of-words vectors, edges denote hyperlinks, and labels indicate webpage types.

Wikipedia Network Datasets: We use the Chameleon and Squirrel datasets, where each graph represents CS department webpages as nodes, with bag-of-words features, hyperlink edges, and labels indicating webpage types.

C.2 Model size and computational time for citation datasets

Table 5 compares the model size and computational time for experiments on citation datasets in Section 6.2.

	# Parameters	Training Time (s)	Inference Time (ms)
Cora			
GCN	100,423	8.4	1.6
GCNII	110,535	10.0	2.1
GCNII	708,743	57.6	12.3
GCNII-SCT	1,237,127	110.3	29.6
EGNN	712,839	65.6	14.4
EGNN-SCT	316,551	24.8	4.5
Citeseer			
GCN	245,638	8.3	1.5
GCN-SCT	301,830	15.5	4.0
GCNII	999,174	57.6	12.3
GCNII-SCT	1,001,222	65.9	15.7
EGNN	739,078	39.6	7.2
EGNN-SCT	540,934	24.0	5.8
PubMed			
GCN	40,451	9.0	1.8
GCN-SCT	40,707	11.1	2.2
GCNII	326,659	98.2	12.8
GCNII-SCT	590,851	71.7	17.4
EGNN	592,899	93.7	2.5
EGNN-SCT	130,563	16.0	3.1
Coauthor-Physics			
GCN	547,141	35.2	8.0
GCN-SCT	547,397	33.9	8.3
GCNII	555,333	49.1	10.3
GCNII-SCT	555,461	67.0	9.5
EGNN	672,069	176.4	47.9
EGNN-SCT	572,229	51.7	14.8
Ogbn-arxiv			
GCN	27,240	50.4	21.1
GCN-SCT	28,392	62.6	24.4
GCNII	76,392	205.4	94.8
GCNII-SCT	80,616	253.0	108.9
EGNN	77,416	206.8	98.0
EGNN-SCT	81,640	254.0	112.3

Table 5: Number of model parameters for varying numbers of layers using the optimal model hyperparameters. The SCT is added at each layer, and the size of the additional parameters scales with the number of eigenvectors with an eigenvalue of one for matrix $\boldsymbol{G}$ in equation 2.

Parameter	**Values**
Learning Rate	$\{1e\text{-}4, 1e\text{-}3, 1e\text{-}2\}$
Weight Decay (FC)	$\{0, 1e\text{-}4, 5e\text{-}4, 1e\text{-}3, 5e\text{-}3, 1e\text{-}2\}$
Weight Decay (Conv)	$\{0, 1e\text{-}4, 5e\text{-}4, 1e\text{-}3, 5e\text{-}3, 1e\text{-}2\}$
Dropout	$\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$
Hidden Channels	$\{16, 32, 64, 128\}$
GCNII-α	$\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$
GCNII-θ	$\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$
EGNN-$c_{\max}$	$\{0.5, 1.0, 1.5, 2.0\}$
EGNN-α	$\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$
EGNN-θ	$\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$

Table 6: Hyperparameter grid search for Table 1.

C.3 Additional Section 6.2 details for citation datasets

Table 6 lists the hyperparameters used in the grid search for generating the results in Table 1. Table 8 reports classification accuracy across model depths using ReLU and leaky ReLU activations.

Layers	2	4	16	32
Cora				
EGNN/EGNN-SCT	83.2/**83.4**	84.2/**84.3**	85.4/**85.5**	85.3/**85.5**
Citeseer				
EGNN/EGNN-SCT	72.0/**72.1**	71.9/**72.3**	72.4/**72.6**	72.3/**72.8**
PubMed				
EGNN/EGNN-SCT	79.2/**79.4**	79.5/**79.8**	**80.1**/**80.1**	80.0/**80.2**
Coauthor-Physics				
EGNN/EGNN-SCT	92.6/**92.8**	92.9/**93.0**	93.1/**93.3**	**93.3**/**93.3**
Ogbn-arxiv				
EGNN/EGNN-SCT	68.4/**68.5**	71.1/**71.3**	72.7/**73.0**	72.7/**72.9**

Table 7: Test accuracy for EGNN and EGNN-SCT using the SReLU activation function of varying depth on citation networks with the split discussed in Section 6.2. (Unit:%)

C.3.1 Vanishing gradients

Figure 4 illustrates the vanishing gradient problem in training deep GCNs—with and without SCT—compared to GCNII and EGNN. It plots $||\partial \boldsymbol{H}^{\text{out}}/\partial \boldsymbol{H}^{l}||$ for layers $l \in [0, 32]$ over 100 training epochs. Subfigures (a) and (b) show that GCN and GCN-SCT suffer from vanishing gradients. In contrast, (c) and (e) demonstrate that GCNII and EGNN avoid this issue by connecting $\boldsymbol{H}^0$ to every layer, maintaining nonzero gradients in early layers. Interestingly, adding SCT to GCNII and EGNN (shown in (d) and (f)) amplifies intermediate gradients as training progresses.

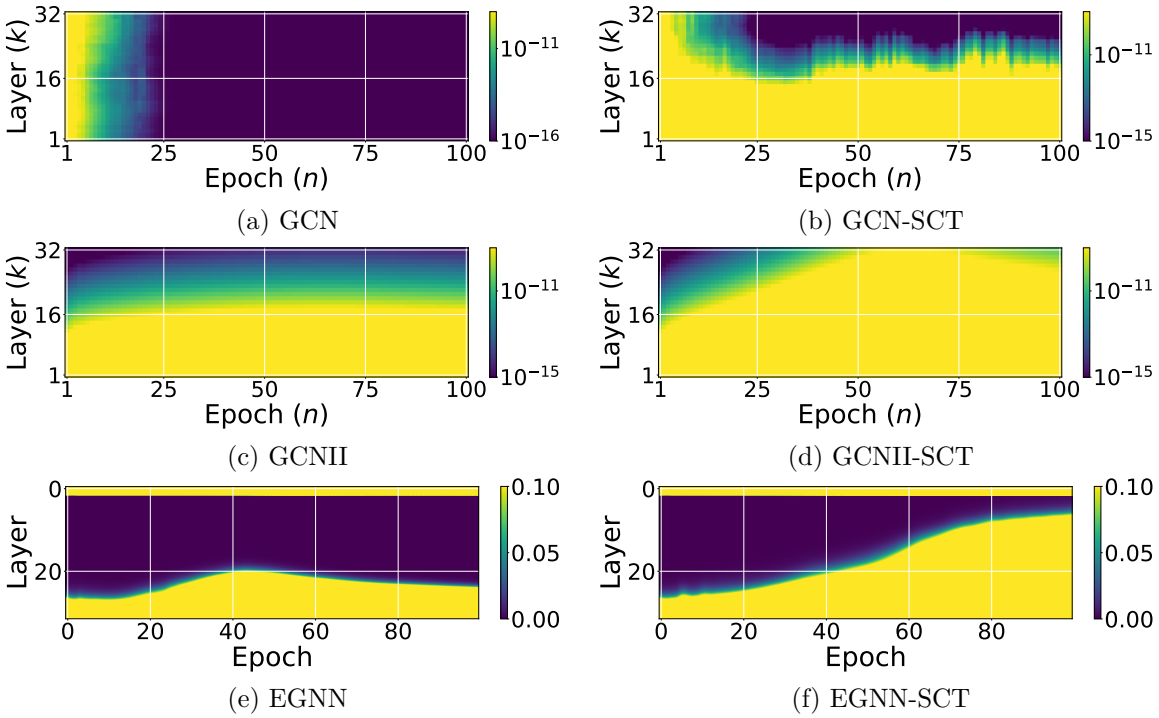


Figure 4: We visualize training gradients $||\partial \boldsymbol{H}^{\text{out}}/\partial \boldsymbol{H}^{l}||$ for layers $l \in [0, 32]$ over 100 epochs on the Citeseer dataset. All models use 32 layers with 16 hidden dimensions. Subfigure (a) shows that GCN suffers from vanishing gradients, while (c) GCNII and (e) EGNN maintain nonzero gradients due to skip connections to $\boldsymbol{H}^0$. Although SCT does not resolve vanishing gradients in (b) GCN-SCT, it increases intermediate gradient norms in (d) GCNII-SCT and (f) EGNN-SCT as training progresses.

Cora								
	ReLU				leaky ReLU			
Layers	2	4	16	32	2	4	16	32
GCN-SCT	81.2	80.3	71.4	67.2	82.9	82.8	68.0	65.5
GCNII-SCT	83.5	83.8	82.7	83.3	83.8	84.8	84.8	85.5
EGNN-SCT	84.1	83.8	82.3	80.8	83.7	84.5	83.3	82.0
Citeseer								
	ReLU				leaky ReLU			
Layers	2	4	16	32	2	4	16	32
GCN-SCT	69.0	67.3	51.5	50.3	69.9	67.7	55.4	51.0
GCNII-SCT	72.8	72.8	72.8	73.3	72.8	72.9	73.8	72.7
EGNN-SCT	72.5	72.0	70.2	71.8	73.1	71.7	72.6	72.9
PubMed								
	ReLU				leaky ReLU			
Layers	2	4	16	32	2	4	16	32
GCN-SCT	79.4	78.2	75.9	77.0	79.8	78.4	76.1	76.9
GCNII-SCT	79.7	80.1	80.7	80.7	79.6	80.0	80.3	80.7
EGNN-SCT	79.7	80.1	80.0	80.4	79.8	80.4	80.3	80.2
Coauthor-Physics								
	ReLU				leaky ReLU			
Layers	2	4	16	32	2	4	16	32
GCN-SCT	91.8 ± 1.6	91.6 ± 3.0	44.5 ± 13.0	42.6 ± 17.0	92.6 ± 1.6	92.5 ± 5.9	50.9 ± 15.0	43.6 ± 16.0
GCNII-SCT	94.4 ± 0.4	93.5 ± 1.2	93.7 ± 0.7	93.8 ± 0.6	94.0 ± 0.4	94.2 ± 0.3	93.3 ± 0.7	94.1 ± 0.3
EGNN-SCT	93.6 ± 0.7	94.1 ± 0.4	93.4 ± 0.8	93.8 ± 1.3	93.9 ± 0.7	94.0 ± 0.7	94.0 ± 0.7	93.3 ± 0.9
Ogbn-arxiv								
	ReLU				leaky ReLU			
Layers	2	4	16	32	2	4	16	32
GCN-SCT	71.7 ± 0.3	72.6 ± 0.3	71.4 ± 0.2	71.9 ± 0.3	72.1 ± 0.3	72.7 ± 0.3	72.3 ± 0.2	72.3 ± 0.3
GCNII-SCT	71.4 ± 0.3	72.1 ± 0.3	72.2 ± 0.2	71.8 ± 0.2	72.0 ± 0.3	72.2 ± 0.2	72.4 ± 0.3	72.1 ± 0.3
EGNN-SCT	68.5 ± 0.6	71.0 ± 0.5	72.8 ± 0.5	72.1 ± 0.6	67.7 ± 0.5	71.3 ± 0.5	72.3 ± 0.5	72.3 ± 0.5

Table 8: Test accuracy for models of varying depth using ReLU or leaky ReLU activation on citation network datasets, based on the split described in Section 6.2.

C.4 Additional Section 6.2 details for other datasets

Table 9 reports mean test accuracy and standard deviation over 10 folds for SCT-based models on the WebKB and WikipediaNetwork datasets. Table 10 shows average per-epoch time for models with 8 layers. These results indicate that integrating SCT incurs minimal computational overhead.

	Cornell	**Texas**	**Wisconsin**	**Chameleon**	**Squirrel**
GCN-SCT	55.95 ± 8.5	62.16 ± 5.7	54.71 ± 4.4	38.44 ± 4.3	35.31 ± 1.9
GCNII-SCT	75.41 ± 2.2	83.34 ± 4.5	86.08 ± 3.8	64.52 ± 2.2	47.51 ± 1.4

Table 9: Mean $\pm$ standard deviation test accuracy from 10-fold cross-validation on five heterophilic datasets with fixed 48/32/20

	Cornell	**Texas**	**Wisconsin**	**Chameleon**	**Squirrel**
GCN Kipf & Welling (2017)	0.011	0.013	0.012	0.011	0.022
GCNII Chen et al. (2020b)	0.017	0.018	0.017	0.013	0.022
GCN-SCT	0.015	0.017	0.015	0.011	0.023
GCNII-SCT	0.017	0.018	0.017	0.020	0.025

Table 10: Average per-epoch computation time on five heterophilic datasets with fixed 48/32/20% splits. All models use 8 layers with 16 hidden channels. (Unit: second)