On Asymptotic Behaviors of Graph CNNs from Dynamical Systems Perspective

Kenta Oono
The University of Tokyo
Preferred Networks, Inc.
Tokyo, Japan
kenta_oono@mist.i.u-tokyo.ac.jp

Taiji Suzuki
The University of Tokyo
RIKEN Center for Advanced Intelligence Project
Tokyo, Japan
taiji@mist.i.u-tokyo.ac.jp

Abstract

Graph Convolutional Neural Networks (graph CNNs) are a promising deep learning approach for analyzing graph-structured data. However, it is known that they do not improve (or sometimes worsen) their predictive performance as we pile up more layers and make them deeper. To tackle this problem, we investigate the expressive power of graph CNNs by analyzing their asymptotic behaviors as the layer size tends to infinity. Our strategy is to generalize the forward propagation of a Graph Convolutional Network (GCN), which is one of the most popular graph CNN variants, as a specific dynamical system. In the case of GCNs, we show that when the weights satisfy the conditions determined by the spectra of the (augmented) normalized Laplacian, the output of GCNs exponentially approaches the set of signals that carry only information of the connected components and node degrees for distinguishing nodes. Our theory enables us to directly relate the expressive power of GCNs with the topological information of the underlying graphs, which is inherent in the graph spectra. To demonstrate this, we characterize the asymptotic behavior of GCNs on the Erdős – Rényi graph. We show that when the Erdős – Rényi graph is sufficiently dense and large, a wide range of GCNs on them suffers from this “information loss" in the limit of infinite layers with high probability. Furthermore, our theory provides principled guidelines for the weight normalization of graph CNNs. We experimentally confirmed that weight scaling based on our theory enhanced the predictive performance of GCNs in real data.

1 Introduction

Motivated by the success of Deep Learning (DL), several attempts have been made to apply DL models to non-Euclidean data, particularly, graph-structured data such as chemical compounds, social networks, and polygons. Recently, *Graph Convolutional Neural Networks* (graph CNNs) [5, 15, 19, 23, 30, 33, 39, 45, 54, 56] have emerged as a promising approach. They have outperformed conventional machine-learning methods in the application of chemical compounds [15], knowledge graphs [45], and scene graphs [57], to name a few; see surveys [55, 61, 62] for recent advances. However, despite their practical popularity, theoretical research of graph CNNs has been less explored.

The characterization of DL model *expressive power*, i.e. , to identify what function classes DL models can (approximately) represent, is a fundamental question in theoretical research of DL. Many studies have been conducted for Fully Connected Neural Networks (FNNs) [4, 13, 26, 27, 38, 47, 58] and Convolutional Neural Networks (CNNs) [41, 43, 63]. For such models, we have theoretical and empirical justification for deep and non-linear architectures: DL models can enhance representation power by stacking many layers and adding non-linear functions (activation functions) in-between [7, 49, 64]. However, the situation seems to be different for graph CNNs. Several papers have

Preprint. Under review.

reported severe performance degradation when stacking many layers [30, 55]. For non-linearity, [54] reported that graph CNNs achieved comparable performance even if they lacked intermediate non-linear functions. These studies posed a question about the current architecture and made us aware of theoretical analysis for the expressive power of graph CNNs.

In this paper, we investigated the expressive power of graph CNNs by analyzing their asymptotic behaviors as the layer size goes to infinity. Our theorems give new theoretical conditions under which neither layer stacking nor non-linearity contribute to improving of the expressive power. We consider a specific dynamics that have a transition defining a Markov process and the forward propagation of a Graph Convolutional Network (GCN) [30], which is one of the most popular graph CNN variants, as special cases. We prove that under certain conditions, the dynamics exponentially shrink to a subspace that is invariant under the dynamics. In the case of GCN, the invariant space is a set of signals that have "no information" other than connected components and node degrees for distinguishing nodes. In addition, the distance between the dynamics and the invariant space exponentially decreases at the rate of $O((s\lambda)^L)$ where s is the maximum singular values of weights, λ is a value determined by the spectra of the (augmented) normalized Laplacian of the underlying graphs, and L is the layer size. Our theorem implies that under certain conditions, GCNs asymptotically lose information for node classification tasks where each sample is represented as a node in a graphs and the goal is to predict the nodes' properties. See Sections 3.3 (general case) and 4 (GCN case) for precise statements.

We can interpret our theorem as the generalization of the well-known property of Markov processes that if a finite and discrete Markov process is irreducible and aperiodic, it exponentially converges to a unique equilibrium and the convergence rate is determined by the eigenvalues of its transition matrix (see, e.g., [11]). However, as opposed to the Markov process case, which is a linear dynamics, the existence of intermediate non-linear functions complicates the analysis, as with other DL models. We overcame this problem by leveraging the combination of the ReLU activation function [32] and the positivity of eigenvectors of the Laplacian associated with the smallest eigenvalues.

Our theory enables us to investigate the asymptotic behavior of graph CNNs via the spectral distribution of the underlying graphs. To demonstrate this, we take GCNs defined on the Erdős – Rényi graph $G_{N,p}$, which has N nodes and each edge appears independently with probability p, as an example. We prove that if $\frac{\log N}{pN} = o(1)$ as a function of N, any GCN whose weights have maximum singular values at most $C\sqrt{\frac{Np}{\log(N/\varepsilon)}}$ approaches the "information-less" invariant space with probability at least $1-\varepsilon$, where C is a universal constant. Intuitively, if the graph on which graph CNNs are defined is sufficiently dense, graph convolution operations mix signals on nodes fast and hence the feature maps lose information for distinguishing nodes quickly.

In summary, our contributions are as follows

- We relate the asymptotic behavior of graph CNNs with the topological information of underlying graphs via the spectral distribution of the (augmented) normalized Laplacian.
- We prove that if the weights of a GCN satisfy conditions determined by the graph spectra, the output of the GCN carries no information other than the node degrees and connected components for discriminating nodes when the layer size goes to infinity (Corollary 4).
- We apply our theory to Erdős – Rényi graphs as an example and show that when the underlying graph is sufficiently dense and large, a wide range of GCNs suffers from the information loss (Theorem 2).
- We propose a principled guideline for weight normalization of graph CNNs and empirically confirm it using real data

2 Related Work

MPNN-type Graph CNNs. Since many variants of graph CNNs have been researched, several unified formulations of graph CNNs have been proposed [5, 19]. Our approach is the closest to the formulation of Message Passing Neural Network (MPNN) [19]. MPNN consists of update and readout operations: The update operation computes the node representations from the representations of neighboring nodes, while the readout operation aggregates representations of all nodes to compute the representation for the whole graph. Neural Finger Print (NFP) [15], Gated Graph Sequence Neural Networks (GGNN) [33], Graph Attention Network (GAT) [51], and Graph Convolutional

Network (GCN) [30] fall into this formulation. Among others, GCN is the most important application of our theory. We are interested in GCN because it is one of the most widely used graph CNNs. For example, several deep graph generative models such as [21, 60] have used GCNs as a building block, particularly as an encoder for input graphs or a discriminator of Generative Adversarial Networks (GAN) [20] for graphs. In addition, GCN is interesting from a theoretical research perspective because in addition to an MPNN-type graph CNN, we can interpret GCN as a simplification of spectral-type graph CNNs [14, 25], that make use of graph Laplacian.

Our approach, which considers the asymptotic behaviors graph CNNs as the layer size goes to infinity, is similar to [44], one of the earliest works about graph CNNs. They obtained node representations by iterating message passing between nodes until they reached convergence. Their formulation is general in that we can use any local aggregation operation as long as it is a *contraction map*. Our theory differs from theirs in that we proved that the output of a graph CNN approaches to a certain space even if the local aggregation function is not necessarily a contraction map.

Expressive Power of Graph CNNs. Several studies have focused on theoretical analysis and the improvement of graph CNN expressive power. For example, [56] proved that graph CNNs are no less powerful than the Weisfeiler – Lehnman (WL) isomorphism test [53] and proposed a Graph Isomorphism Network (GIN), that is as powerful as the WL test. Although they experimentally showed that GIN has improved accuracy in supervised learning tasks, their analysis was restricted to the graph isomorphism problem, which only judges whether two graphs are isomorphic. Hence, it is not trivial to evaluate the expression ability of graph CNNs quantitatively from their theory.

Expressive Power of Invariant Neural Networks. Most graph CNNs are invariant under node permutations, i.e., the output of a graph CNN does not depend on indexing of nodes. Several studies have generalized such invariance to neural networks that are invariant under group actions (*invariant NNs*) [35, 31]. For example, the invariance of graph CNNs corresponds to the permutation group. Regarding the expressive power of invariant NNs, [59] proved a universal approximation theorem for arbitrary compact groups. However, unlike usual graph CNNs, invariant NNs have components that compute polynomial invariants. Likewise, [35] proved a universal approximation theorem for NNs invariant under an arbitrary subgroup of the permutation group. However, the NNs they considered were higher-order, which are generally different from graph CNNs, which are first-order. Although they gave a necessity condition (2-closedness) that the first order invariant NNs is a universal approximator, it is unknown whether the 2-closedness condition is sufficient and we do not know whether the graph CNNs satisfy the condition.

Depth and Expressive Power of Graph CNNs. For ordinal DL models such as FNNs or CNNs, we have both theoretical and empirical justification of deep and non-linear architectures for enhancing of the expressive power. For example, [49] showed that deep FNNs can approximate specific function classes more efficiently than shallow FNNs. Since CNNs have at least as much representation power as FNNs [41, 43], CNNs also have these characteristics in common. Practically, we use Residual Network [24] with over 100 layers for image recognition. Further, sophisticated initialization enables us to train vanilla CNNs with over 10,000 layers [7]. In contrast, several studies have witnessed serious performance degradation when stacking many layers on graph CNNs [30, 55]. Regarding non-linearity, [54] empirically showed that graph CNNs achieve comparable performance even if we omit intermediate non-linearity. These observations gave us questions about the current models of deep graph CNNs in terms of their expressive power.

Stability of Subspaces. [28] generalized the notion of the stability of dynamical systems from equilibrium points to subspaces and characterized the condition in which such subspaces will exist in linear dynamical systems. Since GCN is non-linear dynamics, their analysis cannot be applied, whereas our analysis overcame the non-linearity.

3 Problem Setting and Main Results

3.1 Notations

Let $\mathbb{N}$ and $\mathbb{N}_+$ be the set of non-negative and positive integers, respectively. For $N \in \mathbb{N}_+$, we denote the set of positive intergers less than or equal to N by $[N] := \{1, \ldots, N\}$. For a vector $v \in \mathbb{R}^N$, we write $v \geq 0$ if and only if $v_n \geq 0$ for all $n \in [N]$. Similarly, for a matrix $X \in \mathbb{R}^{N \times C}$, we write $X \geq 0$ if and only if $X_{nc} \geq 0$ for all $n \in [N]$ and $c \in [C]$. We call such a vector (resp. matrix) a *non-negative* vector (resp. matrix). $\langle \cdot, \cdot \rangle$ denotes the inner product of vectors or matrices, depending on the context. For example, $\langle u, v \rangle := u^\top v = \sum_{n=1}^N u_n v_n$ for $u, v \in \mathbb{R}^N$ and $\langle X, Y \rangle := \mathrm{tr}(X^T Y) = \sum_{n=1}^N \sum_{c=1}^C X_{nc} Y_{nc}$ for $X, Y \in \mathbb{R}^{N \times C}$. $\mathbb{1}_P$ equals to 1 if the proposition P is true else 0. For vectors $v \in \mathbb{R}^N$ and $w \in \mathbb{R}^C$, $v \otimes w \in \mathbb{R}^{N \times C}$ denotes the Kronecker product of v and w defined by $(v \otimes w)_{nc} := v_n w_c$. For $X \in \mathbb{R}^{N \times C}$, $\|X\|_{\mathrm{F}} := \langle X, X \rangle^{1/2}$ denotes the Frobenius norm of X. For a vector $v \in \mathbb{R}^N$, $\mathrm{diag}(v) := (v_n \delta_{nm})_{n,m \in [N]} \in \mathbb{R}^{N \times N}$ denotes the diagonalization of v where δ_{nm} is Kronecker's delta, which is 1 when $n = m$ and 0 otherwise. For $N \in \mathbb{N}^+$, $I_N \in \mathbb{R}^{N \times N}$ denotes the identity matrix of size N. For a linear operator $P : \mathbb{R}^N \to \mathbb{R}^M$ and a subset $V \subset \mathbb{R}^N$, we denote the restriction of P to V by $P|_V : V \to \mathbb{R}^M$. We also write $P|_V$ even if we restrict the range of $P|_V$ to a subset that includes $P|_V(V)$.

3.2 Dynamical Systems

Although we are mainly interested in GCNs, we develop our theory more generally using dynamical systems. We will specialize to the GCNs in Section 4.

For $N, C \in \mathbb{N}_+$, let $P \in \mathbb{R}^{N \times N}$ be a symmetric matrix and $W_l \in \mathbb{R}^{C \times C}$ for $l \in \mathbb{N}_+$. We denote the maximum singular value of W_l by s_l and set $s := \sup_{l \in \mathbb{N}_+} s_l$. We define $f_l : \mathbb{R}^{N \times C} \to \mathbb{R}^{N \times C}$ by $f_l(X) := \sigma(PXW_l)$, where $\sigma : \mathbb{R}^{N \times C} \to \mathbb{R}^{N \times C}$ is an element-wise ReLU function [32] defined by $\sigma(X)_{nc} := \max(X_{nc}, 0)$ for $n \in [N], c \in [C]$. We consider the dynamics $X_{l+1} := f_l(X_l)$ with some initial value $X_1 \in \mathbb{R}^{N \times C}$. We are interested in the asymptotic behavior of X_l as $l \to \infty$. For $M \leq N$, let U be a M-dimensional subspace of $\mathbb{R}^N$. We assume U and P satisfy the following properties that generalize the situation where U is the eigenspace associated with the smallest eigenvalue of a (normalized) graph Laplacian Δ (that is, zero) and P is a polynomial of Δ.

Assumption 1. *U has an orthonormal basis $(e_m)_{m \in [M]}$ that consists of non-negative vectors.*

Assumption 2. *U is invariant under P, i.e., if $u \in U$, then $Pu \in U$.*

We endow $\mathbb{R}^N$ with the ordinal inner product and denote the orthogonal complement of U by $U^\perp := \{u \in \mathbb{R}^N \mid \langle u, v \rangle = 0, \forall v \in U\}$. By the symmetry of P, we can show that $U^\perp$ is invariant under P, too (Proposition 2 of Appendix D.1). Therefore, we can regard P as a linear mapping $P|_{U^\perp} : U^\perp \to U^\perp$. We denote the operator norm of $P|_{U^\perp}$ by λ. When U is the eigenspace associated with the smallest eigenvalue of Δ and P is $g(\Delta)$ where g is a polynomial, then, λ corresponds to $\lambda = \sup_\mu |g(\mu)|$ where sup ranges over all eigenvalues except the smallest one.

3.3 Main Results

Let $\mathcal{M}$ be a subspace of $\mathbb{R}^{N \times C}$ defined by $\mathcal{M} := U \otimes \mathbb{R}^C = \{\sum_{m=1}^M e_m \otimes w_m \mid w_m \in \mathbb{R}^C\}$ where $(e_m)_{m \in [M]}$ is the orthonormal basis of U appeared in Assumption 1. For $X \in \mathbb{R}^{N \times C}$, we denote the distance (induced from the Frobenius norm) from X to $\mathcal{M}$ by $d_{\mathcal{M}}(X) := \inf\{\|X - Y\|_{\mathrm{F}} \mid Y \in \mathcal{M}\}$. With these preparations, we introduce the main theorem of the paper.

Theorem 1. *For any $l \in \mathbb{N}_+$, we have $d_{\mathcal{M}}(f_l(X)) \leq s_l \lambda d_{\mathcal{M}}(X)$ for any $X \in \mathbb{R}^{N \times C}$.*

We can show that both of the linear operation $X \mapsto PXW_l$ and the non-linear operation $X \mapsto \sigma(X)$ decrease the distance $d_{\mathcal{M}}$. The theorem is the direct consequence of them. We use the non-negativity of e_m to prove the latter claim (Lemma 2); see Appendix A for the complete proof. We also discuss the strictness of Theorem 1 in Appendix D.3.

By setting $d_{\mathcal{M}}(X) = 0$, this theorem implies that $\mathcal{M}$ is invariant under f_l. In addition, if the maximum value of singular values are small, X_l asymptotically approaches $\mathcal{M}$ in the sense of [28] for any initial value X_1. We say $(X_l)_{l \in \mathbb{N}_+}$ *exponentially approaches* $\mathcal{M}$ if and only if there exists $a \in [0, 1)$ such that $d_{\mathcal{M}}(X_l) = O(a^l)$.

Corollary 1. *$\mathcal{M}$ is invariant under f_l for any $l \in \mathbb{N}_+$. That is, if $X \in \mathcal{M}$, then we have $f_l(X) \in \mathcal{M}$.*

Corollary 2. *$d_{\mathcal{M}}(X_l) = O((s\lambda)^l)$. In particular, if $s\lambda < 1$, then X_l exponentially approaches $\mathcal{M}$ as $l \to \infty$ for any initial value X_1.*

Suppose the operator norm of $P|_U : U \to U$ is no larger than λ, then, under the assumption of $s\lambda < 1$, X_l converges to 0, the trivial fixed point (see Proposition 3 of Appendix D.2). Therefore, we are interested in the case where the operator norm of $P|_U$ is strictly larger than λ. Further, we take U

as the direct sum of eigenspaces associated with the largest M eigenvalues of P. We restate Theorem 1 specialized to this situation. Note that the eigenvalues of P is real since P is symmetric.

Corollary 3. *Let $\lambda_1 \leq \cdots \leq \lambda_N$ be the eigenvalue of P, sorted in ascending order. Suppose the multiplicity of the largest eigenvalue λ_N is $M(\leq N)$, i.e., $\lambda_{N-M} < \lambda_{N-M+1} = \cdots = \lambda_N$. We define $\lambda := \max_{n \in [N-M]} |\lambda_n|$ and assume $\lambda < |\lambda_N|$. Let U to be the eigenspace associated with λ_N. We assume that U has an orthonormal basis that consists of non-negative vectors. Then, we have $d_{\mathcal{M}}(X_{l+1}) \leq s_l \lambda d_{\mathcal{M}}(X_l)$ where $\mathcal{M} := U \otimes \mathbb{R}^C$.*

Remark 1. *It is known that any Markov process on finite states converges to a unique distribution (equilibrium) if it is irreducible and aperiodic (see e.g., [40]). Theorem 1 includes this proposition as a special case with $M = 1$, $C = 1$, and $W_l = 1$ for all $l \in \mathbb{N}_+$. This is essentially the direct consequence of Perron – Frobenius' theorem (see e.g., [37]). See Appendix E for detail.*

4 Application to GCN

Several researchers have reported that the predictive accuracy of node classification degrades as the layer size increases [30, 55]. We formulate GCN [30] without readout operations [19] using the dynamical system defined in the previous section and derive a sufficient condition in terms of the spectra of underlying graphs in which layer stacking and non-linearly harm the node classification.

Let $G = (V, E)$ be an undirected graph where V is a set of nodes and E is a set of edges. Let $N = |V|$ be the number of nodes in G. We fix an order on V and identify V with $[N]$. In the case of GCN, we associate C dimensional signal to each node. X in the previous section corresponds to concatenation of the signals. GCN iteratively updates signals on V using the connection information P and weights W_l.

Let $A := (\mathbb{1}_{\{(i,j)\in E\}})_{i,j\in[N]} \in \mathbb{R}^{N\times N}$ be the adjacency matrix and $D := \mathrm{diag}(\deg(i)_{i\in[N]}) \in \mathbb{R}^{N\times N}$ be the degree matrix of G where $\deg(i) := |\{j \in V \mid (i,j) \in E\}|$ is the degree of node i. Let $\tilde{A} := A + I_N$, $\tilde{D} := D + I_N$ be the adjacent and degree matrix of graph G augmented with self-loops. We define the *augmented* normalized Laplacian [54] of G by $\tilde{\Delta} := I_N - \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$ and set $P := I_N - \tilde{\Delta}$. Let $L, C \in \mathbb{N}_+$ be the layer and channel sizes, respectively. For weights $W_l \in \mathbb{R}^{C\times C}$ ($l \in [L]$), we define a GCN associated with G by $f = f_L \circ \cdots \circ f_1$ where $f_l : \mathbb{R}^{N\times C} \to \mathbb{R}^{N\times C}$ is defined by $f_l(X) := \sigma(PXW_l)$. We are interested in the asymptotic behavior of the output X_L of the GCN as $L \to \infty$.

Suppose G has M connected components and let $V = V_1 \sqcup \cdots \sqcup V_M$ be the decomposition of the node set V into connected components. We denote an indicator vector of connected component $m \in [M]$ by $\mathbf{1}_m := (\mathbb{1}_{\{n\in V_m\}})_{n\in[N]} \in \mathbb{R}^N$. The following proposition shows that GCN satisfies the assumption of Corollay 3 (see Appendix B for proof).

Proposition 1. *Let $\lambda_1 \leq \cdots \leq \lambda_N$ be the eigenvalue of P sorted in ascending order. Then, we have $-1 < \lambda_1$, $\lambda_{N-M} < 1$, and $\lambda_{N-M+1} = \cdots = \lambda_N = 1$. In particular, we have $\lambda := \max_{n=1,\ldots,N-M} |\lambda_n| < 1$. Further, $e_m := \tilde{D}^{\frac{1}{2}}\mathbf{1}_m$ for $m \in [M]$ are the basis of the eigenspace associated with the eigenvalue 1.*

Corollary 4. *Let $\mathcal{M} := \{\sum_{k=N-M+1}^{N} e_k \otimes w_k \mid w_k \in \mathbb{R}^C\}$. If $s\lambda < 1$, then, the output X_l of the l-th layer of GCN on G exponentially approaches $\mathcal{M}$ as $l \to \infty$ for any initial value X_1.*

In the context of node classification tasks, we can interpret this corollary as the "information loss" of GCN in the limit of infinite layers. For any $X \in \mathcal{M}$, if two nodes $i, j \in V$ are in a same connected component and their degrees are identical, then the column vectors of X that corresponds to nodes i and j are identical. This means that we cannot distinguish these nodes using X. In this sense, $\mathcal{M}$ only has information about connected components and node degrees and X_l exponentially approaches such "information-less" states as $l \to \infty$. As we wrote in the previous section, X_l converges to 0 when $s < 1$ (remember $\lambda_N = 1$). An interesting point is that even if $s \geq 1$, X_l suffers from the aforementioned information loss.

We note that the rate $s\lambda$ in Corollary 4 depends on the spectra of the augmented normalized Laplacian, which is determined by the topology of the underlying graph G. Hence, our result explicitly relates the topological information of graphs and asymptotic behaviors of graph CNNs.

5 Asymptotic Behavior of GCN on Erdős – Rényi Graph

Corollary 4 gives us a way to characterize the asymptotic behaviors of GCNs via the spectral distribution of the underlying graphs. Specifically, consider a graph G with M connected components. Let $0 = \tilde{\mu}_1 = \cdots = \tilde{\mu}_M < \tilde{\mu}_{M+1} \leq \cdots \leq \tilde{\mu}_N < 2$ be the eigenvalue of the augmented normalized Laplacian of G (see, Proposition 1) and set $\lambda := \min_{m=M+1,\ldots,N} |1 - \tilde{\mu}_m|^{-1} > 1$. By Corollary 4, the output of GCN "loses information" as the limit of the layer size goes to infinity when the largest singular values of weights are strictly smaller than λ. Therefore, the closer the positive eigenvalues μ_m are to 1, the wider range of GCNs that satisfy the assumption of the corollary.

To demonstrate this, we investigate the spectral distribution when the underlying graph is an Erdős – Rényi graph [16, 17], that has N nodes and where the edges between two distinct nodes appear independently with probability $p \in [0, 1]$, as an example. [10] showed that when $\frac{\log N}{Np} = o(1)$, the eigenvalues of the normalized Laplacian Δ except for the smallest one converge to 1 with high probability (see Theorem 2 therein). [9, 12] also proved similar theorems. Since that the normalized Laplacian of the complete graph with N nodes has eigenvalues 0 with multiplicity 1 and 1 with multiplicity $N - 1$, we can interpret this theorem that the denser the Erdős-Rényi graph, the more closely it approaches the complete graph in terms of the spectral distribution. We can show that the spectra of the augmented normalized Laplacian behaves similarly (Lemma 5 of Appendix C). By combining this fact with the discussion of the previous paragraph, we obtain the asymptotic behavior of GCNs on the Erdős – Rényi graph. See Appendix C for the complete proof.

Theorem 2. *Consider GCN on the Erdős-Rényi graph $G_{N,p}$ such that $\frac{\log N}{Np} = o(1)$ as a function of N. For any $\varepsilon > 0$, if the supremum s of the maximum singular values of weights in the GCN satisfies $s < \frac{1}{7}\sqrt{\frac{Np-p+1}{\log(4N/\varepsilon)}}$, then, for sufficiently large N, the GCN satisfies the condition of Corollary 4 with probability at least $1 - \varepsilon$.*

We note that the upper bound $\frac{1}{7}\sqrt{\frac{Np-p+1}{\log(4N/\varepsilon)}} \to \infty$ as $N \to \infty$ under the condition $\frac{\log N}{Np} = o(1)$. Therefore, Theorem 2 implies that when a graph on which GCN is defined is sufficiently dense and large, the output of GCN tends to fail to distinguish nodes if the scale of its weights is not extremely large. We also note that the denser the underlying graph, the more GCNs suffer from information loss. Intuitively, when a graph is dense, graph convolution operations mix signals on nodes and move them closer to each other quickly. Therefore, this result affirms the hypothesis that deep graph CNNs perform badly due to information loss via signal mixing by graph convolutions.

6 Experiments

6.1 Visualization of the One-step Transition

We numerically investigate how the transition $f(X) = \sigma(PXW)$ changes inputs using the vector field $V(X) := f(X) - X$ [1]. For this purpose, we set $N = 2$, $M = 1$, and $C = 1$. Let $\lambda_1 \leq \lambda_2$ be the eigenvalues of P. We choose W as $|\lambda_2|^{-1} \leq W < |\lambda_1|^{-1}$ such that Theorem 1 is applicable but is not reduced to the trivial situation (i.e., $W < |\lambda_2|^{-1}$). We will choose the eigenvector $e \in \mathbb{R}^2$ associated with λ_2 in two ways as described below. See Appendix G.1 for the concrete values of P, e, and W.

First, we choose the eigenvector e to be non-negative in order to satisfy the assumption of Theorem 1 (**Case 1**). Figure 1 (left) shows the vector field V for this case. We can see that V faces toward the direction of the invariant space $\mathcal{M}$ at every point, which means that the transition function f uniformly decreases the distance from V. This observation is consistent with the consequence of the theorems.

Next, we choose the eigenvector $e = [e_1 \quad e_2]^\top$ such that the signs of e_1 and e_2 differ (**Case 2**); this violates the assumption of Theorem 1. Figure 1 (right) is the visualization of V for this case. V crosses $\mathcal{M}$, showing that $\mathcal{M}$ is not invariant under f. In addition, the direction of V is opposite to $\mathcal{M}$ at some points, which means that f does not uniformly decrease the distance from $\mathcal{M}$. These observations show that we cannot remove the non-negativity assumption of e from Theorem 1.

[1] Since we consider the one-step transition only, we omit the subscript l from f_l, X_l, and W_l.

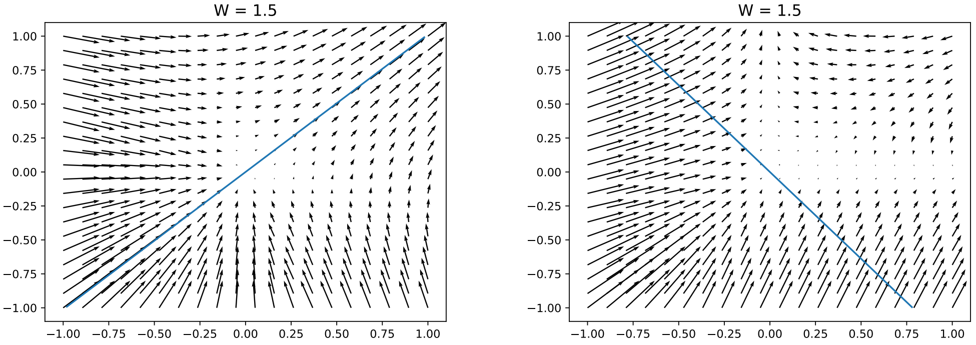

Figure 1: Visualization of vector field V induced by the one-step transition. Straight lines indicate the subspace $\mathcal{M}$. Left: **Case 1**. Right: **Case 2**

6.2 Effect of Maximum Singular Values on Performance

Corollary 4 implies that if the maximum singular values s of the weights are less than the threshold λ^{-1} (defined in Section 3.2 for the general case and Section 4 for the GCN case), the output of GCN loses information for node classification in the limit of the infinite layers. Therefore, if s is in that region, we cannot expect deep GCN to achieve good prediction accuracy. Conversely, if we can successfully train the model, s should avoid the region $s \leq \lambda^{-1}$. We empirically confirm these hypotheses using real datasets in this section.

We use Cora [36, 46], which is a standard citation network dataset. The task is to classify the genre of papers out of seven classes using the occurrence of words and the citation relationship. We regard the citation relationship as a graph that consists of papers and links and apply graph CNNs. This is a transductive learning [42] setting because we can use node properties of the validation and test data during training. As implied from the discussion in Section 5, Corollary 4 can support a wide range of GCNs when the underlying graph is relatively dense; we can also observe this phenomenon in Theorem 2. However, the Cora dataset is too sparse to examine the aforementioned hypotheses — the range of the maximum singular values s for which Corollary 4 gives a non-trivial result is $1 \leq s < \lambda^{-1} \approx 1 + 3.62 \times 10^{-3}$ for the original Cora dataset. Therefore, we make a noisy version of CiteSeer by randomly adding edges to the graph. We call the resultant dataset *Noisy CiteSeer*. Through this manipulation, we can expand the range of s to $1 \leq s < \lambda^{-1} \approx 1.11$.

Figure 2 (left) shows the accuracy for the test dataset in terms of the maximum singular values and the number of graph convolution layers. We can observe that when GCNs whose maximum singular value s is out of the region $s < \lambda^{-1}$ (i.e., $s = 3, 10$), outperforms those inside the region (i.e., $s = 0.5, 1.05$) in almost all configurations. Furthermore, we note that the accuracy of GCN with $s = 10$ is better than GCN without normalization (*Unnormalized*). Figure 2 (right) shows the transition of the maximum singular values of the weights during training when we use three-layered GCN. See Figure 7 – 9 in Appendix H.2.1 for the results using other layer sizes and other datasets. We can observe that the maximum singular value s does not shrink to the region $s \leq \lambda^{-1}$. On the contrary, when the layer size is small and predictive accuracy is high, GCNs gradually increase s from the initial value and avoid the region. We conducted the experiment results using the other two citation networks (CiteSeer [18, 46] dataset and the Cora dataset with more random edges) and observe similar results; see Appendix G.2 and H.2 for details. In conclusion, the experiment results are consistent with the theorems we obtained.

7 Discussion

Applicability to Graph CNNs on Sparse Graphs. We showed in Section 5 that for the Erdős – Rényi graph $G_{N,p}$ such that $\frac{\log N}{Np} = o(1)$, if GCN on $G_{N,p}$ has weights for which the maximum singular values is smaller than $s_0 := C\sqrt{\frac{Np}{\log(N/\varepsilon)}}$ (C is a universal constant), it satisfies the assumption of Corollary 4 with probability $1 - \varepsilon$. This means that when the underlying Erdős-

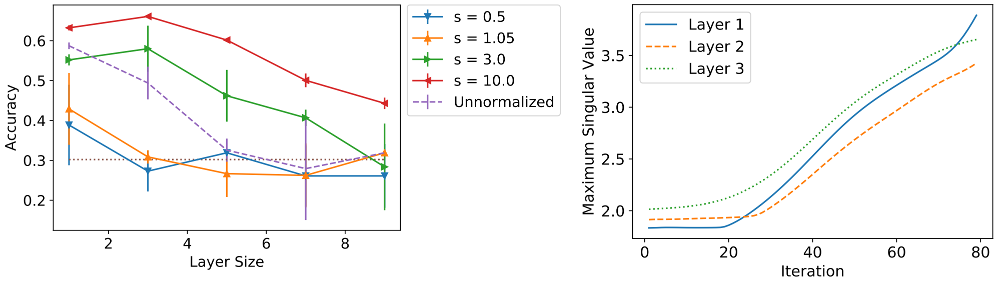

Figure 2: Left: Effect of the maximum singular values on weights on model performance. The horizontal dotted line indicates the chance rate (30.2%). The error bar is the standard deviation of 3 trials. Right: Transition of maximum singular values during training. Best view in color.

Rényi graph is sufficiently dense, the “forbidden area" of $s \leq s_0$ is relatively large. In addition, in Section 6.2 we empirically confirmed that if the weights of GCN avoid this area, GCN can have good prediction accuracy. However, real-world graphs are sometimes not as sparse, which means that Corollary 4 is applicable to very limited GCNs on these graphs. Theoretically, [12] proved that if the expected average degree of $G_{N,p}$ is bounded, then, the smallest positive eigenvalue of the normalized Laplacian of $G_{N,p}$ is $o(1)$ with high probability. The asymptotic behavior of graph CNNs on such sparse graphs is left for future research.

GCN with Large Weights. Our theory suggests that the maximum singular values of weights in GCN should be larger than a threshold determined by the spectral distribution of the graphs underneath, otherwise GCNs suffer from information loss for node classification in the limit of infinite layers. From a statistical learning theory perspective, we can interpret that too-small singular values lead to poor expressive power. However, if the scale of weights gets larger, the model complexity of the function class represented by graph CNNs increases. Since too-large model complexity leads to large generalization error, we conjecture that the graph CNNs with too-large weights also perform poorly. Therefore, a trade-off should exist between the expressive power and model complexity and there should be a “sweet spot" on the weight scale that balances the two.

Limitations in Graph CNN Architectures. Our analysis is limited to graph CNNs with the ReLU activation function. We implicitly use the property of ReLU that it is a projection onto the cone $\{X \geq 0\}$ (see Lemma 2 of Appendix A). This fact enables the ReLU function to get along with the non-negativity of eigenvectors associated with the largest eigenvalues. Therefore, it is far from trivial to extend our result to other activation functions such as the sigmoid function or Leaky ReLU [34]. Another point is that our dynamical system only considers the update operation [19] of graph CNNs and does not take readout operations into consideration. In particular, we cannot directly apply our theory to graph classification tasks in which each sample is represented as a graph.

8 Conclusion

In this paper, to understand the empirically observed phenomena that deepening graph CNNs does not improve their predictive performance, we analyzed asymptotic behaviors of graph CNNs by interpreting them as a dynamical system that includes GCN and Markov process as special cases. We gave theoretical conditions under which GCNs suffer from the information loss in the limit of infinite layers. Practically, our theory gives a principled guideline for how to determine the scale of weights of graph CNNs. We empirically showed that the weight normalization implied by our theory performed well in real datasets. Our theory directly related the expressive power of graph CNNs and topological information of the underlying graphs via spectra of the Laplacian. This enabled us to leverage spectral and random graph theory to analyze the expressive power of graph CNNs. Actually, we considered GCN on the Erdős – Rényi graph as an example and showed that when the underlying graph is sufficiently dense and large, a wide range of GCNs on the graph suffer from information loss. One promising direction of future research is to analyze the optimization of graph CNNs and statistical properties such as the generalization power [52] via spectral and random graph theories. We hope that this paper can be used as a first step in this direction.

Acknowledgments

We thank Katsuhiko Ishiguro for providing the part of code for the experiments, Kohei Hayashi and Haru Negami Oono for giving us feedback and comments on the draft. TS was partially supported by MEXT Kakenhi (26280009, 15H05707, 18K19793 and 18H03201), Japan Digital Design and JSTCREST.

References

[1] Chainer Chemistry: Chainer extension library for biology and chemistry. `https://github.com/pfnet-research/chainer-chemistry`. Accessed: 2019-05-03.

[2] Optuna: A hyperparameter optimization framework. `https://optuna.org`. Accessed: 2019-05-03.

[3] D. Angluin and L. G. Valiant. Fast probabilistic algorithms for hamiltonian circuits and matchings. *Journal of Computer and system Sciences*, 18(2):155–193, 1979.

[4] A. R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information theory*, 39(3):930–945, 1993.

[5] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

[6] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. Algorithms for hyper-parameter optimization. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 2546–2554. Curran Associates, Inc., 2011.

[7] M. Chen, J. Pennington, and S. Schoenholz. Dynamical isometry and a mean field theory of RNNs: Gating enables signal propagation in recurrent neural networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 873–882. PMLR, 10–15 Jul 2018.

[8] F. Chung and L. Lu. Connected components in random graphs with given expected degree sequences. *Annals of combinatorics*, 6(2):125–145, 2002.

[9] F. Chung, L. Lu, and V. Vu. The spectra of random graphs with given expected degrees. *Internet Mathematics*, 1(3):257–275, 2004.

[10] F. Chung and M. Radcliffe. On the spectra of general random graphs. *the electronic journal of combinatorics*, 18(1):215, 2011.

[11] F. R. Chung and F. C. Graham. *Spectral graph theory*. Number 92. American Mathematical Soc., 1997.

[12] A. Coja-Oghlan. On the laplacian eigenvalues of $G_{n,p}$. *Combinatorics, Probability and Computing*, 16(6):923–946, 2007.

[13] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

[14] M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3844–3852. Curran Associates, Inc., 2016.

[15] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams. Convolutional networks on graphs for learning molecular fingerprints. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2224–2232. Curran Associates, Inc., 2015.

[16] P. Erdös and A. Rényi. On random graphs I. *Publicationes Mathematicae (Debrecen)*, 6:290–297, 1959.

[17] E. N. Gilbert. Random graphs. *The Annals of Mathematical Statistics*, 30(4):1141–1144, 1959.

[18] C. L. Giles, K. D. Bollacker, and S. Lawrence. Citeseer: an automatic citation indexing system. In *Proceedings of the third ACM conference on Digital libraries*, pages 89–98. ACM, 1998.

[19] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1263–1272. PMLR, 2017.

[20] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[21] A. Grover, A. Zweig, and S. Ermon. Graphite: Iterative generative modeling of graphs. *arXiv preprint arXiv:1803.10459*, 2018.

[22] T. Hagerup and C. Rüb. A guided tour of Chernoff bounds. *Information processing letters*, 33(6):305–308, 1990.

[23] W. Hamilton, Z. Ying, and J. Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems 30*, pages 1024–1034. Curran Associates, Inc., 2017.

[24] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[25] M. Henaff, J. Bruna, and Y. LeCun. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*, 2015.

[26] K. Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.

[27] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[28] C. Johnson. Stabilization of linear dynamical systems with respect to arbitrary linear subspaces. *Journal of Mathematical Analysis and Applications*, 44(1):175–186, 1973.

[29] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[30] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

[31] R. Kondor and S. Trivedi. On the generalization of equivariance and convolution in neural networks to the action of compact groups. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 2747–2755. PMLR, 2018.

[32] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[33] Y. Li, R. Zemel, and M. Brockschmidt. Gated graph sequence neural networks. In *International Conference on Learning Representations*, 2016.

[34] A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *in ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013.

[35] H. Maron, E. Fetaya, N. Segol, and Y. Lipman. On the universality of invariant networks. *arXiv preprint arXiv:1901.09342*, 2019.

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

[37] C. D. Meyer. *Matrix analysis and applied linear algebra*, volume 71. Siam, 2000.

[38] H. N. Mhaskar. Approximation properties of a multilayered feedforward artificial neural network. *Advances in Computational Mathematics*, 1(1):61–80, 1993.

[39] H. Nguyen, S. Maeda, and K. Oono. Semi-supervised learning of hierarchical representations of molecules using neural message passing. *arXiv preprint arXiv:1711.10168*, 2017.

[40] J. R. Norris. *Markov chains*. Number 2. Cambridge university press, 1998.

[41] K. Oono and T. Suzuki. Approximation and non-parametric estimation of ResNet-type convolutional neural networks. *arXiv preprint arXiv:1903.10047*, 2019.

[42] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[43] P. Petersen and F. Voigtlaender. Equivalence of approximation by convolutional neural networks and fully-connected networks. *arXiv preprint arXiv:1809.00973*, 2018.

[44] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009.

[45] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, and M. Welling. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607. Springer, 2018.

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

[47] S. Sonoda and N. Murata. Neural network with unbounded activation functions is universal approximator. *Applied and Computational Harmonic Analysis*, 43(2):233–268, 2017.

[48] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[49] M. Telgarsky. Benefits of depth in neural networks. In *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 1517–1539. PMLR, 23–26 Jun 2016.

[50] S. Tokui, K. Oono, S. Hido, and J. Clayton. Chainer: a next-generation open source framework for deep learning. In *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS)*, 2015.

[51] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

[52] S. Verma and Z.-L. Zhang. Stability and generalization of graph convolutional neural networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019.

[53] B. Weisfeiler and L. A.A. A reduction of a graph to a canonical form and an algebra arising during this reduction. *Nauchno-Technicheskaya Informatsia*, 2(9):12–16, 1968.

[54] F. Wu, T. Zhang, A. H. d. Souza Jr, C. Fifty, T. Yu, and K. Q. Weinberger. Simplifying graph convolutional networks. *arXiv preprint arXiv:1902.07153*, 2019.

[55] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu. A comprehensive survey on graph neural networks. *arXiv preprint arXiv:1901.00596*, 2019.

[56] K. Xu, W. Hu, J. Leskovec, and S. Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.

[57] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh. Graph R-CNN for scene graph generation. In *Proceedings of the European Conference on Computer Vision*, pages 670–685, 2018.

[58] D. Yarotsky. Error bounds for approximations with deep ReLU networks. *Neural Networks*, 94:103–114, 2017.

[59] D. Yarotsky. Universal approximations of invariant maps by neural networks. *arXiv preprint arXiv:1804.10306*, 2018.

[60] J. You, B. Liu, Z. Ying, V. Pande, and J. Leskovec. Graph convolutional policy network for goal-directed molecular graph generation. In *Advances in Neural Information Processing Systems 31*, pages 6410–6421. Curran Associates, Inc., 2018.

[61] S. Zhang, H. Tong, J. Xu, and R. Maciejewski. Graph convolutional networks: Algorithms, applications and open challenges. In *International Conference on Computational Social Networks*, pages 79–91. Springer, 2018.

[62] Z. Zhang, P. Cui, and W. Zhu. Deep learning on graphs: A survey. *arXiv preprint arXiv:1812.04202*, 2018.

[63] D.-X. Zhou. Universality of deep convolutional neural networks. *arXiv preprint arXiv:1805.10769*, 2018.

[64] P. Zhou and J. Feng. Understanding generalization and optimization performance of deep CNNs. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5960–5969. PMLR, 2018.

Appendix

A Proof of Theorem 1

As we wrote in the main article, it is enough to show the following lemmas (definition of miscellaneous variables are as in Section 3.2).

Lemma 1. *For any $X \in \mathbb{R}^{N \times C}$, we have $d_{\mathcal{M}}(PXW_l) \leq d_{\mathcal{M}}(X)$.*

Lemma 2. *For any $X \in \mathbb{R}^{N \times C}$, we have $d_{\mathcal{M}}(\sigma(X)) \leq d_{\mathcal{M}}(X)$.*

Proof of Lemma 1. Since P is a symmetric linear operator on $U^\perp$, we can choose the orthonormal basis $(e_m)_{m=M+1,\ldots,N}$ of $U^\perp$ consisting of the eigenvalue of $P|_{U^\perp}$. Let λ_m be the eigenvalue of P to which e_m is associated ($m = M+1, \ldots, N$). Note that since the operator norm of $P|_{U^\perp}$ is λ, we have $|\lambda_m| \leq \lambda$ for all $m = M+1, \ldots, N$. Since $(e_m)_{m \in [N]}$ forms the orthonormal basis of $\mathbb{R}^N$, we can uniquely write $X \in \mathbb{R}^{N \times C}$ as $X = \sum_{m=1}^N e_m \otimes w_m$ for some $w_m \in \mathbb{R}^C$. Then, we have

$d^2_{\mathcal{M}}(X) = \sum_{m=M+1}^{N} \|w_m\|^2$ where $\|\cdot\|$ is the 2-norm of a vector. On the other hand, we have

$$
\begin{aligned}
PXW_l &= \sum_{m=1}^{N} (Pe_m) \otimes (W_l^\top w_m) \\
&= \sum_{m=1}^{M} (Pe_m) \otimes (W_l^\top w_m) + \sum_{m=M+1}^{N} (Pe_m) \otimes (W_l^\top w_m) \\
&= \sum_{m=1}^{M} (Pe_m) \otimes (W_l^\top w_m) + \sum_{m=M+1}^{N} e_m \otimes (\lambda_m W_l^\top w_m)
\end{aligned}
$$

Since U is invariant under P, for any $m \in [M]$, we can write Pe_m as a linear combination of $e_n (n \in [M])$. Therefore, we have $d^2_{\mathcal{M}}(PXW_l) = \sum_{m=M+1}^{N} \|\lambda_m W_l^\top w_m\|^2$. Then, we obtain the desired inequality as follows:

$$
\begin{aligned}
d^2_{\mathcal{M}}(PXW_l) &= \sum_{m=M+1}^{N} \|\lambda_m W_l^\top w_m\|^2 \\
&\leq \lambda^2 \sum_{m=M+1}^{N} \|W_l^\top w_m\|^2 \\
&\leq \lambda^2 s_l^2 \sum_{m=M+1}^{N} \|w_m\|^2 \\
&= \lambda^2 s_l^2 d^2_{\mathcal{M}}(X).
\end{aligned}
$$

□

Proof of Lemma 2. We choose $(e_m)_{m=N-M+1,\ldots,N}$ as in the proof of Lemma 1. We denote $X = (X_{nc})_{n\in[N],c\in[C]}$ and $e_n = (e_{mn})_{m\in[N]}$, respectively. Let $(e'_c)_{c\in[C]}$ be the standard basis of $\mathbb{R}^C$. Then, $(e_n \otimes e'_c)_{n\in[N],c\in[C]}$ is the orthonormal basis of $\mathbb{R}^{N\times C}$, endowed with the standard inner product as a Euclid space. Therefore, we can decompose X as $X = \sum_{n=1}^{N}\sum_{c=1}^{C} a_{nc} e_n \otimes e'_c$ where $a_{nc} = \langle X, e_n \otimes e'_c\rangle = \sum_{m=1}^{N} X_{mc} e_{mn}$. Then, we have $d^2_{\mathcal{M}}(X) = \sum_{n=M+1}^{N} \|\sum_{c=1}^{C} a_{nc} e'_c\|^2$, which is further transformed as

$$
\begin{aligned}
d^2_{\mathcal{M}}(X) &= \sum_{n=M+1}^{N} \left\| \sum_{c=1}^{C} a_{nc} e'_c \right\|^2 \\
&= \sum_{n=M+1}^{N} \sum_{c=1}^{C} a_{nc}^2 \\
&= \sum_{c=1}^{C} \left(\sum_{n=1}^{N} a_{nc}^2 - \sum_{n=1}^{M} a_{nc}^2 \right) \\
&= \sum_{c=1}^{C} \left(\|X_{\cdot c}\|^2 - \sum_{n=1}^{M} \langle X_{\cdot c}, e_n\rangle^2 \right),
\end{aligned}
$$

where $X_{\cdot c}$ is the c-th column vector of X. Similarly, we have

$$
d^2_{\mathcal{M}}(\sigma(X)) = \sum_{c=1}^{C} \left(\|X^+_{\cdot c}\|^2 - \sum_{n=1}^{M} \langle X^+_{\cdot c}, e_n\rangle^2 \right),
$$

where we denote $\sigma(X) = (X^+_{nc})_{n\in[N],c\in[C]}$ in shorthand. Therefore, the inequality follow from the following lemma. □

Lemma 3. *Let $x \in \mathbb{R}^N$ and $v_1, \ldots, v_M \in \mathbb{R}^N$ be orthonormal vectors (i.e., $\langle v_m, v_n\rangle = \delta_{mn}$) satisfying $v_m \geq 0$ for all $m \in [M]$. Then, we have $\|x\|^2 - \sum_{m=1}^{M}\langle x, v_m\rangle^2 \geq \|x^+\|^2 - \sum_{m=1}^{M}\langle x^+, v_m\rangle^2$ where $x^+ := \max(x, 0)$ for $x \in \mathbb{R}$.*

Proof. The value $\|y\|^2 - \sum_{m=1}^{M} \langle y, u_m \rangle^2$ is invariant under simultaneous coordinate permutation of y and u_m's. Therefore, we can assume without loss of generality that the coordinate of x are sorted: $x_1 \leq \ldots \leq x_L < 0 \leq x_{L+1} \leq \cdots \leq x_N$ for some $L \leq N$. Then, we have

$$\|x\|^2 - \|x^+\|^2 = \sum_{n=1}^{L} x_n^2. \tag{1}$$

When $L = 0$, the sum in the right hand side is treated as 0. On the other hand, writing as $v_m = (v_{nm})_{n \in [N]}$, direct calculation shows

$$\sum_{m=1}^{M} \langle x, v_m \rangle^2 - \langle x^+, v_m \rangle^2 = \sum_{m=1}^{M} \left(\left(\sum_{n=1}^{L} x_n v_{nm} \right)^2 - 2 \sum_{n=1}^{L} \sum_{l=L+1}^{N} x_n x_l v_{nm} v_{lm} \right). \tag{2}$$

Let $I_m := \{n \in [N] \mid v_{nm} > 0\}$ be the support of v_m for $m \in [M]$. We note that if $m \neq m' \in [M]$, we have $I_m \cap I_{m'} = \emptyset$ since if there existed $n \in I_m \cap I_{m'}$, we have

$$0 = \langle v_m, v_{m'} \rangle \geq v_{nm} v_{nm'} > 0,$$

which is contradictory. Therefore,

$$\begin{aligned} \sum_{m=1}^{N} \left(\sum_{n=1}^{L} x_n v_{nm} \right)^2 &= \sum_{m=1}^{N} \left(\sum_{n \in I_m \cap [L]} x_n v_{nm} \right)^2 \\ &\leq \sum_{m=1}^{N} \left(\sum_{n \in I_m \cap [L]} x_n^2 \right) \left(\sum_{n \in I_m \cap [L]} v_{nm}^2 \right) \quad (\because \text{Cauchy–Schwarz inequality}) \\ &\leq \sum_{m=1}^{N} \left(\sum_{n \in I_m \cap [L]} x_n^2 \right) \quad (\because \|v_m\|^2 = 1) \\ &\leq \sum_{n=1}^{L} x_n^2. \end{aligned} \tag{3}$$

We used the fact that I_m's are disjoint and $v_{nm} = 0$ if $n \notin \cup_m I_m$ in the first equality above. Further, we have $x_n x_l v_{nm} v_{lm} \leq 0$ for $1 \leq n \leq L$ and $L + 1 \leq l \leq N$ by the definition of L and non-negativity of v_m. By combining (1), (2), and (3), we have

$$\sum_{m=1}^{M} \langle x, v_m \rangle^2 - \langle x^+, v_m \rangle^2 \leq \sum_{n=1}^{L} x_n^2 = \|x\|^2 - \|x^+\|^2.$$

□

B Proof of Proposition 1

Proof. Let $\tilde{\mu}_1 \leq \cdots \leq \tilde{\mu}_N$ be the eigenvalue of the augmented normalized Laplacian $\tilde{\Delta}$, sorted in ascending order. Since $P = I_N - \tilde{\Delta}$, it is enough to show $\tilde{\mu}_1 = \cdots = \tilde{\mu}_M = 0$, $\tilde{\mu}_{M+1} > 0$, and $\tilde{\mu}_N < 2$. For the first two, the statements are equivalent to that $\tilde{\Delta}$ is positive semi-definite and that the multiplicity of the eigenvalue 0 is same as the number of connected components [2]. This is well-known for Laplacian or its normalized version (see, e.g., [11]) and the proof for $\tilde{\Delta}$ is similar. By direct calculation, we have

$$x^\top \tilde{\Delta} x = \frac{1}{2} \sum_{i,j=1}^{N} a_{ij} \left(\frac{x_i}{\sqrt{d_i + 1}} - \frac{x_j}{\sqrt{d_j + 1}} \right)^2$$

for any $x = \begin{bmatrix} x_1 & \cdots & x_N \end{bmatrix}^\top \in \mathbb{R}^N$. Therefore, $\tilde{\Delta}$ is positive semi-definite and hence $\tilde{\mu}_1 \geq 0$.

[2] The former statement is identical to Lemma 1 and latter one is the extension of Lemma 2 of [54].

Suppose temporally that G is connected. If $x \in \mathbb{R}^N$ is an eigenvector associated to 0, then, by the aformentioned calculation, $\frac{x_i}{\sqrt{d_i+1}}$ and $\frac{x_j}{\sqrt{d_j+1}}$ must be same for all pairs (i, j) such that $a_{ij} > 0$. However, since G is connected, $\frac{x_i}{\sqrt{d_i+1}}$ must be same value for all $i \in [N]$. That means the multiplicity of the eigenvalue 0 is 1 and any eigenvector associated to 0 must be proportional to $\tilde{D}^{\frac{1}{2}}\mathbf{1}$. Now, suppose G has M connected components $V_1, \ldots, V_M$. Let $\tilde{\Delta}_m$ be the augmented normalized Laplacians corresponding to each connected component V_m for $m \in [M]$. By the aformentioned discussion, $\tilde{\Delta}_m$ has the eigenvalue 0 with multiplicity 1. Since $\tilde{\Delta}$ is the direct sum of $D\tilde{e}lta'_ms$, the eigenvalue of $\tilde{\Delta}$ is the union of those for $\tilde{\Delta}_m$'s. Therefore, $\tilde{\Delta}$ has the eigenvalue 0 with multiplicity M and $e_m = \tilde{D}^{\frac{1}{2}}\mathbf{1}_m$'s are the orthogonal basis of the eigenspace.

Finally, we prove $\tilde{\mu}_N < 2$. Let μ_N be the largest eigenvalue of the normalized Laplacian $\Delta = D^{-\frac{1}{2}}(D - A)D^{-\frac{1}{2}}$, where $D^{-\frac{1}{2}} \in \mathbb{R}^{N \times N}$ is the diagonal matrix defined by

$$D_{ii}^{-\frac{1}{2}} = \begin{cases} \deg(i)^{-\frac{1}{2}} & (\text{if } \deg(i) \neq 0) \\ 0 & (\text{if } \deg(i) = 0) \end{cases}.$$

Note that $D^{-\frac{1}{2}}D^{\frac{1}{2}}$ nor $D^{\frac{1}{2}}D^{-\frac{1}{2}}$ are not equal to the identity matrix I_N in general. However, we have

$$L = D^{\frac{1}{2}}D^{-\frac{1}{2}}LD^{-\frac{1}{2}}D^{\frac{1}{2}} \tag{4}$$

where $L = D - A$ is the (unnormalized) Laplacian. Therefore, we have

$$\begin{aligned}
\tilde{\mu}_N &= \max_{x \neq 0} \frac{x^\top \tilde{\Delta} x}{\|x\|} \\
&= \max_{x \neq 0} \frac{x^\top \tilde{D}^{-\frac{1}{2}} L \tilde{D}^{-\frac{1}{2}} x}{\|x\|} \\
&= \max_{x \neq 0} \frac{x^\top \tilde{D}^{-\frac{1}{2}} D^{\frac{1}{2}} D^{-\frac{1}{2}} L D^{-\frac{1}{2}} D^{\frac{1}{2}} \tilde{D}^{-\frac{1}{2}} x}{\|x\|} \quad (\because (4)) \\
&= \max_{x \neq 0} \frac{(D^{\frac{1}{2}}\tilde{D}^{-\frac{1}{2}}x)^\top \Delta (D^{\frac{1}{2}}\tilde{D}^{-\frac{1}{2}}x)}{\|x\|} \\
&= \max_{\substack{x \neq 0 \\ D^{\frac{1}{2}}\tilde{D}^{-\frac{1}{2}}x \neq 0}} \frac{(D^{\frac{1}{2}}\tilde{D}^{-\frac{1}{2}}x)^\top \Delta (D^{\frac{1}{2}}\tilde{D}^{-\frac{1}{2}}x)}{\|x\|} \\
&= \max_{\substack{x \neq 0 \\ D^{\frac{1}{2}}\tilde{D}^{-\frac{1}{2}}x \neq 0}} \frac{(D^{\frac{1}{2}}\tilde{D}^{-\frac{1}{2}}x)^\top \Delta (D^{\frac{1}{2}}\tilde{D}^{-\frac{1}{2}}x)}{\|D^{\frac{1}{2}}\tilde{D}^{-\frac{1}{2}}x\|} \frac{\|D^{\frac{1}{2}}\tilde{D}^{-\frac{1}{2}}x\|}{\|x\|} \\
&\leq \max_{\substack{x \neq 0 \\ D^{\frac{1}{2}}\tilde{D}^{-\frac{1}{2}}x \neq 0}} \frac{(D^{\frac{1}{2}}\tilde{D}^{-\frac{1}{2}}x)^\top \Delta (D^{\frac{1}{2}}\tilde{D}^{-\frac{1}{2}}x)}{\|D^{\frac{1}{2}}\tilde{D}^{-\frac{1}{2}}x\|} \max_{\substack{x \neq 0 \\ D^{\frac{1}{2}}\tilde{D}^{-\frac{1}{2}}x \neq 0}} \frac{\|D^{\frac{1}{2}}\tilde{D}^{-\frac{1}{2}}x\|}{\|x\|} \\
&\leq \max_{y \neq 0} \frac{y^\top \Delta y}{\|y\|} \max_{x \neq 0} \frac{\|D^{\frac{1}{2}}\tilde{D}^{-\frac{1}{2}}x\|}{\|x\|} \\
&= \mu_N \max_{n \in [N]} \left(\frac{\deg(i)}{\deg(i) + 1} \right)^{\frac{1}{2}} \\
&\leq \mu_N.
\end{aligned}$$

Therefore, we have $\tilde{\mu}_N \leq \mu_N$[3]. Since $\max_{i \in [N]} \left(\frac{\deg(i)}{\deg(i)+1} \right)^{\frac{1}{2}} < 1$, the equality $\tilde{\mu}_N = \mu_N$ holds if and only if $\mu_N = 0$, that is, G has N connected components. On the other hand, it is known that $\mu_N \leq 2$ and the equality holds if and only if G has non-trivial bipartite graph as a connected component (see, e.g., [11]). Therefore, $\tilde{\mu}_N = \mu_N$ and $\mu_N = 2$ does not hold simultaneously and we obtain $\mu_N < 2$. □

[3] Theorem 1 of [54] showed that this inequality strictly holds when G is simple and connected. We do not require this assumption.

C Proof of Theorem 2

We follow the proof of Theorem 2 of [10]. The idea is to relate the spectral distribution of the normalized Laplacian with that of its expected version. Since we can compute the latter one explicitly for the Erdős-Rényi graph, we can derive the convergence of spectra. We employ this technique and derive similar conclusion for the augmented normalized Laplacian.

First, we consider genral random graphs not restricted to Erdős-Rényi graphs. Let $N \in \mathbb{N}_+$, and $P = (p_{ij})_{i,j \in [N]}$ be a non-negative symmetric matrix (meaning that $p_{ij} \geq 0$ for any $i, j \in [N]$). Let G be an undirected random graph with N nodes such that an edge between i and j is independently present with probability p_{ij}. Let A and D be the adjacency and the degree matrices of G, respectively (that is, $A_{ij} \sim \mathrm{Ber}(p_{ij})$, i.i.d.). Define the expected node degree of node i by $t_i := \sum_{j=1}^N p_{ij}$. Let $\tilde{A} := A + I_N$, $\tilde{D} := D + I_N$ and define $\bar{A} := \mathbb{E}[\tilde{A}] = P + I_N$ and $\bar{D} := \mathbb{E}[\tilde{D}] = \mathrm{diag}(t_1, \ldots, t_N) + I_N$ correspondingly. We define the augmented normalized Laplacian $\tilde{\Delta}$ of G by $\tilde{\Delta} := I_N - \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}$ and its expected version by $\bar{\Delta} := I_N - \bar{D}^{-\frac{1}{2}} \bar{A} \bar{D}^{-\frac{1}{2}}$ [4]. For a symmetric matrix $X \in \mathbb{R}^N$, we define its eigenvalues, sorted in ascending order by $\lambda_1(X) \leq \cdots \leq \lambda_N(X)$ and its operator norm by $\|X\| = \max_{n \in [N]} |\lambda_n(X)|$.

Lemma 4 (Ref. [10] Theorem 2)**.** *Let $\delta := \min_{n \in [N]} t_n$ be the minimum expected degree of G. Set $k(\varepsilon) := 3(1 + \log(4/\varepsilon))$. Then, for any $\varepsilon > 0$, if $\delta + 1 > k(\varepsilon) \log N$, we have*

$$\max_{n \in [N]} \left| \lambda_n(\tilde{\Delta}) - \lambda_n(\bar{\Delta}) \right| \leq 4 \sqrt{\frac{3 \log(4N/\varepsilon)}{\delta + 1}}$$

with probability at least $1 - \varepsilon$.

Proof. By Weyl's theorem, we have $\max_{n \in [N]} \left| \lambda_n(\tilde{\Delta}) - \lambda_n(\bar{\Delta}) \right| \leq \|\tilde{\Delta} - \bar{\Delta}\|$. Therefore, it is enough to bound $\|\tilde{\Delta} - \bar{\Delta}\|$. Let $C := I_N - \bar{D}^{-\frac{1}{2}} \tilde{A} \bar{D}$. By the triangular inequality, we have $\|\tilde{\Delta} - \bar{\Delta}\| \leq \|\tilde{\Delta} - C\| + \|C - \bar{\Delta}\|$. We will bound these terms respectively.

First, we bound $\|C - \bar{\Delta}\|$. Direct calculation shows $C - \bar{\Delta} = -\bar{D}^{-\frac{1}{2}}(A - P)\bar{D}^{-\frac{1}{2}}$. Let $E^{ij} \in \mathbb{R}^{N \times N}$ be a matrix defined by

$$(E^{ij})_{kl} = \begin{cases} 1 & \text{if } (i = k \text{ and } i = l) \text{ or } (i = l \text{ and } j = k), \\ 0 & \text{otherwise.} \end{cases}$$

We define the random variable Y_{ij} by

$$Y_{ij} := \frac{A_{ij} - p_{ij}}{\sqrt{t_i + 1}\sqrt{t_j + 1}} E^{ij}.$$

Then, Y_{ij}'s are independent and we have $C - \bar{\Delta} = \sum_{i,j=1}^N Y_{ij}$. To apply Theorem 5 of [10] to Y_{ij}'s, we bound $\|Y_{ij} - \mathbb{E}[Y_{ij}]\|$ and $\| \sum_{i,j=1}^N \mathbb{E}[Y_{ij}^2] \|$. First, we have

$$\|Y_{ij} - \mathbb{E}[Y_{ij}]\| = \|Y_{ij}\| \leq \frac{\|E^{ij}\|}{\sqrt{t_i + 1}\sqrt{t_j + 1}} \leq (\delta + 1)^{-1}.$$

Since

$$\mathbb{E}[Y_{ij}^2] = \frac{p_{ij} - p_{ij}^2}{(t_i + 1)(t_j + 1)} \begin{cases} E^{ii} + E^{jj} & (\text{if } i \neq j), \\ E^{ii} & (\text{if } i = j), \end{cases}$$

[4] Note that $\mathbb{E}[\tilde{\Delta}] \neq \bar{\Delta}$ in general due to the dependence between $\tilde{A}$ and $\tilde{D}$.

we have

$$\begin{aligned}\left\|\sum_{i,j=1}^{N}\mathbb{E}[Y_{ij}^2]\right\| &= \left\|\sum_{i,j=1}^{N}\frac{p_{ij}-p_{ij}^2}{(t_i+1)(t_j+1)}E^{ii}\right\| \\ &= \max_{i\in[N]}\left(\sum_{j=1}^{N}\frac{p_{ij}-p_{ij}^2}{(t_i+1)(t_j+1)}\right) \\ &\le \max_{i\in[N]}\left(\sum_{j=1}^{N}\frac{p_{ij}}{(t_i+1)(t_j+1)}\right) \\ &\le (\delta+1)^{-1}.\end{aligned}$$

By letting $a \leftarrow \sqrt{\frac{3\log(4N/\varepsilon)}{\delta+1}}$, $M \leftarrow (\delta+1)^{-1}$, $v^2 \leftarrow (\delta+1)^{-1}$ and applying Theorem 5 of [10], we have

$$\begin{aligned}\Pr(\|C-\bar{\Delta}\| > a) &\le 2N\exp\left(-\frac{a^2}{2(\delta+1)^{-1}+2(\delta+1)^{-1}a/3}\right) \\ &\le 2N\exp\left(-\frac{3\log(4N/\varepsilon)}{2(1+a/3)}\right).\end{aligned}$$

By the definition of $k(\varepsilon)$, we have $a<1$ if $\delta+1>k(\varepsilon)\log n$. For such δ, we have

$$\begin{aligned}\Pr(\|C-\bar{\Delta}\| > a) &\le 2N\exp\left(-\frac{3\log(4N/\varepsilon)}{2(1+a/3)}\right) \\ &\le 2N\exp\left(-\log(4N/\varepsilon)\right) \quad (\because a<1) \\ &= \frac{\varepsilon}{2}.\end{aligned} \tag{5}$$

Next, we bound $\|\tilde{\Delta}-C\|$. First, since $a<1$, by Chernoff bound (see, e.g. [3, 22])), we have

$$\begin{aligned}\Pr(|d_i-t_i| > a(t_i+1)) &\le 2\exp\left(-\frac{a^2(t_i+1)}{3}\right) \\ &\le 2\exp\left(-\frac{a^2(\delta+1)}{3}\right) \\ &= \frac{\varepsilon}{2N}.\end{aligned}$$

Therefore, if $|d_i-t_i|\le a(t_i+1)$, then we have

$$\begin{aligned}\left|\sqrt{\frac{d_i+1}{t_i+1}}-1\right| &\le \left|\frac{d_i+1}{t_i+1}-1\right| \quad (\because |\sqrt{x}-1|\le|x-1| \text{ for } x\ge 0) \\ &= \left|\frac{d_i-t_i}{t_i+1}\right| \\ &\le a.\end{aligned}$$

Therefore, by union bound, we have

$$\|\bar{D}^{-\frac{1}{2}}\tilde{D}^{\frac{1}{2}}-I_N\| = \max_{i\in[N]}\left|\sqrt{\frac{d_i+1}{t_i+1}}-1\right| \le a$$

with probability at least $1-\varepsilon/2$. Further, since the eigenvalue of the augmented normalized Laplacian is in $[0,2]$ by the proof of Proposition 1, we have $\|I_N-\tilde{\Delta}\|\le 1$. By combining them, we have

$$\begin{aligned}\|\tilde{\Delta}-C\| &= \|(\bar{D}^{-\frac{1}{2}}\tilde{D}^{\frac{1}{2}}-I_N)(I_N-\tilde{\Delta})\tilde{D}^{\frac{1}{2}}\bar{D}^{-\frac{1}{2}}+(I_N-\tilde{\Delta})(I-\tilde{D}^{\frac{1}{2}}\bar{D}^{-\frac{1}{2}})\| \\ &\le \|(\bar{D}^{-\frac{1}{2}}\tilde{D}^{\frac{1}{2}}-I_N\|\|\tilde{D}^{\frac{1}{2}}\bar{D}^{-\frac{1}{2}}\|+\|I-\tilde{D}^{\frac{1}{2}}\bar{D}^{-\frac{1}{2}}\| \\ &\le a(a+1)+a.\end{aligned} \tag{6}$$

From (5) and (6), we have

$$\begin{aligned}
\|\tilde{\Delta} - \bar{\Delta}\| &\leq \|\tilde{\Delta} - C\| + \|C - \bar{\Delta}\| \\
&\leq a + a(a+1) + a \\
&\leq a^2 + 3a \\
&\leq 4a \quad (\because a < 1)
\end{aligned}$$

with probability at least $1 - \varepsilon$ by union bound. □

Let $N \in \mathbb{N}_+$ and $p > 0$. In the case of the Erdős-Rényi graph $G_{N,p}$, we should set $P = p(J_N - I_N)$ where $J_N \in \mathbb{R}^{N \times N}$ are the all-one matrix. Then, we have $\bar{A} = pJ_N + (1-p)I_N$, $\bar{D} = (Np - p + 1)I_N$, and $\bar{\Delta} = \frac{p}{Np-p+1}(NI_N - J_N)$. Since the eigenvalue of J_N is N (with multiplicity 1) and 0 (with multiplicity $N-1$), the eigenvalue of $\bar{\Delta}$ is 0 (with multiplicity 1) and $\frac{Np}{Np-p+1}$ (with multiplicity $N-1$). For $G_{N,p}$, δ is the expected average degree $(N-1)p$. Hence, we have the following lemma from Lemma 4:

Lemma 5. *Let $\tilde{\Delta}$ be its augmented normalized Laplacian of the Erdős-Rényi graph $G_{N,p}$. For any $\varepsilon > 0$, if $\frac{Np-p+1}{\log N} > k(\varepsilon) := 3(1 + \log(4/\varepsilon))$, then, with probability at least $1 - \varepsilon$, we have*

$$\max_{i=2,\ldots,N} \left| \lambda_i(\tilde{\Delta}) - \frac{Np}{Np-p+1} \right| \leq 4\sqrt{\frac{3\log(4N/\varepsilon)}{Np-p+1}}.$$

Corollary 5. *Consider GCN on $G_{N,p}$. Let W_l be the weight of the l-th layer of GCN and s_l be the maximum singular value of W_l for $l \in \mathbb{N}_+$. Set $s := \sup_{l \in \mathbb{N}_+}$. Let $\varepsilon > 0$. We define $k(\varepsilon) := 3(1 + \log(4/\varepsilon))$ and $l(N, p, \varepsilon) = \frac{1-p}{Np-p+1} + 4\sqrt{\frac{3\log(4N/\varepsilon)}{Np-p+1}}$. If $\frac{Np-p+1}{\log N} > k(\varepsilon)$ and $s \leq l(N, \varepsilon)^{-1}$, then, GCN on $G_{N,p}$ satisfies the assumption of the Corollary 4 with probability at least $1 - \varepsilon$.*

Proof of Theorem 2. Since $\frac{\log N}{Np} = o(1)$, for fixed ε, we have

$$\frac{Np - p + 1}{\log N} > \frac{Np}{\log N} > k(\varepsilon)$$

for sufficiently large N. Further, $Np \to \infty$ as $N \to \infty$ when $\frac{\log N}{Np} = o(1)$. Therefore, we have

$$\frac{(1-p)^2}{Np - p + 1} \leq \frac{1}{Np} \leq (7 - 4\sqrt{3})^2 \log\left(\frac{4N}{\varepsilon}\right)$$

for sufficiently large N. Hence.

$$\frac{1-p}{Np - p + 1} \leq (7 - 4\sqrt{3})\sqrt{\frac{\log(4N/\varepsilon)}{Np - p + 1}}.$$

Therefore, we have $l(N, p, \varepsilon) \leq 7\sqrt{\frac{\log(4N/\varepsilon)}{Np-p+1}}$. Therefore, if $s \leq \frac{1}{7}\sqrt{\frac{Np-p+1}{\log(4N/\varepsilon)}}$, then we have $s \leq l(N, p, \varepsilon)^{-1}$. □

D Miscellaneous Propositions

D.1 Invariance of Orthogonal Complement Space

Proposition 2. *Let $P \in \mathbb{R}^{N \times N}$ be a symmetric matrix, treated as a linear operator $P : \mathbb{R}^N \to \mathbb{R}^N$. If a subspace $U \subset \mathbb{R}^N$ is invariant under P (i.e., if $u \in U$, then $Pu \in U$), then, $U^\perp$ is invariant under P, too.*

Proof. For any $u \in U^\perp$ and $v \in U$, by symmetry of P, we have

$$\langle Pu, v \rangle = (Pu)^\top v = u^\top P^\top v = u^\top Pv = \langle u, Pv \rangle.$$

Since U is an invariant space of P, we have $Pv \in U$. Hence, we have $\langle u, Pv \rangle = 0$ because $u \in U^\perp$. We obtain $Pu \in U^\perp$ by the definition of $U^\perp$. □

D.2 Convergence to Trivial Fixed Point

Let $P \in \mathbb{R}^{N \times N}$ be a symmetric matrix, $W_l \in \mathbb{R}^{C \times C}$, s_l be the maximum singular value of W_l for $l \in \mathbb{N}_+$. We define $f_l : \mathbb{R}^{N \times C} \to \mathbb{R}^{N \times C}$ by $f_l(X) := \sigma(PXW_l)$ where σ is the element-wise ReLU function.

Proposition 3. *Suppose further that the operator norm of* P *is no less than* λ*, then we have* $\|f_l(X)\|_{\mathrm{F}} \leq s_l \lambda \|X\|_{\mathrm{F}}$ *for any* $l \in \mathbb{N}_+$*. In particular, let* $s := \sup_{l \in \mathbb{N}_+} s_l$*. If* $s\lambda < 1$*, then,* X_l *exponentially approaches* 0 *as* $l \to \infty$*.*

Proof. Since λ is the operator norm of $P|_{U^\perp}$, the assumption implies that the operator norm of P itself is no less than λ. Therefore, we have $\|PXW_l\|_{\mathrm{F}} \leq \lambda \|XW_l\|_{\mathrm{F}} \leq s_l \lambda \|X\|_{\mathrm{F}}$. On the other hand, since $\sigma(x)^2 \leq x^2$ for any $x \in \mathbb{R}$, we have $\|\sigma(X)\|_{\mathrm{F}} \leq \|X\|_F$ for any $X \in \mathbb{R}^{N \times C}$. Combining the two, we have $\|f_l(X)\|_{\mathrm{F}} \leq \|PXW_l\|_{\mathrm{F}} \leq s_l \lambda \|X\|_{\mathrm{F}}$. □

D.3 Strictness of Theorem 1

Theorem 1 implies that if $s\lambda \leq 1$, then, one-step transition f_l does not increase the distance to $\mathcal{M}$. In this section, we first prove that this theorem is strict in the sense that, there exists a situation in which $s_l \lambda > 1$ holds and the distance $d_{\mathcal{M}}$ increases by one-step transition f_l at some point X.

Set $N \leftarrow 2$, $C \leftarrow 1$, and $M \leftarrow 1$ in Section 3.2. For $\mu, \lambda > 0$, we set

$$P \leftarrow \begin{bmatrix} \mu & 0 \\ 0 & \lambda \end{bmatrix}, e \leftarrow \begin{bmatrix} 1 \\ 0 \end{bmatrix}, U \leftarrow \left\{ \begin{bmatrix} x \\ y \end{bmatrix} \mid y = 0 \right\}.$$

Then, by definition, we can check that the 3-tuple (P, e, U) satisfies the Assumptions 1 and 2. Set $\mathcal{M} := U \otimes \mathbb{R} = U$ and choose $W \in \mathbb{R}$ so that $W > \lambda^{-1}$. Finally define $f : \mathbb{R}^{N \times C} \to \mathbb{R}^{N \times C}$ by $f(X) := \sigma(PXW)$ where σ is the element-wise ReLU function.

Proposition 4. *We have* $d_{\mathcal{M}}(f(X)) > d_{\mathcal{M}}(X)$ *for any* $X = [x_1 \quad x_2]^\top \in \mathbb{R}^2$ *such that* $x_2 > 0$*.*

Proof. By definition, we have $d_{\mathcal{M}}(X) = |x_2|$. On the other hand, direct calculation shows that $f_l(X) = \left[(W\mu X_1)^+ \quad (W\lambda X_2)^+\right]^\top$ and $d_{\mathcal{M}}(f_l(X)) = (W\lambda X_2)^+$ where $x^+ := \max(x, 0)$ for $x \in \mathbb{R}$. Since $W > \lambda^{-1}$ and $x_2 > 0$, we have $d_{\mathcal{M}}(f_l(X)) > d_{\mathcal{M}}(X)$. □

Next, we prove the non-strictness of Theorem 1 in the sense that there exists a situation in which $s_l \lambda > 1$ holds and the distance $d_{\mathcal{M}}$ uniformly decreases by f_l. Again, we set Set $N \leftarrow 2$, $C \leftarrow 1$, and $M \leftarrow 1$. Let $\lambda \in (1, 2)$ and set

$$P \leftarrow \frac{\lambda}{2} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}, e \leftarrow \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, U \leftarrow \left\{ \begin{bmatrix} x \\ y \end{bmatrix} \mid x = y \right\}$$

Then, we can directly show that 3-tuple (P, e, U) satisfies the Assumptions 1 and 2. Set $W \leftarrow 1$.

Proposition 5. *We have* $W\lambda > 1$ *and* $d_{\mathcal{M}}(f_l(X)) < d_{\mathcal{M}}(X)$ *for all* $X \in \mathbb{R}^2$*.*

Proof. First, note that $e' := \frac{1}{\sqrt{2}} [1 \quad -1]^\top$ is the eigenvector of P associated to λ: $Pe' = \lambda e'$. For $X = ae + be'$ $(a, b > 0)$, the distance between X and $\mathcal{M}$ is $d_{\mathcal{M}}(X) = |b|$. On the other hand, by direct computation, we have

$$f(X) = \sigma(PXW) = \begin{cases} \begin{bmatrix} 0 & \frac{\lambda b}{\sqrt{2}} \end{bmatrix}^\top & \text{(if } b \geq 0\text{)}, \\ \begin{bmatrix} \frac{-\lambda b}{\sqrt{2}} & 0 \end{bmatrix}^\top & \text{(if } b < 0\text{)}. \end{cases}$$

Therefore, the distance between $f(X)$ and $\mathcal{M}$ is $d_{\mathcal{M}}(f(X)) = \lambda |b| / 2$. Since $\lambda < 2$, we have $d_{\mathcal{M}}(f(X)) < d_{\mathcal{M}}(X)$ for any $X \in \mathbb{R}^2$. □

We also show that the non-negativity of e (Assumption 1) is *not* a redundant condition in Section 6.1.

E Relation to Markov Process

It is known that any Markov process on finite states converges to a unique distribution (*equilibrium*) if it is irreducible and aperiodic (see, e.g., [40]). As we see in this section, this theorem is the special case of Corollary 3.

Let $S := \{1, \ldots, N\}$ be a finite discrete state space. Consider a Markov process on S characterized by a symmetric transition matrix $P = (p_{ij})_{i,j \in [N]} \in \mathbb{R}^{N \times N}$ such that $P \geq 0$ and $P\mathbf{1} = \mathbf{1}$ where $\mathbf{1}$ is the all-one vector. We interpret p_{ij} as the transition probability from a state i to j. We associate P with a graph $G_P = (V_P, E_P)$ by $V_P = [N]$ and $(i, j) \in E_P$ if and only if $p_{ij} > 0$. Since P is symmetric, we can regard G_P as an undirected graph. We assume P is irreducible and aperiodic [5]. Perron – Frobenius' theorem (see, e.g., [37]) implies that P satisfy the assumption of Corollary 3 with $M = 1$.

Proposition 6 (Perron – Frobenius)**.** *Let the eigenvalues of P be $\lambda_1 \leq \cdots \leq \lambda_N$. Then, we have $-1 < \lambda_1$, $\lambda_{N-1} < 1$, and $\lambda_N = 1$. Further, there exists unique vector $e \in \mathbb{R}^N$ such that $e \geq 0$, $\|e\| = 1$, and e is the eigenvector for the eivenvalue* 1.

Corollary 6. *Let $\lambda := \max_{n=1,\ldots,N-1} |\lambda_n| (< 1)$ and $\mathcal{M} := \{e \otimes w \mid w \in \mathbb{R}^C\}$ If $s\lambda < 1$, then, for any initial valueX_1, X_l exponentially approaches $\mathcal{M}$ as $l \to \infty$.*

If we set $C = 1$ and $W_l = 1$ for all $l \in \mathbb{N}_+$, then, we can inductively show that $X_l \geq 0$ for any $l \geq 2$. Therefore, we can interpret X_l as a measure on S. Suppose further that we take the initial value X_1 as $X_1 \geq 0$ and $X_1^\top \mathbf{1} = 1$ so that we can interpret X_1 as a probability distribution on S. Then, we can inductively show that $X_l \geq 0$, $X_l^\top \mathbf{1} = 1$ (i.e., X_l is a probability distribution on S), and $X_{l+1} = \sigma(PX_lW_l) = PX_l$ for all $l \in \mathbb{N}_+$. In conclusion, the corollary is reduced to the fact that if a finite and discrete Markov process is irreducible and aperiodic, any initial probability distribution converges exponentially to an equibrilium. In addition, the the rate λ corresponds to the *mixing time* of the Markov process.

F GCN Defined Using Normalized Laplacian

In Section 4, we defined P using the augmented normalized Laplacian $\tilde{\Delta}$ by $P = I_N - \tilde{\Delta}$. We can alternatively use the usual normalized Laplacian Δ instead of the augmented one to define P and want to apply the theory developed in Section 3.2. We write the normalized Laplacian version as $P_\Delta := I_N - \Delta$. The only obstacle is that the smallest eigenvalue λ_1 of P_Δ can be equal to -1, while that of P is strictly larger than -1 (see, Proposition 1). This corresponds to that fact the largest eigenvalue of $\tilde{\Delta}$ is strictly smaller than 2, while that for Δ can be 2. It is known that the largest eigenvalue of Δ is 2 if and only if the graph has a non-trivial bipartite connected component (see, e.g., [11]). Therefore, we can develop a theory using the normalized Laplacian instead of the augmented one in parallel for such a graph G.

In Section 5, we characterized the asymptotic behavior of GCN defined by the augmented normalized Laplacian via its spectral distribution (Lemma 5 of Appendix C). We can derive a similar claim for GCN defined via the normalized Laplacian using the original theorem for the normalized Laplacian in [10] (Theorem 7 therein). The normalized Laplacian version of GCN is advantegeous over the one made from the augmented one because we know its spectral distribution for broader range of random graphs. For example, [10] proved the convergence of the spectral distribution of the normalized Laplacian for Chung-Lu's model [8], which includes power law graphs as a special case (see, Theorem 4 of [10]).

[5] A symmetric matrix A is called *irreducible* if and only if G_A is connected. We say a graph G is *aperiodic* if the greatest common divisor of length of all loops in G is 1. A symmetric matrix A is aperiodic if the graph G_A induced by A is aperiodic.

G Details of Experiment Settings

G.1 Experiment of Section 6.1

We set the eigenvalue of P to $\lambda_1 = 0.5$ and $\lambda_2 = 1.0$ and randomly generate P until the eigenvector e associated to λ_2 satisfies the condition of each case described in the main article. We set $W = 1.5$. We use the following values for each case as P and e.

G.1.1 Case 1

$$P = \begin{bmatrix} 0.7469915 & 0.2499819 \\ 0.2499819 & 0.7530085 \end{bmatrix}, \quad e = \begin{bmatrix} 0.7028392 \\ -0.71134876 \end{bmatrix}.$$

G.1.2 Case 2

$$P = \begin{bmatrix} 0.6899574 & -0.2426827 \\ -0.2426827 & 0.8100426 \end{bmatrix}, \quad e = \begin{bmatrix} 0.61637234 \\ -0.78745485 \end{bmatrix}.$$

G.2 Experiment of Section 6.2

G.2.1 Dataset

We used the Cora [36, 46] and CiteSeer [18, 46] datasets for experiments. We obtained the preprocessed dataset from the code repository of [30][6].

The Cora dataset is a citation network dataset consisting of 2708 papers and 5429 links. Each paper is represented as the occurence of 1433 unique words and is associated to one of 7 genres (Case Based, Genetic Algorithms, Neural Networks, Probabilistic Methods, Reinforcement Learning, Rule Learning, Theory). The graph made from the citation links has 78 connected components and the smallest positive eigenvalue of the augmented Normalized Laplacian is approximately $\tilde{\mu} = 3.62 \times 10^{-3}$. Therefore, the upper bound of Corollary 4 is $\lambda^{-1} = (1-\tilde{\mu})^{-1} \approx 1 + 3.62 \times 10^{-3}$. 818 out of 2708 samples are labelled as "Probabilistic Methods", which is the largest proportion. Therefore, the chance rate is $818/2708 = 30.2\%$.

The CiteSeer dataset is a citation network dataset consisting of 3312 papers and 4732 links. Each paper is represented as the occurence of 3703 unique words and is associated to one of 6 genres (Agents, AI, DB, IR, ML, HCI). The graph made from the citation links has 438 connected components and the smallest positive eigenvalue of the augmented Normalized Laplacian is approximately $\tilde{\mu} = 1.25 \times 10^{-3}$. Therefore, the upper bound of Corollary 4 is $\lambda^{-1} = (1-\tilde{\mu})^{-1} \approx 1 + 1.25 \times 10^{-3}$. 701 out of 2708 samples are labelled as "IR", which is the largest proportion. Therefore, the chance rate is $701/3312 = 21.1\%$.

G.2.2 Noisy Citation Networks

We created two datasets from the Cora dataset: Noisy Cora 2500 and Noisy Cora 5000. *Noisy Cora 2500* is made from the Cora dataset by uniformly randomly adding 2500 edges, respectively. Since some random edges are overlapped with existing edges, the number of newly-added edges is 2495 in total. We only changes the underlying graph from the Cora dataset and do not change word occurences (feature vectors) and genres (labels). The underlying graph of the Noisy Cora dataset has two connected components and the smallest positive eigenvalue is $\tilde{\mu} \approx 9.62 \times 10^{-2}$. Therefore, the threshold of the maximum singular values of in Corollary 4 is increased to $\lambda^{-1} = (1 - \tilde{\mu})^{-1} \approx 1.11$. Similarly, *Noisy Cora 5000* is made by adding 5000 edges uniformaly randomly. The number of newly added edges is 4988 and the graph is connected (i.e., it has only 1 connected component). $\tilde{\mu}$ and λ are $\tilde{\mu} \approx 1.32 \times 10^{-1}$ and $\lambda = (1 - \tilde{\mu})^{-1} \approx 1.15$, respectively.

We made the noisy version of CiteSeer (*Noisy CiteSeer*), too, by adding 5000 edges uniformly randomly to the CiteSeer dataset. The number of newly-added edges was 4991 in total. This manipulation reduced the number of connected component of the graph to 3. $\tilde{\mu}$ and λ are $\tilde{\mu} \approx 1.11 \times 10^{-1}$ and $\lambda^{-1} = (1 - \tilde{\mu})^{-1} \approx 1.13$, respectively.

[6] https://github.com/tkipf/gcn

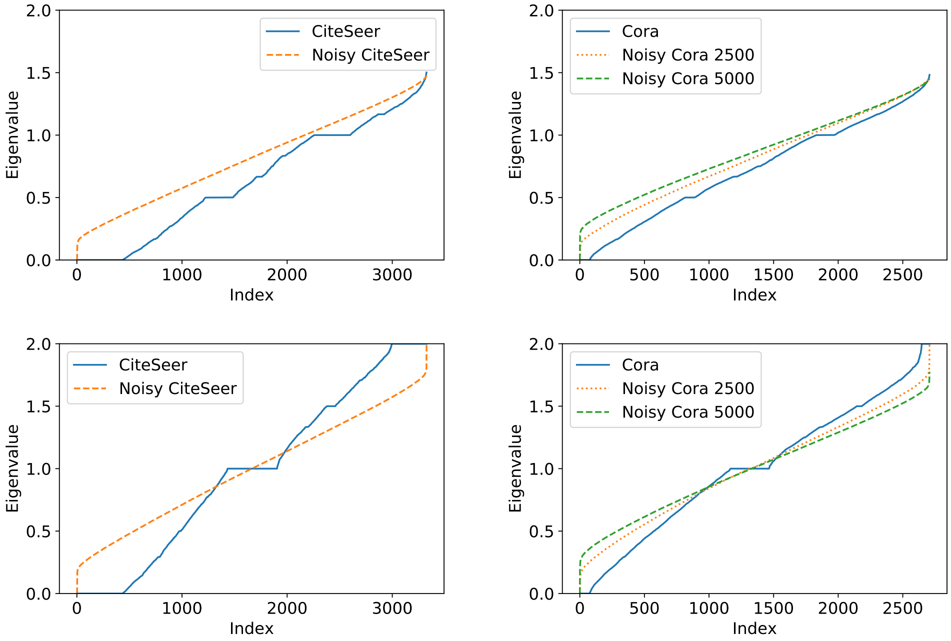

Figure 3: Spectral distribution of Laplacian for the citation network datasets. Top: autmented Normalized Laplacian. Bottom: normalized Laplacian. Left: CiteSeer and Noisy CiteSeer. Right: Cora and Noisy Cora (2500, 5000). Best view in color.

Figure 3 shows the spectral distribution of the autmented normalized Laplacian for CiteSeer and Cora, and their noisy variants. For comparison, we show in Figure 3 the spectral distribution of the normalized Laplacian for these datasets.

G.2.3 Model Architecture

We used GCN consisting of a single node embedding layer, one to nine graph convolution layers, and a readout operation [19], which is a linear transformation common to all nodes in our case. We applied softmax function to the output of GCN. The output dimension of GCN is same as the number of classes (i.e., seven for Noisy Cora 2500/5000 and six for Noisy CiteSeer). We treated the number of units in each graph convolution layer as a hyperparameter. Optionally, we specified the maximum singular values s of graph convolution layers. The choice of s is either 0.5 (smaller than 1), s_1 (in the interval $\{1 \leq s < \lambda^{-1}\}$), 3 and 10 (larger than λ^{-1}). We used $s_1 = 1.05$ for Noisy Cora 2500 and $s_1 = 1.1$ for Noisy Cora 5000 and Noisy CiteSeer so that s_1 is not close to the edges of the the interval $\{1 \leq s < \lambda^{-1}\}$.

G.2.4 Performance Evaluation Procedure

We split all nodes in a graph (either Noisy Cora 2500/5000 or Noisy CiteSeer) into training, validation, and test sets. Data split is the same as the one done by [30]. We trained the model three times for each choice of hyperparemeters using the training set and defined the objective function as the average accuracy on the validation set. We chose the combination of hyperparameters that achieves the best value of objective function. We evaluate the accuracy of the test dataset three times using the chosen combination of hyperparameters and computed their average and the standard deviation.

G.2.5 Training

At initialization, we sample parameters from the i.i.d. Gaussian distribution. If the scale of maximum singular values s is specified, we subsequently scale weight matrices of graph convolution layers so that their maximum singular values are normalized to s. The loss function is defined as the sum of the cross entropy loss for all training nodes. We train the model using the one of gradient-based

Table 1: Hyperparameters of the experiment in Section 6.2. $X \sim \mathrm{LogUnif}[10^a, 10^b]$ denotes the random variable $\log_{10} X$ obeys the uniform distribution over $[a, b]$. “Learning rate" corresponds to α when “Optimization algorithm" is Adam [29].

Name	Value
Unit size	$\{10, 20, \ldots, 500\}$
Epoch	$\{10, 20, \ldots, 100\}$
Optimization algorithm	$\{\mathrm{SGD}, \mathrm{MomentumSGD}, \mathrm{Adam}\}$
Learning rate	$\mathrm{LogUnif}[10^{-5}, 10^{-2}]$

optimization methods described in Table 1. If the maximum singular values s is specified, we normalize weight matrices of graph convolution layers at every iteration so that their maximum singular values are normalized to s.

G.2.6 Hyperpramenters

Table 1 shows the set of hyperparameters from which we chose. Since we compute the representations of all nodes at once at each iteration, each epoch consists of 1 iteration. We employ Tree-structured Parzen Estimator [6] for hyperparameter optimization.

G.2.7 Implementation

We used Chainer Chemistry [1], which is an extension library for the deep learning framework Chainer [50], to implement GCN and Optuna [2] for hyperparameter tuning. We conducted experiments in a signel machine which has 2 Intel(R) Xeon(R) Gold 6136 CPU@3.00GHz (24 cores), 192 GB memory (DDR4), and 3 GPGPUs (NVIDIA Tesla V100). Our implementation achieved 68.1% with Dropout [48] (2 graph convolution layers) and 64.2% without Dropout (1 graph convolution layer) on the test dataset. These are slightly worse than the accuracy reported in [30], but are still comparable with it.

H Additional Experiment Results

H.1 Experiment of Section 6.1

We show the vector field V for various W in Figure 4 (**Case 1**) and Figure 5 (**Case 2**). Parameters other than W are same as those specified in Section 6.1 and Appendix G.1.

H.2 Experiment of Section 6.2

H.2.1 Predictive Accuracy

Figure 6 shows the comparison of predictive performance in terms the maximum singular value and layer size when the dataset is Noisy Cora 5000 (left) and Noisy Citeseer (right), respectively. Concrete values are available in Table 2.

H.2.2 Transition of Maximum Singular Values

Figure 7 – Figure 9 show the transition of weight of graph convolution layers during training when the dataset is Noisy Cora 2500, Noisy Cora 5000, and Noisy CiteSeer, respectively. We note that the result of 3-layered GCN from the Noisy Cora 2500 is identical to Figure 2 (right) of the main article.

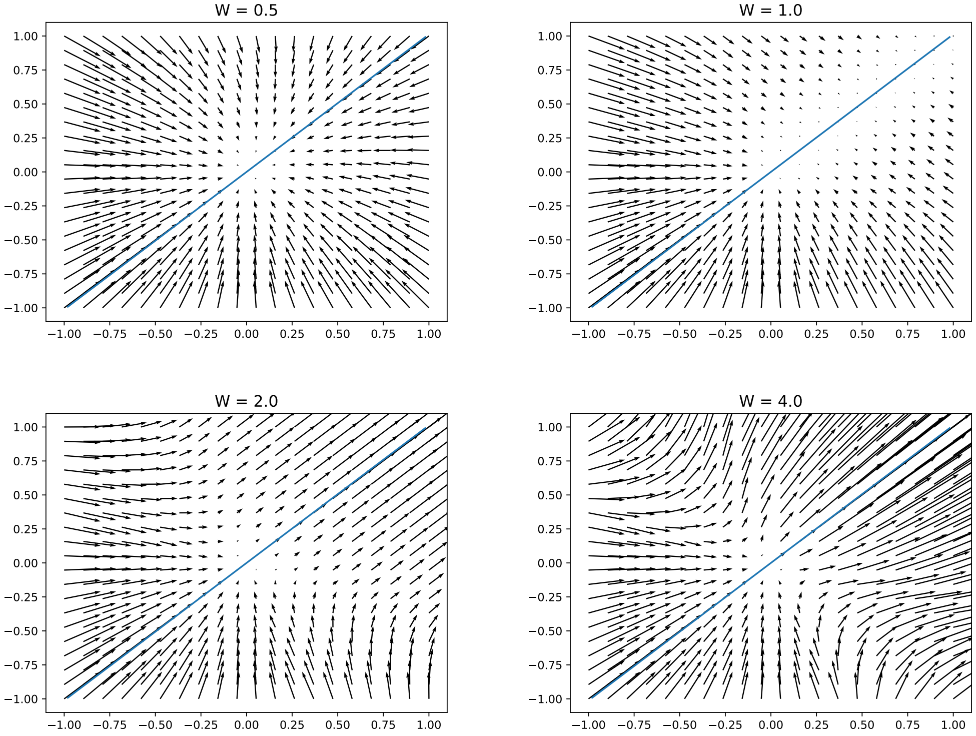

Figure 4: Vector field V for various W for **Case 1**. Top left: $W = 0.5$. Top right: $W = 1.0$. Bottom left: $W = 2.0$. Bottom right: $W = 4.0$.

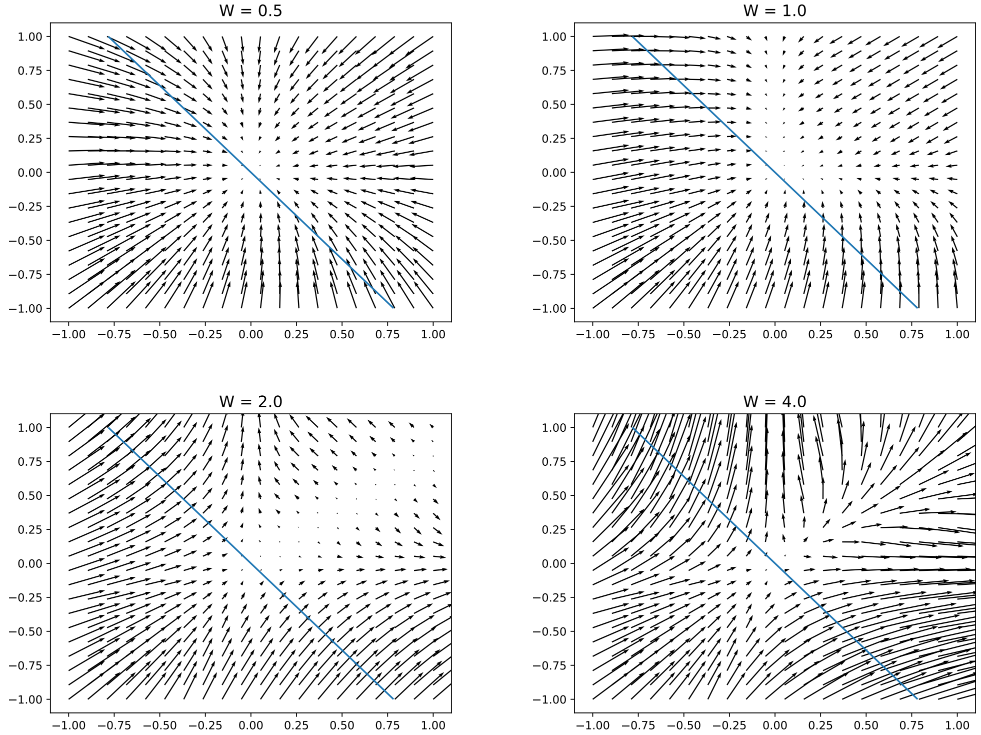

Figure 5: Vector field V for various W for **Case 2**. Top left: $W = 0.5$. Top right: $W = 1.0$. Bottom left: $W = 2.0$. Bottom right: $W = 4.0$.

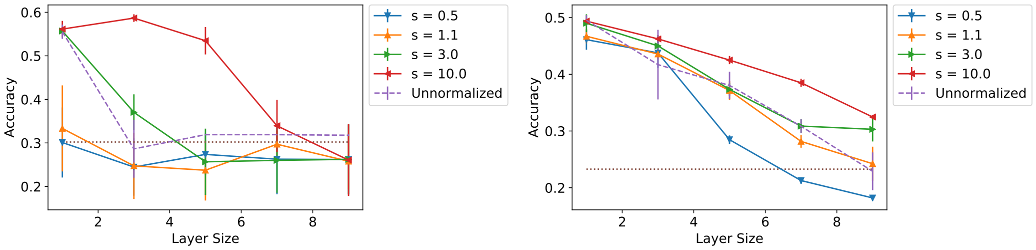

Figure 6: Effect of the maximum singular values of weights on predictive performance. Horizontal dotted lines indicate the chance rates (30.2% for Noisy Cora 5000 and 21.2% for Noisy CiteSeer). The error bar is the standard deviation of 3 trials. Left: Noisy Cora 5000. Right: Noisy CiteSeer. Best view in color.

Table 2: Comparison of performance in terms of maximum singular value of weights and layer size. “U" in the right most column indicates the accuracy of GCN without weight normalization. Top: Noisy CiteSeer. Bottom: Noisy Cora.

Noisy Cora 2500

	Maximum Singular Value				
Depth	1	1.05	3	10	U
1	0.389 ± 0.101	0.429 ± 0.090	0.552 ± 0.014	0.632 ± 0.007	0.587 ± 0.008
3	0.273 ± 0.051	0.309 ± 0.017	0.580 ± 0.058	0.661 ± 0.003	0.494 ± 0.041
5	0.319 ± 0.000	0.267 ± 0.059	0.462 ± 0.065	0.602 ± 0.004	0.326 ± 0.029
7	0.261 ± 0.076	0.262 ± 0.080	0.407 ± 0.021	0.501 ± 0.017	0.279 ± 0.129
9	0.261 ± 0.080	0.319 ± 0.000	0.284 ± 0.109	0.443 ± 0.014	0.319 ± 0.000

Noisy Cora 5000

	Maximum Singular Value				
Depth	1	1.1	3	10	U
1	0.301 ± 0.080	0.333 ± 0.099	0.557 ± 0.004	0.561 ± 0.019	0.555 ± 0.016
3	0.245 ± 0.066	0.247 ± 0.076	0.370 ± 0.041	0.587 ± 0.009	0.286 ± 0.066
5	0.274 ± 0.048	0.237 ± 0.070	0.257 ± 0.076	0.535 ± 0.031	0.319 ± 0.000
7	0.263 ± 0.080	0.297 ± 0.031	0.260 ± 0.074	0.339 ± 0.060	0.319 ± 0.000
9	0.262 ± 0.081	0.258 ± 0.064	0.262 ± 0.080	0.261 ± 0.082	0.318 ± 0.002

Noisy CiteSeer

	Maximum Singular Value				
Depth	0.5	1.1	3	10	U
1	0.461 ± 0.018	0.467 ± 0.012	0.490 ± 0.016	0.494 ± 0.006	0.495 ± 0.009
3	0.438 ± 0.027	0.436 ± 0.010	0.450 ± 0.019	0.462 ± 0.007	0.417 ± 0.061
5	0.285 ± 0.008	0.371 ± 0.016	0.373 ± 0.011	0.425 ± 0.007	0.380 ± 0.024
7	0.213 ± 0.006	0.282 ± 0.011	0.309 ± 0.012	0.385 ± 0.007	0.308 ± 0.012
9	0.182 ± 0.005	0.242 ± 0.030	0.303 ± 0.021	0.325 ± 0.003	0.229 ± 0.033

Noisy Cora 2500

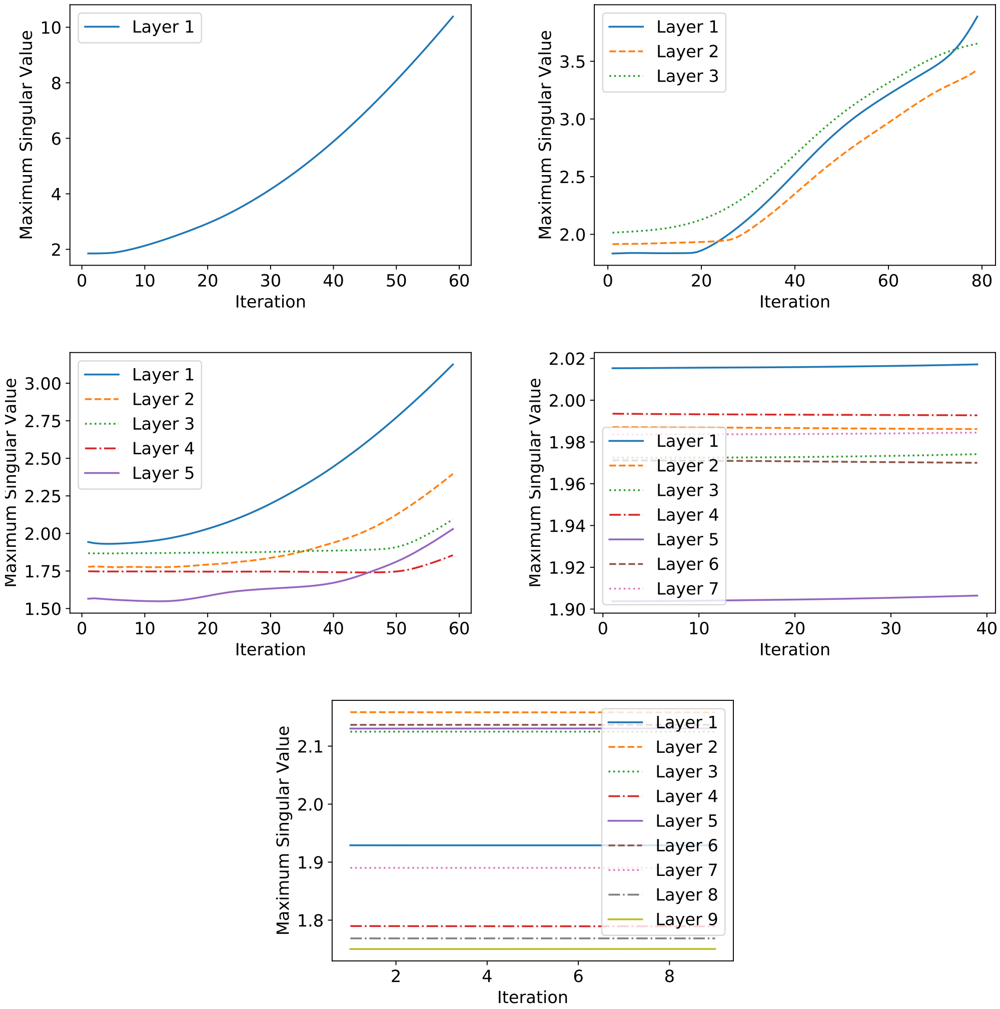

Figure 7: Transition of maximum singular values of GCN during training using Noisy Cora 2500. Top left: 1 layer. Top right: 5 layers. Bottom left: 7 layers. Bottom right: 9 layers.

Noisy Cora 5000

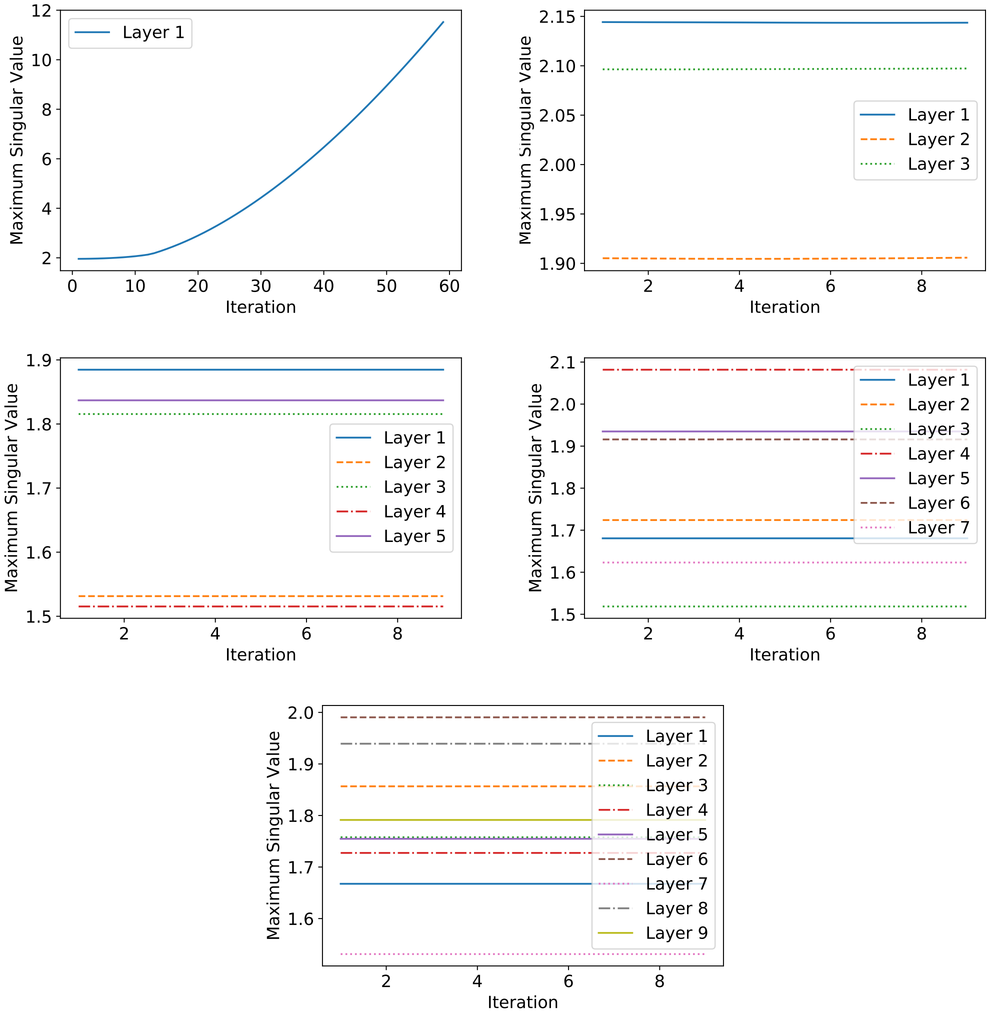

Figure 8: Transition of maximum singular values of GCN during training using Noisy Cora 5000. Top left: 1 layer. Top right: 3 layers. Middle left: 5 layers. Middle right: 7 layers. Bottom: 9 layers.

Noisy CiteSeer

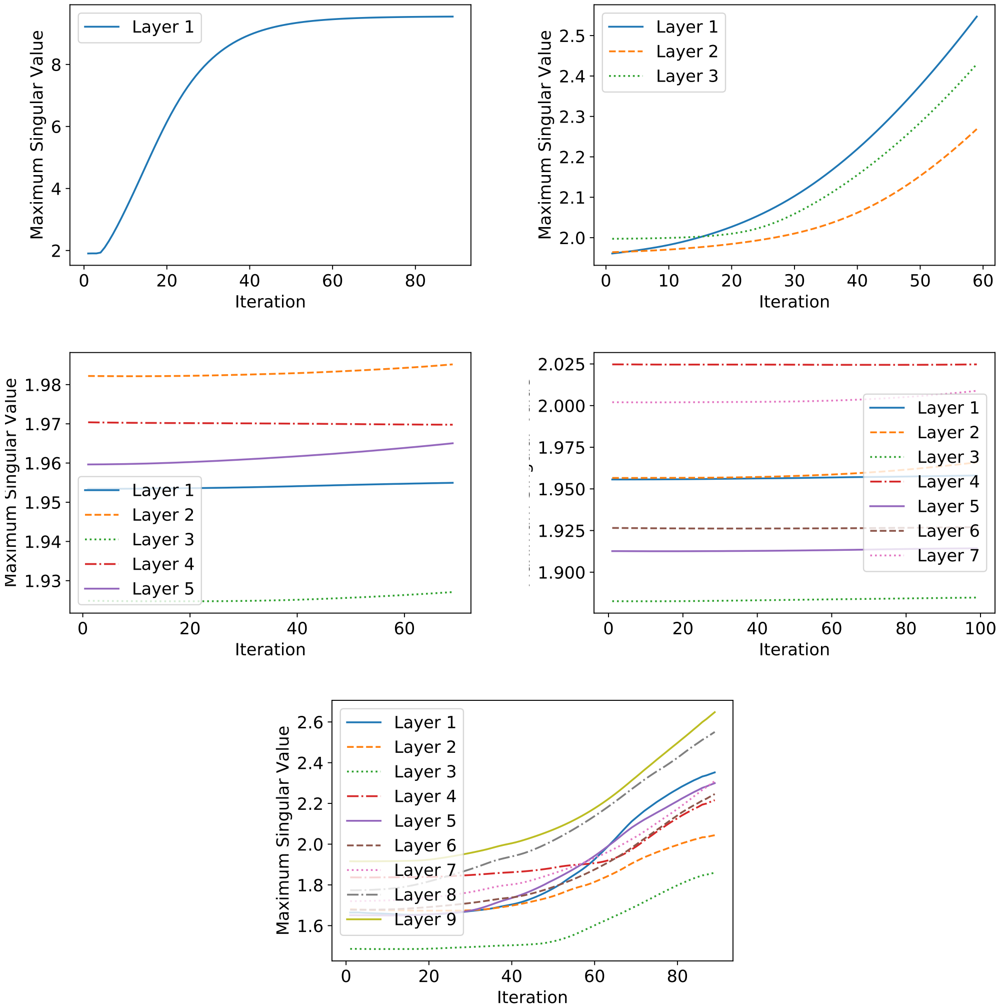

Figure 9: Transition of maximum singular values of GCN during training using Noisy CiteSeer. Top left: 1 layer. Top right: 3 layers. Middle left: 5 layers. Middle right: 7 layers. Bottom: 9 layers.