Generalization Error of Graph Neural Networks in the Mean-field Regime

Gholamali Aminian [*1] **Yixuan He** [*2] **Gesine Reinert** [1 2] **Łukasz Szpruch** [1 3] **Samuel N. Cohen** [1 4]

Abstract

This work provides a theoretical framework for assessing the generalization error of graph neural networks in the over-parameterized regime, where the number of parameters surpasses the quantity of data points. We explore two widely utilized types of graph neural networks: graph convolutional neural networks and message passing graph neural networks. Prior to this study, existing bounds on the generalization error in the over-parametrized regime were uninformative, limiting our understanding of over-parameterized network performance. Our novel approach involves deriving upper bounds within the mean-field regime for evaluating the generalization error of these graph neural networks. We establish upper bounds with a convergence rate of $O(1/n)$, where n is the number of graph samples. These upper bounds offer a theoretical assurance of the networks' performance on unseen data in the challenging over-parameterized regime and overall contribute to our understanding of their performance.

1. Introduction

Graph Neural Networks (GNNs) have received increasing attention due to their exceptional ability to extract meaningful representations from data structured in the form of graphs (Kipf & Welling, 2016; Veličković et al., 2017; Gori et al., 2005; Bronstein et al., 2017; Battaglia et al., 2018). Consequently, GNNs have achieved state-of-the-art performance across various domains, including but not limited to social networks (Hamilton et al., 2017; Fan et al., 2019), recommendation systems (Ying et al., 2018a; Wang et al., 2018) and computer vision (Monti et al., 2017). Despite the success of GNN models, explaining their empirical generalization performance remains a challenge within the domain of GNN learning theory.

A central concern in GNN learning theory is to understand the efficacy of a GNN learning algorithm when applied to data that it has not been previously exposed to. This evaluation is typically carried out by investigating the generalization error, which quantifies the disparity between the algorithm's performance, as assessed through a risk function, on the training data set and its performance on previously unseen data drawn from the same underlying distribution. This paper focuses on the generalization error for GNNs in scenarios with an overabundance of model parameters, potentially surpassing the number of available training data points, with a particular focus on tasks related to classifying graphs. In this regard, our research endeavors to shed light on the generalization performance exhibited by over-parameterized GNN models in the mean-field regime (Mei et al., 2018) for graph classification tasks.

We draw inspiration from the recent advancements in the mean-field perspective regarding the training of neural networks, as proposed in a body of literature (Mei et al., 2018; Chizat & Bach, 2018; Mei et al., 2019; Hu et al., 2019; Tzen & Raginsky, 2020). These works propose to frame the process of attaining optimal weights in one-hidden-layer neural networks as a sampling challenge. Within the mean-field regime, the learning algorithm endeavors to discern the optimal distribution within the parameter space, rather than solely concentrating on achieving optimal parameter values. A central question driving our research is how the mean-field perspective can provide further insights into the generalization behavior of over-parameterized GNN models. Our contributions here can be summarized as follows:

- We provide upper bounds on the generalization error for graph classification tasks in GNN models, including graph convolutional networks and message passing graph neural networks in the mean-field regime, via two different approaches: functional derivatives and Rademacher complexity based on symmetrized KL divergence.
- Using the approach based on functional derivatives, we derive an upper bound with convergence rate of $O(1/n)$, where n is the number of graph samples.
- The effects of different readout functions and aggregation functions on the generalization error of GNN

[*]Equal contribution [1]The Alan Turing Institute, London, United Kingdom [2]Department of Statistics, University of Oxford, Oxford, United Kingdom [3]School of Mathematics, University of Edinburgh [4]Mathematical Institute, University of Oxford, Oxford, United Kingdom. Correspondence to: Gholamali Aminian <gaminian@turing.ac.uk>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

models are studied.

- We carry out an empirical analysis on both synthetic and real-world data sets.

Notations: We adopt the following convention for random variables and their distributions in the sequel. A random variable is denoted by an upper-case letter (e.g., Z), its space of possible values is denoted with the corresponding calligraphic letter (e.g., $\mathcal{Z}$), and an arbitrary value of this variable is denoted with the lower-case letter (e.g., z). We denote the set of integers from 1 to N by $[N] \triangleq \{1, \ldots, N\}$; the set of measures over a space $\mathcal{X}$ with finite variance is denoted $\mathcal{P}(\mathcal{X})$. For a matrix $\mathbf{X} \in \mathbb{R}^{k\times q}$, we denote the $i-$th row of the matrix by $\mathbf{X}[i,:]$. The Euclidean norm of a vector $X \in \mathbb{R}^q$ is $\|X\|_2 := (\sum_{j=1}^q x_j^2)^{1/2}$. For a matrix $\mathbf{Y} \in \mathbb{R}^{k\times q}$, we let $\|\mathbf{Y}\|_\infty := \max_{1\leq j\leq k} \sum_{i=1}^q |\mathbf{Y}[j,i]|$ and $\|\mathbf{Y}\|_F := \sqrt{\sum_{j=1}^k \sum_{i=1}^q \mathbf{Y}^2[j,i]}$. We write δ_z for a Dirac measure supported at z. The KL-divergence between two probability distributions on $\mathbb{R}^d$ with densities $p(x)$ and $q(x)$, so that $q(x) > 0$ when $p(x) > 0$, is $\mathrm{KL}(p\|q) := \int_{\mathbb{R}^d} p(x)\log(p(x)/q(x))\mathrm{d}x$ (with $0/0 := 0$); the symmetrized KL divergence is $\mathrm{KL}_{\mathrm{sym}}(p\|q) := \mathrm{KL}(p\|q) + \mathrm{KL}(q\|p)$. A comprehensive notation table is provided in the Appendix (App.) A.

2. Our Model

2.1. Preliminaries

We first introduce the functional linear derivative, see (Cardaliaguet et al., 2019).

Definition 2.1. (Cardaliaguet et al., 2019) A functional $U : \mathcal{P}(\mathbb{R}^n) \to \mathbb{R}$ admits a linear derivative if there is a map $\frac{\delta U}{\delta m} : \mathcal{P}(\mathbb{R}^n) \times \mathbb{R}^n \to \mathbb{R}$ which is continuous on $\mathcal{P}(\mathbb{R}^n)$, such that $|\frac{\delta U}{\delta m}(m,a)| \leq \mathrm{C}(1+|a|^2)$ for a constant $C \in \mathbb{R}^+$ and, for all $m, m' \in \mathcal{P}(\mathbb{R}^n)$, it holds that

$$U(m') - U(m) = \int_0^1 \int_{\mathbb{R}^n} \frac{\delta U}{\delta m}(m_\lambda, a)\,(m'-m)(da)\,\mathrm{d}\lambda,$$

where $m_\lambda = m + \lambda(m'-m)$.

Graph data samples and learning algorithm: We consider graph classification for undirected graphs with N nodes which have no self-loops or multiple edges. Inputs to GNNs are graph samples, which are comprised of their node features and graph adjacency matrices. We denote the space of all adjacency matrices and node feature matrices for a graph classification task with maximum number of nodes $N_{\max}$, maximum node degree $d_{\max}$ and minimum node degree $d_{\min}$ by $\mathcal{A}$ and $\mathcal{F}$, respectively. The input pair of a graph sample with N nodes is denoted by $\mathbf{X} = (\mathbf{F}, \mathbf{A})$, where $\mathbf{F} \in \mathcal{F} \subset \mathbb{R}^{N\times k}$ denotes a node feature matrix with feature dimension k per node and $\mathbf{A} \in \mathcal{A} \subset \{0,1\}^{N\times N}$ denotes the graph adjacency matrix. The GNN output (label) is denoted by $y \in \mathcal{Y}$ where $\mathcal{Y} = \{-1, 1\}$ for binary classification. Define $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X} := \mathcal{F} \times \mathcal{A}$. Let $\mathbf{Z}_n = \{Z_i\}_{i=1}^n \in \mathcal{Z}$ denote the training set, where the $i-$th graph sample is $Z_i = (\mathbf{X}_i = (\mathbf{F}_i, \mathbf{A}_i), Y_i)$. We assume that $\mathbf{Z}_n$ are i.i.d. random vectors such that $Z_i \sim \mu \in \mathcal{P}(\mathcal{Z})$. Its empirical measure, $\mu_n := \frac{1}{n}\sum_{i=1}^n \delta_{Z_i}$, is a random element with values in $\mathcal{P}(\mathcal{Z})$. We also assume that a sample $\widehat{Z}_1 = (\widehat{\mathbf{X}}_1, \widehat{Y}_1)$ is available, and this sample is i.i.d. with respect to $\mathbf{Z}_n$. We set $\mu_{n,(1)} := \mu_n + \frac{1}{n}(\delta_{\widehat{Z}_1} - \delta_{Z_1})$. We are interested in learning a parameterized model (or function), $\mathfrak{f}_w : \mathcal{X} \mapsto \mathcal{Y}$ for some parameters $w \in \mathcal{W}$, where $\mathcal{W}$ is the parameter space of the model. Inspired by (Aminian et al., 2023), we define a learning algorithm as a map $m : \mathcal{P}(\mathcal{Z}) \to \mathcal{P}(\mathcal{W}); \mu_n \mapsto m(\mu_n)$, which outputs a probability distribution (measure) on $\mathcal{W}$. For example, when learning using stochastic gradient descent (SGD), the input is random samples, whereas the output is a probability measure on the space of parameters, which are used in SGD.[1]

Graph filters: Graph filters are linear functions of the adjacency matrix, $\mathbf{A}$, defined by $G : \mathcal{A} \mapsto \mathbb{R}^{N\times N}$ where N is the size of the input graph, see (Defferrard et al., 2016). Graph filters model the aggregation of node features in a graph neural network. For example, the symmetric normalized graph filter proposed by (Kipf & Welling, 2016) is $G(\mathbf{A}) = \tilde{L} := \tilde{D}^{-1/2}\tilde{\mathbf{A}}\tilde{D}^{-1/2}$ where $\tilde{\mathbf{A}} = I + \mathbf{A}$, $\tilde{D}$ is the degree-diagonal matrix of $\tilde{\mathbf{A}}$, and I is the identity matrix. Another normalized filter, a.k.a. random walk graph filter (Xu et al., 2019), is $G(\mathbf{A}) = D^{-1}\mathbf{A} + I$, where D is the degree-diagonal matrix of $\mathbf{A}$. The mean-aggregator is also a well-known aggregator defined as $G(\mathbf{A}) = \tilde{D}^{-1}(\mathbf{A} + I)$. The sum-aggregator graph filter is defined by $G(\mathbf{A}) = (\mathbf{A} + I)$. Let us define $R_{\max}(G(\mathbf{A}))$ as the maximum rank of graph filter $G(\mathbf{A})$ over all adjacency matrices in the data set. We let

$$G_{\max} = \min(\|G(\mathcal{A})\|_\infty^{\max}, \|G(\mathcal{A})\|_F^{\max}), \qquad (1)$$

where $\|G(\mathcal{A})\|_\infty^{\max} = \max_{\mathbf{A}_q\in\mathcal{A},\mu(f_q,\mathbf{A}_q)>0} \|G(\mathbf{A}_q)\|_\infty$ and $\|G(\mathcal{A})\|_F^{\max} = \max_{\mathbf{A}_q\in\mathcal{A},\mu(f_q,\mathbf{A}_q)>0} \|G(\mathbf{A}_q)\|_F$.

2.2. Problem Formulation

In this study, we investigate two prominent GNN architectures: one-hidden-layer **Graph Convolutional Networks** (GCN) by Kipf & Welling (2016), and one-hidden-layer **Message Passing Graph Neural Networks** (MPGNN) by Dai et al. (2016) and Gilmer et al. (2017). The GCN model is constructed by summating multiple neurons, while MPGNN relies on the summation of multiple Message Passing and Updating (MPU) units.

[1] Due to a probability measure for the random initialization, the final output is also a probability distribution over space of parameters.

Neuron unit: Here we define the one-hidden-layer GCN (Zhang et al., 2020) consisting of h neuron (hidden) units. The i–th neuron unit is defined by parameters $W_c[i,:] := (W_{1,c}(i), W_{2,c}(i)) \in \mathcal{W}_c \subset \mathcal{S}^{k+1}$, where $\mathcal{W}_c$ is the parameter space of one neuron unit, $W_{1,c} \in \mathcal{S}^k$ are parameters connecting the aggregated node features to the neuron unit, $W_{2,c} \in \mathcal{S}$ is the parameter connecting the output of neuron unit to output and $\mathcal{S} \subset \mathbb{R}$ is a bounded set. We define $w_{2,c} := \sup_{W_{2,c}\in\mathcal{S}} W_{2,c}$ and $\|W_{1,c,m}\|_2 = \sup_{W_{1,c}\in\mathcal{S}^k} \|W_{1,c}\|_2$. An aggregation layer aggregates the features of neighboring nodes of each node via a graph filter, $G(\cdot)$. In a GCN, the output of the $i-$th neuron unit (Kipf & Welling, 2016) for the $j-$th node in a graph sample $\mathbf{X}_q = \{\mathbf{F}_q, \mathbf{A}_q\}$ is

$$\phi_c(W_c[i,:], G(\mathbf{A}_q)[j,:]\mathbf{F}_q) := W_{2,c}(i)\varphi\big(G(\mathbf{A}_q)[j,:]\mathbf{F}_q \cdot W_{1,c}(i)\big),$$

where $\varphi : \mathbb{R} \mapsto \mathbb{R}$ is the activation function. The empirical measure over the parameters of the neurons is then $m_h^c(\mu_n) := \frac{1}{h}\sum_{i=1}^h \delta_{(W_{1,c}(i), W_{2,c}(i))}$, where $\{(W_{1,c}(i), W_{2,c}(i))\}_{i=1}^h$ depend on μ_n.

MPU unit: Several structures of MPGNNs have been proposed by (Li et al., 2022; Liao et al., 2020) and (Garg et al., 2020) for analyzing the generalization error, with inspiration drawn from previous works such as (Dai et al., 2016; Gilmer et al., 2017). In this paper, we utilize the MPGNN model introduced in (Li et al., 2022) and (Liao et al., 2020). The parameters for the $i-$th Message Passing and Updating unit are denoted as $W_p[i,:] := \big(W_{1,p}(i), W_{2,p}(i), W_{3,p}(i)\big) \in \mathcal{W}_p \subset \mathcal{S}^{2k+1}$, where $W_{2,p} \in \mathcal{S}$, $W_{1,p}, W_{3,p} \in \mathcal{S}^k$, and $\mathcal{S} \subset \mathbb{R}$ is a bounded set. We define $w_{2,p} := \sup_{W_{2,p}\in\mathcal{S}} W_{2,p}$, $\|W_{1,p,m}\|_2 = \sup_{W_{1,p}\in\mathcal{S}^k} \|W_{1,p}\|_2$ and $\|W_{3,p,m}\|_2 = \sup_{\|W_{3,p}\|_2\in\mathcal{S}^k} \|W_{3,p}\|_2$. The output of the $i-$th MPU unit for the $j-$th node in a graph sample $\mathbf{X}_q = (\mathbf{F}_q, \mathbf{A}_q)$ with graph filter $G(\cdot)$ is

$$\phi_p(W_p[i,:], G(\mathbf{A}_q)[j,:]\mathbf{F}_q) := W_{2,p}(i)\kappa\Big(\mathbf{F}_q[j,:] \cdot W_{3,p}(i) + \rho(G(\mathbf{A}_q)[j,:]\zeta(\mathbf{F}_q)) \cdot W_{1,p}(i)\Big),$$

where the non-linear functions $\zeta : \mathbb{R}^{N\times k} \mapsto \mathbb{R}^{N\times k}$, $\rho : \mathbb{R}^k \mapsto \mathbb{R}^k$, and $\kappa : \mathbb{R} \mapsto \mathbb{R}$ may be chosen from non-linear options such as Tanh or Sigmoid. For an MPGNN the parameter space is $\mathcal{W}_p = \mathbb{R}^{2k+1}$ and the corresponding empirical measure is $m_h^p(\mu_n) := \frac{1}{h}\sum_{i=1}^h \delta_{(W_{1,p}(i), W_{2,p}(i), W_{3,p}(i))}$, where $\{(W_{1,p}(i), W_{2,p}(i), W_{3,p}(i))\}_{i=1}^h$ depend on μ_n.

One-hidden-layer generic GNN model: Inspired by the neuron unit in GCNs and the MPU unit in MPGNNs, we introduce a generic model for GNNs, which can be applied to GCNs and MPGNNs. For each unit in a generic GNN with parameter W, which belongs to the parameter space $\mathcal{W}$, the empirical measure $m_h(\mu_n) = \frac{1}{h}\sum_{i=1}^h \delta_{W_i}$ is a measure on the parameter space of a unit. We denote the unit function for the j-th node by

$$(W, G(\mathbf{A})[j,:]\mathbf{F}) \mapsto \phi(W, G(\mathbf{A})[j,:]\mathbf{F}).$$

The output of the network for the $j-$th node of a graph sample, $\mathbf{X}_q = (\mathbf{F}_q, \mathbf{A}_q)$, can then be represented as

$$\begin{aligned} &\frac{1}{h}\sum_{i=1}^h \phi\left(W_i, G(\mathbf{A}_q)[j,:]\mathbf{F}_q\right) \\ &= \int \phi\left(w, G(\mathbf{A}_q)[j,:]\mathbf{F}_q\right) m_h(\mu_n)(\mathrm{d}w) \\ &= \mathbb{E}_{W\sim m_h(\mu_n)}[\phi\left(W, G(\mathbf{A}_q)[j,:]\mathbf{F}_q\right)], \end{aligned} \tag{2}$$

where W_i is the parameter of the $i-$th unit. The final step is the pooling of the node features across all nodes for each graph sample as the output of generic GNNs. For this purpose, we introduce the readout function (a.k.a. pooling layer) $\Psi : \mathcal{P}(\mathcal{W}) \times \mathcal{X} \mapsto \mathbb{R}$ as follows:

$$\Psi(m_h(\mu_n), \mathbf{X}_q) := \psi\Big(\sum_{j=1}^N \mathbb{E}_{W\sim m_h(\mu_n)}[\phi\left(W, G(\mathbf{A}_q)[j,:]\mathbf{F}_q\right)]\Big), \tag{3}$$

where $\psi : \mathbb{R} \mapsto \mathbb{R}$. In this work, we consider the mean-readout and sum-readout functions by taking $\psi(x) = \frac{x}{N}$ and $\psi(x) = x$, respectively. For a GCN and an MPGNN, the outputs of the model after aggregation are, respectively:

$$\Psi_c\left(m_h^c(\mu_n), \mathbf{X}_q\right) := \psi\Big(\sum_{j=1}^N \mathbb{E}_{W_c\sim m_h^c(\mu_n)}\left[\phi_c(W_c, G(\mathbf{A}_q)[j,:]\mathbf{F}_q)\right]\Big), \tag{4}$$

$$\Psi_p\left(m_h^p(\mu_n), \mathbf{X}_q\right) := \psi\Big(\sum_{j=1}^N \mathbb{E}_{W_p\sim m_h^p(\mu_n)}\Big[\phi_p(W_p, G(\mathbf{A}_q)[j,:]\mathbf{F}_q)\Big]\Big). \tag{5}$$

Loss function: With $\mathcal{Y}$ the label space, the loss function $\ell : \mathbb{R} \times \mathcal{Y} \to \mathbb{R}$ is denoted as $\ell(\widehat{y}, y)$, where $\widehat{y}$ is defined in (4) and (5) for a GCN and for an MPGNN, respectively; the loss function is assumed to be convex. For binary classification, we take loss functions of the form $\ell(\widehat{y}, y) = \mathfrak{h}(y\widehat{y})$, where $\mathfrak{h}(\cdot)$ represents a margin-based loss function (Bartlett et al., 2006); examples include the logistic loss function $\mathfrak{h}(y\widehat{y}) = \log(1 + \exp(-y\widehat{y}))$, the exponential loss function $\mathfrak{h}(y\widehat{y}) = \exp(-y\widehat{y})$, and the square loss function $(1 - y\hat{y})^2$. Liao et al. (2020) and Garg et al. (2020) studied a γ-margin loss inspired by Neyshabur et al. (2018).

Over-parameterized one-hidden-layer generic GNN: As discussed in (Hu et al., 2019; Mei et al., 2018), based on stochastic gradient descent dynamics in one-hidden-layer neural networks for a large number of hidden neurons and a

small step size, the random empirical measure can be well approximated by a probability measure. The same behavior (due to the law of large numbers) also holds for one-hidden-layer GNNs. Therefore, for an over-parameterized one-hidden-layer generic GNN, as the number of hidden units h (width of the hidden layer) increases, under some assumptions the distribution $m_h(\mu_n)$ converges to a continuous distribution over the parameter space of the unit.

True and empirical risks: The true risk function (expected loss) based on the loss function ℓ, with a data measure μ and a parameter measure $m \in \mathcal{P}(\mathcal{W})$, is

$$\mathrm{R}(m, \mu) := \int_{\mathcal{X}\times\mathcal{Y}} \ell\Big(\Psi(m, \mathbf{x}), y\Big)\mu(\mathrm{d}\mathbf{x}, \mathrm{d}y)\,. \tag{6}$$

When the parameter measure is $m(\mu_n)$, which depends on an empirical measure, we obtain for example $\mathrm{R}(m^c(\mu_n), \mu)$ and $\mathrm{R}(m^p(\mu_n), \mu)$, as the true risks for a GCN and an MPGNN, respectively, given the observations which are encoded in the empirical measure μ_n.

The empirical risk is given by

$$\mathrm{R}(m, \mu_n) := \frac{1}{n}\sum_{i=1}^{n} \ell\Big(\Psi(m, \mathbf{x}_i), y_i\Big). \tag{7}$$

Note that $\mathrm{R}(m^c(\mu_n), \mu_n)$ and $\mathrm{R}(m^p(\mu_n), \mu_n)$ are empirical risks for a GCN and an MPGNN, respectively.

Generalization Error: We would like to study the performance of the model trained with the empirical data set $\mathbf{Z}_n = \{(\mathbf{X}_i, Y_i)\}_{i=1}^n$ and evaluated against the true measure μ, using the expected generalization error

$$\overline{\mathrm{gen}}(\mu_n), \mu) := \mathbb{E}_{\mathbf{Z}_n}\Big[\mathrm{gen}(\mu_n), \mu)\Big], \tag{8}$$

where $\mathrm{gen}(\mu_n), \mu) = \mathrm{R}(m(\mu_n), \mu) - \mathrm{R}(m(\mu_n), \mu_n)$. The generalization errors for a GCN and an MPGNN are, respectively, $\overline{\mathrm{gen}}(m^c(\mu_n), \mu)$ and $\overline{\mathrm{gen}}(m^p(\mu_n), \mu)$.

3. Related Works

Graph Classification and Generalization Error: Different learning theory methods, e.g., VC-dimension, Rademacher complexity, and PAC-Bayesian, have been used to understand the generalization error of graph classification tasks. In particular, (Scarselli et al., 2018) studied the generalization error via VC-dimension analysis. (Garg et al., 2020) provided some data-dependent generalization error bounds for MPGNNs for binary graph classification via VC-dimension analysis. The PAC-Bayesian approach has also been applied to two GNN models, including GCNs and MPGNNs, but only for hidden layers of bounded width, by (Liao et al., 2020) and (Ju et al., 2023). Considering a large random graph model, (Maskey et al., 2022) proposed a continuous MPGNN and provided a generalization error upper bound for graph classification. An extension of the neural tangent kernel for GNN as a graph neural tangent kernel was proposed by Du et al. (2019), where a high-probability upper bound on the true risk of their approach is derived. Our work differs from the above approaches as we study the generalization error of graph classification tasks under an over-parameterized regime for GCNs and MPGNNs where the width of the hidden layer is infinite. A detailed comparison is provided in Sec. 4.5.

Generalization and over-parameterization: In the under-parameterized regime, that is, when the number of model parameters is significantly less than the number of training data points, the theory of the generalization error has been well-developed (Vapnik & Chervonenkis, 2015; Bartlett & Mendelson, 2002). However, in the over-parameterized regime, this theory fails. Indeed, deep neural network models can achieve near-zero training loss and still perform well on out-of-sample data (Belkin et al., 2019; Spigler et al., 2019; Bartlett et al., 2021). There are three primary strategies for studying and modeling the over-parameterized regime: the neural tangent kernel (NTK) (Jacot et al., 2018b), random feature (Mei & Montanari, 2022), and mean-field (Mei et al., 2018). The NTK approach, also known as lazy training, for one-hidden-layer neural networks, utilizes the fact that a one-hidden-layer neural network can be expressed as a linear model under certain assumptions. The random feature model is similar to the NTK approach but assumes constant weights in the single hidden layer of the neural network. The mean-field approach utilizes the exchangeability of neurons to work with the distribution of a single neuron's parameters. Recently, Nishikawa et al. (2022), Nitanda et al. (2022) and Aminian et al. (2023) studied the generalization error of the one-hidden layer neural network in the mean-field regime, via Rademacher complexity analysis and differential calculus over the measure space, respectively. Nevertheless, the analysis of over-parameterization within GNNs remains to be thoroughly unexplored. Recently, inspired by NTK, (Du et al., 2019) proposed a graph neural tangent kernel (GNTK) method, which is equivalent to some over-parameterized GNN models with some modifications. However, GNTK is different from GCNs and MPGNNs. For a deeper understanding of the over-parameterized regime for such GNN models, i.e., GCNs and MPGNNs, we need to delve into the generalization error analysis. For this endeavor, the mean-field methodology for GNN models is promising for an analytical approach, especially when compared to NTK and random feature models (Fang et al., 2021). Hence, we use this methodology.

4. The Generalization Error of KL-Regularized Empirical Risk Minimization

In this section, we derive an upper bound on the solution of the KL-regularized risk minimization problem. The KL-

regularised empirical risk minimizer for parameter measure m on $\mathcal{W}$ and empirical measure μ_n is defined as

$$\mathcal{V}^{\alpha}(m, \mu_n) = \mathrm{R}(m, \mu_n) + \frac{1}{\alpha}\mathrm{KL}(m\|\pi), \tag{9}$$

where $\pi(w)$, $w \in \mathcal{W}$, is a prior on the parameter measure with (assumed finite) KL divergence between parameter measure m and prior measure, i.e., $\mathrm{KL}(m\|\pi) < \infty$, and α is a parameter (the inverse temperature). It has been shown in (Hu et al., 2019) that under a convexity assumption on the risk, a minimizer, denoted by $\mathrm{m}^{\alpha}(\mu_n)$, of $\mathcal{V}^{\alpha}(\cdot, \mu_n)$, exists, is unique, and satisfies the equation,

$$\mathrm{m}^{\alpha}(\mu_n) = \frac{\pi}{S_{\alpha,\pi}(\mu_n)} \exp\left\{ -\alpha \frac{\delta \mathrm{R}(\mathrm{m}^{\alpha}, \mu_n, w)}{\delta m} \right\}, \tag{10}$$

where $S_{\alpha,\pi}(\mu_n)$ is the normalizing constant to ensure that $\mathrm{m}^{\alpha}(\mu_n)$ integrates to 1.

For an example of a training algorithm that yields this parameter measure, consider one-hidden-layer (graph) neural networks in the context of the mean-field limit $\lim_{h\to\infty} m_h(\mu_n)$. We denote the Gibbs measure related to an over-parameterized one-hidden-layer GCN and an over-parameterized one-hidden-layer MPGNN by $\mathrm{m}^{\alpha,c}(\mu_n) \in \mathcal{P}(\mathcal{W}_c)$ and $\mathrm{m}^{\alpha,p}(\mu_n) \in \mathcal{P}(\mathcal{W}_p)$, respectively. We also consider $W_c = (W_{1,c}, W_{2,c})$ and $W_p = (W_{1,p}, W_{2,p}, W_{3,p})$.

The conditional expectation in (4) and (5) taken over parameters distributed according to $\mathrm{m}^{\alpha}(\mu_n)$, given the empirical measure μ_n, corresponds to taking an average over samples drawn from the learned parameter measure. This connection has been exploited in the mean-field models of one-hidden-layer neural networks (Hu et al., 2019; Mei et al., 2019; Tzen & Raginsky, 2020).

The following assumptions are needed for our main results.

Assumption 4.1 (Loss function)**.** The gradient of the loss function $(\hat{y}, y) \mapsto \ell(\hat{y}, y)$ with respect to $\hat{y}$ is continuous and uniformly bounded for all $\hat{y}, y \in \mathcal{Y}$, i.e., there is a constant $M_{\ell'}$ such that $|\partial_{\hat{y}}\ell(\hat{y}, y)| \leq M_{\ell'}$.[2] Furthermore, we assume that the loss function is convex with respect to $\hat{y}$.

Assumption 4.2 (Bounded loss function)**.** There is a constant $M_{\ell} > 0$ such that the loss function, $(\hat{y}, y) \mapsto \ell(\hat{y}, y)$ satisfies $0 \leq \ell(\hat{y}, y) \leq M_{\ell}$.

Assumption 4.3 (Unit function)**.** The unit function $(w, G(\mathbf{A})[j,:]\mathbf{f}) \mapsto \phi(w, G(\mathbf{A})[j,:]\mathbf{f})$ with graph filter $G(\cdot)$ is uniformly bounded, i.e, there is a constant M_{ϕ} such that $\sup_{w\in\mathcal{W},\mathbf{f}\in\mathcal{F},\mathbf{A}\in\mathcal{A}} |\phi(w, G(\mathbf{A})[j,:]\mathbf{f})| \leq M_{\phi}$ for all $j \in [N]$ and $\mathbf{f} \in \mathcal{F}$.

Assumption 4.4 (Readout function)**.** There is a constant L_{ψ} such that $|\psi(x_1) - \psi(x_2)| \leq L_{\psi}|x_1 - x_2|$ for all $x_1, x_2 \in \mathbb{R}$, and $\psi(0) = 0$.

[2]For an L-Lipschitz-continuous loss function, $M_{\ell'} = L$.

For mean-readout and sum-readout functions, we have $L_{\psi} = 1/N$ and $L_{\psi} = 1$, respectively.

Assumption 4.5 (Bounded node features)**.** For every graph, the node features are contained in an ℓ_2-ball of radius B_f. In particular, $\|\mathbf{F}[i;]\|_2 \leq B_f$ for all $i \in [N]$.

Assumptions Discussion: The requirements of Lipschitz continuity and convexity (Assumption 4.1) for the loss function are met by both logistic and square losses, provided the inputs and unit functions are bounded. The condition of boundedness for unit functions (Assumption 4.3) holds under the premise of bounded inputs and a bounded parameter space for either neuron units or MPU units. Notably, with a bounded activation function, bounding parameters only in the last layer suffices. The sum and mean readout functions satisfy the conditions of Lipschitz continuity and are zero-centered (Assumption 4.4). The bounded feature (Assumption 4.5) is a widely accepted assumption, which can be achieved through input normalization. Note that, similar assumptions have been imposed in the generalization error analysis of graph neural network literature, as discussed in (Liao et al., 2020), (Ju et al., 2023), (Garg et al., 2020) and (Maskey et al., 2022) to facilitate the theoretical analysis. While these assumptions can be relaxed, we adhere to the current forms for clarity and simplicity.

To establish an upper bound on the generalization error of the generic GNN, we employ two approaches: **(a)** we compute an upper bound for the expected generalization error using functional derivatives in conjunction with symmetrized KL divergence, and **(b)** we establish a high-probability upper bound using Rademacher Complexity along with symmetrized KL divergence.

4.1. Generalization Error via Functional Derivative

We first derive two propositions to obtain intermediate bounds on the generalization error in terms of KL divergence. The proofs of this section are provided in App. C. The notation is provided in Sec. 2.1.

Proposition 4.6. *Let Assumptions 4.1, 4.3, and 4.4 hold. Let $\mathrm{m}(\mu_n) \in \mathcal{P}(\mathcal{W})$. Then, for the generalization error of a generic GNN model,*

$$\begin{aligned}\overline{\mathrm{gen}}(\mathrm{m}(\mu_n), \mu) \leq &\left(M_{\ell'} L_{\psi} N_{\max} M_{\phi}/\sqrt{2}\right) \\ &\times \mathbb{E}_{\mathbf{Z}_n, \widehat{Z}_1}\left[\sqrt{\mathrm{KL}\big(\mathrm{m}(\mu_n)\|\mathrm{m}(\mu_{n,(1)})\big)}\right],\end{aligned} \tag{11}$$

where $N_{\max}$ is the maximum number of nodes among all graph samples.

Remark 4.7. For the mean-readout function, $L_{\psi} = \frac{1}{N}$ and the upper bound in Proposition 4.6 can be represented as,

$$\begin{aligned}\overline{\mathrm{gen}}(\mathrm{m}(\mu_n), \mu) \leq &\left(M_{\ell'} M_{\phi}/\sqrt{2}\right) \\ &\times \mathbb{E}_{\mathbf{Z}_n, \widehat{Z}_1}\left[\sqrt{\mathrm{KL}\big(\mathrm{m}(\mu_n)\|\mathrm{m}(\mu_{n,(1)})\big)}\right].\end{aligned} \tag{12}$$

Proposition 4.6 holds for all $\mathrm{m}(\mu_n) \in \mathcal{P}_2(\mathcal{W})$. Using the functional derivative, the following proposition provides a lower bound on the generalization error of the Gibbs measure from (10).

Proposition 4.8. *Let Assumptions 4.1 hold. Then, for the Gibbs measure $m^\alpha(\mu_n)$ in (10),*

$$\overline{\mathrm{gen}}(\mathrm{m}^\alpha(\mu_n), \mu) \geq \frac{n}{2\alpha}\mathbb{E}_{\mathbf{Z}_n, \widehat{Z}_1}\left[\mathrm{KL}_{\mathrm{sym}}\left(\mathrm{m}^\alpha(\mu_n) \| \mathrm{m}^\alpha(\mu_{n,(1)})\right)\right].$$

Using Proposition 4.6 for the Gibbs measure and Proposition 4.8, we can derive the following upper bound on the generalization error for a generic GNN.

Theorem 4.9 (Generalization error and generic GNNs)**.** *Let Assumptions 4.1, 4.3 and 4.4 hold. Then, for the generalization error of the Gibbs measure $\mathrm{m}^\alpha(\mu_n)$ in (10),*

$$\overline{\mathrm{gen}}(\mathrm{m}^\alpha(\mu_n), \mu) \leq \frac{\alpha C}{n},$$

where $C = (M_{\ell'} M_\phi L_\psi N_{\max})^2$ does not depend on n.

Remark 4.10 (Readout-function comparison). In Theorem 4.9, the mean-readout function with $L_\psi = 1/N$ exhibits a tighter upper bound when compared to the sum-readout function with $L_\psi = 1$. For the mean-readout function, in Theorem 4.9 we have $C = (M_{\ell'} M_\phi)^2$.

Remark 4.11 (Comparison with (Aminian et al., 2023)). In (Aminian et al., 2023, Theorem 3.4), an exact representation of the generalization error in terms of the functional derivative of the parameter measure with respect to the data measure, i.e., $\frac{\delta \mathrm{m}}{\delta \mu}(\mu_n, z)(\mathrm{d}w)$, is provided. Then, for the Gibbs measure, in (Aminian et al., 2023, Lemma D.4), an upper bound on the generalization error requires to compute $\frac{\delta \mathrm{m}}{\delta \mu}(\mu_n, z)(\mathrm{d}w)$, the functional derivative of parameter measure for the data measure. Instead, we use the convexity of the loss function concerning the parameter measure, Proposition 4.8, and the general upper bound on the generalization error, Proposition 4.6, to establish an upper bound on the generalization error of the Gibbs measure, Theorem 4.9, via symmetrized KL divergence properties. Not only is this simpler, but, more importantly, our framework enables us to derive non-trivial upper bounds on the generalization error of the graph neural network (GNN) based on specific graph properties, such as $d_{\max}$, $d_{\min}$, and $R_{\max}$, e.g., see Remarks 4.20 and 4.21.

Remark 4.12 (Comparison with (Aminian et al., 2021)). In (Aminian et al., 2021, Theorem 1), an exact characterization of the expected generalization error of the Gibbs algorithm in terms of the symmetrized KL information (Aminian et al., 2015) is derived. Then, Aminian et al. (2021) derived an upper bound on the expected generalization error of the Gibbs algorithm. However, as discussed in (Aminian et al., 2023, Appendix F), the Gibbs algorithm is different from the Gibbs measure and the generalization error analysis of the Gibbs algorithm can not be applied to the mean-field regime.

4.2. Generalization Error via Rademacher Complexity

Inspired by the Rademacher complexity analysis in (Nishikawa et al., 2022; Nitanda et al., 2022), we derive a high-probability upper bound on the generalization error in the mean-field regime.

Proposition 4.13 (Upper bound on the symmetrized KL divergence)**.** *Under Assumptions 4.1, 4.3, and 4.4,*

$$\mathrm{KL}_{\mathrm{sym}}(\mathrm{m}^\alpha(\mu_n) \| \pi) \leq 2 N_{\max} M_\phi M_{\ell'} L_\psi \alpha.$$

Combining Proposition 4.13 with (Chen et al., 2020, Lemma 5.5), uniform bound and Talagrand's contraction lemma (Mohri et al., 2018), we can derive a high-probability upper bound on the generalization error via Rademacher complexity analysis. The details are provided in App. D.

Theorem 4.14 (Generalization error upper bound via Rademacher complexity)**.** *Let Assumptions 4.1, 4.3, 4.4, and 4.2 hold. Then, for any $\delta \in (0,1)$, with probability at least $1-\delta$, under the distribution of $P_{\mathbf{Z}_n}$,*

$$\mathrm{gen}(\mathrm{m}^\alpha(\mu_n), \mu) \leq 4 N_{\max} M_\phi M_{\ell'} L_\psi \sqrt{\frac{N_{\max} M_\phi M_{\ell'} L_\psi \alpha}{n}} + 3 M_\ell \sqrt{\frac{\log(2/\delta)}{2n}}.$$

Remark 4.15 (Comparison with (Chen et al., 2020; Nishikawa et al., 2022; Nitanda et al., 2022)). In (Chen et al., 2020; Nishikawa et al., 2022; Nitanda et al., 2022), it is assumed that there exists a "true" distribution $m_{\mathrm{true}} \in \mathcal{P}(\mathcal{W})$ which satisfies $\ell(\Psi(m_{\mathrm{true}}, \mathbf{X}_i), y_i) = 0$ for all $(\mathbf{X}_i, y_i) \in \mathcal{Z}$ where $\mu(\mathbf{X}_i, y_i) > 0$ and the KL-divergence between the true distribution and the prior distribution is finite, i.e., $\mathrm{KL}(m_{\mathrm{true}} \| \pi) < \infty$. In particular, the authors in (Chen et al., 2020) contributed Theorem 4.5 to provide an upper bound for one-hidden layer neural networks in terms of the chi-square divergence, $\chi^2(m_{\mathrm{true}} \| \pi)$, which is unknown. In addition, in (Nishikawa et al., 2022; Nitanda et al., 2022), the authors proposed upper bounds in terms of $\mathrm{KL}(m_{\mathrm{true}} \| \pi)$ which cannot be computed. To address this issue, our main contribution in comparison with (Chen et al., 2020; Nishikawa et al., 2022; Nitanda et al., 2022) is the utilization of Proposition 4.13, to obtain a parametric upper bound that can be efficiently computed numerically, overcoming the challenges posed by the unknown term.

Remark 4.16 (Comparison with Theorem 4.9). The convergence rate of the generalization error upper bound in

Theorem 4.14 is $O(1/\sqrt{n})$, while the upper bound in Theorem 4.9 achieves a faster convergence rate of $O(1/n)$.

4.3. Over-Parameterized One-Hidden-Layer GCN

For the over-parameterized one-hidden-layer GCN, we first show the boundedness of the unit function as in Assumption 4.3 under an additional assumption.

Assumption 4.17 (Activation functions in neuron units)**.** The activation function $\varphi : \mathbb{R} \mapsto \mathbb{R}$ is L_φ-Lipschitz [3], so that $|\varphi(x_1) - \varphi(x_2)| \leq L_\varphi |x_1 - x_2|$ for all $x_1, x_2 \in \mathbb{R}$, and zero-centered, i.e., $\varphi(0) = 0$.

Lemma 4.18 (Upper Bound on the GCN Unit Output)**.** *Let Assumptions 4.5 and 4.17 hold. For a graph sample $(\mathbf{A}_q, \mathbf{F}_q)$ with N nodes and a graph filter $G(\cdot)$, the following upper bound holds on the summation of GCN neuron units over all nodes:*

$$\sum_{j=1}^{N} |\phi_c(W_c, G(\mathbf{A}_q)[j,:]\mathbf{F}_q)| \leq N w_{2,c} L_\varphi \|W_{1,c}\|_2 B_f G_{\max}.$$

Combining Lemma 4.18 with Theorem 4.9, we can derive an upper bound on the generalization error of an over-parameterized one-hidden-layer GCN.

Proposition 4.19 (Generalization error and GCN)**.** *In a GCN model with the mean-readout function, under the combined assumptions for Theorem 4.9 and Lemma 4.18, the following upper bound holds on the generalization error of the Gibbs measure $\mathrm{m}^{\alpha,c}(\mu_n)$,*

$$\overline{\mathrm{gen}}(\mathrm{m}^{\alpha,c}(\mu_n), \mu) \leq \frac{\alpha M_c^2 M_{\ell'}^2 G_{\max}^2}{n},$$

where $M_c = w_{2,c} L_\varphi \|W_{1,c}\|_2 B_f$.

Remark 4.20 (Graph filter and $\|G(\mathbf{A})\|_\infty$). Proposition 4.19 shows that choosing the graph filter with smaller $\|G(\mathcal{A})\|_\infty^{\max}$ can affect the upper bound on the generalization error of GCN. In particular, it shows the effect of the aggregation of the input features on the generalization error upper bound. For example, if we consider the graph filter $G(\mathbf{A}) = \tilde{L}$, then we have $\|G(\mathcal{A})\|_\infty^{\max} \leq \sqrt{(d_{\max}+1)/(d_{\min}+1)}$, for sum-aggregation $G(\mathbf{A}) = \mathbf{A} + I$ we have $\|G(\mathcal{A})\|_\infty^{\max} \leq d_{\max} + 1$, and for random-walk, i.e., $G(\mathbf{A}) = D^{-1}\mathbf{A} + I$, we have $\|G(\mathcal{A})\|_\infty^{\max} = 2$.

Remark 4.21 (Graph filter and $\|G(\mathbf{A})\|_F$). Similarly, choosing the graph filter with a smaller $\|G(\mathcal{A})\|_F^{\max}$ value can affect the upper bound on the generalization error of GCN. For example, if we consider the graph filter $G(\mathbf{A}) = \tilde{L}$, then we have $\|G(\mathcal{A})\|_F^{\max} \leq \sqrt{R_{\max}(\tilde{L})}$ and for random walk graph filter $G(\mathbf{A}) = \tilde{D}^{-1}\mathbf{A} + I$ we have $\|G(\mathcal{A})\|_F^{\max} \leq 2\sqrt{R_{\max}(\tilde{D}^{-1}\mathbf{A} + I)}$.

In a similar approach to Proposition 4.19, we can derive an upper bound on generalization error of GCN by combining Lemma 4.18 with Theorem 4.14.

4.4. Over-Parameterized One-Hidden-Layer MPGNN

Similar to GCN, we next investigate the boundedness of the unit function as in Assumption 4.3 for MPGNN; again we make an additional assumption.

Assumption 4.22 (Non-linear functions in the MPGNN units)**.** The non-linear functions $\zeta : \mathbb{R}^{N\times k} \mapsto \mathbb{R}^{N\times k}$, $\rho : \mathbb{R}^k \mapsto \mathbb{R}^k$ and $\kappa : \mathbb{R} \mapsto \mathbb{R}$ satisfy $\zeta(\mathbf{0}^{N\times k}) = \mathbf{0}^{N\times k}$, $\rho(\mathbf{0}^k) = \mathbf{0}^k$ and $\kappa(0) = 0$, and are Lipschitz with parameters L_ζ, L_ρ, and L_κ under vector 2-norm, respectively.

Similarly to GCN, we provide the following upper bound on the MPGNN unit output.

Lemma 4.23 (Upper Bound on the MPGNN Unit Output)**.** *Let Assumptions 4.5 and 4.22 hold. For a graph sample $(\mathbf{A}_q, \mathbf{F}_q)$ and a graph filter $G(\cdot)$, for the MPU units over all nodes,*

$$\sum_{j=1}^{N} |\phi_p(W_p, G(\mathbf{A}_q)[j,:]\mathbf{F}_q)| \leq w_{2,p} L_\kappa B_f (\|W_{3,p,m}\|_2 + L_\rho L_\zeta G_{\max} \|W_{1,p,m}\|_2).$$

Proposition 4.24 (Generalization error and MPGNN)**.** *In an MPGNN with the mean-readout function, under the combined assumptions for Theorem 4.9 and Lemma 4.23,*

$$\overline{\mathrm{gen}}(\mathrm{m}^{\alpha,p}(\mu_n), \mu) \leq \frac{\alpha M_p^2 M_{\ell'}^2}{n}.$$

with $M_p = w_{2,p} L_\kappa B_f (\|W_{3,p}\|_2 + G_{\max} L_\rho L_\zeta \|W_{1,p}\|_2)$.

Similar discussions as in Remark 4.20 and Remark 4.21 hold for the effect of graph filter choices in an MPGNN. In a similar approach to Proposition 4.24, we can derive an upper bound on the generalization error of MPGNN by combining Lemma 4.23 with Theorem 4.14.

4.5. Comparison to Previous Works

We compare our generalization error upper bound with other generalization error upper bounds derived by the VC-dimension approach (Scarselli et al., 2018), bounding Rademacher Complexity (Garg et al., 2020), PAC-Bayesian bounds via perturbation analysis (Liao et al., 2020), PAC-Bayesian bounds based on the Hessian matrix of the loss (Ju et al., 2023), and probability bounds for continuous MPGNNs on large random graphs (Maskey et al., 2022).

[3] For the Tanh activation function, we have $L_\varphi = 1$.

Table 1: Comparison of generalization bounds in over-parameterized one-hidden layer GCN. The width of the hidden layer, the number of training samples, maximum and minimum degree of graph data set are denoted as h, n, $d_{\max}$ and $d_{\min}$, respectively. We have $\tilde{d}_{\max} = d_{\max} + 1$ and $\tilde{d}_{\min} = d_{\min} + 1$. We examine three types of upper bounds: Probability (P), High-Probability (HP) and Expected (E).“N/A” means not applicable.

Approach	$\tilde{d}_{\max}, \tilde{d}_{\min}$	h	n	Bound Type
VC-Dimension (Scarselli et al., 2018)	N/A	$O(h^4)$	$O(1/\sqrt{n})$	HP
Rademacher Complexity (Garg et al., 2020)	$O(\tilde{d}_{\max}\log^{1/2}(\tilde{d}_{\max}))$	$O(h\sqrt{\log(h)})$	$O(1/\sqrt{n})$	HP
PAC-Bayesian (Liao et al., 2020)	$O(\tilde{d}_{\max})$	$O(\sqrt{h\log(h)})$	$O(1/\sqrt{n})$	HP
PAC-Bayesian(Ju et al., 2023)	N/A	$O(\sqrt{h})$	$O(1/\sqrt{n})$	HP
Continuous MPGNN (Maskey et al., 2022)	N/A	N/A	$O(1/\sqrt{n})$	P
Rademacher Complexity (this paper, Theorem 4.14)	$O\big((\tilde{d}_{\max}/\tilde{d}_{\min})^{3/4}\big)$	N/A	$O(1/\sqrt{n})$	HP
Functional Derivative (this paper, Theorem 4.9)	$O(\tilde{d}_{\max}/\tilde{d}_{\min})$	**N/A**	$\boldsymbol{O(1/n)}$	E

We also compare with our upper bound in App.D obtained via Rademacher complexity in App.D. To compare different bounds, we analyze their convergence rate concerning the width of the hidden layer (h) and the number of training samples (n). We also examine the type of bounds on the generalization error, including high-probability bounds, probability bounds, and expected bounds. In high-probability bounds, the upper bound depends on $\log(1/\delta)$ for $\delta \in (0, 1)$, as opposed to $1/\delta$ in probability bounds. For small δ, the high probability bounds are tighter with respect to probability bounds. We use a fixed α in Theorem 4.9 for our comparison in Table 1. More discussion is found in App. E.

As shown in Tab. 1, the upper bounds in (Scarselli et al., 2018; Garg et al., 2020; Liao et al., 2020) and (Ju et al., 2023) are vacuous for infinite width ($h \to \infty$) of one hidden layer. We also provide results for other non-linear functions that are functions of the sum of the final node representations. (Maskey et al., 2022) proposed an expected upper bound on the square of the generalization error of continuous MPGNNs, which is independent of the width of the layers. Then, via the Markov inequality, they provide a probability upper bound on the generalization error of continuous MPGNNs by considering the mean-readout, obtaining a convergence rate of $O(1/\sqrt{n})$. While the random graph model in (Maskey et al., 2022) is based on graphons, here we do not assume that the graph samples arise from a specific random graph model. To the best of our knowledge, this is the first work to represent an upper bound on the generalization error with the convergence rate of $O(1/n)$.

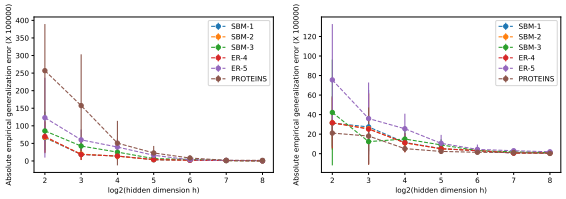
Inspired by the NTK approach of (Jacot et al., 2018a), (Du et al., 2019) proposed the graph neural tangent kernel (GNTK) as a GNN model for layers of infinite width. They provided a high-probability upper bound on the true risk based on (Bartlett & Mendelson, 2002) for the sum-readout function, for their proposed structure, GNTK, which is different from GCNs and MPGNNs. However, as noted in (Fang et al., 2021), the neural tangent kernel has some limitations for over-parameterized analysis of neural networks when compared to mean-field analysis. Therefore, we do not compare with (Du et al., 2019). Finally, the upper bound in (Verma & Zhang, 2019) focuses on stability analysis for one-hidden-layer GNNs in the context of semi-supervised node classification tasks. In (Verma & Zhang, 2019), the data samples are node features rather than a graph, and the node features within each graph sample are assumed to be i.i.d., whereas we only assume that the graph samples themselves are i.i.d. and, therefore, we do not compare with (Verma & Zhang, 2019).

5. Experiments

Our investigation focuses on the over-parameterized one-hidden-layer case in the context of GNNs, such as GCNs and MPGNNs. Prior works on graph classification tasks (Garg et al., 2020; Liao et al., 2020; Ju et al., 2023; Maskey et al., 2022; Du et al., 2019) have demonstrated that the upper bounds on generalization error tend to increase as the number of layers in the network grows. An examination of the over-parameterized one-hidden-layer case may be particularly instructive in elucidating the generalization error performance of GNNs in the context of graph classification tasks.

For this purpose, we investigate the effect of the number of hidden neurons h on the true generalization error of GCNs and MPGNNs for the (semi-)supervised graph classification task on both synthetic and real-world data sets. We use a supervised ratio of $\beta_{\text{sup}} \in \{0.7, 0.9\}$ for $h \in \{4, 8, 16, 32, 64, 126, 256\}$ for our experiments detailed in App. F. For synthetic data sets, we generate three types of Stochastic Block Models (SBMs) and two types of Erdős-Rényi (ER) models, with 200 graphs for each type. We also conduct experiments on a bioinformatics data set called PROTEINS (Borgwardt et al., 2005). Details on implementation, data sets, and additional results are in App. F.

In our experiments, we use the logistic loss $\ell(\Psi(\mathrm{m}, \mathbf{x}), y) = \log(1 + \exp(-\Psi(\mathrm{m}, \mathbf{x})y))$ for binary classification, where y is the true label and $\Psi(\mathrm{m}, \mathbf{x})$ is the mean- or sum-readout function for GCNs and MPGNNs. The empirical risk is



(a) GCN. (b) MPGNN.

Figure 1: Absolute empirical generalization error ($\times 10^5$) for different widths h of the hidden layer. We employ a mean-readout function and a supervised ratio of $\beta_{\text{sup}} = 0.7$, for GCN and MPGNN. Values are averaged over ten runs. Error bars indicate one standard deviation.

then $\frac{1}{n}\sum_{i=1}^{n} \log(1 + \exp(-\Psi(\mathrm{m}, \mathbf{x}_i)y_i))$.

From Fig. 1 (and App. F with detailed mean and standard deviation values), we observe a consistent trend: as the value of h increases, the absolute generalization error decreases. This observation shows that the upper bounds dependent on the width of the layer fail to capture the trend of generalization error in the over-parameterized regime. We provide extended actual absolute generalization errors in this section and the values of our upper bounds are provided in App. F.

6. Conclusions and Future Work

This work develops generalization error upper bounds for one-hidden-layer GCNs and MPGNNs for graph classification in the over-parameterized regime. Our analysis is based on a mean-field approach. Our upper bound on the generalization error of one-hidden-layer GCNs and MPGNNs with the KL-regularized empirical risk minimization is of the order of $O(1/n)$, where n is the number of graph data samples, in the mean-field regime. This order is a significant improvement over previous work, see Table 1.

The main limitation of our work is that it considers only one hidden layer for graph convolutional networks and message-passing graph neural networks. Inspired by Sirignano & Spiliopoulos (2019), we aim to apply our approach to deep graph neural networks and investigate the effect of depth on the generalization performance. Furthermore, we plan to expand the current framework to study the generalization error of hypergraph neural networks, using the framework introduced in (Feng et al., 2019).

Acknowledgements

Gholamali Aminian, Gesine Reinert, Łukasz Szpruch and Samuel N. Cohen acknowledge the support of the UKRI Prosperity Partnership Scheme (FAIR) under EPSRC Grant EP/V056883/1 and the Alan Turing Institute. Yixuan He is supported by a Clarendon scholarship from University of Oxford. Gesine Reinert is also supported in part by EPSRC grants EP/W037211/1 and EP/R018472/1. For the purpose of Open Access, the authors have applied a CC BY public copyright licence to any Author Accepted Manuscript (AAM) version arising from this submission.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning by providing a theoretical underpinning. There are many indirect potential societal consequences of our work through applications of empirical risk, see for example discussions in Abrahamsson & Johansson (2006) and Tran et al. (2021), and we believe that no direct consequences warrant being highlighted.

References

Abrahamsson, M. and Johansson, H. Risk preferences regarding multiple fatalities and some implications for societal risk decision making—an empirical study. *Journal of risk research*, 9(7):703–715, 2006.

Allen-Zhu, Z. and Li, Y. What can resnet learn efficiently, going beyond kernels? *Advances in Neural Information Processing Systems*, 32, 2019.

Aminian, G., Arjmandi, H., Gohari, A., Nasiri-Kenari, M., and Mitra, U. Capacity of diffusion-based molecular communication networks over lti-poisson channels. *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, 1(2):188–201, 2015.

Aminian, G., Bu, Y., Toni, L., Rodrigues, M., and Wornell, G. An exact characterization of the generalization error for the gibbs algorithm. *Advances in Neural Information Processing Systems*, 34:8106–8118, 2021.

Aminian, G., Cohen, S. N., and Szpruch, Ł. Mean-field analysis of generalization errors. *arXiv preprint arXiv:2306.11623*, 2023.

Bartlett, P. L. and Mendelson, S. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

Bartlett, P. L., Jordan, M. I., and McAuliffe, J. D. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

Bartlett, P. L., Montanari, A., and Rakhlin, A. Deep learning: a statistical viewpoint. *Acta numerica*, 30:87–201, 2021.

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

Belkin, M., Hsu, D., Ma, S., and Mandal, S. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

Borgwardt, K. M., Ong, C. S., Schönauer, S., Vishwanathan, S., Smola, A. J., and Kriegel, H.-P. Protein function prediction via graph kernels. *Bioinformatics*, 21(suppl_1): i47–i56, 2005.

Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A., and Vandergheynst, P. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4): 18–42, 2017.

Cangea, C., Veličković, P., Jovanović, N., Kipf, T., and Liò, P. Towards sparse hierarchical graph classifiers. *arXiv preprint arXiv:1811.01287*, 2018.

Cao, Y. and Gu, Q. Generalization bounds of stochastic gradient descent for wide and deep neural networks. *Advances in neural information processing systems*, 32, 2019.

Cardaliaguet, P., Delarue, F., Lasry, J.-M., and Lions, P.-L. *The Master Equation and the Convergence Problem in Mean Field Games*. Princeton University Press, 2019.

Chen, Z., Cao, Y., Gu, Q., and Zhang, T. A generalized neural tangent kernel analysis for two-layer neural networks. *Advances in Neural Information Processing Systems*, 33: 13363–13373, 2020.

Chizat, L. and Bach, F. A note on lazy training in supervised differentiable programming.(2018). *arXiv preprint arXiv:1812.07956*, 1812.

Chizat, L. and Bach, F. On the global convergence of gradient descent for over-parameterized models using optimal transport. *arXiv preprint arXiv:1805.09545*, 2018.

Cong, W., Ramezani, M., and Mahdavi, M. On provable benefits of depth in training graph convolutional networks. *Advances in Neural Information Processing Systems*, 34: 9936–9949, 2021.

Dai, H., Dai, B., and Song, L. Discriminative embeddings of latent variable models for structured data. In *International conference on machine learning*, pp. 2702–2711. PMLR, 2016.

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

Du, S. S., Hou, K., Salakhutdinov, R. R., Poczos, B., Wang, R., and Xu, K. Graph neural tangent kernel: Fusing graph neural networks with graph kernels. *Advances in Neural Information Processing Systems*, 32, 2019.

El-Yaniv, R. and Pechyony, D. Stable transductive learning. In *Learning Theory: 19th Annual Conference on Learning Theory, COLT 2006, Pittsburgh, PA, USA, June 22-25, 2006. Proceedings 19*, pp. 35–49. Springer, 2006.

Esser, P., Chennuru Vankadara, L., and Ghoshdastidar, D. Learning theory can (sometimes) explain generalisation in graph neural networks. *Advances in Neural Information Processing Systems*, 34:27043–27056, 2021.

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

Fang, C., Dong, H., and Zhang, T. Mathematical models of overparameterized neural networks. *Proceedings of the IEEE*, 109(5):683–703, 2021.

Feng, Y., You, H., Zhang, Z., Ji, R., and Gao, Y. Hypergraph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, 2019.

Gao, H. and Ji, S. Graph u-nets. In *International Conference on Machine Learning*, pp. 2083–2092. PMLR, 2019.

Garg, V., Jegelka, S., and Jaakkola, T. Generalization and representational limits of graph neural networks. In *International Conference on Machine Learning*, pp. 3419–3430. PMLR, 2020.

GHORBANI, B., MEI, S., MISIAKIEWICZ, T., and MONTANARI, A. Linearized two-layers neural networks in high dimension. *The Annals of Statistics*, 49(2):1029–1054, 2021.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. Neural message passing for quantum chemistry. In *International conference on machine learning*, pp. 1263–1272. PMLR, 2017.

Gori, M., Monfardini, G., and Scarselli, F. A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pp. 729–734. IEEE, 2005.

Hagberg, A., Swart, P., and S Chult, D. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

Hamilton, W., Ying, Z., and Leskovec, J. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

Hamilton, W. L. Graph representation learning. *Synthesis Lectures on Artifical Intelligence and Machine Learning*, 14(3):1–159, 2020.

Hu, K., Ren, Z., Siska, D., and Szpruch, L. Mean-field Langevin dynamics and energy landscape of neural networks. *arXiv preprint arXiv:1905.07769*, 2019.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. *arXiv preprint arXiv:1806.07572*, 2018a.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in Neural Information Processing Systems*, 31, 2018b.

Ju, H., Li, D., Sharma, A., and Zhang, H. R. Generalization in graph neural networks: Improved pac-bayesian bounds on graph diffusion. *arXiv preprint arXiv:2302.04451*, 2023.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2016.

Li, H., Wang, M., Liu, S., Chen, P.-Y., and Xiong, J. Generalization guarantee of training graph convolutional networks with graph topology sampling. In *International Conference on Machine Learning*, pp. 13014–13051. PMLR, 2022.

Liao, R., Urtasun, R., and Zemel, R. A pac-bayesian approach to generalization bounds for graph neural networks. In *International Conference on Learning Representations*, 2020.

Lv, S. Generalization bounds for graph convolutional neural networks via Rademacher complexity. *arXiv preprint arXiv:2102.10234*, 2021.

Maskey, S., Levie, R., Lee, Y., and Kutyniok, G. Generalization analysis of message passing neural networks on large random graphs. In *Advances in Neural Information Processing Systems*, 2022.

Mei, S. and Montanari, A. The generalization error of random features regression: Precise asymptotics and the double descent curve. *Communications on Pure and Applied Mathematics*, 75(4):667–766, 2022.

Mei, S., Montanari, A., and Nguyen, P.-M. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33): E7665–E7671, 2018.

Mei, S., Misiakiewicz, T., and Montanari, A. Mean-field theory of two-layers neural networks: dimension-free bounds and kernel limit. In *Conference on Learning Theory*, pp. 2388–2464. PMLR, 2019.

Mesquita, D., Souza, A., and Kaski, S. Rethinking pooling in graph neural networks. *Advances in Neural Information Processing Systems*, 33:2220–2231, 2020.

Meyer, C. D. and Stewart, I. *Matrix analysis and applied linear algebra*. SIAM, 2023.

Mohri, M., Rostamizadeh, A., and Talwalkar, A. *Foundations of machine learning*. MIT press, 2018.

Monti, F., Boscaini, D., Masci, J., Rodola, E., Svoboda, J., and Bronstein, M. M. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5115–5124, 2017.

Neyshabur, B., Bhojanapalli, S., and Srebro, N. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. In *International Conference on Learning Representations*, 2018.

Nishikawa, N., Suzuki, T., Nitanda, A., and Wu, D. Two-layer neural network on infinite dimensional data: global optimization guarantee in the mean-field regime. In *Advances in Neural Information Processing Systems*, 2022.

Nitanda, A., Wu, D., and Suzuki, T. Particle dual averaging: optimization of mean field neural network with global convergence rate analysis. *Journal of Statistical Mechanics: Theory and Experiment*, 2022(11):114010, 2022.

Oono, K. and Suzuki, T. Optimization and generalization analysis of transduction through gradient boosting and application to multi-scale graph neural networks. *Advances in Neural Information Processing Systems*, 33: 18917–18930, 2020.

Polyanskiy, Y. and Wu, Y. *Information Theory: From Coding to Learning*. Cambridge University Press, 2022.

Scarselli, F., Tsoi, A. C., and Hagenbuchner, M. The vapnik–chervonenkis dimension of graph and recursive neural networks. *Neural Networks*, 108:248–259, 2018.

Shalev-Shwartz, S. and Ben-David, S. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Sirignano, J. and Spiliopoulos, K. Mean field analysis of deep neural networks. *arXiv preprint arXiv:1903.04440*, 2019.

Spigler, S., Geiger, M., d'Ascoli, S., Sagun, L., Biroli, G., and Wyart, M. A jamming transition from under-to over-parametrization affects generalization in deep learning.

Journal of Physics A: Mathematical and Theoretical, 52 (47):474001, 2019.

Suzuki, T. and Nitanda, A. Deep learning is adaptive to intrinsic dimensionality of model smoothness in anisotropic besov space. *Advances in Neural Information Processing Systems*, 34:3609–3621, 2021.

Tang, H. and Liu, Y. Towards understanding the generalization of graph neural networks. *arXiv preprint arXiv:2305.08048*, 2023.

Tran, C., Dinh, M., and Fioretto, F. Differentially private empirical risk minimization under the fairness lens. *Advances in Neural Information Processing Systems*, 34: 27555–27565, 2021.

Tzen, B. and Raginsky, M. A mean-field theory of lazy training in two-layer neural nets: entropic regularization and controlled mckean-vlasov dynamics. *arXiv preprint arXiv:2002.01987*, 2020.

Vapnik, V. N. and Chervonenkis, A. Y. On the uniform convergence of relative frequencies of events to their probabilities. In *Measures of complexity*, pp. 11–30. Springer, 2015.

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

Verma, S. and Zhang, Z.-L. Stability and generalization of graph convolutional neural networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1539–1548, 2019.

Vinyals, O., Bengio, S., and Kudlur, M. Order matters: Sequence to sequence for sets. *arXiv preprint arXiv:1511.06391*, 2015.

Wainwright, M. J. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge university press, 2019.

Wang, J., Huang, P., Zhao, H., Zhang, Z., Zhao, B., and Lee, D. L. Billion-scale commodity embedding for e-commerce recommendation in alibaba. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 839–848, 2018.

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

Yang, G. and Hu, E. J. Feature learning in infinite-width neural networks. *arXiv preprint arXiv:2011.14522*, 2020.

Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., and Leskovec, J. 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, 2018a.

Ying, Z., You, J., Morris, C., Ren, X., Hamilton, W., and Leskovec, J. Hierarchical graph representation learning with differentiable pooling. *Advances in neural information processing systems*, 31, 2018b.

Zhang, S., Wang, M., Liu, S., Chen, P.-Y., and Xiong, J. Fast learning of graph neural networks with guaranteed generalizability: one-hidden-layer case. In *International Conference on Machine Learning*, pp. 11268–11277. PMLR, 2020.

Zhou, X. and Wang, H. The generalization error of graph convolutional networks may enlarge with more layers. *Neurocomputing*, 424:97–106, 2021.

A. Preliminaries

Notations in this paper are summarized in Table 2.

Table 2: Summary of notations in the paper

Notation	Definition	Notation	Definition
$\mathcal{W}$	Parameter space of the model	$\mathcal{F}$	Feature matrices space
$\mathcal{A}$	Adjacency matrices space	$\mathbf{F}$	Matrix of feature nodes of a graph sample
$\mathbf{A}$	Adjacency matrix of a graph sample	$\mathbf{X}$	input graph sample where $\mathbf{X} = (\mathbf{A}, \mathbf{F})$
Y	Label of input graph	Z_i	i-th graph sample $(\mathbf{X}, Y)$
$\mathbf{Z}_n$	The set of data training samples	D	Degree matrix of A
$\tilde{L}$	Symmetric normalized graph filter	n	Number of graph data samples
$N_{\max}$	Maximum number of nodes of all graph samples	$d_{\max}$	Maximum node degree of all graph samples
$d_{\min}$	Minimum node degree of all graph samples	$G(\mathbf{A})$	Graph filter with input matrix $\mathbf{A}$
μ_n	Empirical data measure	$\mu_{n,(1)}$	Replace-one sample empirical data measure
M_ϕ	Bound on unit function	$M_{\ell'}$	Bound on $\lvert\partial_{\hat{y}}\ell(\hat{y}, y)\rvert$
L_ζ	Lipschitz parameter of function $\zeta(\cdot)$	L_ρ	Lipschitz parameter of function $\rho(\cdot)$
L_κ	Lipschitz parameter of function $\kappa(\cdot)$	L_ψ	Lipschitz parameter of readout function $\psi(\cdot)$
L_φ	Lipschitz parameter of activation function $\psi(\cdot)$	B_f	Bound on node features
α	Inverse temperature	h	Width of hidden layer
π	Prior measure over parameters	m^α	The Gibbs measure for General GNN model
$\mathrm{m}^{\alpha,c}$	The Gibbs measure for GCN model	$\mathrm{m}^{\alpha,p}$	The Gibbs measure for MPGNN model
m_{true}	True distribution over data samples	$G_{\max}$	$\min(\lVert G(\mathcal{A})\rVert_\infty^{\max}, \lVert G(\mathcal{A})\rVert_F^{\max})$
$\Psi(\cdot)$	Readout function	$(W_{1,c}, W_{2,c})$	Parameters of Neuron unit
$(W_{1,p}, W_{2,p}, W_{3,p})$	Parameters of MPU unit	$\overline{\text{gen}}(\mathrm{m}(\mu_n), \mu)$	Generalization error under parameter measure $\mathrm{m}(\mu_n)$
β_{sup}	Supervised ratio	$\lVert\mathbf{Y}\rVert_F$	$\sqrt{\sum_{j=1}^k \sum_{i=1}^q \mathbf{Y}^2[j,i]}$
$\lVert\mathbf{Y}\rVert_\infty$	$\max_{1\le j\le k} \sum_{i=1}^q \lvert\mathbf{Y}[j,i]\rvert$	$\mathrm{KL}(p\lVert q)$	$\int_{\mathbb{R}^d} p(x)\log(p(x)/q(x))\mathrm{d}x$
$\mathrm{KL}_{\text{sym}}(p\lVert q)$	$\mathrm{KL}(p\lVert q) + \mathrm{KL}(q\lVert p)$	$\phi(\cdot)$	Unit function

Let us recapitulate all the assumptions required for our proofs.

Assumption 4.1 (Loss function)**.** *The loss function, $(\hat{y}, y) \mapsto \ell(\hat{y}, y)$, satisfies the following conditions,*

(i) The gradient of the loss function $(\hat{y}, y) \mapsto \ell(\hat{y}, y)$ with respect to $\hat{y}$ is continuous and uniformly bounded for all $\hat{y}, y \in \mathcal{Y}$, i.e., there is a constant $M_{\ell'}$ such that $\lvert\partial_{\hat{y}}\ell(\hat{y}, y)\rvert \le M_{\ell'}$.

(ii) We assume that the loss function is convex with respect to $\hat{y}$.

Assumption 4.3 (Unit function)**.** *The unit function $(w, G(\mathbf{A})[j,:]\mathbf{f}) \mapsto \phi(w, G(\mathbf{A})[j,:]\mathbf{f})$ with graph filter $G(\cdot)$ is uniformly bounded, i.e, there is a constant M_ϕ such that $\sup_{w\in\mathcal{W},\mathbf{f}\in\mathcal{F},\mathbf{A}\in\mathcal{A}} \lvert\phi(w, G(\mathbf{A})[j,:]\mathbf{f})\rvert \le M_\phi$ for all $j \in [N]$ and $\mathbf{f} \in \mathcal{F}$.*

Assumption 4.4 (Readout function)**.** *The function $\psi : \mathbb{R} \mapsto \mathbb{R}$ is L_ψ-Lipschitz-continuous, i.e., there is a constant L_ψ such that $\lvert\psi(x_1) - \psi(x_2)\rvert \le L_\psi\lvert x_1 - x_2\rvert$ for all $x_1, x_2 \in \mathbb{R}$, and zero-centered, i.e., $\psi(0) = 0$.*

Assumption 4.5 (Bounded node features)**.** *For every graph, the node features are contained in an ℓ_2-ball of radius B_f. In particular, $\lVert\mathbf{F}[i;]\rVert_2 \le B_f$ for all $i \in [N]$.*

Assumption 4.17 (Activation functions in neuron unit)**.** *The activation function $\varphi : \mathbb{R} \mapsto \mathbb{R}$ is L_φ-Lipschitz, so that $\lvert\varphi(x_1) - \varphi(x_2)\rvert \le L_\varphi\lvert x_1 - x_2\rvert$ for all $x_1, x_2 \in \mathbb{R}$, and zero-centered, i.e., $\phi(0) = 0$.*

Assumption 4.22 (Non-linear functions in the MPGNN unit)**.** *The non-linear functions $\zeta : \mathbb{R}^{N\times k} \mapsto \mathbb{R}^{N\times k}$, $\rho : \mathbb{R}^k \mapsto \mathbb{R}^k$ and $\kappa : \mathbb{R} \mapsto \mathbb{R}$ are zero-centered, i.e., $\zeta(\mathbf{0}^{N\times k}) = \mathbf{0}^{N\times k}$, $\rho(\mathbf{0}^k) = \mathbf{0}^k$ and $\kappa(0) = 0$, and Lipschitz with parameters L_ζ, L_ρ and L_κ under vector 2-norm[4], respectively.*

Total variation distance: The total variation distance between two densities $p(x)$ and $q(x)$, is defined as $\mathbb{TV}(p, q) := \frac{1}{2}\int_{\mathbb{R}^d} \lvert p(x) - q(x)\rvert\mathrm{d}x$.

The following lemmas are needed for our proofs.

[4]For function $\zeta(\cdot)$, we consider that it is Lipschitz under a vector 2-norm,

Lemma A.1 (Donsker's representation of KL divergence)**.** *Let us consider the variational representation of the KL divergence between two probability distributions m_1 and m_2 on a common space ψ given by Polyanskiy & Wu (2022),*

$$\mathrm{KL}(m_1||m_2) = \sup_f \int_\psi f dm_1 - \log \int_\psi e^f dm_2, \tag{13}$$

where $f \in \mathfrak{F} = \{f : \psi \to \mathbb{R} \text{ s.t. } \mathbb{E}_{m_2}[e^f] < \infty\}$.

Lemma A.2 (Kantorovich-Rubenstein duality of total variation distance)**.** *The Kantorovich-Rubenstein duality (variational representation) of the total variation distance is as follows (Polyanskiy & Wu, 2022):*

$$\mathbb{TV}(m_1, m_2) = \frac{1}{2L} \sup_{g \in \mathcal{G}_L} \left\{ \mathbb{E}_{Z\sim m_1}[g(Z)] - \mathbb{E}_{Z\sim m_2}[g(Z)] \right\}, \tag{14}$$

where $\mathcal{G}_L = \{g : \mathcal{Z} \to \mathbb{R}, ||g||_\infty \leq L\}$.

Lemma A.3 (Bound on infinite norm of the symmetric normalized graph filter)**.** *Consider a graph sample G with adjacency matrix A. For the symmetric normalized graph filter, we have $\left\|\tilde{D}^{-1/2}(A+I)\tilde{D}^{-1/2}\right\|_\infty \leq \sqrt{\frac{d_{\max}+1}{d_{\min}+1}}$ where $d_{\max}$ is the maximum degree of graph G with adjacency matrix A.*

Proof. Recall that $\tilde{A} = A + I$. We have,

$$\begin{aligned}
\left\|\tilde{L}\right\|_\infty &= \left\|\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}\right\|_\infty \\
&= \max_{i\in[N]} \sum_{j=1}^{N} \frac{\tilde{A}_{ij}}{\sqrt{d_i+1}\sqrt{d_j+1}} \\
&\leq \frac{1}{\sqrt{d_{\min}+1}} \max_{i\in[N]} \sum_{j=1}^{N} \frac{\tilde{A}_{ij}}{\sqrt{d_i+1}} \\
&\leq \sqrt{\frac{d_{\max}+1}{d_{\min}+1}}.
\end{aligned} \tag{15}$$

□

Lemma A.4 (Meyer & Stewart, 2023)**.** *For a matrix $\mathbf{Y} \in \mathbb{R}^{k\times q}$, we have $\|\mathbf{Y}\|_F \leq \sqrt{R}\|\mathbf{Y}\|_2$, where r is the rank of $\mathbf{Y}$ and $\|\mathbf{Y}\|_2$ is the 2-norm of $\mathbf{Y}$ which is equal to the maximum singular value of $\mathbf{Y}$.*

Lemma A.5 (Verma & Zhang, 2019)**.** *For the symmetric normalized graph filter, i.e., $\tilde{L} = \tilde{D}^{-1/2}\tilde{\mathbf{A}}\tilde{D}^{-1/2}$, we have $\left\|\tilde{L}\right\|_2 = 1$. For the random walk graph filter, i.e., $D^{-1}\mathbf{A} + I$, we have $\left\|D^{-1}\mathbf{A} + I\right\|_2 = 2$, where $\|A\|_2$ is the 2-norm of matrix A.*

Lemma A.6 (Hoeffding lemma (Wainwright, 2019))**.** *For bounded random variable, $a \leq X \leq b$, and all $\lambda \in \mathbb{R}$, we have,*

$$\mathbb{E}[\exp(\lambda(X - \mathbb{E}[X]))] \leq \exp\big(\lambda(b-a)^2/8\big). \tag{16}$$

Lemma A.7 (Pinsker's inequality)**.** *The following upper bound holds on the total variation distance between two measures m_1 and m_2 (Polyanskiy & Wu, 2022):*

$$\mathbb{TV}(m_1, m_2) \leq \sqrt{\frac{\mathrm{KL}(m_1\|m_2)}{2}}. \tag{17}$$

In the following, we apply some lemmata from Aminian et al. (2023) where we use $\ell\big(\Psi(\mathrm{m}(\mu_n), \widehat{\mathbf{x}}_1), \widehat{\mathbf{y}}_1\big)$ instead of $\ell\big(\mathrm{m}(\mu_n), \widehat{Z}_1\big)$ in Aminian et al. (2023).

Lemma A.8. *(Aminian et al., 2023, Proposition 3.3) Given Assumption 4.1, the following lower bound holds on the generalization error of a Generic GNN,*

$$\overline{\mathrm{gen}}(\mathrm{m}^\alpha(\mu_n), \mu) \geq \mathbb{E}_{\mathbf{Z}_n, \widehat{Z}_1}\Big[\int_{\mathcal{W}} \partial_{\hat{y}}\ell\big(\Psi(\mathrm{m}(\mu_{n,(1)}), \widehat{\mathbf{x}}_1), \widehat{\mathbf{y}}_1\big) \frac{\delta\Psi}{\delta \mathrm{m}}(\mathrm{m}(\mu_{n,(1)}), \widehat{\mathbf{x}}_1, w)(\mathrm{m}(\mu_n) - \mathrm{m}(\mu_{n,(1)}))(\mathrm{d}w)\Big].$$

Remark A.9 (Another representation of Lemma A.8). Due to the fact that data samples and $\widehat{Z}_1$ are i.i.d., Lemma A.8 can be represented as follows,

$$\overline{\mathrm{gen}}(\mathrm{m}^{\alpha}(\mu_n),\mu) \geq \mathbb{E}_{\mathbf{Z}_n,\widehat{Z}_1}\Big[\int_{\mathcal{W}} \partial_{\hat{y}}\ell\big(\Psi(\mathrm{m}(\mu_n),\mathbf{x}_1),\mathbf{y}_1\big)\frac{\delta\Psi}{\delta \mathrm{m}}(\mathrm{m}(\mu_n),\mathbf{x}_1,w)(\mathrm{m}(\mu_{n,(1)})-\mathrm{m}(\mu_n))(\mathrm{d}w)\Big].$$

Lemma A.10. *(Aminian et al., 2023, Theorem 3.2) Given Assumption 4.1.i, the following representation of the generalization error holds:*

$$\overline{\mathrm{gen}}(\mathrm{m}^{\alpha}(\mu_n),\mu) = \mathbb{E}_{\mathbf{Z}_n,\widehat{Z}_1}\Big[\int_0^1\int_{\mathcal{W}} \partial_{\hat{y}}\ell\big(\Psi(\mathrm{m}_\lambda(\mu_n),\widehat{\mathbf{x}}_1),\widehat{\mathbf{y}}_1\big)\frac{\delta\Psi}{\delta \mathrm{m}}(\mathrm{m}_\lambda(\mu_n),\widehat{\mathbf{x}}_1,w)(\mathrm{m}(\mu_n)-\mathrm{m}(\mu_{n,(1)}))(\mathrm{d}w)\mathrm{d}\lambda\Big]$$

where $\mathrm{m}_\lambda(\mu_n) = \mathrm{m}(\mu_{n,(1)}) + \lambda(\mathrm{m}(\mu_n)-\mathrm{m}(\mu_{n,(1)}))$ *for* $\lambda \in [0,1]$.

The following preliminaries are needed for our Rademacher complexity analysis.

Rademacher complexity: For a hypothesis set, $\mathcal{H}$ of functions $f_h : \mathcal{Z} \mapsto \mathbb{R}$ and set $\boldsymbol{\sigma} = \{\sigma_i\}_{i=1}^n$, the empirical Rademacher complexity $\hat{\mathfrak{R}}_{\mathrm{Z}}(\mathcal{H})$ with respect to set $\mathbf{Z}_n$ is defined as:

$$\hat{\mathfrak{R}}_{\mathbf{Z}_n}(\mathcal{H}) := \mathbb{E}_{\boldsymbol{\sigma}}\Big[\sup_{f_h\in\mathcal{H}}\frac{1}{n}\sum_{i=1}^n \sigma_i f_h(Z_i)\Big], \tag{18}$$

where $\sigma = \{\sigma_i\}_{i=1}^n$ are i.i.d random variables and $\sigma_i \in \{-1,1\}$ for all $i \in [r]$ with equal probability.

In addition to the previous assumptions, for Rademacher complexity analysis we also need the following assumption.

Assumption 4.2 (Bounded loss function)**.** *The loss function,* $(\hat{y},y) \mapsto \ell(\hat{y},y)$*, is bounded for all* $\hat{y},y \in \mathcal{Y}$*, i.e.,* $0 \leq \ell(\hat{y},y) \leq M_\ell$.

Lemma A.11 (Uniform bound (Mohri et al., 2018))**.** *Let* $\mathcal{F}_u$ *be the set of functions* $f : \mathcal{Z} \to [0, M_\ell]$ *and* μ *be a distribution over* $\mathcal{Z}$*. Let* $S = \{z_i\}_{i=1}^n$ *be a set of size* n *i.i.d. drawn from* $\mathcal{Z}$*. Then, for any* $\delta \in (0,1)$*, with probability at least* $1-\delta$ *over the choice of* S*, we have*

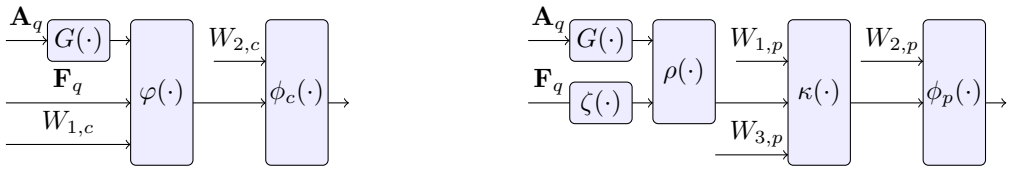
$$\sup_{f\in\mathcal{F}_n}\left\{\mathbb{E}_{Z\sim\mu}[f(Z)] - \frac{1}{n}\sum_{i=1}^n f(z_i)\right\} \leq 2\hat{\mathfrak{R}}_S(\mathcal{F}_n) + 3M_\ell\sqrt{\frac{1}{2n}\log\frac{2}{\delta}}.$$

The contraction lemma helps us to estimate the Rademacher complexity.

Lemma A.12 (Talagrand's contraction lemma (Shalev-Shwartz & Ben-David, 2014))**.** *Let* $\phi_i : \mathbb{R} \to \mathbb{R}$ $(i \in \{1,\dots,n\})$ *be* L*-Lipschitz functions and* $\mathcal{F}_n$ *be a set of functions from* $\mathcal{Z}$ *to* $\mathbb{R}$*. Then it follows that for any* $\{z_i\}_{i=1}^n \subset \mathcal{Z}$*,*

$$\mathbb{E}_\sigma\left[\sup_{f\in\mathcal{F}_n}\frac{1}{n}\sum_{i=1}^n \sigma_i\phi_i(f(z_i))\right] \leq L\mathbb{E}_\sigma\left[\sup_{f\in\mathcal{F}_r}\frac{1}{n}\sum_{i=1}^n \sigma_i f(z_i)\right].$$

The units of GCNs and MPGNN are shown in Figure 2.



(a) Neuron Unit (b) MPU Unit

Figure 2: Units of GCN and MPGNN

B. Other Related Works

Mean-field: Our study employs the mean-field framework utilized in a recent line of research (Chizat & Bach, 2018; Mei et al., 2018; 2019; Sirignano & Spiliopoulos, 2019; Hu et al., 2019). The convergence of gradient descent for training one-hidden layer NNs with infinite width under certain structural assumptions is established by Chizat & Bach (2018). The study of Mei et al. (2018) proved the global convergence of noisy stochastic gradient descent and established approximation bounds between finite and infinite neural networks. Furthermore, Mei et al. (2019) demonstrated that this approximation error can be independent of the input dimension in certain cases, and established that the residual dynamics of noiseless gradient descent are close to the dynamics of NTK-based kernel regression under some conditions.

Graph Representation Learning: Numerous Graph Neural Networks (GNNs) have emerged for graph-based tasks, spanning node, edge, and graph levels. GCNs, introduced in Kipf & Welling (2016), simplify the Cheby-Filter from Defferrard et al. (2016) for one-hop neighbors. MPGNNs, as proposed in Gilmer et al. (2017), outline a general GNN framework, treating graph convolutions as message-passing among nodes and edges. For graph-level tasks, like graph classification, a typical practice involves applying a graph readout (pooling) layer after graph filtering layers, composed of graph filters. The readout (pooling) layer aggregates node representations to create a graph-wide embedding. Common graph readout choices encompass set pooling methods like direct sum application, mean (Hamilton, 2020), or maximum (Mesquita et al., 2020), as well as combinations of LSTMs with attention (Vinyals et al., 2015) and graph coarsening techniques leveraging graph structure (Ying et al., 2018b; Cangea et al., 2018; Gao & Ji, 2019). In this paper, we investigate how mean and sum impact the generalization errors of GCNs and MPGNNs.

Node Classification and Generalization Error: For node classification tasks, Verma & Zhang (2019) discussed the generalization error under node classification for GNNs via algorithm stability analysis. The work by Zhou & Wang (2021) extended the results in Verma & Zhang (2019) and found that increasing the depth of GCN enlarges its generalization error for the node classification task. A Rademacher complexity analysis was applied to GCNs for the node classification task by Lv (2021). Based on transductive Rademacher complexity, a high-probability upper bound on the generalization error of the transductive node classification task was proposed by Esser et al. (2021). The transductive modeling of node classification was studied in Oono & Suzuki (2020) and Tang & Liu (2023). Cong et al. (2021) presented an upper bound on the generalization error of GCNs for node classification via transductive uniform stability, building on the work of El-Yaniv & Pechyony (2006). In contrast, our research focuses on the task of graph classification, which involves (semi-)supervised learning on graph data samples rather than semi-supervised learning for node classification.

Neural Tangent Kernels: The neural tangent kernel (NTK) model, as described by Jacot et al. (2018a), elucidates the learning dynamics inherent in neural networks when subjected to appropriate scaling conditions. This explication relies on the linearization of learning dynamics in proximity to its initialization. Conclusive evidence pertaining to the (quantitative) global convergence of gradient-based techniques for neural networks has been established for both regression problems (Suzuki & Nitanda, 2021) and classification problems (Cao & Gu, 2019). The NTK model, founded upon linearization, is constrained in its ability to account for the phenomenon of "feature learning" within neural networks, wherein parameters exhibit the capacity to traverse and adapt to the inherent structure of the learning problem. Fundamental to this analysis is the linearization of training dynamics, necessitating the imposition of appropriate scaling conditions on the model (Chizat & Bach, 1812). Consequently, this framework proves inadequate for elucidating the feature learning aspect of neural networks (Yang & Hu, 2020; Fang et al., 2021). Notably, empirical investigations have demonstrated the superior expressive power of deep learning over kernel methods concerning approximation and estimation errors (GHORBANI et al., 2021). In certain contexts, it has been observed that neural networks, optimized through gradient-based methodologies, surpass the predictive performance of the NTK model, and more broadly, kernel methods, concerning generalization error or true risk (Allen-Zhu & Li, 2019).

C. Proofs and details of Section 4

C.1. Generic GNN

Proposition 4.6. ***(restated)*** *Let Assumptions 4.1, 4.3, and 4.4 hold. Then, the following bound holds on the generalization error of generic GNN,*

$$\left|\overline{\mathrm{gen}}(\mathrm{m}(\mu_n),\mu)\right| \leq \left(M_{\ell'}L_{\psi}N_{\max}M_{\phi}/\sqrt{2}\right)\mathbb{E}_{\mathbf{Z}_n,\widehat{Z}_1}\left[\sqrt{\mathrm{KL}\big(\mathrm{m}(\mu_n)\|\mathrm{m}(\mu_{n,(1)})\big)}\right]. \tag{19}$$

In the following, we provide two technical proofs for Proposition 4.6.

Proof of Proposition 4.6 via Lemma A.1 and Lemma A.6. From Lemma A.1, the following representation of KL divergence holds between two probability distributions m_1 and m_2 on a common space $\mathcal{W}$,

$$\mathrm{KL}(m_1\|m_2) = \sup_{f\in\mathfrak{F}} \mathbb{E}_{W\sim m_1}[f(W)] - \log\left(\mathbb{E}_{W\sim m_2}[\exp\{f(W)\}]\right), \tag{20}$$

where $f \in \mathfrak{F} = \{f : \mathcal{W} \to \mathbb{R} \text{ s.t. } \mathbb{E}_{W\sim m_2}[e^{f(W)}] < \infty\}$. Lemma A.10 yields

$$\begin{aligned}
&\overline{\mathrm{gen}}(\mathrm{m}^\alpha(\mu_n), \mu)\\
&= \mathbb{E}_{\mathbf{Z}_n,\widehat{Z}_1}\Big[\int_0^1 \int_{\mathcal{W}} \partial_{\hat{y}}\ell\big(\Psi(\mathrm{m}_\lambda(\mu_n), \widehat{\mathbf{x}}_1), \widehat{\mathbf{y}}_1\big)\frac{\delta\Psi}{\delta \mathrm{m}}(\mathrm{m}_\lambda(\mu_n), \widehat{\mathbf{x}}_1, w)(\mathrm{m}(\mu_n) - \mathrm{m}(\mu_{n,(1)}))(\mathrm{d}w)\mathrm{d}\lambda\Big]\\
&= \mathbb{E}_{\mathbf{Z}_n,\widehat{Z}_1}\Big[\int_0^1 \partial_{\hat{y}}\ell\big(\Psi(\mathrm{m}_\lambda(\mu_n), \widehat{\mathbf{x}}_1), \widehat{\mathbf{y}}_1\big)\Big(\mathbb{E}_{W\sim \mathrm{m}(\mu_n)}\Big[\frac{\delta\Psi}{\delta \mathrm{m}}(\mathrm{m}_\lambda(\mu_n), \widehat{\mathbf{x}}_1, W)\Big]\\
&\qquad - \mathbb{E}_{W\sim \mathrm{m}(\mu_{n,(1)})}\Big[\frac{\delta\Psi}{\delta \mathrm{m}}(\mathrm{m}_\lambda(\mu_n), \widehat{\mathbf{x}}_1, W)\Big]\Big)\mathrm{d}\lambda\Big].
\end{aligned}$$

For the function $f(w) = \lambda_1 \partial_{\hat{y}}\ell\big(\Psi(\mathrm{m}_\lambda(\mu_n), \widehat{\mathbf{x}}_1), \widehat{\mathbf{y}}_1\big)\frac{\delta\Psi}{\delta \mathrm{m}}(\mathrm{m}_\lambda(\mu_n), \widehat{\mathbf{x}}_1, w)$, due to Assumptions 4.1, 4.3 and 4.4, we have $\Big|\partial_{\hat{y}}\ell\big(\Psi(\mathrm{m}_\lambda(\mu_n), \widehat{\mathbf{x}}_1), \widehat{\mathbf{y}}_1\big)\frac{\delta\Psi}{\delta \mathrm{m}}(\mathrm{m}_\lambda(\mu_n), \widehat{\mathbf{x}}_1, w)\Big| \le M_{\ell'} L_\psi N_{\max} M_\phi$. Hence $f \in \mathfrak{F}$.

From (20), for $m_1 = \mathrm{m}(\mu_n)$ and $m_2 = \mathrm{m}(\mu_{n,(1)})$, we have for all $\lambda_1 \in \mathbb{R}$

$$\begin{aligned}
&\lambda_1 \mathbb{E}_{W\sim \mathrm{m}(\mu_n)}\Big[\Big(\partial_{\hat{y}}\ell\big(\Psi(\mathrm{m}_\lambda(\mu_n), \widehat{\mathbf{x}}_1)], \widehat{\mathbf{y}}_1\big)\frac{\delta\Psi}{\delta \mathrm{m}}(\mathrm{m}_\lambda(\mu_n), \widehat{\mathbf{x}}_1, W)\Big)\Big]\\
&\le \mathrm{KL}(\mathrm{m}(\mu_n)\|\mathrm{m}(\mu_{n,(1)})) + \log\Big(\mathbb{E}_{W\sim \mathrm{m}^\alpha(\mu_{n,(1)})}\Big[\exp\Big\{\lambda_1\Big(\partial_{\hat{y}}\ell\big(\Psi(\mathrm{m}_\lambda(\mu_n), \widehat{\mathbf{x}}_1), \widehat{\mathbf{y}}_1\big)\frac{\delta\Psi}{\delta \mathrm{m}}(\mathrm{m}_\lambda(\mu_n), \widehat{\mathbf{x}}_1, W)\Big)\Big\}\Big]\Big).
\end{aligned} \tag{21}$$

With Lemma A.6,

$$\begin{aligned}
&\mathbb{E}_{W\sim \mathrm{m}^\alpha(\mu_{n,(1)})}\Big[\exp\Big\{\lambda_1\Big(\partial_{\hat{y}}\ell\big(\Psi(\mathrm{m}_\lambda(\mu_n), \widehat{\mathbf{x}}_1), \widehat{\mathbf{y}}_1\big)\Big(\frac{\delta\Psi}{\delta \mathrm{m}}(\mathrm{m}_\lambda(\mu_n), \widehat{\mathbf{x}}_1, W) - \mathbb{E}_{W\sim \mathrm{m}^\alpha(\mu_{n,(1)})}\Big[\frac{\delta\Psi}{\delta \mathrm{m}}(\mathrm{m}_\lambda(\mu_n), \widehat{\mathbf{x}}_1, W)\Big]\Big)\Big)\Big\}\Big]\\
&\le \exp\Big\{\frac{\lambda_1^2\sigma_1^2}{2}\Big\},
\end{aligned} \tag{22}$$

where $\sigma_1 = M_{\ell'} L_\psi N_{\max} M_\phi$. Combining (22) with (21), we can derive the following:

$$\begin{aligned}
&\lambda_1\Big(\partial_{\hat{y}}\ell\big(\Psi(\mathrm{m}_\lambda(\mu_n), \widehat{\mathbf{x}}_1), \widehat{\mathbf{y}}_1\big)\Big(\mathbb{E}_{W\sim \mathrm{m}(\mu_n)}\Big[\frac{\delta\Psi}{\delta \mathrm{m}}(\mathrm{m}_\lambda(\mu_n), \widehat{\mathbf{x}}_1, W)\Big] - \mathbb{E}_{W\sim \mathrm{m}(\mu_{n,(1)})}\Big[\frac{\delta\Psi}{\delta \mathrm{m}}(\mathrm{m}_\lambda(\mu_n), \widehat{\mathbf{x}}_1, W)\Big]\Big)\Big)\\
&\le \mathrm{KL}(\mathrm{m}(\mu_n)\|\mathrm{m}(\mu_{n,(1)})) + \frac{\lambda_1^2\sigma_1^2}{2},
\end{aligned}$$

This is a nonnegative parabola in λ_1, whose discriminant must be nonpositive. Therefore, we have,

$$\begin{aligned}
&\Big|\partial_{\hat{y}}\ell\big(\Psi(\mathrm{m}_\lambda(\mu_n), \widehat{\mathbf{x}}_1), \widehat{\mathbf{y}}_1\big)\Big(\mathbb{E}_{W\sim \mathrm{m}(\mu_n)}\Big[\frac{\delta\Psi}{\delta \mathrm{m}}(\mathrm{m}_\lambda(\mu_n), \widehat{\mathbf{x}}_1, W)\Big] - \mathbb{E}_{W\sim \mathrm{m}(\mu_{n,(1)})}\Big[\frac{\delta\Psi}{\delta \mathrm{m}}(\mathrm{m}_\lambda(\mu_n), \widehat{\mathbf{x}}_1, W)\Big]\Big)\Big|\\
&\le \sqrt{2\sigma_1^2 \mathrm{KL}(\mathrm{m}(\mu_n)\|\mathrm{m}(\mu_{n,(1)}))}.
\end{aligned} \tag{23}$$

The upper bound in (23) holds for all $\lambda \in [0,1]$. Therefore, combining with Lemma A.10, we have,

$$\begin{aligned}
&\overline{\mathrm{gen}}(\mathrm{m}^\alpha(\mu_n),\mu)\\
&= \mathbb{E}_{\mathbf{Z}_n,\widehat{Z}_1}\Big[\int_0^1 \partial_{\hat{y}}\ell\big(\Psi(\mathrm{m}_\lambda(\mu_n),\widehat{\mathbf{x}}_1),\widehat{\mathbf{y}}_1\big)\Big(\mathbb{E}_{W\sim\mathrm{m}(\mu_n)}\Big[\frac{\delta\Psi}{\delta\mathrm{m}}(\mathrm{m}_\lambda(\mu_n),\widehat{\mathbf{x}}_1,W)\Big]\\
&\qquad - \mathbb{E}_{W\sim\mathrm{m}(\mu_{n,(1)})}\Big[\frac{\delta\Psi}{\delta\mathrm{m}}(\mathrm{m}_\lambda(\mu_n),\widehat{\mathbf{x}}_1,W)\Big]\Big)\mathrm{d}\lambda\Big]\\
&\le \mathbb{E}_{\mathbf{Z}_n,\widehat{Z}_1}\Bigg[\int_0^1 \Big|\partial_{\hat{y}}\ell\big(\Psi(\mathrm{m}_\lambda(\mu_n),\widehat{\mathbf{x}}_1),\widehat{\mathbf{y}}_1\big)\Big(\mathbb{E}_{W\sim\mathrm{m}(\mu_n)}\Big[\frac{\delta\Psi}{\delta\mathrm{m}}(\mathrm{m}_\lambda(\mu_n),\widehat{\mathbf{x}}_1,W)\Big]\\
&\qquad - \mathbb{E}_{W\sim\mathrm{m}(\mu_{n,(1)})}\Big[\frac{\delta\Psi}{\delta\mathrm{m}}(\mathrm{m}_\lambda(\mu_n),\widehat{\mathbf{x}}_1,W)\Big]\Big)\Big|\mathrm{d}\lambda\Bigg]\\
&\le \mathbb{E}_{\mathbf{Z}_n,\widehat{Z}_1}\Big[\sqrt{2\sigma_1^2\mathrm{KL}(\mathrm{m}(\mu_n)\|\mathrm{m}(\mu_{n,(1)}))}\Big].
\end{aligned} \tag{24}$$

This completes the proof. □

Proof of Proposition 4.6 via Lemma A.2 and Lemma A.7. From Lemma A.10, it yields,

$$\begin{aligned}
&\overline{\mathrm{gen}}(\mathrm{m}^\alpha(\mu_n),\mu)\\
&= \mathbb{E}_{\mathbf{Z}_n,\widehat{Z}_1}\Big[\int_0^1\int_{\mathcal{W}} \partial_{\hat{y}}\ell\big(\Psi(\mathrm{m}_\lambda(\mu_n),\widehat{\mathbf{x}}_1),\widehat{\mathbf{y}}_1\big)\frac{\delta\Psi}{\delta\mathrm{m}}(\mathrm{m}_\lambda(\mu_n),\widehat{\mathbf{x}}_1,w)(\mathrm{m}(\mu_n)-\mathrm{m}(\mu_{n,(1)}))(\mathrm{d}w)\mathrm{d}\lambda\Big]\\
&= \mathbb{E}_{\mathbf{Z}_n,\widehat{Z}_1}\Big[\int_0^1 \partial_{\hat{y}}\ell\big(\Psi(\mathrm{m}_\lambda(\mu_n),\widehat{\mathbf{x}}_1),\widehat{\mathbf{y}}_1\big)\Big(\mathbb{E}_{W\sim\mathrm{m}(\mu_n)}\Big[\frac{\delta\Psi}{\delta\mathrm{m}}(\mathrm{m}_\lambda(\mu_n),\widehat{\mathbf{x}}_1,W)\Big]\\
&\qquad - \mathbb{E}_{W\sim\mathrm{m}(\mu_{n,(1)})}\Big[\frac{\delta\Psi}{\delta\mathrm{m}}(\mathrm{m}_\lambda(\mu_n),\widehat{\mathbf{x}}_1,W)\Big]\Big)\mathrm{d}\lambda\Big].
\end{aligned}$$

For the function $f(w) = \lambda_1\partial_{\hat{y}}\ell\big(\Psi(\mathrm{m}_\lambda(\mu_n),\widehat{\mathbf{x}}_1),\widehat{\mathbf{y}}_1\big)\frac{\delta\Psi}{\delta\mathrm{m}}(\mathrm{m}_\lambda(\mu_n),\widehat{\mathbf{x}}_1,w)$, due to Assumptions 4.1, 4.3 and 4.4, we have $\Big|\partial_{\hat{y}}\ell\big(\Psi(\mathrm{m}_\lambda(\mu_n),\widehat{\mathbf{x}}_1),\widehat{\mathbf{y}}_1\big)\frac{\delta\Psi}{\delta\mathrm{m}}(\mathrm{m}_\lambda(\mu_n),\widehat{\mathbf{x}}_1,w)\Big| \le M_{\ell'}L_\psi N_{\max}M_\phi$. Hence, from lemma A.2, it yields,

$$\big|\overline{\mathrm{gen}}(\mathrm{m}^\alpha(\mu_n),\mu)\big| \le M_{\ell'}L_\psi N_{\max}M_\phi\mathbb{E}_{\mathbf{Z}_n,\widehat{Z}_1}\Big[\mathbb{TV}(\mathrm{m}(\mu_n),\mathrm{m}(\mu_{n,(1)}))\Big]. \tag{25}$$

Using Lemma A.7 completes the proof. □

Proposition 4.8. ***(restated)*** *Let Assumptions 4.1 hold. Then, the following lower bound holds on the generalization error of the Gibbs measure $m^\alpha(\mu_n)$,*

$$\overline{\mathrm{gen}}(\mathrm{m}^\alpha(\mu_n),\mu) \ge \frac{n}{2\alpha}\mathbb{E}_{\mathbf{Z}_n,\widehat{Z}_1}\Big[\mathrm{KL}_{\mathrm{sym}}\big(\mathrm{m}^\alpha(\mu_n)\|\mathrm{m}^\alpha(\mu_{n,(1)})\big)\Big].$$

Proof. For simplicity of proof, we abbreviate

$$\tilde{\ell}(m,z) := \ell\big(\Psi(m,x),y\big), \tag{26}$$

$$\frac{\delta\tilde{\ell}}{\delta m}(m,z,w) := \partial_{\hat{y}}\ell(\Psi(m,x),y)\frac{\delta\Psi}{\delta m}(m,x,w), \tag{27}$$

where $z=(x,y)$ and (27) follows from chain rule. From Assumption 4.1, where the loss function is convex with respect to $\Psi(m,x)$ and due to the fact that $\Psi(m,x)$ is linear with respect to parameter measure m, then we have the convexity of $\tilde{\ell}(m,z)$ with respect to parameter measure. Recall that from (10) we have

$$\mathrm{m}^\alpha(\mu_n) = \frac{\pi}{S_{\alpha,\pi}(\mu_n)}\exp\Bigg\{-\alpha\Bigg[\frac{\delta\mathrm{R}(\mathrm{m}^\alpha,\mu_n,w)}{\delta m}\Bigg]\Bigg\},$$

and

$$\mathrm{m}^{\alpha}(\mu_{n,(1)}) = \frac{\pi}{S_{\alpha,\pi}(\mu_{n,(1)})} \exp\Big\{ -\alpha \Big[\frac{\delta \mathrm{R}(\mathrm{m}^{\alpha}, \mu_{n,(1)}, w)}{\delta m} \Big] \Big\}.$$

We need to compute the expectation of $\mathrm{KL}_{\mathrm{sym}}\big(\mathrm{m}^{\alpha}(\mu_{n,(1)}) \| \mathrm{m}^{\alpha}(\mu_n)\big)$ where $\mathrm{KL}_{\mathrm{sym}}(p\|q) = \mathrm{KL}(p\|q) + \mathrm{KL}(q\|p)$. For that purpose,

$$\begin{aligned}
&\mathbb{E}_{\mathbf{Z}_n, \widehat{Z}_1}\Big[\mathrm{KL}\big(\mathrm{m}^{\alpha}(\mu_n)\|\mathrm{m}^{\alpha}(\mu_{n,(1)})\big) + \mathrm{KL}\big(\mathrm{m}^{\alpha}(\mu_{n,(1)})\|\mathrm{m}^{\alpha}(\mu_n)\big)\Big] \\
&= \mathbb{E}_{\mathbf{Z}_n, \widehat{Z}_1}\Big[\int_{\mathcal{W}} \log\big(\mathrm{m}^{\alpha}(\mu_n)/\mathrm{m}^{\alpha}(\mu_{n,(1)})\big)(\mathrm{m}^{\alpha}(\mu_n) - \mathrm{m}^{\alpha}(\mu_{n,(1)}))(\mathrm{d}w)\Big] \\
&= \mathbb{E}_{\mathbf{Z}_n, \widehat{Z}_1}\Big[\int_{\mathcal{W}} \log\big(S_{\alpha,\pi}(\mu_{n,(1)})/S_{\alpha,\pi}(\mu_n)\big)(\mathrm{m}^{\alpha}(\mu_n) - \mathrm{m}^{\alpha}(\mu_{n,(1)}))(\mathrm{d}w)\Big] \\
&\quad + \mathbb{E}_{\mathbf{Z}_n, \widehat{Z}_1}\Big[\mathbb{E}_{W\sim \mathrm{m}^{\alpha}(\mu_{n,(1)})}\Big[\pi(W)\Big] - \mathbb{E}_{W\sim \mathrm{m}^{\alpha}(\mu_n)}\Big[\pi(W)\Big]\Big] \\
&\quad + \alpha\Big(\mathbb{E}_{\mathbf{Z}_n, \widehat{Z}_1}\Big[\mathbb{E}_{W\sim \mathrm{m}^{\alpha}(\mu_n)}\Big[\frac{\delta \mathrm{R}}{\delta m}(\mathrm{m}^{\alpha}(\mu_{n,(1)}), \mu_{n,(1)}, W) - \frac{\delta \mathrm{R}}{\delta m}(\mathrm{m}^{\alpha}(\mu_n), \mu_n, W)\Big]\Big] \\
&\quad - \mathbb{E}_{\mathbf{Z}_n, \widehat{Z}_1}\Big[\mathbb{E}_{W\sim \mathrm{m}^{\alpha}(\mu_{n,(1)})}\Big[\frac{\delta \mathrm{R}}{\delta m}(\mathrm{m}^{\alpha}(\mu_{n,(1)}), \mu_{n,(1)}, W) - \frac{\delta \mathrm{R}}{\delta m}(\mathrm{m}^{\alpha}(\mu_n), \mu_n, W)\Big]\Big]\Big).
\end{aligned}$$

Let us define the following terms,

$$\begin{aligned}
I_1 &:= \mathbb{E}_{\mathbf{Z}_n, \widehat{Z}_1}\Big[\int_{\mathcal{W}} \log\big(S_{\alpha,\pi}(\mu_{n,(1)})/S_{\alpha,\pi}(\mu_n)\big)(\mathrm{m}^{\alpha}(\mu_n) - \mathrm{m}^{\alpha}(\mu_{n,(1)}))(\mathrm{d}w)\Big], \\
I_2 &:= \mathbb{E}_{\mathbf{Z}_n, \widehat{Z}_1}\Big[\mathbb{E}_{W\sim \mathrm{m}^{\alpha}(\mu_{n,(1)})}\Big[\pi(W)\Big] - \mathbb{E}_{W\sim \mathrm{m}^{\alpha}(\mu_n)}\Big[\pi(W)\Big]\Big], \\
I_3 &:= \mathbb{E}_{\mathbf{Z}_n, \widehat{Z}_1}\Big[\mathbb{E}_{W\sim \mathrm{m}^{\alpha}(\mu_n)}\Big[\frac{\delta \mathrm{R}}{\delta m}(\mathrm{m}^{\alpha}(\mu_{n,(1)}), \mu_{n,(1)}, W)\Big] - \mathbb{E}_{W\sim \mathrm{m}^{\alpha}(\mu_{n,(1)})}\Big[\frac{\delta \mathrm{R}}{\delta m}(\mathrm{m}^{\alpha}(\mu_{n,(1)}), \mu_{n,(1)}, W)\Big]\Big], \\
I_4 &:= \mathbb{E}_{\mathbf{Z}_n, \widehat{Z}_1}\Big[\mathbb{E}_{W\sim \mathrm{m}^{\alpha}(\mu_{n,(1)})}\Big[\frac{\delta \mathrm{R}}{\delta m}(\mathrm{m}^{\alpha}(\mu_n), \mu_n, W)\Big] - \mathbb{E}_{W\sim \mathrm{m}^{\alpha}(\mu_n)}\Big[\frac{\delta \mathrm{R}}{\delta m}(\mathrm{m}^{\alpha}(\mu_n), \mu_n, W)\Big]\Big].
\end{aligned} \tag{28}$$

Note that $\log\big(S_{\alpha,\pi}(\mu_{n,(1)})/S_{\alpha,\pi}(\mu_n)\big)$ is not a function of parameters. Therefore, we have,

$$\begin{aligned}
I_1 &= \mathbb{E}_{\mathbf{Z}_n, \widehat{Z}_1}\Big[\int_{\mathcal{W}} \log\big(S_{\alpha,\pi}(\mu_{n,(1)})/S_{\alpha,\pi}(\mu_n)\big)(\mathrm{m}^{\alpha}(\mu_n) - \mathrm{m}^{\alpha}(\mu_{n,(1)}))(\mathrm{d}w)\Big] \\
&= 0.
\end{aligned} \tag{29}$$

Also $\pi(W)$ is not a function of data samples, therefore, we have

$$\begin{aligned}
I_2 &= \mathbb{E}_{\mathbf{Z}_n, \widehat{Z}_1}\Big[\mathbb{E}_{W\sim \mathrm{m}^{\alpha}(\mu_{n,(1)})}\Big[\pi(W)\Big] - \mathbb{E}_{W\sim \mathrm{m}^{\alpha}(\mu_n)}\Big[\pi(W)\Big]\Big] \\
&= 0.
\end{aligned} \tag{30}$$

By considering,

$$\frac{\delta \mathrm{R}}{\delta m}(\mathrm{m}^{\alpha}(\mu_n), \mu_n, w) = \frac{1}{n}\frac{\delta \tilde{\ell}}{\delta m}(\mathrm{m}^{\alpha}(\mu_n), Z_1, w) + \frac{1}{n}\sum_{i=2}^{n}\frac{\delta \tilde{\ell}}{\delta m}(\mathrm{m}^{\alpha}(\mu_n), Z_i, w), \tag{31}$$

$$\frac{\delta \mathrm{R}}{\delta m}(\mathrm{m}^{\alpha}(\mu_{n,(1)}), \mu_{n,(1)}, w) = \frac{1}{n}\frac{\delta \tilde{\ell}}{\delta m}(\mathrm{m}^{\alpha}(\mu_{n,(1)}), \widehat{Z}_1, w) + \frac{1}{n}\sum_{i=2}^{n}\frac{\delta \tilde{\ell}}{\delta m}(\mathrm{m}^{\alpha}(\mu_{n,(1)}), Z_i, w), \tag{32}$$

then, we have,

$$\begin{aligned}
I_3 &= \mathbb{E}_{\mathbf{Z}_n, \widehat{Z}_1}\Big[\mathbb{E}_{W\sim \mathrm{m}^{\alpha}(\mu_n)}\Big[\frac{\delta \mathrm{R}}{\delta m}(\mathrm{m}^{\alpha}(\mu_{n,(1)}), \mu_{n,(1)}, W)\Big] - \mathbb{E}_{W\sim \mathrm{m}^{\alpha}(\mu_{n,(1)})}\Big[\frac{\delta \mathrm{R}}{\delta m}(\mathrm{m}^{\alpha}(\mu_{n,(1)}), \mu_{n,(1)}, W)\Big]\Big] \\
&= \frac{1}{n}\mathbb{E}_{\mathbf{Z}_n, \widehat{Z}_1}\Big[\mathbb{E}_{W\sim \mathrm{m}^{\alpha}(\mu_n)}\Big[\frac{\delta \tilde{\ell}}{\delta m}(\mathrm{m}^{\alpha}(\mu_{n,(1)}), \widehat{Z}_1, W)\Big] - \mathbb{E}_{W\sim \mathrm{m}^{\alpha}(\mu_{n,(1)})}\Big[\frac{\delta \tilde{\ell}}{\delta m}(\mathrm{m}^{\alpha}(\mu_{n,(1)}), \widehat{Z}_1, W)\Big]\Big] \\
&\quad + \frac{1}{n}\sum_{i=2}^{n}\mathbb{E}_{\mathbf{Z}_n, \widehat{Z}_1}\Big[\mathbb{E}_{W\sim \mathrm{m}^{\alpha}(\mu_n)}\Big[\frac{\delta \tilde{\ell}}{\delta m}(\mathrm{m}^{\alpha}(\mu_{n,(1)}), Z_i, W)\Big] - \mathbb{E}_{W\sim \mathrm{m}^{\alpha}(\mu_{n,(1)})}\Big[\frac{\delta \tilde{\ell}}{\delta m}(\mathrm{m}^{\alpha}(\mu_{n,(1)}), Z_i, W)\Big]\Big].
\end{aligned} \tag{33}$$

$$
\begin{aligned}
I_4 &= \mathbb{E}_{\mathbf{Z}_n,\widehat{Z}_1}\Big[\mathbb{E}_{W\sim \mathrm{m}^\alpha(\mu_{n,(1)})}\Big[\frac{\delta \mathrm{R}}{\delta m}(\mathrm{m}^\alpha(\mu_n),\mu_n,W)\Big]-\mathbb{E}_{W\sim \mathrm{m}^\alpha(\mu_n)}\Big[\frac{\delta \mathrm{R}}{\delta m}(\mathrm{m}^\alpha(\mu_n),\mu_n,W)\Big]\Big] \\
&= \frac{1}{n}\mathbb{E}_{\mathbf{Z}_n,\widehat{Z}_1}\Big[\mathbb{E}_{W\sim \mathrm{m}^\alpha(\mu_{n,(1)})}\Big[\frac{\delta \tilde{\ell}}{\delta m}(\mathrm{m}^\alpha(\mu_n),Z_1,W)\Big]-\mathbb{E}_{W\sim \mathrm{m}^\alpha(\mu_{n,(1)})}\Big[\frac{\delta \tilde{\ell}}{\delta m}(\mathrm{m}^\alpha(\mu_n),Z_1,W)\Big]\Big] \\
&\quad+\frac{1}{n}\sum_{i=2}^{n}\mathbb{E}_{\mathbf{Z}_n,\widehat{Z}_1}\Big[\mathbb{E}_{W\sim \mathrm{m}^\alpha(\mu_{n,(1)})}\Big[\frac{\delta \tilde{\ell}}{\delta m}(\mathrm{m}^\alpha(\mu_n),Z_i,W)\Big]-\mathbb{E}_{W\sim \mathrm{m}^\alpha(\mu_n)}\Big[\frac{\delta \tilde{\ell}}{\delta m}(\mathrm{m}^\alpha(\mu_n),Z_i,W)\Big]\Big].
\end{aligned}
\tag{34}
$$

For $2 \le j \le n$ due to the fact that data samples are i.i.d, we have:

$$
\mathbb{E}_{\mathbf{Z}_n,\widehat{Z}_1}[\tilde{\ell}(\mathrm{m}(\mu_n),Z_j)] = \mathbb{E}_{\mathbf{Z}_n,\widehat{Z}_1}[\tilde{\ell}(\mathrm{m}(\mu_{n,(1)}),Z_j)]. \tag{35}
$$

Via the convexity of the loss function, $\tilde{\ell}(m,z)$, with respect to the parameter measure, m, from Lemma A.8, it holds that,

$$
\begin{aligned}
&\mathbb{E}_{\mathbf{Z}_n,\widehat{Z}_1}[\tilde{\ell}(\mathrm{m}(\mu_n),Z_j)-\tilde{\ell}(\mathrm{m}(\mu_{n,(1)}),Z_j)] \\
&\ge \mathbb{E}_{\mathbf{Z}_n,\widehat{Z}_1}\Big[\int \frac{\delta \tilde{\ell}}{\delta m}(\mathrm{m}(\mu_{n,(1)}),Z_j,w)(\mathrm{m}(\mu_n)-\mathrm{m}(\mu_{n,(1)}))(\mathrm{d}w)\Big].
\end{aligned}
\tag{36}
$$

Therefore, we obtain for $2 \le j \le n$,

$$
0 \ge \mathbb{E}_{\mathbf{Z}_n,\widehat{Z}_1}\Big[\int \frac{\delta \tilde{\ell}}{\delta m}(\mathrm{m}(\mu_{n,(1)}),Z_j,w)(\mathrm{m}(\mu_n)-\mathrm{m}(\mu_{n,(1)}))(\mathrm{d}w)\Big]. \tag{37}
$$

Similarly, we can show that, for $2 \le j \le n$,

$$
0 \ge \mathbb{E}_{\mathbf{Z}_n,\widehat{Z}_1}\Big[\int \frac{\delta \tilde{\ell}}{\delta m}(\mathrm{m}(\mu_n),Z_j,w)(\mathrm{m}(\mu_{n,(1)})-\mathrm{m}(\mu_n))(\mathrm{d}w)\Big]. \tag{38}
$$

Combining (37) and (38) with (33) and (34), the following holds:

$$
I_3 \le \frac{1}{n}\mathbb{E}_{\mathbf{Z}_n,\widehat{Z}_1}\Big[\mathbb{E}_{W\sim \mathrm{m}^\alpha(\mu_n)}\Big[\frac{\delta \tilde{\ell}}{\delta m}(\mathrm{m}^\alpha(\mu_{n,(1)}),\widehat{Z}_1,W)\Big]-\mathbb{E}_{W\sim \mathrm{m}^\alpha(\mu_{n,(1)})}\Big[\frac{\delta \tilde{\ell}}{\delta m}(\mathrm{m}^\alpha(\mu_{n,(1)}),\widehat{Z}_1,W)\Big]\Big], \tag{39}
$$

and

$$
I_4 \le \frac{1}{n}\mathbb{E}_{\mathbf{Z}_n,\widehat{Z}_1}\Big[\mathbb{E}_{W\sim \mathrm{m}^\alpha(\mu_{n,(1)})}\Big[\frac{\delta \tilde{\ell}}{\delta m}(\mathrm{m}^\alpha(\mu_n),Z_1,W)\Big]-\mathbb{E}_{W\sim \mathrm{m}^\alpha(\mu_n)}\Big[\frac{\delta \tilde{\ell}}{\delta m}(\mathrm{m}^\alpha(\mu_n),Z_1,W)\Big]\Big]. \tag{40}
$$

Therefore, from (29), (30), (39), and (40) we have,

$$
\begin{aligned}
&\mathbb{E}_{\mathbf{Z}_n,\widehat{Z}_1}\Big[\mathrm{KL}\big(\mathrm{m}^\alpha(\mu_n)\|\mathrm{m}^\alpha(\mu_{n,(1)})\big)+\mathrm{KL}\big(\mathrm{m}^\alpha(\mu_{n,(1)})\|\mathrm{m}^\alpha(\mu_n)\big)\Big] \\
&= I_1+I_2+\alpha(I_3+I_4) \\
&\le \frac{\alpha}{n}\mathbb{E}_{\mathbf{Z}_n,\widehat{Z}_1}\Big[\mathbb{E}_{W\sim \mathrm{m}^\alpha(\mu_n)}\Big[\frac{\delta \tilde{\ell}}{\delta m}(\mathrm{m}^\alpha(\mu_{n,(1)}),\widehat{Z}_1,W)\Big]-\mathbb{E}_{W\sim \mathrm{m}^\alpha(\mu_{n,(1)})}\Big[\frac{\delta \tilde{\ell}}{\delta m}(\mathrm{m}^\alpha(\mu_{n,(1)}),\widehat{Z}_1,W)\Big]\Big] \\
&+\frac{\alpha}{n}\mathbb{E}_{\mathbf{Z}_n,\widehat{Z}_1}\Big[\mathbb{E}_{W\sim \mathrm{m}^\alpha(\mu_{n,(1)})}\Big[\frac{\delta \tilde{\ell}}{\delta m}(\mathrm{m}^\alpha(\mu_n),Z_1,W)\Big]-\mathbb{E}_{W\sim \mathrm{m}^\alpha(\mu_n)}\Big[\frac{\delta \tilde{\ell}}{\delta m}(\mathrm{m}^\alpha(\mu_n),Z_1,W)\Big]\Big].
\end{aligned}
\tag{41}
$$

From Lemma A.8, we have:

$$
\begin{aligned}
&\mathbb{E}_{\mathbf{Z}_n,\widehat{Z}_1}\Big[\mathbb{E}_{W\sim \mathrm{m}^\alpha(\mu_n)}\Big[\frac{\delta \tilde{\ell}}{\delta m}(\mathrm{m}^\alpha(\mu_{n,(1)}),\widehat{Z}_1,W)\Big]-\mathbb{E}_{W\sim \mathrm{m}^\alpha(\mu_{n,(1)})}\Big[\frac{\delta \tilde{\ell}}{\delta m}(\mathrm{m}^\alpha(\mu_{n,(1)}),\widehat{Z}_1,W)\Big]\Big] \\
&\le \overline{\mathrm{gen}}(\mathrm{m}^\alpha(\mu_n),\mu).
\end{aligned}
\tag{42}
$$

Similarly from Remark A.9, we have,

$$
\begin{aligned}
&\mathbb{E}_{\mathbf{Z}_n,\widehat{Z}_1}\Big[\mathbb{E}_{W\sim \mathrm{m}^\alpha(\mu_{n,(1)})}\Big[\frac{\delta \tilde{\ell}}{\delta m}(\mathrm{m}^\alpha(\mu_n),Z_1,W)\Big]-\mathbb{E}_{W\sim \mathrm{m}^\alpha(\mu_n)}\Big[\frac{\delta \tilde{\ell}}{\delta m}(\mathrm{m}^\alpha(\mu_n),Z_1,W)\Big]\Big] \\
&\le \overline{\mathrm{gen}}(\mathrm{m}^\alpha(\mu_n),\mu).
\end{aligned}
\tag{43}
$$

The final results holds by combining (42) and (43) with (41). □

Theorem 4.9. ***(restated)*** *Let Assumptions 4.1, 4.3 and 4.4 hold. Then, the following upper bound holds on the generalization error of the Gibbs measure, i.e.,* $\mathrm{m}^{\alpha}(\mu_n)$*,*

$$\overline{\mathrm{gen}}(\mathrm{m}^{\alpha}(\mu_n), \mu) \leq \frac{\alpha C}{n},$$

where $C = (M_{\ell'} M_{\phi} L_{\psi} N_{\max})^2$*.*

Proof.

$$\begin{aligned}
&\frac{n}{2\alpha}\mathbb{E}_{\mathbf{Z}_n, \widehat{Z}_1}\Big[\mathrm{KL}\big(\mathrm{m}^{\alpha}(\mu_n)\|\mathrm{m}^{\alpha}(\mu_{n,(1)})\big)\Big] \\
&\leq \frac{n}{2\alpha}\mathbb{E}_{\mathbf{Z}_n, \widehat{Z}_1}\Big[\mathrm{KL}_{\mathrm{sym}}\big(\mathrm{m}^{\alpha}(\mu_n)\|\mathrm{m}^{\alpha}(\mu_{n,(1)})\big)\Big] && (44)\\
&\leq \overline{\mathrm{gen}}(\mathrm{m}^{\alpha}(\mu_n), \mu) && (45)\\
&\leq \big(M_{\ell'} L_{\psi} N_{\max} M_{\phi}/\sqrt{2}\big)\mathbb{E}_{\mathbf{Z}_n, \widehat{Z}_1}\Big[\sqrt{\mathrm{KL}\big(\mathrm{m}(\mu_n)\|\mathrm{m}(\mu_{n,(1)})\big)}\Big] && (46)\\
&\leq \big(M_{\ell'} L_{\psi} N_{\max} M_{\phi}/\sqrt{2}\big)\sqrt{\mathbb{E}_{\mathbf{Z}_n, \widehat{Z}_1}\big[\mathrm{KL}\big(\mathrm{m}(\mu_n)\|\mathrm{m}(\mu_{n,(1)})\big)\big]}, && (47)
\end{aligned}$$

where (44), (45), (46), (47), follow from the fact that $\mathrm{KL}(\cdot\|\cdot) \leq \mathrm{KL}_{\mathrm{sym}}(\cdot\|\cdot)$, Proposition 4.8, Proposition 4.6 and Jensen-Inequality, respectively. Therefore, we have,

$$\sqrt{\mathbb{E}_{\mathbf{Z}_n, \widehat{Z}_1}\big[\mathrm{KL}\big(\mathrm{m}(\mu_n)\|\mathrm{m}(\mu_{n,(1)})\big)\big]} \leq \Big(\frac{\sqrt{2}\alpha M_{\ell'} L_{\psi} N_{\max} M_{\phi}}{n}\Big). \tag{48}$$

The final result follows from combining (48) with the following inequality,

$$\overline{\mathrm{gen}}(\mathrm{m}^{\alpha}(\mu_n), \mu) \leq \big(M_{\ell'} L_{\psi} N_{\max} M_{\phi}/\sqrt{2}\big)\sqrt{\mathbb{E}_{\mathbf{Z}_n, \widehat{Z}_1}\big[\mathrm{KL}\big(\mathrm{m}(\mu_n)\|\mathrm{m}(\mu_{n,(1)})\big)\big]}.$$

□

C.2. GCN

Lemma 4.18. ***(restated)*** *Let Assumptions 4.5 and 4.17 hold. For a graph sample,* $(\mathbf{A}_q, \mathbf{F}_q)$ *with* N *nodes, and a graph filter* $G(\cdot)$*, the following upper bound holds on summation of GCN neuron units over all nodes:*

$$\sum_{j=1}^{N} |\phi_c(W_c, G(\mathbf{A}_q)[j,:]\mathbf{F}_q)| \leq N w_{2,c} L_{\varphi} \|W_{1,c}\|_2 B_f \min(\|G(\mathbf{A}_q)\|_{\infty}, \|G(\mathbf{A}_q)\|_F).$$

Proof. Recall that $\|G(\mathbf{A}_q)\|_{\infty} = \max_j \sum_{i=1}^{N} |G(\mathbf{A}_q)[j,i]|$, $\|G(\mathbf{A}_q)\|_F = \sqrt{\sum_{j=1}^{N}\sum_{i=1}^{N}(G(\mathbf{A}_q)[j,i])^2}$ and

$\|W_{1,c,m}\|_2 = \sup_{W_{1,c}\in\mathcal{S}^k} \|W_{1,c}\|_2$. Then, we have,

$$\begin{aligned}
\sum_{j=1}^{N} |\phi_c(W_c, G(\mathbf{A}_q)[j,:]\mathbf{F}_q)|)| &\leq \sum_{j=1}^{N} \sup_{W_{2,c}\in\mathcal{S}} |W_{2,c}||\varphi(G(\mathbf{A}_q)[j,:]\mathbf{F}_q).W_{1,c})| \\
&\leq \sum_{j=1}^{N} \sup_{W_{1,c}\in\mathcal{S}^k} w_{2,c}|\varphi(G(\mathbf{A}_q)[j,:]\mathbf{F}_q).W_{1,c,m})| \\
&\leq N w_{2,c} L_\varphi \|W_{1,c,m}\|_2 \max_j \|G(\mathbf{A}_q)[j,:]\mathbf{F}_q\|_2 \\
&\leq N w_{2,c} L_\varphi \|W_{1,c,m}\|_2 \max_j \left\| \sum_{i=1}^{N} G(\mathbf{A}_q)[j,i]\mathbf{F}_q[:,i] \right\|_2 \\
&\leq N w_{2,c} L_\varphi \|W_{1,c,m}\|_2 \max_j \Big(\sum_{i=1}^{N} |G(\mathbf{A}_q)[j,i]| \|\mathbf{F}_q[:,i]\|_2 \Big) \\
&\leq N w_{2,c} L_\varphi \|W_{1,c,m}\|_2 B_f \max_j \Big(\sum_{i=1}^{N} |G(\mathbf{A}_q)[j,i]| \Big) \\
&\leq N w_{2,c} L_\varphi \|W_{1,c,m}\|_2 B_f \|G(\mathbf{A}_q)\|_\infty .
\end{aligned} \tag{49}$$

We also have,

$$\begin{aligned}
\sum_{j=1}^{N} |\phi_c(W_c, G(\mathbf{A}_q)[j,:]\mathbf{F}_q)|)| &\leq \sum_{j=1}^{N} \sup_{W_{2,c}\in\mathcal{S}} |W_{2,c}||\varphi(G(\mathbf{A}_q)[j,:]\mathbf{F}_q).W_{1,c})| \\
&\leq \sum_{j=1}^{N} \sup_{W_{1,c}\in\mathcal{S}^k} w_{2,c}|\varphi(G(\mathbf{A}_q)[j,:]\mathbf{F}_q).W_{1,c,m})| \\
&\leq w_{2,c} L_\varphi \|W_{1,c,m}\|_2 \sum_{j=1}^{N} \|G(\mathbf{A}_q)[j,:]\mathbf{F}_q\|_2 \\
&\leq w_{2,c} L_\varphi \|W_{1,c,m}\|_2 \sum_{j=1}^{N} \left\| \sum_{i=1}^{N} G(\mathbf{A}_q)[j,i]\mathbf{F}_q[:,i] \right\|_2 \\
&\leq w_{2,c} L_\varphi \|W_{1,c,m}\|_2 \sum_{j=1}^{N} \sum_{i=1}^{N} |G(\mathbf{A}_q)[j,i]| \|\mathbf{F}_q[:,i]\|_2 \\
&\leq w_{2,c} L_\varphi \|W_{1,c,m}\|_2 B_f \sum_{j=1}^{N} \Big(\sum_{i=1}^{N} |G(\mathbf{A}_q)[j,i]| \Big) \\
&\leq N w_{2,c} L_\varphi \|W_{1,c,m}\|_2 B_f \|G(\mathbf{A}_q)\|_F .
\end{aligned} \tag{50}$$

This completes the proof. □

Proposition 4.19. ***(restated)*** *In a GCN with mean-readout function, under the combined assumptions for Theorem 4.9 and Lemma 4.18, the following upper bound holds on the generalization error of the Gibbs measure* $\mathrm{m}^{\alpha,c}(\mu_n)$*,*

$$\overline{\mathrm{gen}}(\mathrm{m}^{\alpha,c}(\mu_n), \mu) \leq \frac{\alpha M_c^2 M_{\ell'}^2 G_{\max}^2}{n} ,$$

where $M_c = w_{2,c} L_\varphi \|W_{1,c}\|_2 B_f$, $G_{\max} = \min\left(\|G(\mathcal{A})\|_\infty^{\max}, \|G(\mathcal{A})\|_F^{\max}\right)$, $\|G(\mathcal{A})\|_\infty^{\max} = \max_{\mathbf{A}_q\in\mathcal{A},\mu(f_q,\mathbf{A}_q)>0} \|G(\mathbf{A}_q)\|_\infty$ *and* $\|G(\mathcal{A})\|_F^{\max} = \max_{\mathbf{A}_q\in\mathcal{A},\mu(f_q,\mathbf{A}_q)>0} \|G(\mathbf{A}_q)\|_F$.

Proof. The result follows directly by combining Lemma 4.18 with Theorem 4.9 and assuming the mean-readout function. □

For sum-readout, we can modify the result as follows,

Corollary C.1 (GCN and sum-readout)**.** *In a GCN with the sum-readout function, under the combined assumptions for Theorem 4.9 and Lemma 4.18, the following upper bound holds on the generalization error of the Gibbs measure* $\mathrm{m}^{\alpha,c}(\mu_n)$*,*

$$\overline{\mathrm{gen}}(\mathrm{m}^{\alpha,c}(\mu_n),\mu) \leq \frac{\alpha M_c^2 M_{\ell'}^2 G_{\max}^2}{n},$$

where $M_c = w_{2,c} N_{\max} L_\varphi \|W_{1,c}\|_2 B_f$*,* $G_{\max} = \min\left(\|G(\mathcal{A})\|_\infty^{\max}, \|G(\mathcal{A})\|_F^{\max}\right)$*,* $\|G(\mathcal{A})\|_\infty^{\max} = \max_{\mathbf{A}_q \in \mathcal{A}, \mu(f_q,\mathbf{A}_q)>0} \|G(\mathbf{A}_q)\|_\infty$ *and* $\|G(\mathcal{A})\|_F^{\max} = \max_{\mathbf{A}_q \in \mathcal{A}, \mu(f_q,\mathbf{A}_q)>0} \|G(\mathbf{A}_q)\|_F$*.*

We can present a modified version of Proposition 4.19 that accommodates a bounded activation function. It is important to note that our analysis assumes the activation function to be Lipschitz continuous, a condition that holds for the popular Tanh function. However, for scenarios where the activation function is bounded, i.e.,

$$\sup_{\mathbf{A},\mathbf{F}} |\varphi(G(\mathbf{A})[j,:]\mathbf{F})| \leq M_\varphi$$

we can propose an updated upper bound in Proposition 4.19 as follows,

Corollary C.2. *Let us assume the same assumptions in Proposition 4.19 and a bounded activation function, i.e.,* $\sup_{\mathbf{A},\mathbf{F}} |\varphi(G(\mathbf{A})[j,:]\mathbf{F})| \leq M_\varphi$ *for* $j \in [N]$ *in GCN. Then the following upper bound holds on the generalization error of the Gibbs measure,* $\mathrm{m}^{\alpha,c}(\mu_n)$*,*

$$\overline{\mathrm{gen}}(\mathrm{m}^{\alpha,c}(\mu_n),\mu) \leq \frac{\alpha M_c^2 M_{\ell'}^2}{n},$$

where $M_c = w_{2,c} \min\left(M_\varphi, L_\varphi \|W_{1,c}\|_2 G_{\max} B_f\right)$*,* $G_{\max} = \min\left(\|G(\mathcal{A})\|_\infty^{\max}, \|G(\mathcal{A})\|_F^{\max}\right)$*,* $\|G(\mathcal{A})\|_\infty^{\max} = \max_{\mathbf{A}_q \in \mathcal{A}, \mu(f_q,\mathbf{A}_q)>0} \|G(\mathbf{A}_q)\|_\infty$ *and* $\|G(\mathcal{A})\|_F^{\max} = \max_{\mathbf{A}_q \in \mathcal{A}, \mu(f_q,\mathbf{A}_q)>0} \|G(\mathbf{A}_q)\|_F$*.*

C.3. MPGNN

Lemma 4.23. ***(restated)*** *Let Assumptions 4.5 and 4.22 hold. For a graph sample* $(\mathbf{A}_q, \mathbf{F}_q)$ *and a graph filter* $G(\cdot)$*, the following upper bound holds on the summation of MPU units over all nodes:*

$$\sum_{j=1}^{N} |\phi_p(W_p, G(\mathbf{A}_q)[j,:]\mathbf{F}_q)| \leq w_{2,p} L_\kappa B_f (\|W_{3,p,m}\|_2 + L_\rho L_\zeta G_{\max} \|W_{1,p,m}\|_2). \tag{51}$$

Proof. Recall that $\|G(\mathbf{A}_q)\|_\infty = \max_j \sum_{i=1}^N |G(\mathbf{A}_q)[j,i]|$, $\|G(\mathbf{A}_q)\|_F = \sqrt{\sum_{j=1}^N \sum_{i=1}^N (G(\mathbf{A}_q)[j,i])^2}$, $\|W_{1,p,m}\|_2 = \sup_{W_{1,c} \in \mathcal{S}^k} \|W_{1,p}\|_2$ and $\|W_{3,p,m}\|_2 = \sup_{W_{1,c} \in \mathcal{S}^k} \|W_{3,p}\|_2$. Then, we have,

$$\begin{aligned}
\sum_{j=1}^{N} |\phi_p(W_p, G(\mathbf{A}_q)[j,:]\mathbf{F}_q)| &\leq \sup_{W_{2,p} \in \mathcal{S}} |W_{2,p}| L_\kappa \Big(\|F_q[j,:]W_{3,p} + \rho(G(\mathbf{A}_q)[j,:]\zeta(\mathbf{F}_q))W_{1,p}\|_2 \Big) \\
&\leq \sum_{j=1}^{N} \sup_{W_{3,p}, W_{1,p} \in \mathcal{S}^k} w_{2,p} L_\kappa \Big(\|F_q[j,:]W_{3,p}\|_2 + \|\rho(G(\mathbf{A}_q)[j,:]\zeta(\mathbf{F}_q))W_{1,p}\|_2 \Big) \\
&\leq \sum_{j=1}^{N} w_{2,p} L_\kappa \Big(\|F_q[j,:]\|_2 \|W_{3,p,m}\|_2 + \|\rho(G(\mathbf{A}_q)[j,:]\zeta(\mathbf{F}_q))\|_2 \|W_{1,p,m}\|_2 \Big) \\
&\leq \sum_{j=1}^{N} w_{2,p} L_\kappa \Big(\|F_q[j,:]\|_2 \|W_{3,p,m}\|_2 + L_\rho L_\zeta \|G(\mathbf{A}_q)[j,:]\mathbf{F}_q\|_2 \|W_{1,p,m}\|_2 \Big) \\
&\leq N w_{2,p} L_\kappa B_f (\|W_{3,p,m}\|_2 + L_\rho L_\zeta \|G(\mathbf{A}_q)\|_\infty \|W_{1,p,m}\|_2),
\end{aligned}$$

Similar to Lemma 4.18, we can show that,

$$\sum_{j=1}^{N} |\phi_p(W_p, G(\mathbf{A}_q)[j,:]\mathbf{F}_q)| \leq N w_{2,p} L_\kappa B_f (\|W_{3,p,m}\|_2 + L_\rho L_\zeta \|G(\mathbf{A}_q)\|_F \|W_{1,p,m}\|_2).$$

□

Proposition 4.24. ***(restated)*** *In an MPGNN with the mean-readout function, under the combined assumptions for Theorem 4.9 and Lemma 4.23, the following upper bound holds on the generalization error of the Gibbs measure* $\mathrm{m}^{\alpha,p}(\mu_n)$*,*

$$\overline{\mathrm{gen}}(\mathrm{m}^{\alpha,p}(\mu_n),\mu) \leq \frac{\alpha M_p^2 M_{\ell'}^2}{n},$$

with $M_p = w_{2,p} L_\kappa B_f(\|W_{3,p}\|_2 + G_{\max} L_\rho L_\zeta \|W_{1,p}\|_2)$*,* $G_{\max} = \min\left(\|G(\mathcal{A})\|_\infty^{\max}, \|G(\mathcal{A})\|_F^{\max}\right)$*,* $\|G(\mathcal{A})\|_\infty^{\max} = \max_{\mathbf{A}_q \in \mathcal{A}, \mu(f_q, \mathbf{A}_q)>0} \|G(\mathbf{A}_q)\|_\infty$ *and* $\|G(\mathcal{A})\|_F^{\max} = \max_{\mathbf{A}_q \in \mathcal{A}, \mu(f_q, \mathbf{A}_q)>0} \|G(\mathbf{A}_q)\|_F$*.*

Proof. The result follows directly by combining Lemma 4.23 with Theorem 4.9 and assuming the mean-readout function. □

For the sum-readout function, similar to Corollary C.1, we have,

Corollary C.3 (MPGNN and sum-readout)**.** *In an MPGNN with the sum-readout function, under the combined assumptions for Theorem 4.9 and Lemma 4.23, the following upper bound holds on the generalization error of the Gibbs measure* $\mathrm{m}^{\alpha,p}(\mu_n)$*,*

$$\overline{\mathrm{gen}}(\mathrm{m}^{\alpha,p}(\mu_n),\mu) \leq \frac{\alpha M_p^2 M_{\ell'}^2}{n}.$$

with $M_p = w_{2,p} N_{\max} L_\kappa B_f(\|W_{3,p}\|_2 + G_{\max} L_\rho L_\zeta \|W_{1,p}\|_2)$*,* $G_{\max} = \min\left(\|G(\mathcal{A})\|_\infty^{\max}, \|G(\mathcal{A})\|_F^{\max}\right)$*,* $\|G(\mathcal{A})\|_\infty^{\max} = \max_{\mathbf{A}_q \in \mathcal{A}, \mu(f_q, \mathbf{A}_q)>0} \|G(\mathbf{A}_q)\|_\infty$ *and* $\|G(\mathcal{A})\|_F^{\max} = \max_{\mathbf{A}_q \in \mathcal{A}, \mu(f_q, \mathbf{A}_q)>0} \|G(\mathbf{A}_q)\|_F$*.*

In a similar approach to the proof in Corollary C.2, we can derive an upper bound on the generalization error of MPGNN based on the upper bound on the $\kappa(.)$ function.

Corollary C.4. *Let us assume the same assumptions in Proposition 4.19 and the bounded* $\kappa(.)$ *function, i.e.,* $\sup_{x\in\mathbb{R}} |\kappa(x)| \leq M_\kappa$ *, in MPGNN. Then the following upper bound holds on the generalization error of the Gibbs measure,* $\mathrm{m}^{\alpha,p}(\mu_n)$*,*

$$\overline{\mathrm{gen}}(\mathrm{m}^{\alpha,p}(\mu_n),\mu) \leq \frac{\alpha M_p^2 M_{\ell'}^2}{n},$$

with $M_p = w_{2,p} \min\left(M_\kappa, L_\kappa B_f(\|W_{3,p}\|_2 + G_{\max} L_\rho L_\zeta \|W_{1,p}\|_2)\right)$*,* $G_{\max} = \min\left(\|G(\mathcal{A})\|_\infty^{\max}, \|G(\mathcal{A})\|_F^{\max}\right)$*,* $\|G(\mathcal{A})\|_\infty^{\max} = \max_{\mathbf{A}_q \in \mathcal{A}, \mu(f_q, \mathbf{A}_q)>0} \|G(\mathbf{A}_q)\|_\infty$ *and* $\|G(\mathcal{A})\|_F^{\max} = \max_{\mathbf{A}_q \in \mathcal{A}, \mu(f_q, \mathbf{A}_q)>0} \|G(\mathbf{A}_q)\|_F$*.*

C.4. Details of Remark 4.20 and Remark 4.21

Using Lemma A.3, we have $\|G(\mathcal{A})\|_\infty^{\max} \leq \sqrt{(d_{\max}+1)/(d_{\min}+1)}$. Similar to Lemma A.3, for sum-aggregation $G(\mathbf{A}) = \mathbf{A} + I$ we have $\|G(\mathcal{A})\|_\infty^{\max} \leq d_{\max} + 1$, and for random-walk $G(\mathbf{A}) = D^{-1}\mathbf{A} + I$ we have $\|G(\mathcal{A})\|_\infty^{\max} = 2$.

Regarding to $\left\|G(\tilde{L})\right\|_F^{\max}$, using Lemma A.4, we have $\left\|G(\tilde{L})\right\|_F^{\max} \leq \sqrt{R_{\max}(\tilde{L})}\left\|G(\tilde{L})\right\|_2^{max}$. Then, via Lemma A.5, we have $\left\|G(\tilde{L})\right\|_2 = 1$. Similarly, for the random walk graph filter $G(\mathbf{A}) = \tilde{D}^{-1}\mathbf{A} + I$ we have $\|G(\mathcal{A})\|_F^{\max} \leq \left\|G(\tilde{D}^{-1}\mathbf{A} + I)\right\|_2^{max} \sqrt{R_{\max}(\tilde{D}^{-1}\mathbf{A} + I)}$ and $\left\|G(\tilde{D}^{-1}\mathbf{A} + I)\right\|_2^{max} = 2$.

D. Generalization Error Upper Bound via Rademacher Complexities

Inspired by the Rademacher Complexity analysis in (Nishikawa et al., 2022) and (Nitanda et al., 2022), we provide an upper bound on the generalization error of an over-parameterized one-hidden generic GNN model via Rademacher complexity analysis.

For Rademacher complexity analysis, we define the following hypothesis set for generic GNN functions based on the mean-field regime characterized by KL divergence between (m, π),

$$\mathcal{F}_{\mathrm{KL}}(H) \triangleq \left\{ \sum_{j=1}^{N} \mathbb{E}_{W\sim m}\left[\phi\left(W, G(\mathbf{A}_q)[j,:]\mathbf{F}_q\right)\right] : \mathrm{KL}(m\|\pi) \leq H \right\}. \tag{52}$$

To establish an upper bound on the generalization error of the generic GNN, we first use the following lemma to bound the Rademacher complexity of hypothesis set $\mathcal{F}_{\mathrm{KL}}(H)$.

Lemma D.1. *(Chen et al., 2020, Based on Lemma 5.5) Under Assumption 4.3, the following bound holds on the empirical Rademacher complexity of the hypothesis set $\mathcal{F}_{\mathrm{KL}}(H)$,*

$$\hat{\mathfrak{R}}_{\mathbf{Z}_n}(\mathcal{F}_{\mathrm{KL}}(H)) \leq N M_\phi \sqrt{\frac{2H}{n}}. \tag{53}$$

Proof. Without loss of generality, we denote $\phi_T(W, \mathbf{X}_i) := \sum_{j=1}^N \phi\left(W, G(\mathbf{A}_i)[j,:]\mathbf{F}_i\right)$. Then, with (52),

$$\mathcal{F}_{\mathrm{KL}}(H) = \left\{\mathbb{E}_{W\sim m}\Big[\phi_T(W, \mathbf{X}_q)\Big] : \mathrm{KL}(m\|\pi) \leq H\right\}.$$

From Donsker's representation of KL for the definition of $\hat{\mathfrak{R}}_{\mathbf{Z}_n}(\mathcal{F}_{\mathrm{KL}}(H))$ and considering a constant $\lambda > 0$, we have

$$\begin{aligned}
\hat{\mathfrak{R}}_{\mathbf{Z}_n}(\mathcal{F}_{\mathrm{KL}}(H)) &= \frac{1}{\lambda}\mathbb{E}_\sigma\left[\sup_{\substack{m:\mathrm{KL}(m\|\pi)\leq H,\\ m<<\pi}} \frac{\lambda}{n}\sum_{i=1}^n \sigma_i \mathbb{E}_{W\sim m}\phi_T(W, \mathbf{X}_i)\right] \\
&\leq \frac{1}{\lambda}\mathbb{E}_\sigma\left[\sup_{\substack{m:\mathrm{KL}(m\|\pi)\leq H,\\ m<<\pi}} \left\{\mathrm{KL}(m\|\pi) + \log\left(\mathbb{E}_{W\sim\pi}\exp\left(\frac{\lambda}{n}\sum_{i=1}^n \sigma_i\phi_T(W, \mathbf{X}_i)\right)\right)\right\}\right] \\
&\leq \frac{1}{\lambda}\mathbb{E}_\sigma\left[H + \log\left(\mathbb{E}_{W\sim\pi}\exp\left(\frac{\lambda}{n}\sum_{i=1}^n \sigma_i\phi_T(W, \mathbf{X}_i)\right)\right)\right] \\
&\leq \frac{1}{\lambda}\left[H + \log\left(\mathbb{E}_{W\sim\pi}\mathbb{E}_\sigma\exp\left(\frac{\lambda}{n}\sum_{i=1}^n \sigma_i\phi_T(W, \mathbf{X}_i)\right)\right)\right].
\end{aligned}$$

Here we applied Jensen's inequality in the last line. Note that, $\{\sigma_i\}_{i=1}^n$ are i.i.d. Rademacher random variables. Then, from the Hoeffding inequality with respect to Rademacher random variables, we have

$$\mathbb{E}_\sigma \exp\left(\frac{\lambda}{n}\sum_{i=1}^n \sigma_i\phi_T(W, \mathbf{X}_i)\right) \leq \exp\left\{\frac{\lambda^2}{2n^2}\sum_{i=1}^n \phi_T^2(W, \mathbf{X}_i)\right\}.$$

Note that, we have $\phi_T(W, \mathbf{X}_i) \leq N M_\phi$. Therefore, we have,

$$\begin{aligned}
\hat{\mathfrak{R}}_{\mathbf{Z}_n}(\mathcal{F}_{\mathrm{KL}}(H)) &\leq \frac{1}{\lambda}\Big(H + \frac{N^2 M_\phi^2 \lambda^2}{2n}\Big) \\
&= \frac{H}{\lambda} + \frac{N^2 M_\phi^2 \lambda}{2n},
\end{aligned}$$

which is minimized at $\lambda = \sqrt{\frac{2nH}{N^2 M_\phi^2}}$, yielding

$$\hat{\mathfrak{R}}_{\mathbf{Z}_n}(\mathcal{F}_{\mathrm{KL}}(H)) \leq N M_\phi \sqrt{\frac{2H}{n}}.$$

□

Remark D.2 (Comparison with Lemma 5.5 in (Chen et al., 2020)). In (Chen et al., 2020, Lemma 5.5), the authors assume a Gaussian prior π. However, in Lemma D.1, we do not assume a Gaussian prior. Instead, we assume a bounded unit output.

Note that, in Nishikawa et al. (2022) and Nitanda et al. (2022), it is assumed that there exists a "true" distribution $m_{\mathrm{true}} \in \mathcal{P}(\mathcal{W})$ which satisfies $\ell(\Psi(m_{\mathrm{true}}, \mathbf{X}_i), y_i) = 0$ for all $(\mathbf{X}_i, y_i) \in \mathcal{Z}$ where $\mu(\mathbf{X}_i, y_i) > 0$ and therefore we have $\mathrm{R}(m_{\mathrm{true}}, \mu_n) = 0$. In addition, it is assumed that the KL-divergence between the true distribution and the prior distribution is finite, i.e., $\mathrm{KL}(m_{\mathrm{true}}\|\pi) < \infty$. Due to the fact that the Gibbs measure is the minimizer of (9), $\mathcal{V}^\alpha(m, \mu_n)$, we have

$$\frac{1}{\alpha}\mathrm{KL}(\mathrm{m}^\alpha(\mu_n)\|\pi) \leq \mathrm{R}(\mathrm{m}^\alpha(\mu_n), \mu_n) + \frac{1}{\alpha}\mathrm{KL}(\mathrm{m}^\alpha(\mu_n)\|\pi) \tag{54}$$

$$\leq \mathrm{R}(m_{\mathrm{true}}, \mu_n) + \frac{1}{\alpha}\mathrm{KL}(m_{\mathrm{true}}\|\pi) \tag{55}$$

$$\leq \frac{1}{\alpha}\mathrm{KL}(m_{\mathrm{true}}\|\pi). \tag{56}$$

Therefore, the value of H is estimated by $\mathrm{KL}(m_{\mathrm{true}}\|\pi)$ in Nishikawa et al. (2022) and Nitanda et al. (2022). However, m_{true} is unknown and cannot be computed.

Considering the Gibbs measure, we provide the following proposition to estimate the value of H for the hypothesis set $\mathcal{F}_{\mathrm{KL}}(H)$ in terms of known parameters of problem formulation. This is our main theoretical contribution in the area of Rademacher Complexity analysis; it is related to results by (Chen et al., 2020; Nishikawa et al., 2022; Nitanda et al., 2022).

Proposition 4.13 (Upper bound on the symmetrized KL divergence)**.** *Under Assumptions 4.1, 4.3, and 4.4, the following upper bound holds on the symmetrized KL divergence between the Gibbs measure $\mathrm{m}^\alpha(\mu_n)$ and the prior measure π,*

$$\mathrm{KL}_{\mathrm{sym}}(\mathrm{m}^\alpha(\mu_n)\|\pi) \leq 2NM_\phi M_{\ell'} L_\psi \alpha.$$

Proof. The functional derivative of the empirical risk of a generic GNN concerning a measure m is

$$\frac{\delta \mathrm{R}(\mathrm{m}^\alpha(\mu_n), \mu_n, w)}{\delta m} = \frac{1}{n}\sum_{i=1}^{n} \partial_{\hat{y}}\ell\big(\mathbb{E}_{W\sim \mathrm{m}^\alpha(\mu_n)}[\Phi(W, \mathbf{x}_i)], y_i\big)\Phi(w, \mathbf{x}_i).$$

Note that $\frac{\delta \mathrm{R}(\mathrm{m}^\alpha(\mu_n), \mu_n, w)}{\delta m}$ is $NM_\phi M_{\ell'} L_\psi$-Lipschitz with respect to w under the metric $d(w, w') = \mathbb{1}_{w\neq w'}$. Recall that

$$\mathrm{m}^\alpha(\mu_n) = \frac{\pi}{S_{\alpha,\pi}(\mu_n)} \exp\Big\{ -\alpha \frac{\delta \mathrm{R}(\mathrm{m}^\alpha(\mu_n), \mu_n, w)}{\delta m}\Big\}.$$

We can compute the symmetrized KL divergence as follows:

$$\begin{aligned}
\mathrm{KL}_{\mathrm{sym}}(\mathrm{m}^\alpha(\mu_n)\|\pi) &= \mathrm{KL}(\mathrm{m}^\alpha(\mu_n)\|\pi) + \mathrm{KL}(\pi\|\mathrm{m}^\alpha(\mu_n)) \\
&= \mathbb{E}_{\mathrm{m}^\alpha(\mu_n)}\Big[\log\Big(\frac{\mathrm{m}^\alpha(\mu_n)}{\pi}\Big)\Big] + \mathbb{E}_{\pi}\Big[\log\Big(\frac{\pi}{\mathrm{m}^\alpha(\mu_n)}\Big)\Big] \\
&= \mathbb{E}_{\pi}[\log(S_{\alpha,\pi})] - \mathbb{E}_{\mathrm{m}^\alpha(\mu_n)}[\log(S_{\alpha,\pi})] \\
&\quad + \alpha\mathbb{E}_{\mathrm{m}^\alpha(\mu_n)}\left[\frac{\delta \mathrm{R}(\mathrm{m}^\alpha(\mu_n), \mu_n, w)}{\delta m}\right] - \alpha\mathbb{E}_{\pi}\left[\frac{\delta \mathrm{R}(\mathrm{m}^\alpha(\mu_n), \mu_n, w)}{\delta m}\right] \\
&\leq \alpha\left|\mathbb{E}_{\mathrm{m}^\alpha(\mu_n)}\left[\frac{\delta \mathrm{R}(\mathrm{m}^\alpha(\mu_n), \mu_n, w)}{\delta m}\right] - \mathbb{E}_{\pi}\left[\frac{\delta \mathrm{R}(\mathrm{m}^\alpha(\mu_n), \mu_n, w)}{\delta m}\right]\right| \\
&\leq 2\alpha L_R \mathbb{TV}(\mathrm{m}^\alpha(\mu_n), \pi),
\end{aligned}$$

where $L_R = NM_\phi M_{\ell'} L_\psi$. The last and second to the last inequalities follow from the total variation distance representation, (14), and the fact that

$$\mathbb{E}_{\pi}[\log(S_{\alpha,\pi})] - \mathbb{E}_{\mathrm{m}^\alpha(\mu_n)}[\log(S_{\alpha,\pi})] = 0\,.$$

Using that $\mathbb{TV}(\mathrm{m}^\alpha(\mu_n), \pi) \leq 1$ completes the proof. □

Combining Lemma D.1, Proposition 4.13, Lemma A.12 (the Uniform bound), and Lemma A.12 (Talagrand's contraction), we now derive the following upper bound on the generalization error for a generic GNN in the mean-field regime.

Theorem 4.14 (Generalization error upper bound via Rademacher complexity)**.** *Let Assumptions 4.1, 4.3, 4.4, and 4.2 hold. Then, for any $\delta \in (0, 1)$, with probability at least $1 - \delta$, the following upper bound holds on the generalization error of the Gibbs measure, i.e., $\mathrm{m}^\alpha(\mu_n)$, under the distribution of $P_{\mathbf{Z}_n}$,*

$$\mathrm{gen}(\mathrm{m}^\alpha(\mu_n), \mu) \leq 4N_{\max}M_\phi M_{\ell'} L_\psi \sqrt{\frac{N_{\max}M_\phi M_{\ell'} L_\psi \alpha}{n}} + 3M_\ell\sqrt{\frac{\log(2/\delta)}{2n}}\,.$$

Proof. From the uniform bound (Lemma A.11) and Talagrand's contraction lemma (Lemma A.12), we have for any $\delta \in (0,1)$

$$\mathrm{gen}(\mathrm{m}^{\alpha}(\mu_n), \mu) \leq 2M_{\ell'} L_{\psi} \hat{\mathfrak{R}}_{\mathbf{Z}_n}(\mathcal{F}_{\mathrm{KL}}(H)) + 3M_{\ell}\sqrt{\frac{1}{2n}\log\left(\frac{2}{\delta}\right)}. \tag{57}$$

Combining Lemma D.1 with (57) results in

$$\mathrm{gen}(\mathrm{m}^{\alpha}(\mu_n), \mu) \leq 2N_{\max} M_{\phi} M_{\ell'} L_{\psi}\sqrt{\frac{2H}{n}} + 3M_{\ell}\sqrt{\frac{1}{2n}\log\left(\frac{2}{\delta}\right)}. \tag{58}$$

Next, we find a suitable value of H to be used in the definition (52) of $\mathcal{F}_{\mathrm{KL}}(H)$. For that purpose, note that

$$\begin{aligned} \mathrm{KL}(\mathrm{m}^{\alpha}(\mu_n)\|\pi) &\leq \mathrm{KL}(\mathrm{m}^{\alpha}(\mu_n)\|\pi) + \mathrm{KL}(\mathrm{m}^{\alpha}(\mu_n)\|\pi) \\ &\leq 2N_{\max} M_{\phi} M_{\ell'} L_{\psi} \alpha \mathbb{TV}(\mathrm{m}^{\alpha}(\mu_n), \pi) \\ &\leq 2N_{\max} M_{\phi} M_{\ell'} L_{\psi} \alpha, \end{aligned} \tag{59}$$

where the last inequality follows from $\mathbb{TV}(\mathrm{m}^{\alpha}(\mu_n), \pi) \leq 1$. Therefore, we choose $H = 2N_{\max} M_{\phi} M_{\ell'} L_{\psi} \alpha$ in (58). This choice completes the proof. □

Note that the upper bound in Theorem 4.14 can be combined with Lemma 4.18 and Lemma 4.23 to provide upper bounds on the generalization error of a GCN and an MPGNN in the mean-field regime, respectively. In addition to the assumptions in Theorem 4.9, however, in Theorem 4.14 we need an extra assumption (Assumption 4.2).

Similar results to Corollary C.1 and Corollary C.3 for Rademacher complexity upper bounds based on sum-readout function can be derived using similar approach. Also, similar to Corollary C.2 and Corollary C.4, we can derive Rademacher complexity upper bounds based on mean-readout function.

E. More Discussion for Table 1

We also examine the dependency of our bound for the maximum/minimum node degree of graph data set $(d_{\max}, d_{\min})$ in Table 1. In Ju et al. (2023), the upper bound is dependent on the spectral norm (L_2 norm of the graph filter) which is independent of $(d_{\max}, d_{\min})$. The upper bound proposed by (Maskey et al., 2022), is dependent on $D_{\mathcal{X}}$ and the dimension of the space of graphon instead of the maximum and minimum degree. Note that, our bounds also depend on the maximum rank of the adjacency matrix in the graph data set.

F. Experiments

F.1. Implementation Details

Hardware and setup. Experiments were conducted on two compute nodes, each with 8 Nvidia Tesla T4 GPUs, 96 Intel Xeon Platinum 8259CL CPUs @ 2.50GHz and 378GB RAM. With this setup, all experiments were completed within one day. Note that we have six data sets, each with seven values of the width of the hidden layer, h, for two different supervised ratio values, β_{sup}, on three model types (GCN, GCN_RW[5], and MPGNN) with two different readout functions (mean and sum), for ten different random seeds. Therefore, there are a total of $6 \times 7 \times 2 \times 3 \times 2 \times 10 = 5,040$ single runs in our empirical analysis.

Code. Implementation code is provided at `https://github.com/SherylHYX/GNN_MF_GE`. We thank the authors of (Liao et al., 2020) for kindly sharing their code with us.

Training. We train 200 epochs for synthetic data sets and 50 epochs for PROTEINS. The batch size is 128 for all data sets.

[5]GCN with random walk as aggregation method

Regularization. Inspired by Chen et al. (2020), we propose the following regularized empirical risk minimization for GCN.

$$\mathrm{R}(m_h^c(\mu_n), \mu_n) = \frac{1}{n}\sum_{i=1}^{n} \ell\big(\mathbb{E}[\Psi_c(m_h^c(\mu_n), \mathbf{x}_i)], y_i\big) + \frac{1}{h\alpha}\sum_{i=1}^{h} \frac{\|W_c[i,:]\|_2^2}{2}, \tag{60}$$

where $W_c[i,:]$ denotes the parameters of the $i-$th neuron unit. For MPGNN, we consider the following regularized empirical risk minimization:

$$\mathrm{R}(m_h^p(\mu_n), \mu_n) = \frac{1}{n}\sum_{i=1}^{n} \ell\big(\Psi_p(m_h^p(\mu_n), \mathbf{x}_i), y_i\big) + \frac{1}{h\alpha}\sum_{i=1}^{h} \frac{\|W_p[i,:]\|_2^2}{2}, \tag{61}$$

where $W_p[i,:]$ is the parameters of $i-$th MPU unit.

Optimizer. Taking the regularization term into account, we use Stochastic Gradient Descend (SGD) from PyTorch as the optimizer and ℓ_2 regularization with weight decay $\frac{1}{h\alpha}$ to avoid overfitting, where h is the width of the hidden layer, and α is a tuning parameter which we set to be 100. We use a learning rate of 0.005 and a momentum of 0.9 throughout.

Tanh function. For ease of bound computation, we use Tanh for GCN as the activation function and for MPGNN as the non-linear function $\kappa(.)$.

F.2. Data Sets

We generate five synthetic data sets from two random graph models using NetworkX (Hagberg et al., 2008). The first three synthetic data sets correspond to Stochastic Block Models (SBMs) and the remaining two correspond to Erdős-Rényi (ER) models. The synthetic models have the following settings:

1. Stochastic-Block-Models-1 (SBM-1), where each graph has 100 nodes, two blocks with size 40 and 60, respectively. The edge probability matrix is
$$\begin{bmatrix} 0.25 & 0.13 \\ 0.13 & 0.37 \end{bmatrix}.$$
2. Stochastic-Block-Models-2 (SBM-2), where each graph has 100 nodes, and three blocks with sizes $25, 25,$ and 50, respectively. The edge probability matrix is
$$\begin{bmatrix} 0.25 & 0.05 & 0.02 \\ 0.05 & 0.35 & 0.07 \\ 0.02 & 0.07 & 0.40 \end{bmatrix}.$$
3. Stochastic-Block-Models-3 (SBM-3), where each graph has 50 nodes, and three blocks with sizes $15, 15,$ and 20, respectively. The edge probability matrix is
$$\begin{bmatrix} 0.5 & 0.1 & 0.2 \\ 0.1 & 0.4 & 0.1 \\ 0.2 & 0.1 & 0.4 \end{bmatrix}.$$
4. Erdős-Rényi-Models-4 (ER-4), where each graph has 100 nodes, with edge probability 0.7.
5. Erdős-Rényi-Models-5 (ER-5), where each graph has 20 nodes, with edge probability 0.5.

Each synthetic data set has 200 graphs, the number of classes is 2, and the random train-test split ratio is $\beta_{\text{sup}} : (1 - \beta_{\text{sup}})$, where in our experiments we vary β_{sup} in $\{0.7, 0.9\}$. For each random graph of an individual synthetic data set, we generate the 16-dimension random Gaussian node feature (normalized to have unit ℓ_2 norm) and a binary class label following a uniform distribution. In addition to the synthetic data sets, we have one real-world bioinformatics data set called PROTEINS (Borgwardt et al., 2005). In PROTEINS, nodes are secondary structure elements, and two nodes are connected by an edge if they are neighbors in the amino-acid sequence or in 3D space. Table 3 summarizes the statistics for the data sets in our experiments.

Table 3: Summary statistics for the data sets.

Statistics/Data Set	SBM-1	SBM-2	SBM-3	ER-4	ER-5	PROTEINS
Maximum number of nodes ($N_{\max}$)	100	100	50	100	20	620
Number of graphs	200	200	200	200	200	1113
Feature dimension	16	16	16	16	16	3
Maximum node degree	14	35	22	87	16	25
Minimum node degree	6	1	1	52	2	0

F.3. Bound Computation

As our upper bounds in Corollaries C.2 and C.4 are applicable for the over-parameterized regime in a continuous space of parameters, we estimate the upper bounds for a large number of hidden units, h. For this purpose, we need to compute the following parameters from the model and the data set:

- B_f: We compute the l_2 norm of the node features before GNN aggregation and find the max $\|F[j,:]\|_2$ for all training and test data. This would be the B_f term in Corollaries C.2 and C.4.
- $d_{\max}$: Maximum degree of a node in all graph samples.
- L_φ: Lipschitzness of the activation function. For Tanh, we have $L_\varphi = 1$.
- $w_{2,c}$: For GCN, we can choose the maximum value of $|W_{2,c}|$ as $w_{2,c}$. For MPGNN, we consider the maximum value of $|W_{2,p}|$ among all MPU units as $w_{2,c}$.
- M_ℓ and $M_{\ell'}$: As we consider the Tanh function as our activation function in GCN and as the $\kappa(.)$ function in MPGNN, we have $M_\ell = \log(1+\exp(-w_{2,c} * B))$ and we also have $M_{\ell'} = 1$. Note that the maximum value of the derivative of the logistic loss function $\ell(\Psi(W,\mathbf{x}),y) = \log(1+\exp(-\Psi(W,\mathbf{x})y))$ is 1.
- $\|W_{1,c,m}\|_2$: We consider the maximum value of $\|W_{1,c}\|_2$ among all neuron units.
- $\|W_{1,p,m}\|_2$ and $\|W_{3,p,m}\|_2$: We consider the maximum values of $\|W_{1,p}\|_2$ and $\|W_{3,p}\|_2$ among all MPU units as $\|W_{1,p,m}\|_2$ and $\|W_{3,p,m}\|_2$, respectively.

F.4. Extended Experimental Results

We compute the actual absolute empirical generalization errors (the difference between the test loss value and the training loss value) as well as empirical generalization error bounds for GCN, GCN_RW, and MPGNN with either the mean-readout function or the sum-readout function. Here GCN_RW denotes a variant of GCN where the symmetric normalization of the adjacency matrix is replaced by the random walk normalization. Specifically, Table 4 reports the effect of the width h of the hidden layer on the actual absolute empirical generalization errors for GCN when we employ a mean-readout function, over various data sets and supervised ratio values β_{sup}, where Table 5 and Table 6 reports those for GCN_RW and MPGNN when we employ a mean-readout function, respectively. Tables 7, 8, and 9 report the actual absolute empirical generalization errors for GCN, GCN_RW, and MPGNN with the sum-readout function, respectively. Table 10 reports the empirical generalization error bound values for all three model types for both mean and sum readout functions with hidden size $h = 256$.

F.5. Discussion

In all our experimental findings, increasing the width of the hidden layer leads to a reduction in generalization errors. Regarding the comparison of mean-readout and sum-readout, based on our results in Corollary C.1 and Proposition 4.19 for GCN and Corollary C.3 and Proposition 4.24 for MPGNN, the upper bound on the generalization error of GCN and MPGNN with mean-readout is less than the upper bound on the generalization error with sum-readout function in GCN and MPGNN, respectively. This pattern is similarly evident in the empirical generalization error results presented in Tables 4 and 7 for GCN and Tables 6 and 9 for MPGNN. As shown in Table 10 and Table 11, the upper bounds on the generalization error of GCN under $\tilde{L}$ and random walk as graph filters are similar. The empirical generalization error of GCN under $\tilde{L}$ and random walk in Tables 4 and 5 are also similar. Tables 12 to 15 provides other bound computation results based on smaller h values. We conclude that these are very similar to the bounds computed using $h = 256$, and hence $h = 256$ should be a proper approximation to the mean-field regime.

Table 4: Absolute empirical generalization error ($\times 10^5$) for different widths h of the hidden layer for GCN, for which we employ a mean-readout function and various supervised ratios of β_{sup}. We report the mean plus/minus one standard deviation over ten runs.

Data Set	β_{sup}	$h=4$	$h=8$	$h=16$	$h=32$	$h=64$	$h=128$	$h=256$
SBM-1	0.7	66.37 ± 44.09	18.66 ± 15.44	13.91 ± 9.46	3.42 ± 2.96	2.04 ± 1.68	1.23 ± 0.99	0.29 ± 0.27
SBM-1	0.9	101.34 ± 73.25	36.35 ± 37.01	12.76 ± 14.21	12.00 ± 4.81	3.56 ± 2.69	0.86 ± 0.60	0.60 ± 0.39
SBM-2	0.7	67.27 ± 44.95	19.57 ± 16.12	13.54 ± 9.79	3.79 ± 3.16	2.36 ± 1.71	1.25 ± 0.93	0.31 ± 0.22
SBM-2	0.9	103.32 ± 77.20	36.67 ± 40.96	15.38 ± 15.17	11.85 ± 4.41	3.51 ± 2.41	0.96 ± 0.56	0.60 ± 0.38
SBM-3	0.7	85.51 ± 54.72	42.42 ± 23.06	25.18 ± 10.59	6.72 ± 4.32	4.36 ± 3.63	1.13 ± 0.62	0.33 ± 0.23
SBM-3	0.9	108.83 ± 139.12	61.35 ± 45.92	24.04 ± 22.08	13.07 ± 13.03	4.19 ± 3.06	1.81 ± 1.50	0.73 ± 0.61
ER-4	0.7	69.63 ± 45.13	18.39 ± 15.77	14.04 ± 10.49	3.86 ± 3.36	2.17 ± 1.67	1.24 ± 1.02	0.33 ± 0.25
ER-4	0.9	102.21 ± 79.73	38.18 ± 37.76	13.49 ± 13.99	12.32 ± 4.67	3.52 ± 2.61	0.88 ± 0.56	0.59 ± 0.36
ER-5	0.7	123.16 ± 113.58	60.06 ± 29.55	39.63 ± 23.92	16.21 ± 7.46	3.34 ± 2.76	1.63 ± 1.02	0.59 ± 0.62
ER-5	0.9	203.04 ± 97.39	100.28 ± 95.76	48.95 ± 33.67	21.76 ± 17.30	6.91 ± 6.40	1.98 ± 1.92	1.00 ± 0.62
PROTEINS	0.7	256.91 ± 132.64	157.51 ± 145.88	50.78 ± 63.32	22.77 ± 19.97	8.37 ± 7.71	1.85 ± 1.28	1.72 ± 1.38
PROTEINS	0.9	95.66 ± 85.97	67.25 ± 52.78	26.35 ± 15.27	12.36 ± 7.95	2.18 ± 1.85	1.11 ± 1.16	0.33 ± 0.28

Table 5: Empirical absolute generalization error ($\times 10^5$) for different widths h of the hidden layer for GCN_RW, for which we employ a mean-readout function and various supervised ratios of β_{sup}. We report the mean plus/minus one standard deviation over ten runs.

Data Set	β_{sup}	$h=4$	$h=8$	$h=16$	$h=32$	$h=64$	$h=128$	$h=256$
SBM-1	0.7	66.71 ± 43.86	18.49 ± 15.25	13.85 ± 9.33	3.41 ± 3.08	2.05 ± 1.69	1.23 ± 0.99	0.29 ± 0.26
SBM-1	0.9	101.61 ± 72.92	35.81 ± 36.26	12.65 ± 13.99	12.01 ± 4.85	3.60 ± 2.71	0.85 ± 0.59	0.59 ± 0.39
SBM-2	0.7	66.87 ± 46.15	19.40 ± 16.29	13.70 ± 9.86	3.87 ± 3.37	2.35 ± 1.73	1.24 ± 0.94	0.32 ± 0.22
SBM-2	0.9	102.64 ± 77.20	37.06 ± 40.15	15.15 ± 15.06	11.90 ± 4.43	3.54 ± 2.42	0.95 ± 0.54	0.59 ± 0.38
SBM-3	0.7	84.58 ± 53.86	42.57 ± 22.71	25.41 ± 10.69	6.67 ± 4.39	4.37 ± 3.60	1.13 ± 0.60	0.33 ± 0.22
SBM-3	0.9	107.79 ± 138.73	61.28 ± 46.17	24.22 ± 22.75	13.08 ± 12.93	4.19 ± 3.09	1.81 ± 1.50	0.74 ± 0.63
ER-4	0.7	69.60 ± 45.04	18.40 ± 15.75	14.04 ± 10.50	3.85 ± 3.36	2.17 ± 1.67	1.24 ± 1.02	0.35 ± 0.26
ER-4	0.9	102.20 ± 79.69	38.14 ± 37.83	13.49 ± 13.99	12.32 ± 4.67	3.52 ± 2.61	0.88 ± 0.56	0.59 ± 0.37
ER-5	0.7	121.54 ± 111.05	59.94 ± 29.04	39.85 ± 23.30	16.13 ± 7.67	3.38 ± 2.81	1.62 ± 1.02	0.61 ± 0.62
ER-5	0.9	200.88 ± 97.85	99.61 ± 94.82	49.04 ± 34.17	21.71 ± 17.31	6.93 ± 6.44	1.98 ± 1.92	1.02 ± 0.62
PROTEINS	0.7	257.77 ± 133.07	157.76 ± 146.31	50.83 ± 63.41	22.79 ± 19.97	8.37 ± 7.72	1.85 ± 1.28	1.72 ± 1.38
PROTEINS	0.9	95.82 ± 85.65	67.42 ± 52.74	26.37 ± 15.19	12.38 ± 7.96	2.18 ± 1.84	1.11 ± 1.16	0.32 ± 0.28

Table 6: Empirical absolute generalization error ($\times 10^5$) for different widths h of the hidden layer for MPGNN, for which we employ a mean-readout function and various supervised ratios of β_{sup}. We report the mean plus/minus one standard deviation over ten runs.

Data Set	β_{sup}	$h=4$	$h=8$	$h=16$	$h=32$	$h=64$	$h=128$	$h=256$
SBM-1	0.7	31.39 ± 27.27	27.39 ± 37.85	11.50 ± 9.70	5.13 ± 5.20	3.10 ± 2.19	0.56 ± 0.42	0.77 ± 0.52
SBM-1	0.9	54.32 ± 35.73	24.44 ± 17.76	13.64 ± 8.20	7.76 ± 3.12	2.68 ± 1.35	1.80 ± 1.00	0.87 ± 0.57
SBM-2	0.7	31.59 ± 27.47	25.64 ± 36.66	11.10 ± 9.45	5.03 ± 5.17	3.28 ± 2.19	0.80 ± 0.56	0.77 ± 0.56
SBM-2	0.9	58.06 ± 36.49	26.80 ± 19.50	14.48 ± 8.15	7.38 ± 3.80	2.71 ± 1.58	1.86 ± 0.96	0.88 ± 0.63
SBM-3	0.7	42.21 ± 54.18	12.36 ± 9.99	15.00 ± 12.97	8.99 ± 5.73	3.18 ± 2.86	1.86 ± 1.40	1.10 ± 0.72
SBM-3	0.9	72.91 ± 35.61	26.80 ± 27.89	11.07 ± 6.90	9.62 ± 7.27	3.78 ± 2.42	2.55 ± 2.27	0.86 ± 0.79
ER-4	0.7	31.61 ± 26.04	25.19 ± 36.25	11.17 ± 9.81	5.01 ± 4.88	3.10 ± 2.14	0.64 ± 0.49	0.74 ± 0.56
ER-4	0.9	58.24 ± 34.28	25.84 ± 18.96	13.27 ± 7.39	7.47 ± 3.25	2.71 ± 1.43	1.79 ± 1.01	0.89 ± 0.56
ER-5	0.7	75.54 ± 57.19	35.99 ± 36.89	25.52 ± 15.42	10.60 ± 8.65	4.38 ± 4.97	3.01 ± 1.79	1.93 ± 1.02
ER-5	0.9	112.86 ± 40.24	54.54 ± 62.52	32.05 ± 20.23	13.84 ± 10.95	6.55 ± 4.59	1.96 ± 2.12	1.45 ± 0.75
PROTEINS	0.7	21.09 ± 13.77	18.05 ± 29.26	5.17 ± 3.57	2.33 ± 2.56	1.42 ± 1.23	1.17 ± 1.18	0.22 ± 0.23
PROTEINS	0.9	109.26 ± 71.68	57.88 ± 42.90	25.44 ± 22.23	13.80 ± 10.87	6.96 ± 4.22	4.37 ± 2.09	2.17 ± 1.58

Table 7: Empirical absolute generalization error ($\times\mathbf{10^5}$) for different widths h of the hidden layer for GCN, for which we employ a sum-readout function and various supervised ratios of β_{sup}. We report the mean plus/minus one standard deviation over ten runs.

Data Set	β_{sup}	$h=4$	$h=8$	$h=16$	$h=32$	$h=64$	$h=128$	$h=256$
SBM-1	0.7	32043.32 ± 16566.13	23188.17 ± 11633.58	12027.73 ± 5808.93	3292.38 ± 1512.93	954.86 ± 485.91	309.84 ± 195.80	69.97 ± 53.92
SBM-1	0.9	11014.75 ± 9029.01	8587.25 ± 5798.39	4112.02 ± 2643.86	1061.97 ± 842.66	483.87 ± 310.76	158.10 ± 111.88	59.79 ± 38.18
SBM-2	0.7	29492.62 ± 15154.35	21897.35 ± 10576.63	11528.95 ± 5424.90	3180.05 ± 1592.76	893.67 ± 510.50	302.50 ± 177.62	65.90 ± 47.15
SBM-2	0.9	10939.57 ± 9443.50	8767.78 ± 6205.08	4198.28 ± 2726.61	1029.97 ± 840.74	452.85 ± 299.91	162.18 ± 111.69	59.96 ± 35.63
SBM-3	0.7	24106.77 ± 12571.98	14354.73 ± 5584.92	5151.51 ± 3047.06	1978.71 ± 1092.38	300.63 ± 198.73	124.01 ± 81.82	28.69 ± 19.98
SBM-3	0.9	11512.44 ± 7587.11	6449.65 ± 4699.86	2671.03 ± 2532.28	1033.62 ± 1133.25	274.73 ± 169.96	110.43 ± 83.63	35.22 ± 25.80
ER-4	0.7	31101.54 ± 15568.03	22961.55 ± 10676.48	12133.39 ± 5563.73	3385.93 ± 1638.43	960.17 ± 499.79	311.68 ± 194.79	71.25 ± 53.81
ER-4	0.9	11349.50 ± 9665.34	8915.33 ± 6303.86	4305.80 ± 2789.51	1092.02 ± 868.09	487.43 ± 312.86	159.23 ± 109.41	59.54 ± 36.57
ER-5	0.7	12857.92 ± 6902.99	6270.49 ± 2471.73	2971.75 ± 1483.12	594.54 ± 481.81	222.13 ± 105.15	54.96 ± 21.80	20.09 ± 17.30
ER-5	0.9	3459.34 ± 2468.52	2126.09 ± 1573.64	937.35 ± 715.44	410.62 ± 305.72	161.93 ± 130.67	43.76 ± 37.61	20.60 ± 13.61
PROTEINS	0.7	2901.59 ± 1921.36	2763.40 ± 1750.78	2686.88 ± 1763.09	2639.22 ± 1548.08	2262.60 ± 1594.35	1395.51 ± 1028.10	552.28 ± 408.37
PROTEINS	0.9	5496.67 ± 121.21	5461.98 ± 30.99	5418.24 ± 53.89	5683.96 ± 131.28	6172.92 ± 113.16	4711.74 ± 182.59	2162.89 ± 181.60

Table 8: Empirical absolute generalization error ($\times\mathbf{10^5}$) for different widths h of the hidden layer for GCN_RW, for which we employ a sum-readout function and various supervised ratios of β_{sup}. We report the mean plus/minus one standard deviation over ten runs.

Data Set	β_{sup}	$h=4$	$h=8$	$h=16$	$h=32$	$h=64$	$h=128$	$h=256$
SBM-1	0.7	32202.69 ± 16672.76	23276.51 ± 11663.99	12093.87 ± 5792.22	3319.33 ± 1528.94	959.65 ± 484.71	310.31 ± 195.87	70.17 ± 53.66
SBM-1	0.9	11047.91 ± 9063.60	8605.98 ± 5812.74	4117.02 ± 2640.91	1051.64 ± 836.74	492.20 ± 312.51	157.20 ± 110.82	59.17 ± 37.52
SBM-2	0.7	29464.80 ± 15051.55	21963.74 ± 10570.57	11625.39 ± 5466.76	3205.53 ± 1653.48	900.03 ± 512.29	302.58 ± 179.68	66.34 ± 48.32
SBM-2	0.9	10969.08 ± 9674.08	8786.41 ± 6312.44	4219.59 ± 2785.49	1024.60 ± 849.98	458.05 ± 305.83	161.38 ± 110.93	59.30 ± 35.73
SBM-3	0.7	24204.66 ± 12196.40	14446.60 ± 5486.24	5198.20 ± 3006.84	1986.90 ± 1079.75	304.06 ± 200.22	124.71 ± 82.10	28.79 ± 19.92
SBM-3	0.9	11521.04 ± 7603.94	6424.98 ± 4711.22	2711.02 ± 2530.53	1038.27 ± 1121.70	272.09 ± 168.42	111.38 ± 83.39	35.89 ± 26.55
ER-4	0.7	31094.88 ± 15550.76	22956.86 ± 10665.25	12131.93 ± 5561.57	3385.06 ± 1636.21	960.45 ± 499.69	311.72 ± 194.88	71.27 ± 53.77
ER-4	0.9	11345.07 ± 9661.97	8913.06 ± 6303.20	4303.07 ± 2791.26	1091.45 ± 867.72	487.14 ± 312.64	159.09 ± 109.52	59.55 ± 36.57
ER-5	0.7	12866.49 ± 6889.85	6276.96 ± 2479.75	2982.31 ± 1475.88	590.39 ± 481.24	222.85 ± 104.62	54.99 ± 21.79	20.23 ± 17.47
ER-5	0.9	3435.09 ± 2462.73	2118.73 ± 1592.37	938.36 ± 717.11	408.90 ± 303.14	161.82 ± 132.05	43.84 ± 37.65	20.88 ± 13.62
PROTEINS	0.7	2900.35 ± 1919.12	2763.56 ± 1749.91	2687.83 ± 1762.29	2640.40 ± 1547.35	2262.67 ± 1594.47	1395.50 ± 1028.99	552.57 ± 408.40
PROTEINS	0.9	5501.96 ± 117.09	5470.32 ± 27.49	5425.94 ± 53.69	5690.28 ± 131.20	6181.97 ± 113.28	4720.11 ± 182.87	2166.19 ± 181.78

Table 9: Empirical absolute generalization error ($\times\mathbf{10^5}$) for different widths h of the hidden layer for MPGNN, for which we employ a sum-readout function and various supervised ratios of β_{sup}. We report the mean plus/minus one standard deviation over ten runs.

Data Set	β_{sup}	$h=4$	$h=8$	$h=16$	$h=32$	$h=64$	$h=128$	$h=256$
SBM-1	0.7	6585.43 ± 4142.24	5259.12 ± 5278.60	2267.42 ± 999.08	1870.21 ± 1126.94	863.73 ± 615.32	382.73 ± 199.14	221.63 ± 108.73
SBM-1	0.9	4891.93 ± 4654.50	3103.23 ± 3360.01	1857.44 ± 1867.96	911.86 ± 949.42	444.46 ± 290.26	311.74 ± 232.01	153.68 ± 150.55
SBM-2	0.7	6501.10 ± 4180.16	5044.84 ± 5075.24	2273.68 ± 998.62	1910.63 ± 1122.18	851.73 ± 613.75	390.12 ± 198.82	228.47 ± 108.99
SBM-2	0.9	5001.03 ± 4333.61	3282.25 ± 3330.88	1853.53 ± 1942.58	866.83 ± 949.21	474.32 ± 316.04	318.67 ± 232.69	160.81 ± 156.28
SBM-3	0.7	6344.20 ± 5047.26	2025.66 ± 1769.09	1928.82 ± 1554.69	1118.35 ± 712.80	555.00 ± 350.92	277.29 ± 221.90	145.94 ± 124.73
SBM-3	0.9	3740.18 ± 2378.95	1523.19 ± 1165.79	935.91 ± 700.15	579.09 ± 388.71	247.42 ± 228.09	180.99 ± 110.70	76.95 ± 49.52
ER-4	0.7	6407.19 ± 4182.56	5045.12 ± 5052.70	2261.19 ± 1028.58	1882.95 ± 1098.92	848.09 ± 617.37	383.24 ± 188.61	219.20 ± 108.37
ER-4	0.9	4492.42 ± 4098.88	3340.34 ± 3534.57	1754.96 ± 1911.25	841.93 ± 963.63	450.14 ± 282.38	307.74 ± 232.70	155.84 ± 149.04
ER-5	0.7	1175.50 ± 960.26	1004.51 ± 913.75	838.87 ± 482.98	362.88 ± 261.34	135.72 ± 67.97	112.66 ± 71.25	46.78 ± 37.25
ER-5	0.9	1843.43 ± 1152.07	1115.01 ± 1019.58	748.95 ± 399.20	270.69 ± 242.98	156.37 ± 94.23	77.84 ± 50.28	40.04 ± 18.55
PROTEINS	0.7	1084.56 ± 780.16	1007.44 ± 975.67	1306.72 ± 887.61	1084.37 ± 588.17	1085.74 ± 537.60	843.62 ± 527.95	762.83 ± 537.17
PROTEINS	0.9	558.80 ± 88.90	839.32 ± 97.22	1277.49 ± 147.09	1848.46 ± 100.86	2281.52 ± 64.97	2573.71 ± 48.59	2510.18 ± 29.38

Table 10: Empirical generalization error bounds via functional derivative for width $h = 256$ of the hidden layer for different model types and readout functions, for various supervised ratios of β_{sup}. We report the mean plus/minus one standard deviation over ten runs.

Data Set	β_{sup}	mean readout			sum readout		
		GCN	GCN_RW	MPGNN	GCN	GCN_RW	MPGNN
SBM-1	0.7	0.014 ± 0.001	0.015 ± 0.001	0.244 ± 0.013	$2520476.175 \pm 156887.505$	$2669876.975 \pm 164650.807$	4005.391 ± 220.699
SBM-1	0.9	0.011 ± 0.001	0.012 ± 0.001	0.190 ± 0.010	$1885484.212 \pm 142336.573$	$1996144.750 \pm 150151.799$	3118.617 ± 164.941
SBM-2	0.7	0.023 ± 0.001	0.025 ± 0.001	0.341 ± 0.019	$4089494.725 \pm 260354.119$	$4438159.850 \pm 275790.885$	5603.943 ± 323.262
SBM-2	0.9	0.018 ± 0.001	0.020 ± 0.001	0.266 ± 0.015	$3049281.675 \pm 228393.096$	$3310971.200 \pm 245013.883$	4364.249 ± 244.524
SBM-3	0.7	0.015 ± 0.001	0.016 ± 0.001	0.249 ± 0.013	624203.919 ± 38222.136	657592.244 ± 40341.332	4091.722 ± 236.617
SBM-3	0.9	0.011 ± 0.001	0.012 ± 0.001	0.193 ± 0.010	476286.562 ± 25613.135	501607.622 ± 26855.429	3182.215 ± 179.787
ER-4	0.7	0.005 ± 0.000	0.005 ± 0.000	0.122 ± 0.005	811877.012 ± 49965.822	815294.519 ± 50332.475	1993.695 ± 90.394
ER-4	0.9	0.004 ± 0.000	0.004 ± 0.000	0.095 ± 0.004	607448.919 ± 45159.041	609996.025 ± 45468.797	1552.299 ± 68.335
ER-5	0.7	0.008 ± 0.000	0.009 ± 0.001	0.168 ± 0.008	53656.979 ± 2753.653	58993.716 ± 3014.306	2752.105 ± 142.909
ER-5	0.9	0.006 ± 0.000	0.007 ± 0.000	0.131 ± 0.006	41149.381 ± 2515.410	45243.126 ± 2763.916	2137.883 ± 111.558
PROTEINS	0.7	0.005 ± 0.000	0.005 ± 0.000	0.045 ± 0.003	363676.153 ± 93112.546	363892.670 ± 93165.563	2122.023 ± 27.214
PROTEINS	0.9	0.004 ± 0.000	0.004 ± 0.000	0.035 ± 0.002	402014.978 ± 50920.169	402314.366 ± 50931.723	1657.479 ± 21.872

Table 11: Empirical generalization error bounds via Rademacher Complexities for width $h = 256$ of the hidden layer for different model types and readout functions, for various supervised ratios of β_{sup}. We report the mean plus/minus one standard deviation over ten runs.

Data Set	β_{sup}	mean readout			sum readout		
		GCN	GCN_RW	MPGNN	GCN	GCN_RW	MPGNN
SBM-1	0.7	0.465 ± 0.002	0.465 ± 0.002	3.774 ± 0.063	304382.491 ± 9500.618	304607.622 ± 9518.604	4814.476 ± 84.398
SBM-1	0.9	0.410 ± 0.002	0.410 ± 0.002	3.329 ± 0.055	260826.333 ± 7491.220	260911.930 ± 7550.637	4245.814 ± 74.430
SBM-2	0.7	0.465 ± 0.002	0.465 ± 0.002	3.774 ± 0.063	305824.653 ± 9984.800	305862.869 ± 10206.895	4814.185 ± 84.265
SBM-2	0.9	0.410 ± 0.002	0.410 ± 0.002	3.329 ± 0.055	261423.833 ± 7225.399	261561.489 ± 7270.949	4245.654 ± 74.354
SBM-3	0.7	0.465 ± 0.002	0.465 ± 0.002	3.774 ± 0.063	104673.709 ± 2587.052	104675.531 ± 2549.178	4816.482 ± 84.778
SBM-3	0.9	0.410 ± 0.002	0.410 ± 0.002	3.329 ± 0.055	91149.288 ± 1448.412	91135.775 ± 1449.661	4247.080 ± 74.968
ER-4	0.7	0.465 ± 0.002	0.465 ± 0.002	3.774 ± 0.063	305177.969 ± 10269.198	305189.119 ± 10276.768	4814.444 ± 84.389
ER-4	0.9	0.410 ± 0.002	0.410 ± 0.002	3.329 ± 0.055	261501.731 ± 7297.492	261507.970 ± 7300.305	4245.799 ± 74.568
ER-5	0.7	0.465 ± 0.002	0.465 ± 0.002	3.600 ± 0.137	26211.781 ± 520.483	26209.152 ± 522.064	4557.881 ± 205.714
ER-5	0.9	0.410 ± 0.002	0.410 ± 0.002	3.175 ± 0.120	22835.085 ± 242.080	22832.954 ± 242.523	4014.455 ± 180.352
PROTEINS	0.7	0.197 ± 0.001	0.197 ± 0.001	0.844 ± 0.030	59715.427 ± 7057.934	59730.912 ± 7058.069	2089.489 ± 20.112
PROTEINS	0.9	0.174 ± 0.001	0.174 ± 0.001	0.747 ± 0.026	64516.645 ± 4037.346	64535.982 ± 4035.333	1848.825 ± 18.308

Table 12: Empirical generalization error bounds via functional derivative for width $h = 128$ of the hidden layer for different model types and readout functions, for various supervised ratios of β_{sup}. We report the mean plus/minus one standard deviation over ten runs.

Data Set	β_{sup}	mean readout			sum readout		
		GCN	GCN_RW	MPGNN	GCN	GCN_RW	MPGNN
SBM-1	0.7	0.014 ± 0.001	0.015 ± 0.001	0.244 ± 0.013	$2520476.175 \pm 156887.505$	$2669876.975 \pm 164650.807$	4005.391 ± 220.699
SBM-1	0.9	0.011 ± 0.001	0.012 ± 0.001	0.190 ± 0.010	$1885484.212 \pm 142336.573$	$1996144.750 \pm 150151.799$	3118.617 ± 164.941
SBM-2	0.7	0.023 ± 0.001	0.025 ± 0.001	0.341 ± 0.019	$4089494.725 \pm 260354.119$	$4438159.850 \pm 275790.885$	5603.943 ± 323.262
SBM-2	0.9	0.018 ± 0.001	0.020 ± 0.001	0.266 ± 0.015	$3049281.675 \pm 228393.096$	$3310971.200 \pm 245013.883$	4364.249 ± 244.524
SBM-3	0.7	0.015 ± 0.001	0.016 ± 0.001	0.249 ± 0.013	624203.919 ± 38222.136	657592.244 ± 40341.332	4091.722 ± 236.617
SBM-3	0.9	0.011 ± 0.001	0.012 ± 0.001	0.193 ± 0.010	476286.562 ± 25613.135	501607.622 ± 26855.429	3182.215 ± 179.787
ER-4	0.7	0.005 ± 0.000	0.005 ± 0.000	0.122 ± 0.005	811877.012 ± 49965.822	815294.519 ± 50332.475	1993.695 ± 90.394
ER-4	0.9	0.004 ± 0.000	0.004 ± 0.000	0.095 ± 0.004	607448.919 ± 45159.041	609996.025 ± 45468.797	1552.299 ± 68.335
ER-5	0.7	0.008 ± 0.000	0.009 ± 0.001	0.168 ± 0.008	53656.979 ± 2753.653	58993.716 ± 3014.306	2752.105 ± 142.909
ER-5	0.9	0.006 ± 0.000	0.007 ± 0.000	0.131 ± 0.006	41149.381 ± 2515.410	45243.126 ± 2763.916	2137.883 ± 111.558
PROTEINS	0.7	0.005 ± 0.000	0.005 ± 0.000	0.045 ± 0.003	363676.153 ± 93112.546	363892.670 ± 93165.563	2122.023 ± 27.214
PROTEINS	0.9	0.004 ± 0.000	0.004 ± 0.000	0.035 ± 0.002	402014.978 ± 50920.169	402314.366 ± 50931.723	1657.479 ± 21.872

Table 13: Empirical generalization error bounds via Rademacher Complexities for width $h = 128$ of the hidden layer for different model types and readout functions, for various supervised ratios of β_{sup}. We report the mean plus/minus one standard deviation over ten runs.

Data Set	β_{sup}	mean readout			sum readout		
		GCN	GCN_RW	MPGNN	GCN	GCN_RW	MPGNN
SBM-1	0.7	0.465 ± 0.002	0.465 ± 0.002	3.774 ± 0.063	304382.491 ± 9500.618	304607.622 ± 9518.604	4814.476 ± 84.398
SBM-1	0.9	0.410 ± 0.002	0.410 ± 0.002	3.329 ± 0.055	260826.333 ± 7491.220	260911.930 ± 7550.637	4245.814 ± 74.430
SBM-2	0.7	0.465 ± 0.002	0.465 ± 0.002	3.774 ± 0.063	305824.653 ± 9984.800	305862.869 ± 10206.895	4814.185 ± 84.265
SBM-2	0.9	0.410 ± 0.002	0.410 ± 0.002	3.329 ± 0.055	261423.833 ± 7225.399	261561.489 ± 7270.949	4245.654 ± 74.354
SBM-3	0.7	0.465 ± 0.002	0.465 ± 0.002	3.774 ± 0.063	104673.709 ± 2587.052	104675.531 ± 2549.178	4816.482 ± 84.778
SBM-3	0.9	0.410 ± 0.002	0.410 ± 0.002	3.329 ± 0.055	91149.288 ± 1448.412	91135.775 ± 1449.661	4247.080 ± 74.968
ER-4	0.7	0.465 ± 0.002	0.465 ± 0.002	3.774 ± 0.063	305177.969 ± 10269.198	305189.119 ± 10276.768	4814.444 ± 84.389
ER-4	0.9	0.410 ± 0.002	0.410 ± 0.002	3.329 ± 0.055	261501.731 ± 7297.492	261507.970 ± 7300.305	4245.799 ± 74.568
ER-5	0.7	0.465 ± 0.002	0.465 ± 0.002	3.600 ± 0.137	26211.781 ± 520.483	26209.152 ± 522.064	4557.881 ± 205.714
ER-5	0.9	0.410 ± 0.002	0.410 ± 0.002	3.175 ± 0.120	22835.085 ± 242.080	22832.954 ± 242.523	4014.455 ± 180.352
PROTEINS	0.7	0.197 ± 0.001	0.197 ± 0.001	0.844 ± 0.030	59715.427 ± 7057.934	59730.912 ± 7058.069	2089.489 ± 20.112
PROTEINS	0.9	0.174 ± 0.001	0.174 ± 0.001	0.747 ± 0.026	64516.645 ± 4037.346	64535.982 ± 4035.333	1848.825 ± 18.308

Table 14: Empirical generalization error bounds via functional derivative for width $h = 64$ of the hidden layer for different model types and readout functions, for various supervised ratios of β_{sup}. We report the mean plus/minus one standard deviation over ten runs.

Data Set	β_{sup}	mean readout			sum readout		
		GCN	GCN_RW	MPGNN	GCN	GCN_RW	MPGNN
SBM-1	0.7	0.014 ± 0.001	0.015 ± 0.001	0.244 ± 0.013	$2520476.175 \pm 156887.505$	$2669876.975 \pm 164650.807$	4005.391 ± 220.699
SBM-1	0.9	0.011 ± 0.001	0.012 ± 0.001	0.190 ± 0.010	$1885484.212 \pm 142336.573$	$1996144.750 \pm 150151.799$	3118.617 ± 164.941
SBM-2	0.7	0.023 ± 0.001	0.025 ± 0.001	0.341 ± 0.019	$4089494.725 \pm 260354.119$	$4438159.850 \pm 275790.885$	5603.943 ± 323.262
SBM-2	0.9	0.018 ± 0.001	0.020 ± 0.001	0.266 ± 0.015	$3049281.675 \pm 228393.096$	$3310971.200 \pm 245013.883$	4364.249 ± 244.524
SBM-3	0.7	0.015 ± 0.001	0.016 ± 0.001	0.249 ± 0.013	624203.919 ± 38222.136	657592.244 ± 40341.332	4091.722 ± 236.617
SBM-3	0.9	0.011 ± 0.001	0.012 ± 0.001	0.193 ± 0.010	476286.562 ± 25613.135	501607.622 ± 26855.429	3182.215 ± 179.787
ER-4	0.7	0.005 ± 0.000	0.005 ± 0.000	0.122 ± 0.005	811877.012 ± 49965.822	815294.519 ± 50332.475	1993.695 ± 90.394
ER-4	0.9	0.004 ± 0.000	0.004 ± 0.000	0.095 ± 0.004	607448.919 ± 45159.041	609996.025 ± 45468.797	1552.299 ± 68.335
ER-5	0.7	0.008 ± 0.000	0.009 ± 0.001	0.168 ± 0.008	53656.979 ± 2753.653	58993.716 ± 3014.306	2752.105 ± 142.909
ER-5	0.9	0.006 ± 0.000	0.007 ± 0.000	0.131 ± 0.006	41149.381 ± 2515.410	45243.126 ± 2763.916	2137.883 ± 111.558
PROTEINS	0.7	0.005 ± 0.000	0.005 ± 0.000	0.045 ± 0.003	363676.153 ± 93112.546	363892.670 ± 93165.563	2122.023 ± 27.214
PROTEINS	0.9	0.004 ± 0.000	0.004 ± 0.000	0.035 ± 0.002	402014.978 ± 50920.169	402314.366 ± 50931.723	1657.479 ± 21.872

Table 15: Empirical generalization error bounds via Rademacher Complexities for width $h = 64$ of the hidden layer for different model types and readout functions, for various supervised ratios of β_{sup}. We report the mean plus/minus one standard deviation over ten runs.

Data Set	β_{sup}	mean readout			sum readout		
		GCN	GCN_RW	MPGNN	GCN	GCN_RW	MPGNN
SBM-1	0.7	0.465 ± 0.002	0.465 ± 0.002	3.774 ± 0.063	304382.491 ± 9500.618	304607.622 ± 9518.604	4814.476 ± 84.398
SBM-1	0.9	0.410 ± 0.002	0.410 ± 0.002	3.329 ± 0.055	260826.333 ± 7491.220	260911.930 ± 7550.637	4245.814 ± 74.430
SBM-2	0.7	0.465 ± 0.002	0.465 ± 0.002	3.774 ± 0.063	305824.653 ± 9984.800	305862.869 ± 10206.895	4814.185 ± 84.265
SBM-2	0.9	0.410 ± 0.002	0.410 ± 0.002	3.329 ± 0.055	261423.833 ± 7225.399	261561.489 ± 7270.949	4245.654 ± 74.354
SBM-3	0.7	0.465 ± 0.002	0.465 ± 0.002	3.774 ± 0.063	104673.709 ± 2587.052	104675.531 ± 2549.178	4816.482 ± 84.778
SBM-3	0.9	0.410 ± 0.002	0.410 ± 0.002	3.329 ± 0.055	91149.288 ± 1448.412	91135.775 ± 1449.661	4247.080 ± 74.968
ER-4	0.7	0.465 ± 0.002	0.465 ± 0.002	3.774 ± 0.063	305177.969 ± 10269.198	305189.119 ± 10276.768	4814.444 ± 84.389
ER-4	0.9	0.410 ± 0.002	0.410 ± 0.002	3.329 ± 0.055	261501.731 ± 7297.492	261507.970 ± 7300.305	4245.799 ± 74.568
ER-5	0.7	0.465 ± 0.002	0.465 ± 0.002	3.600 ± 0.137	26211.781 ± 520.483	26209.152 ± 522.064	4557.881 ± 205.714
ER-5	0.9	0.410 ± 0.002	0.410 ± 0.002	3.175 ± 0.120	22835.085 ± 242.080	22832.954 ± 242.523	4014.455 ± 180.352
PROTEINS	0.7	0.197 ± 0.001	0.197 ± 0.001	0.844 ± 0.030	59715.427 ± 7057.934	59730.912 ± 7058.069	2089.489 ± 20.112
PROTEINS	0.9	0.174 ± 0.001	0.174 ± 0.001	0.747 ± 0.026	64516.645 ± 4037.346	64535.982 ± 4035.333	1848.825 ± 18.308