Learning Theory Can (Sometimes) Explain Generalisation in Graph Neural Networks

Pascal Mattia Esser[1] [0000−0002−1235−7918], Leena C. Vankadara[2] [0000−0002−3810−840X], and Debarghya Ghoshdastidar[1,3] [0000−0003−0202−7007]

[1] Technical University of Munich {esser,ghoshdas}@in.tum.de
[2] University of Tübingen leena.chennuru-vankadara@uni-tuebingen.de
[3] Munich Data Science Institute

Abstract. In recent years, several results in the supervised learning setting suggested that classical statistical learning-theoretic measures, such as VC dimension, do not adequately explain the performance of deep learning models which prompted a slew of work in the infinite-width and iteration regimes. However, there is little theoretical explanation for the success of neural networks beyond the supervised setting. In this paper we argue that, under some distributional assumptions, classical learning-theoretic measures can sufficiently explain generalization for graph neural networks in the transductive setting. In particular, we provide a rigorous analysis of the performance of neural networks in the context of transductive inference, specifically by analysing the generalisation properties of graph convolutional networks for the problem of node classification. While VC Dimension does result in trivial generalisation error bounds in this setting as well, we show that transductive Rademacher complexity can explain the generalisation properties of graph convolutional networks for stochastic block models. We further use the generalisation error bounds based on transductive Rademacher complexity to demonstrate the role of graph convolutions and network architectures in achieving smaller generalisation error and provide insights into when the graph structure can help in learning. The findings of this paper could re-new the interest in studying generalisation in neural networks in terms of learning-theoretic measures, albeit in specific problems.
An extended version of this paper was published in NeurIPS 2021.

1 Introduction

Neural networks have found tremendous success in a wide range of practical applications and, in the broader society, it is often considered synonymous to machine learning. The rapid gain in popularity has, however, come at the cost of interpretability and reliability of complex neural network architectures. Hence, there has been an increasing interest in understanding generalization and other theoretical properties of neural networks in the theoretical machine learning community (Feldman 2020; Arora et al. 2019a; Ma et al. 2017; Nagarajan et al. 2019; Theisen et al. 2020; Ghorbani et al. 2020). Most of the existing theory literature focuses on the supervised learning problem, or more precisely, the setting of inductive inference. In contrast, there is a general lack of understanding of transductive problems, in particular the role of unlabeled data in training. Consequently there has also been little progress in rigorously understanding one of widely used tools for transductive inference—Graph neural networks (GNN).

Graph neural networks. GNNs were introduced by Gori et al. (2005) and Scarselli et al. (2009), who used recurrent neural network architectures, for the purpose of transductive inference on graphs, that is, the task of labelling all the nodes of a graph given the graph structure, all node features and labels for few nodes. Broadly, GNNs use a combination of local aggregation of node features and non-linear transformations to predict on unlabelled nodes. In practice, the exact form of aggregation and combination steps varies across architectures to solve domain specific tasks (Kipf et al. 2017; Bruna et al. 2014; Defferrard et al. 2016; Veličković et al. 2018; Xu et al. 2019). While some GNNs focus on the transductive setting, sometimes referred to as semi-supervised node classification,[4] GNNs have also found success in supervised learning, where the task is to label entire graphs, in contrast to labelling nodes in a graph. While the understanding of GNNs is limited, there are empirical approaches to study GNNs in the transductive (Bojchevski et al. 2018) and supervised setting (Zhang et al. 2018; Ying et al. 2018). For an extensive survey on the state of the art of GNNs see for example Wu et al. (2020).

Leaning theoretical analysis of GNNs. While empirical studies provide some insights into the behaviour of machine learning models, rigorous theoretical analysis is the key to deep insights into a model. The focus of this paper is to provide a learning-theoretic analysis of generalisation of GNNs in the transductive setting. Vapnik first studied the problem of transductive inference and provided generalisation bounds for empirical risk minimization (Vapnik 1982; Vapnik 1998). Subsequent works further analysed this setting in transductive regression (Cortes et al. 2007), and derive VC Dimension and Rademacher complexity for transductive classification (Tolstikhin et al. 2016; El-Yaniv et al. 2009). Generalisation error bounds for 1-layer GNNs have been derived in transductive setting based on algorithmic stability (Verma et al. 2019). In contrast, the focus of the current paper is on learning-theoretic measures, which have been previously used to analyse GNNs in a supervised setting. In Scarselli et al. (2018), VC Dimension is derived for a specific class of GNNs and a generalisation error bound is given using node representations. However, their approach of subsuming the graph convolutions under Pfaffian functions does not allow for an explicit representation in terms of the diffusion operator which is important to our presented analysis. Garg et al. (2020) derives the Rademacher complexity

[4] In semi-supervised learning, the learner is given a training set of labeled and unlabeled examples and the goal is to generate a hypothesis that generates predictions on the unseen examples. In transductive learning all features are available to the learner, and the goal is to transfer knowledge from the labeled to the unlabeled data points. The focus of graph-based semi-supervised learning aligns more with the latter setting.

for GNN in a supervised setting with the focus of the equivariant structures of the input graphs and does not allow for an explicit inclusion and analysis of the graph information. Liao et al. (2021) provides PAC-Bayes bounds for GNNs that are tighter than the bounds in Garg et al. (2020).

In the context of this work, especially relevant is Oono et al. (2020b) and Oono et al. (2020a). Oono et al. (2020a) describes the effect of oversmoothing with increasing number of layers. A more detailed comparison to our work is presented in section 2.3. Oono et al. (2020b) analyzes GNNs in the transductive setting. However, they consider a multiscale GCN, and therefore, the analysis is based in a weak-learning/boosting framework where the focus is mostly on exploring the weak learning component, whereas this paper focuses on the specific analysis of the generalization bound and the influence of it's individual components. In addition, we provide a detailed analysis of its dependence on the graph and feature information and provide a more expressive bound by considering generalization under planted models.

Infinite limit analysis. In the broader deep learning, there has been a growing call for alternatives to standard learning-theoretic bounds since they do not adequately capture the behaviour of deep models (Neyshabur et al. 2017). To this end, different limiting case analysis have been introduced. In the context of GNNs, it is known that GNNs have a fundamental connection to belief propagation and message passing (Dai et al. 2016; Gilmer et al. 2017) and some theoretical analyses of GNNs have been based on cavity methods and mean field approaches for supervised (Zhou et al. 2020) and transductive settings (Kawamoto et al. 2019; Chen et al. 2019). The central idea of these approaches is to show results in the limit of the number of iterations. In another limiting setting, Du et al. (2019) study GNNs with infintiely wide hidden layers, and derive corresponding neural tangent kernel (Jacot et al. 2018; Arora et al. 2019b) that can provide generalisation error bounds in the supervised setting. Keriven et al. (2020) derive continuous versions of GNNs applied to large random graphs. While limiting assumptions allow for a theoretical analysis, it is difficult to infer the implications of these results for finite GNNs.

Contributions and paper structure. We reconsider classical learning-theoretic measures to analyse GNNs, with a specific focus on explicitly characterising the influence of the graph information and the network architecture on generalisation. In the process, we show that, under careful construction of the complexity measure and distributional assumptions on the graph data, learning theory can provide insights into the behaviour of GNNs. The main contributions are the following:

1) We introduce a formal setup for graph-based transductive inference, and in Section 2.2, we use this framework to show that VC Dimension based generalisation error bounds are typically loose, except for few trivial cases. This observation is along the lines of existing evidence for neural networks.

2) In Section 2.3, we derive generalization bounds based on the transductive Rademacher complexity. Our results show that these bounds are more informative, suggesting that the correct choice of complexity measure is important.

3) We further refine the generalisation error bounds in Section 3 under a planted model for the graph and features. Such an analysis, under random graphs, is rare in GNN literature. We empirically show that the test error is consistent with the trends predicted by the theoretical bound. Our results suggest that, under distributional assumptions, learning-theoretic bounds can explain behaviour of GNNs.

We conclude in Section 4. All proofs and an overview of the notation are provided in the appendix.

2 Statistical Framework for Transductive Learning on GNN

For a rigorous analysis, we introduce a statistical learning framework for graph based transductive inference in Section 2.1. Based on this, we derive generalisation error bounds based on VC Dimension in Section 2.2 and demonstrate that the bounds have limited expresitivity even under strong assumptions. To overcome this problem we consider transductive Rademacher complexity in Section 2.3. While without further assumptions this bound also gives limited insight, the bound is more expressive and, in Section 3, we show that it can provide meaningful bounds under certain distributional assumptions.

2.1 Framework for Transductive Learning

We briefly recall the framework for supervised binary classification. Let $\mathcal{X} = \mathbb{R}^d$ be the *domain or feature space* and $\mathcal{Y} = \{\pm 1\}$ be the *label set.* The goal is to find a predictor $h : \mathcal{X} \to \mathcal{Y}$ based on m training samples $S \triangleq \{(\boldsymbol{x}_i, y_i)\}_{i=1}^m \subset \mathcal{X} \times \mathcal{Y}$ and a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to [0, \infty)$. In a statistical framework, we assume that S consists independent labelled samples from a distribution $\mathcal{D} = \mathcal{D}_\mathcal{X} \times \eta$, that is, $\boldsymbol{x}_i \sim \mathcal{D}_\mathcal{X}$ and $\boldsymbol{y}_i \sim \eta(\boldsymbol{x}_i)$, where $\eta(\cdot)$ governs the label probability for each feature. The goal of learning is to find h that minimises the *risk / generalisation error* $\mathcal{L}_\mathcal{D}(h) \triangleq \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[\ell(h(\boldsymbol{x}), y)]$. Since, $\mathcal{L}_\mathcal{D}(h)$ cannot be computed without the knowledge of $\mathcal{D}$, one minimises the *empirical risk* over the training sample S as $\mathcal{L}_S(h) \triangleq \frac{1}{m}\sum_{i=1}^m \ell\left(h\left(\boldsymbol{x}_i\right), y_i\right)$.

Transductive learning. In transductive inference, one restricts the domain to be $\mathcal{X} \triangleq \{\boldsymbol{x}_i\}_{i=1}^n$, a finite set of features $\boldsymbol{x}_i \in \mathbb{R}^d$. Without loss of generality, one may assume that the labels $y_1, \ldots, y_m \in \{\pm 1\}$ are known, and the goal is to predict $y_{m+1}, \ldots y_n$. The problem can be reformulated in the statistical learning framework as follows. We define the feature distribution $\mathcal{D}_\mathcal{X}$ to be uniform over the n features, whereas $\boldsymbol{y}_i \sim \eta(\boldsymbol{x}_i)$ for some unknown distribution η. Hence $\mathcal{D} := \mathrm{Unif}([n]) \times \eta$ is the joint distribution on $\mathcal{X} \times \mathcal{Y}$, and the goal is to find a predictor $h : \mathcal{X} \to \mathcal{Y}$ that minimises the *generalisation error* $\mathcal{L}_u(h) \triangleq \frac{1}{n-m}\sum_{i=m+1}^n \ell\left(h(x_i), y_i\right)$. In addition we define the *empirical error* of h to be $\widehat{\mathcal{L}}_m(h) \triangleq \frac{1}{m}\sum_{i=1}^m \ell\left(h(x_i), y_i\right)$ and the full sample error of h to be $\mathcal{L}_n(h) \triangleq \frac{1}{n}\sum_{i=1}^n \ell\left(h(x_i), y_i\right)$, which is defined over both labelled and unlabelled instances. The purpose of this paper is to derive generalisation error bound for graph based transduction of the form

$$\mathcal{L}_u(h) \leq \widehat{\mathcal{L}}_m(h) + \text{complexity term.}$$

The complexity term is typically characterised using learning-theoretic terms such as VC Dimension and Rademacher complexity. For the transductive setting see Tolstikhin et al. (2016), El-Yaniv et al. (2009), and Tolstikhin et al. (2014).

Graph-based transductive learning. A typical view of graph information in transductive inference is as a form of a regularisation (Belkin et al. 2004). In contrast, we view the graph as part of the hypothesis class and derive the impact of the graph information on the complexity term. We assume access to a graph $\mathcal{G}$ with n vertices, corresponding to the respective feature vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, and edge (i, j) denoting similarity of vertices i and j. For ease of exposition, we define the matrix $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ with rows being the n feature vectors of dimension d. We also abuse notation to write a predictor as $h : \mathbb{R}^{n \times d} \to \{\pm 1\}^n$. Furthermore, typically neural networks output a soft predictor in $\mathbb{R}$, that is further transformed into labels through sign or softmax functions. Hence, much of our analysis focuses on predictors $h : \mathbb{R}^{n \times d} \to \mathbb{R}^n$, and corresponding hypothesis class

$$\mathcal{H}_{\mathcal{G}} = \left\{ h : \mathbb{R}^{n \times d} \to \mathbb{R}^n \ : \ h \text{ is parametrized by } \mathcal{G} \right\} \subset \mathbb{R}^{[n]}.$$

When applicable, we denote the hypothesis class of binary predictors obtained through sign function as $\operatorname{sign} \circ \mathcal{H}_{\mathcal{G}} = \{\operatorname{sign}(h) \mid h \in \mathcal{H}_{\mathcal{G}}\}$. Note that $\operatorname{sign} \circ \mathcal{H}_{\mathcal{G}} \subset \mathcal{H}_{\mathcal{G}}$, and hence, VC Dimension or Rademacher complexity bounds for the latter also hold for the hypothesis class of binary predictors. We also note that the presented analysis holds for both sign and sigmoid function for binarisation.

Formal setup of GNNs. We next characterise the hypothesis class for graph neural networks. Consider graph-based neural network model with the propagation rule for layer k denoted by $g_k(\boldsymbol{H}) : \mathbb{R}^{d_{k-1}} \to \mathbb{R}^{d_k}$ with layer wise input matrix $\boldsymbol{H} \in \mathbb{R}^{n \times d_{k-1}}$. Consider a class of GNNs defined over K layers, with dimension of layer $k \in [K]$ being d_k and $\boldsymbol{S} \in \mathbb{R}^{n \times n}$ the graph diffusion operator. Let ϕ denote the point-wise activation function of the network, which we assume to be a Lipschitz function with Lipschitz constant L_ϕ. We assume ϕ to be the same throughout the network. We define the hypothesis class over all K-layer GNNs as:

$$\mathcal{H}^{\phi}_{\mathcal{G}} \triangleq \left\{ h^{\phi}_{\mathcal{G}}(\boldsymbol{X}) = g_K \circ \cdots \circ g_0 \ : \ \mathbb{R}^{n \times d} \to \{\pm 1\}^n \right\} \tag{1}$$

$$\text{with} \quad g_k \triangleq \phi\left(\boldsymbol{b}_k + \boldsymbol{S} g_{k-1}\left(\boldsymbol{H}\right) \boldsymbol{W}_k\right), \ k \in [K], \quad g_0 \triangleq \boldsymbol{X}. \tag{2}$$

where (2) defines the layer wise transformation with $\boldsymbol{W}_k \in \mathbb{R}^{d_{k-1} \times d_k}$ as the trainable weight matrix and $\boldsymbol{b}_k \in \mathbb{R}^{d_k}$ the bias term. Here, the graph is treated as part of the hypothesis class, as indicated by the subscript in $\mathcal{H}^{\phi}_{\mathcal{G}}$. For ease of notation we drop the superscript for non-linearity where it is unambiguous. For the diffusion operator $\boldsymbol{S}$, we consider two main formulations during discussions: $\boldsymbol{S}_{\text{loop}} \triangleq \boldsymbol{A} + \mathbb{I}$ (self loop) and $\boldsymbol{S}_{\text{nor}} \triangleq (\boldsymbol{D} + \mathbb{I})^{-\frac{1}{2}}(\boldsymbol{A} + \mathbb{I})(\boldsymbol{D} + \mathbb{I})^{-\frac{1}{2}}$ (degree normalized (Kipf et al. 2017)) where $\boldsymbol{A}$ denotes the graph adjacency matrix and $\boldsymbol{D}$ is the degree matrix. However, most results are stated for general $\boldsymbol{S}$.

2.2 Generalisation Error-bound using VC Dimension

The main focus of this paper is the notion of generalisation, that is, understanding how well a GNN can predict the classes of an unlabelled set given the training data. We start with one of the most fundamental learning-theoretical concepts in this context which is the Vapnik–Chervonenkis (VC) dimension of a hypothesis class, a measure of the complexity or expressive power of a space of functions learned by a binary classification algorithm. The following result bounds the VC Dimension for the hypothesis class $\mathcal{H}_{\mathcal{G}}^{\phi}$, and use it to derive a generalisation error bound with respect to the full sample error $\mathcal{L}_n$, which is close to the generalisation error for unlabelled examples $\mathcal{L}_u$ when $m \ll n$.

Proposition 1 (Generalisation error bound for GNNs using VC Dimension) *For the hypothesis class over all* ***linear GNNs****, that is $\phi(x) := x$, with binary outputs, the VC Dimension is given by* $\mathrm{VCdim}\big(\operatorname{sign} \circ \mathcal{H}_{\mathcal{G}}^{linear}\big) = \min\big\{d, \mathrm{rank}\big(\boldsymbol{S}\big), \min_{k\in[K-1]}\{d_k\}\big\}$. *Similarly, the VC Dimension for the hypothesis class of GNNs with* ***ReLU non-linearities*** *and binary outputs, can be bounded as* $\mathrm{VCdim}\big(\operatorname{sign} \circ \mathcal{H}_{\mathcal{G}}^{\mathrm{ReLU}}\big) \leq \min\{\mathrm{rank}(\boldsymbol{S}), d_{K-1}\}$.

Using the above bounds, it follows that, for any $\delta \in (0,1)$, the generalisation error for any $h \in \operatorname{sign} \circ \mathcal{H}_{\mathcal{G}}$ satisfies, with probability $1-\delta$,

$$\mathcal{L}_n(h) - \widehat{\mathcal{L}}_m(h) \leq \sqrt{\frac{8}{m}\left(\min\{\mathrm{rank}(\boldsymbol{S}), d_{K-1}\} \cdot \ln(em) + \ln\left(\frac{4}{\delta}\right)\right)}. \tag{3}$$

To interpret Proposition 1, we note that, by introducing the non-linearity, we lose the information about the hidden layers, except the last one and therefore also on the feature dimension. Nevertheless, the information on the graph information (that we are primarily interested in) is preserved. There are two situations that arise. If $d_{K-1} \leq \mathrm{rank}(\boldsymbol{S})$, then, from Proposition 1, the graph information is redundant and one could essentially train a fully connected network without diffusion on the labelled features, and use it to predict on unlabelled features. The graph information has an influence for $\mathrm{rank}(\boldsymbol{S}) < d_{K-1}$. While general statements on the influence of the graph information are difficult, by considering specific assumptions on the graph we can characterise the generalisation error further.

For linear GNN on graph $\mathcal{G}$, one can bound the VC Dimension between those for empty and complete graphs, that is, $\mathrm{VCdim}\big(\operatorname{sign} \circ \mathcal{H}_{\mathrm{complete}}^{\mathrm{linear}}\big) \leq \mathrm{VCdim}\big(\operatorname{sign} \circ \mathcal{H}_{\mathcal{G}}^{\mathrm{linear}}\big) \leq \mathrm{VCdim}\big(\operatorname{sign} \circ \mathcal{H}_{\mathrm{empty}}^{\mathrm{linear}}\big)$. Moreover, for disconnected graphs, $\mathrm{rank}(\boldsymbol{S})$ is related to the number of connected components. Similar observations hold for upper bounds on VC Dimension for ReLU GNNs. Based on this observation for simple settings, it holds that considering graph information in comparison to a fully connected feed forward neural network leads to a decrease in the complexity of the class, and therefore also in the generalisation error. However, the graph $\mathcal{G}$ is connected in most practical scenarios, and even under strong assumptions on the graph, for example under consideration of Erdös-Rényi graphs or stochastic block models, $\mathrm{rank}(\boldsymbol{S}) = O(n)$ (Costello et al. 2008). Therefore, for the case

$d_{K-1} > \operatorname{rank}(\boldsymbol{S}) = O(n)$, Proposition 1 provides a generalisation error bound of $O\left(\sqrt{\frac{n \cdot \ln m}{m}}\right)$, which holds trivially for 0-1 loss as $n > m$. Furthermore, $\operatorname{rank}(\boldsymbol{S})$ is often similar for both self-loop $\boldsymbol{S}_{\text{loop}}$ and degree-normalised diffusion $\boldsymbol{S}_{\text{nor}}$, and hence, the VC Dimension based error bound does not reflect the positive influence of degree normalisation—a fact that can be explained through stability based analysis (Verma et al. 2019).

2.3 Generalisation Error-bound using Transductive Rademacher Complexity

Due to the triviality of VC Dimension based error bounds, we consider generalization error bounds based on transductive Rademacher complexity (TRC). We start by defining TRC that differs from inductive Rademacher complexity by taking the unobserved instances into consideration.

Definition 1 (Transductive Rademacher complexity (El-Yaniv et al. 2009)) *Let $\mathcal{V} \subseteq \mathbb{R}^n$, $p \in [0, 0.5]$ and m the number of labeled points. Let $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_n)^T$ be a vector of independent and identically distributed random variables, where σ_i takes value $+1$ or -1, each with probability p, and 0 with probability $1-2p$. The transductive Rademacher complexity (TRC) of $\mathcal{V}$ is defined as $\mathfrak{R}_{m,n}(\mathcal{V}) \triangleq \left(\frac{1}{m} + \frac{1}{n-m}\right) \cdot \mathop{\mathbb{E}}_{\sigma}\left[\sup_{\mathbf{v} \in \mathcal{V}} \boldsymbol{\sigma}^\top \mathbf{v}\right]$.*

The following result derives a bound for the TRC of GNNs, defined in (1)–(2), and states the corresponding generalization error bound. The bound involves standard matrix norms, such as $\|\cdot\|_\infty$ (maximum absolute row sum) and the 'entrywise' norm, $\|\cdot\|_{2\to\infty}$ (maximum 2-norm of any column).

Theorem 1 (Generalization error bound for GNNs using TRC) *Consider $\mathcal{H}_\mathcal{G}^{\phi,\beta,\omega} \subseteq \mathcal{H}_\mathcal{G}^{\phi}$ such that the trainable parameters satisfy $\|\boldsymbol{b}_k\|_1 \leq \beta$ and $\|\boldsymbol{W}_k\|_\infty \leq \omega$ for every $k \in [K]$. The transductive Rademacher complexity (TRC), $\mathfrak{R}_{m,n}(\mathcal{H}_\mathcal{G}^{\phi,\beta,\omega})$, of the restricted hypothesis class is bounded as*

$$\frac{c_1 n^2}{m(n-m)} \left(\sum_{k=0}^{K-1} c_2^k \|\boldsymbol{S}\|_\infty^k\right) + c_3 c_2^K \|\boldsymbol{S}\|_\infty^K \|\boldsymbol{S}\boldsymbol{X}\|_{2\to\infty} \sqrt{\log(n)}\,, \tag{4}$$

where $c_1 \triangleq 2L_\phi\beta$, $c_2 \triangleq 2L_\phi\omega$, $c_3 \triangleq L_\phi\omega\sqrt{2/d}$ and L_ϕ is Lipschitz constant for activation ϕ.

The bound on TRC leads to a generalisation error bound following El-Yaniv et al. (2009). For any $\delta \in (0,1)$, the generalisation error, $\mathcal{L}_u(h) - \widehat{\mathcal{L}}_m(h)$, for any $h \in \mathcal{H}_\mathcal{G}^{\phi,\beta,\omega}$ satisfies

$$\mathfrak{R}_{m,n}(\mathcal{H}_\mathcal{G}^{\phi,\beta,\omega}) + c_4 \frac{n\sqrt{\min\{m, n-m\}}}{m(n-m)} + c_5 \sqrt{\frac{n}{m(n-m)} \ln\left(\frac{1}{\delta}\right)} \tag{5}$$

with probability $1-\delta$, where c_4, c_5 are absolute constants such that $c_4 < 5.05$ and $c_5 < 0.8$.

The additional terms in (5) are $O\left(\max\left\{\frac{1}{\sqrt{m}}, \frac{1}{\sqrt{n-m}}\right\}\right)$, and hence, we may focus on the upper bound on TRC (4) to understand the influence of the graph diffusion $\boldsymbol{S}$ as well as its interaction with the feature matrix $\boldsymbol{X}$. The bound depends on the choice of ω, and it suggests a natural choice of $\omega = O(1/\|\boldsymbol{S}\|_\infty)$ such that the bound does not grow exponentially with network depth. The subsequent discussions focus on the dependence on $\|\boldsymbol{S}\|_\infty$ and $\|\boldsymbol{SX}\|_{2\to\infty}$, ignoring the role of ω. Few observations are evident from (4), which are also interesting in comparison to existing works.

Role of normalisation. In the case of self-loop, it is easy to see that $\|\boldsymbol{S}_{\text{loop}}\|_\infty = 1 + d_{\max}$, where $d_{\max}$ denotes the maximum degree, and hence, for fixed ω, the bound grows as $O(d_{\max}^K)$. In contrast, for degree normalisation, $\|\boldsymbol{S}_{\text{nor}}\|_\infty = O\left(\sqrt{\frac{d_{\max}}{d_{\min}}}\right)$, and hence, the growth is much smaller (in fact, $\|\boldsymbol{S}_{\text{nor}}\|_\infty = 1$ on regular graphs). It is worth noting that, in the supervised setting, Liao et al. (2021) derived PAC-Bayes for GNN with diffusion $\boldsymbol{S}_{\text{nor}}$, where the bound varies as $O(d_{\max}^K)$. Theorem 1 is tighter in the sense that, for $\boldsymbol{S}_{\text{nor}}$, the error bound has weaker dependence on $d_{\max}$, mainly through $\|\boldsymbol{SX}\|_{2\to\infty}$.

From spectral radius to $\|\boldsymbol{SX}\|_{2\to\infty}$. Previous analyses of GNNs in transductive setting rely on the spectral properties of $\boldsymbol{S}$. For instance, the stability based generalisation error bound for 1-layer GNN in Verma et al. (2019) is $O(\|\boldsymbol{S}\|_2^2)$, where $\|\boldsymbol{S}\|_2$ is the spectral norm. In contrast, Theorem 1 shows TRC $= O(\|\boldsymbol{S}\|_\infty \|\boldsymbol{SX}\|_{2\to\infty})$. This is the first result that explicitly uses the relation between the graph-information and the feature information explicitly via $\|\boldsymbol{SX}\|_{2\to\infty}$. One may note that without node features, that is $\boldsymbol{X} = \mathbb{I}$, we have $\|\boldsymbol{S}\|_{2\to\infty} \leq \|\boldsymbol{S}\|_2 \leq \|\boldsymbol{S}\|_\infty$ and hence, a direct comparison between (5) and $O(\|\boldsymbol{S}\|_2^2)$ bound of Verma et al. (2019) is inconclusive. However, in presence of features $\boldsymbol{X}$, Theorem 1 shows that the bound depends on the alignment between the feature and graph information.

In the presence of graph information we can still express Theorem 1 in terms of spectral components by considering $\|\boldsymbol{SX}\|_{2\to\infty} = \max_j \|(\boldsymbol{SX})_{\cdot j}\|_2 \leq \max_j \|\boldsymbol{S}\|_2 \|\boldsymbol{X}_{\cdot j}\|_2 \leq \|\boldsymbol{S}\|_2 \|\boldsymbol{X}\|_{2\to\infty}$ and $\|\boldsymbol{SX}\|_{2\to\infty}$ which can be bound as $\frac{1}{\sqrt{n}}\|\boldsymbol{S}\|_\infty \leq \|\boldsymbol{S}\|_2$.

Oversmoothing. While the above bound provides a weaker result than (4) it allows to directly connect to the oversmoothing (Li et al. 2018) effect as the diffusion operator in now only included as $\|\boldsymbol{S}\|_2^k,\ k \in [K]$. Therefore with an increasing number of layers (and especially in the setting considered in Oono et al. (2020a) where the number of layers goes to infinity), the information provided by the graph gets oversmoothed and therefore, a loss of information can be observed.

3 Generalization using TRC under Planted Models

The discussion in previous section shows that TRC based generalisation error bound provides some insights into the behaviour of GNNs (example, $\boldsymbol{S}_{\text{nor}}$ is preferred over $\boldsymbol{S}_{\text{loop}}$), but the bound is too general to give insights into the influence of the graph information on the generalisation error. The key quantity of interest is $\|\boldsymbol{SX}\|_{2\to\infty}$, which characterises how the graph and feature information interact. To understand this interaction, we make specific distributional assumptions on both graph and node features. We assume that node features are sampled from a mixture of two d-dimensional isotropic Gaussians (Dasgupta 1999), and graph is independently generated from a two-community stochastic block model (Abbe 2018). Both models have been extensively studied in the context of recovering the latent classes from random observations of features matrix $\boldsymbol{X}$ or adjacency matrix $\boldsymbol{A}$, respectively. Our interest, however, is to quantitatively analyse the influence of graph information when the latent classes in features $\boldsymbol{X}$ and graph $\boldsymbol{A}$ do not align completely. In Section 3.1, we present the model and derive bounds on expected TRC, where the expectation is with respect to random features and graph. We then experimentally illustrate the bounds in Section 3.2, and demonstrate that the corresponding generalisation error bounds indeed capture the trends in performance of GNN.

3.1 Model and Bounds on TRC

We assume that the node features are sampled latent true classes, given a $\boldsymbol{z} = (z_1, \ldots, z_n) \in \{\pm 1\}^n$. The node features are sampled from a Gaussian mixture model (GMM), that is, feature for node-i is sampled as $\boldsymbol{x}_i \sim \mathcal{N}(z_i\boldsymbol{\mu}, \sigma^2\mathbb{I})$ for some $\boldsymbol{\mu} \in \mathbb{R}^d$ and $\sigma \in (0, \infty)$. We express this in terms of $\boldsymbol{X}$ as

$$\boldsymbol{X} = \mathcal{X} + \boldsymbol{\epsilon} \ \in \ \mathbb{R}^{n\times d}, \text{ with } \mathcal{X} = \boldsymbol{z}\boldsymbol{\mu}^\top \ \ \& \ \boldsymbol{\epsilon} = (\epsilon_{ij})_{i\in[n], j\in[d]} \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2). \quad (6)$$

We refer to above as $\boldsymbol{X} \sim$ 2GMM. On the other hand, we assume that graph has two latent communities, characterised by $\boldsymbol{y} \in \{\pm 1\}^n$. The graph is generated from a stochastic block model with two classes (2SBM), where edges (i, j) are added independently with probability $p \in (0, 1]$ if $y_i = y_j$, and with probability $q < [0, p)$ if $y_i \neq y_j$. In other words, we define the random adjacency $\boldsymbol{A} \sim$ 2SBM as a symmetric binary matrix with $\boldsymbol{A}_{ii} = 0$, and $(\boldsymbol{A}_{ij})_{i<j}$ indenpendent such that

$$\boldsymbol{A}_{ij} \sim \text{Bernoulli}(\mathcal{A}_{ij}), \qquad \text{where } \mathcal{A} = \frac{p+q}{2}\boldsymbol{1}\boldsymbol{1}^\top + \frac{p-q}{2}\boldsymbol{y}\boldsymbol{y}^\top - p\mathbb{I}. \quad (7)$$

The choice of two different latent classes $\boldsymbol{z}, \boldsymbol{y} \in \{\pm 1\}^n$ allows study of the case where the graph and feature information of do not align completely. We use $\Gamma = |\boldsymbol{y}^\top \boldsymbol{z}| \in [0, n]$ to quantify this alignment. Assuming $\boldsymbol{y}, \boldsymbol{z}$ are both balanced, that is, $\sum_i y_i = \sum_i z_i = 0$, one can verify that $\|(\mathcal{A} + \mathbb{I})\mathcal{X}\|_{2\to\infty} = \|\boldsymbol{\mu}\|_\infty \left(n(1-p)^2 + \frac{1}{4}n(p-q)^2\Gamma^2 - (p-q)(1-p)\Gamma^2\right)^{1/2}$, which indicates that, for dense graphs ($p, q \gg \frac{1}{n}$), the quantity $\|\boldsymbol{SX}\|_{2\to\infty}$ should typically increase if the latent structure of graph and features are more aligned. This intuition is made precise in the following result that bounds the TRC, in expectation, assuming $\boldsymbol{X} \sim$ 2GMM and $\boldsymbol{A} \sim$ 2SBM.

Theorem 2 (Expected TRC for GNNs under SBM) *Let c_1, c_2 and c_3 as defined in Theorem 1 and $\Gamma \triangleq |\boldsymbol{y}^\top \boldsymbol{z}|$. Let $c_6 \triangleq (1+o(1))$, $c_7 \triangleq (1+ko(1))$, $c_8 \triangleq (1+Ko(1))$. Assuming $p, q \gg \frac{(lnn)^2}{n}$ we can bound the expected TRC for $\boldsymbol{A}$ as defined in (7) and $\boldsymbol{X}$ as defined in (6) as follows:* ***Degree normalized: $\boldsymbol{S} = \boldsymbol{S}_{\boldsymbol{nor}}$***

$$\underset{\substack{\boldsymbol{X}\sim 2\text{GMM} \\ \boldsymbol{A}\sim 2\text{SBM}}}{\mathbb{E}} \left[\mathfrak{R}_{m,n}(\mathcal{H}_{\mathcal{G}}^{\phi,\beta,\omega})\right] \leq \frac{c_1 n^2}{m(n-m)} \left(\sum_{k=0}^{K-1} c_7 c_2^k \left(\frac{p}{q}\right)^{\frac{k}{2}} \right) + c_8 c_3 c_2^K \left(\frac{p}{q}\right)^{\frac{K}{2}} \sqrt{\ln(n)} \times$$
$$\left(c_6 \left\|\boldsymbol{\mu}\right\|_\infty \frac{1+\left(\frac{p-q}{2}\right)^2 \Gamma^2}{\left(\frac{p+q}{2}\right)^2} + c_6 \sqrt{\frac{\ln(n)}{q}} \left\|\boldsymbol{\mu}\right\|_\infty + c_6 \sqrt{\frac{\sigma(1+2\ln(d))}{q}} \right) \tag{8}$$

For space reasons we provide exact formulation of the self loop: $\boldsymbol{S} = \boldsymbol{S}_{loop}$ case in the appendix.

We note that although the above bounds are stated in expectation, they can be translated into high probability bounds. Furthermore the non-triviality of the proof of Theorem 2 stems from bounds on the expectations of matrix norms, which is more complex than the computation above on $\|(\mathcal{A}+\mathbb{I})\mathcal{X}\|_{2\to\infty}$. Theorem 2 can be also translated into bounds on the generalisation gap $\mathcal{L}_u(h) - \widehat{\mathcal{L}}_m(h)$. By considering a planted model we can now further extend the observations of Section 2.2 and 2.3.

Role of normalisation. In the following, we can show that by normalising, the generalisation gap grows slower with increasing graph size. First we compare $\mathbb{E}\left[\|\boldsymbol{S}_{\text{loop}}\|_\infty^k\right] = c_7 (np)^k$ with $\mathbb{E}\left[\|\boldsymbol{S}_{\text{nor}}\|_\infty^k\right] = c_7 \left(p/q\right)^{k/2}$ and observe that by normalising we lose the n term. In addition we can consider $\mathbb{E}\left[\|\boldsymbol{S}\boldsymbol{X}\|_{2\to\infty}\right]$ which is bound by the second line in (8). Again in the first, deterministic, term we observe that the self loop version contains an additional dependency on n. For the two noise terms we can characterize the behaviour in terms of the density of the graph. Let $\rho = O(p), O(q)$ and $\rho \gg \frac{1}{n}$ then we can characterise the *dense setting* as $\rho \asymp \Omega(1)$ and the *sparse setting* as $\rho \asymp O\left(\frac{\ln(n)}{n}\right)$ and observe that in both case the normalised case grows slower with n:

$$\text{Dense: } \mathbb{E}\left[\|\boldsymbol{S}_{\text{loop}}\boldsymbol{X}\|_{2\to\infty}\right] = O(n) \;\&\; \mathbb{E}\left[\|\boldsymbol{S}_{\text{nor}}\boldsymbol{X}\|_{2\to\infty}\right] = O(\sqrt{\ln(n)}) \tag{9}$$

$$\text{Sparse: } \mathbb{E}\left[\|\boldsymbol{S}_{\text{loop}}\boldsymbol{X}\|_{2\to\infty}\right] = O(\sqrt{n\ln(n)}) \;\&\; \mathbb{E}\left[\|\boldsymbol{S}_{\text{nor}}\boldsymbol{X}\|_{2\to\infty}\right] = O(\sqrt{n}) \tag{10}$$

Influence of the graph information. We consider the idea from Section 2.2, to analyse the influence of graph information by comparing the TRC between the case where no graph information is considered, $\boldsymbol{S} = \mathbb{I}$ and $\boldsymbol{S}_{\text{nor}}$. We define the corresponding hypothesis classes as $\mathcal{H}_{\mathbb{I}}^{\phi,\beta,\omega}$ and $\mathcal{H}_{\text{nor}}^{\phi,\beta,\omega}$. Considering the deterministic case ($\boldsymbol{S} = \mathcal{S}, \boldsymbol{X} = \mathcal{X}$) we can observe $\mathfrak{R}_{m,n}(\mathcal{H}_{\mathbb{I}}^{\phi,\beta,\omega}) > \mathfrak{R}_{m,n}(\mathcal{H}_{\text{nor}}^{\phi,\beta,\omega})$ if $\Gamma > O\left(\frac{n}{\sqrt{n\rho+n}}\right)$. Therefore the random graph setting allows us to more precisely characterize under what conditions adding graph information helps.

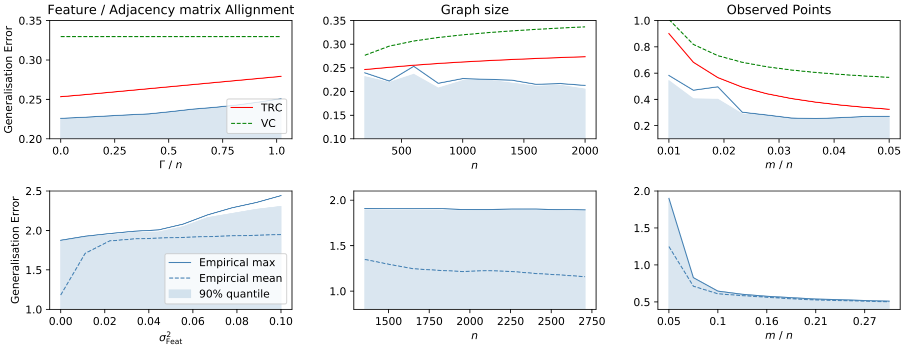

Fig. 1. *Top row* shows experiments for SBM and *bottom row* for Cora. *(left)* Change in the alignment of the features and adjacency matrix. *(middle)* Change of the graph size n. *(right)* Change number of observed points m.

3.2 Experimental Results

While we focus on the theoretical analysis of GNNs, in this section we illustrate that the empirical generalization error follows the trends given by the bounds described in Theorem 2. The bounds in Section 3.1 are derived for binary SBMs so we therefore focus on this setting but in addition also show that those observations extend to real world, multi-class data on the example of the Cora dataset (Rossi et al. 2015). The results are presented in Figure 1. For the SBM we consider a graph with $n = 500, m = 100$ as default. We plot the mean over 5 random initialisation and over several epochs. Note that the range for Cora exceeds $(0, 1)$ as the dataset is multi class and we consider a negative log likelihood loss. For plotting the theoretical bound we can only plot the trend of the bound as the absolute value is out of the $(0, 1)$ range. While this does not allow us to numerically show how tight the bound is in practice, we can still make statements about the influence of the change of parameters, where the experiments validate the constancy between theory and empirical observations [5].

We can first look at the *feature and graph alignment* as characterised through Γ^2 in the TRC based bound (8) and observe that with an increase in the latent structure the generalisation error increases. While this seems to be counterintuitive a possible explanation could be that reduced alignment helps to prevent overfitting and we observe that the slope matches the empirical results. In ad-

[5] Generalisation error bounds, even for simple machine learning models, can exceed 1 due to absolute constants that cannot be precisely estimated. Hence, the point of interest is the dependence of key parameters; for instance, in a supervised setting, the bounds are $O(1/\sqrt{m})$ and typically exceeds 1 for moderate m. This problem is inherent to the bound given in El-Yaniv et al. (2009) that we base our TRC bounds on, as the slack terms can already exceeds 1 and therefore further research on general TRC generalisation gaps is necessary to characterise the absolute gap between theory and experiments.

dition we note that the VC dimension bound (3) does not allow us to model this dependency. For Cora we do not have access to the ground truth for the alignment and therefore can not verify this trend directly. Therefore we simulate a change in the feature structure by adding noise to the feature vector as $\boldsymbol{X} + \epsilon$ where $\epsilon_{i\cdot}$ is *i.i.d.* distributed $\mathcal{N}(0, \sigma^2_{\text{Feat}}\mathbb{I})$ and again observe a similar behaviour to the SBM. To be able to apply the bound to arbitrary graphs an important property is that the bound does not increase drastically with growing *graph size*. We theoretically showed this in the previous section, especially through (9)–(10) and illustrate it in Figure 1 (middle). Empirically for both, SBM and Cora, the generalisation error stays mostly consistent over varying n. Finally for the *number of observed points* we consider a realistic setting of $m \ll n - m$ where we see a sharp decline in the setting of few observed points but then the generalisation error converges which corresponds to the influence of m as described in (8). Practically such an observation can be useful as labeling data can be expensive and such results could be useful to determine a necessary and sufficient number of labeled data to obtain a given level of accuracy.

4 Conclusion

Statistical learning theory has proven to be a successful tool for a complete and rigours analysis of learning algorithms. At the same time research suggests that applied to deep learning models these methods become non-informative. However on the example of GNNs, we demonstrate that classical statistical learning theory can be used under consideration of the right complexity measure and distributional assumptions on the data to provide insight into trends of deep models. Our analysis provides first fundamental results on the influence of different parameters on generalization and opens up different lines of follow up work. As it is not the focus of this paper we consider the bounds on the norms of trainable parameters, ω, β, fixed. However loosening this assumption would allow us to analyse the behaviour of the generalisation error during training and under different optimization approaches. Finally while our analysis focuses on generalisation we suggest that the idea of analysing GNNs under planted models can be extended to other learning-theoretical measures such as stability or model selection as well as the supervised (graph-classification) setting.

References

Abbe, Emmanuel (2018). “Community Detection and Stochastic Block Models: Recent Developments”. In: *Journal of Machine Learning Research.*

Arora, Sanjeev, Nadav Cohen, Noah Golowich, and Wei Hu (2019a). “A Convergence Analysis of Gradient Descent for Deep Linear Neural Networks”. In: *International Conference on Learning Representations.*

Arora, Sanjeev, Simon S. Du, Wei Hu, Zhiyuan Li, Ruslan Salakhutdinov, and Ruosong Wang (2019b). “On Exact Computation with an Infinitely Wide Neural Net”. In: *Advances in Neural Information Processing Systems.*

Belkin, Mikhail, Irina Matveeva, and Partha Niyogi (2004). "Regularization and semi-supervised learning on large graphs". In: *International Conference on Computational Learning Theory.*

Bojchevski, Aleksandar, Oleksandr Shchur, Daniel Zügner, and Stephan Günnemann (2018). "NetGAN: Generating Graphs via Random Walks". In: *International Conference on Machine Learning.*

Bruna, Joan, Wojciech Zaremba, Arthur Szlam, and Yann Lecun (2014). "Spectral networks and locally connected networks on graphs". In: *International Conference on Learning Representations.*

Burges, Chris J.C. (1998). "A Tutorial on Support Vector Machines for Pattern Recognition". In:

Chen, Ming, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li (2020). "Simple and Deep Graph Convolutional Networks". In: *International Conference on Machine Learning.*

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

Cortes, Corinna and Mehryar Mohri (2007). "On Transductive Regression". In: *Advances in Neural Information Processing Systems.*

Costello, Kevin P. and Van H. Vu (2008). "The rank of random graphs". In: *Random Structures & Algorithms.*

Dai, Hanjun, Bo Dai, and Le Song (2016). "Discriminative Embeddings of Latent Variable Models for Structured Data". In: *International Conference on Machine Learning.*

Dasgupta, Sanjoy (1999). "Learning mixtures of Gaussians". In: *Annual Symposium on Foundations of Computer Science.*

Defferrard, Michaël, Xavier Bresson, and Pierre Vandergheynst (2016). "Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering". In: *International Conference on Neural Information Processing Systems.*

Du, Simon S, Kangcheng Hou, Russ R Salakhutdinov, Barnabas Poczos, Ruosong Wang, and Keyulu Xu (2019). "Graph Neural Tangent Kernel: Fusing Graph Neural Networks with Graph Kernels". In: *Advances in Neural Information Processing Systems.*

El-Yaniv, Ran and Dmitry Pechyony (2009). "Transductive Rademacher Complexity and its Applications". In: *Journal of Artificial Intelligence Research.*

Feldman, Vitaly (2020). "Does Learning Require Memorization? A Short Tale about a Long Tail". In: *Annual ACM SIGACT Symposium on Theory of Computing.*

Garg, Vikas, Stefanie Jegelka, and Tommi Jaakkola (2020). "Generalization and Representational Limits of Graph Neural Networks". In: *International Conference on Machine Learning.*

Ghorbani, Behrooz, Song Mei, Theodor Misiakiewicz, and Andrea Montanari (2020). "When Do Neural Networks Outperform Kernel Methods?" In: *Advances in Neural Information Processing Systems.*

Gilmer, Justin, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl (2017). "Neural Message Passing for Quantum Chemistry". In: *International Conference on Machine Learning*.

Gori, Maria Cristina, Gabriele Monfardini, and Franco Scarselli (2005). "A new model for learning in graph domains". In: *IEEE International Joint Conference on Neural Networks*.

Jacot, Arthur, Franck Gabriel, and Clement Hongler (2018). "Neural tangent kernel: Convergence and generalization in neural networks". In: *Advances in neural information processing systems*.

Kawamoto, Tatsuro, Masashi Tsubaki, and Tomoyuki Obuchi (2019). "Mean-field theory of graph neural networks in graph partitioning". In: *Journal of Statistical Mechanics: Theory and Experiment*.

Keriven, Nicolas, Alberto Bietti, and Samuel Vaiter (2020). "Convergence and Stability of Graph Convolutional Networks on Large Random Graphs". In: *Advances in Neural Information Processing Systems*.

Kingma, Diederik P. and Jimmy Ba (2015). "Adam: A Method for Stochastic Optimization". In: *International Conference on Learning Representations*.

Kipf, Thomas N. and Max Welling (2017). "Semi-Supervised Classification with Graph Convolutional Networks". In: *International Conference on Learning Representations*.

Li, Qimai, Zhichao Han, and Xiao ming Wu (2018). *Deeper Insights Into Graph Convolutional Networks for Semi-Supervised Learning*.

Liao, Renjie, Raquel Urtasun, and Richard Zemel (2021). "A PAC-Bayesian Approach to Generalization Bounds for Graph Neural Networks". In: *International Conference on Learning Representations*.

Ma, Siyuan and Mikhail Belkin (2017). "Diving into the shallows: a computational perspective on large-scale shallow learning". In: *Advances in Neural Information Processing Systems*.

Nagarajan, Vaishnavh and J. Zico Kolter (2019). "Uniform convergence may be unable to explain generalization in deep learning". In: *Advances in Neural Information Processing Systems*.

Neyshabur, Behnam, Srinadh Bhojanapalli, David Mcallester, and Nati Srebro (2017). "Exploring Generalization in Deep Learning". In: *Advances in Neural Information Processing Systems*.

Oono, Kenta and Taiji Suzuki (2020a). "Graph Neural Networks Exponentially Lose Expressive Power for Node Classification". In: *International Conference on Learning Representations*.

— (2020b). "Optimization and Generalization Analysis of Transduction through Gradient Boosting and Application to Multi-scale Graph Neural Networks". In: *Advances in Neural Information Processing Systems*.

Rossi, Ryan A. and Nesreen K. Ahmed (2015). "The Network Data Repository with Interactive Graph Analytics and Visualization". In: *AAAI Conference on Artificial Intelligence*.

Scarselli, Franco, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini (2009). "The Graph Neural Network Model". In: *IEEE Transactions on Neural Networks.*

Scarselli, Franco, Ah Chung Tsoi, and Markus Hagenbuchner (2018). In: *The Vapnik–Chervonenkis dimension of graph and recursive neural networks.*

Shalev-Shwartz, Shai and Shai Ben-David (2014). *Understanding Machine Learning: From Theory to Algorithms.* Cambridge University Press.

Theisen, Ryan, Jason M. Klusowski, and Michael W. Mahoney (2020). "Good linear classifiers are abundant in the interpolating regime". In: *Computing Research Repository.*

Tolstikhin, Ilya, Gilles Blanchard, and Marius Kloft (2014). "Localized Complexities for Transductive Learning". In: *Conference on Learning Theory.*

Tolstikhin, Ilya O. and David Lopez-Paz (2016). "Minimax Lower Bounds for Realizable Transductive Classification". In: *ArXiv.* Vol. 1602.03027.

Vapnik, V. N. and A. Ya. Chervonenkis (1971). "On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities". In: *Theory of Probability & Its Applications.*

Vapnik, Vladimir (1982). "Estimation of Dependences Based on Empirical Data". In: *Springer Series in Statistics.*

Vapnik, V.N. (1998). "Statistical Learning Theory". In: *A Wiley-Interscience publication.*

Veličković, Petar, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio (2018). "Graph Attention Networks". In: *International Conference on Learning Representations.*

Verma, Saurabh and Zhi-Li Zhang (2019). "Stability and Generalization of Graph Convolutional Neural Networks". In: *Computing Research Repository.*

Wu, Zonghan, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S Yu (2020). "A Comprehensive Survey on Graph Neural Networks". In: *IEEE transactions on neural networks and learning systems.*

Xu, Keyulu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka (2019). "How Powerful are Graph Neural Networks?" In: *International Conference on Learning Representations.*

Ying, Rex, Jiaxuan You, Christopher Morris, Xiang Ren, William L. Hamilton, and Jure Leskovec (2018). "Hierarchical Graph Representation Learning with Differentiable Pooling". In: *Advances in Neural Information Processing Systems.*

Zhang, Muhan, Zhicheng Cui, Marion Neumann, and Yixin Chen (2018). "An End-to-End Deep Learning Architecture for Graph Classification". In: *AAAI Conference on Artificial Intelligence.*

Zhou, Pengfei, Tianyi Li, and Pan Zhang (2020). "Phase transitions and optimal algorithms for semisupervised classifications on graphs: From belief propagation to graph convolution network". In: *Physical Review Research.*

Appendix

In the appendix we provide the following additional information and proofs.
A: Notation
B: Influence of Depth and Residual Connections on the Generalisation Error
C: Proof Proposition 1 — Generalisation error bound for GNNs using VC-Dimension
D: Proof Theorem 1 — Generalization error bound for GNNs using TRC
E: Proof Theorem 3 — TRC for Residual GNNs
F: Proof Theorem 2 — Expected TRC for GNNs under SBM
G: Experimental Details

A Notation

Let $[n] := 1, 2, \ldots n$. We represent a graph $\mathcal{G}$ by its adjacency matrix $\boldsymbol{A}$, and use $\mathbb{I}$ to denote an identity matrix. For any vertex i, $i \sim j := \{j \mid \boldsymbol{A}_{ij} = 1\}$ is the set indices adjacent to i. We use $\| \cdot \|_p$ to denote the p-norm for vectors and induced p-norm for matrices. We consider standard matrix norms, such as $\| \cdot \|_\infty$ (maximum absolute row sum) and the 'entrywise' norm, $\| \cdot \|_{2\to\infty}$ (maximum 2-norm of any column). Function classes are denoted as $\mathcal{H}$ or $\mathcal{F}$, indexed depending on parameters that are included in the hypothesis class. We define the fully connected graph as $\mathcal{G} =: K_{\mathcal{G}}$, the empty graph (without any edges) as $\mathcal{G} =: \emptyset$. Note that if we consider a graph with only self loops ($\mathcal{G} := \emptyset$) the GNN becomes equivalent to a fully connected neural network. We consider point wise activation functions $\phi(\cdot) : \mathbb{R} \to \mathbb{R}$. In this context we define the rectified linear unit as $\mathrm{ReLU}(x) := \max\{0, x\}$.

B Influence of Depth and Residual Connections on the Generalisation Error

While for standard neural networks increasing the depth is a common approach for increasing the performance, this idea becomes more complex in the context of GNNs as each layer contains a left multiplication of the diffusion operator and we can therefore observe an over-smoothing effect (Li et al. 2018) — the repeated multiplication of the diffusion operator in each layer spreads the feature information such that it converges to be constant over all nodes. To overcome this problem, empirical works suggest the use of residual connections (Kipf et al. 2017; Chen et al. 2020), such that by adding connections from previous layers the network retains some feature information. In this section we investigate this approach in the TRC setting. In Section B.1 we provide the TRC bound for GNN with skip connections and show that it improves the generalisation error compared to vanilla GNNs. In Section B.2 we illustrate this bounds empirically.

B.1 Model and bounds on TRC for GNN with Residual connections

While there is a wide range of residual connections, introduced in recent years we follow the idea presented in Chen et al. (2020) where a GNN as defined in (2) is extended by an interpolation over parameter α with the features. This setup is especially interesting as it captures the idea of preserving the influence of the feature information more than residual definition that only connect to the previous layer. Formally we can now write the layer wise propagation rule as

$$g_{k+1} \triangleq \phi\left((1-\alpha)\left(\boldsymbol{b}_k + \boldsymbol{S} g_k\left(\boldsymbol{H}\right)\boldsymbol{W}_k\right) + \alpha g_0\left(\boldsymbol{H}\right)\right), \qquad \text{with } \alpha \in (0,1). \tag{11}$$

We can now derive a generalization error bound similar to Theorem 1 for the Residual network.

Theorem 3 (TRC for Residual GNNs) *Consider a Residual network as defined in* (11) *and* $\mathcal{H}_{\mathcal{G}}^{\phi,\beta,\omega} \subset \mathcal{H}_{\mathcal{G}}^{\phi}$ *such that the trainable parameters satisfy* $\|\boldsymbol{b}_k\|_1 \leq \beta$ *and* $\|\boldsymbol{W}_k\|_\infty \leq \omega$ *for every* $k \in [K]$. *Then with* $\alpha \in (0,1)$ *and* $c_1 \triangleq 2L_\phi\beta$, $c_2 \triangleq 2L_\phi\omega$, $c_3 \triangleq L_\phi\omega\sqrt{2/d}$ *the TRC of the restricted class or Residual GNNs is bounded as*

$$\begin{aligned}\mathfrak{R}_{m,n}(\mathcal{H}_{\mathcal{G}}^{\phi,\beta,\omega}) \ \leq \ & \frac{((1-\alpha)c_1 + \alpha 2L_\phi \|\boldsymbol{X}\|_\infty)n^2}{m(n-m)}\left(\sum_{k=0}^{K-1}(1-\alpha)c_2^k \|\boldsymbol{S}\|_\infty^k\right) \\ & + \alpha 2L_\phi \|\boldsymbol{X}\|_\infty + (1-\alpha)c_3 c_2^K \|\boldsymbol{S}\|_\infty^K \|\boldsymbol{S}\boldsymbol{X}\|_{2\to\infty}\sqrt{\log(n)}\end{aligned} \tag{12}$$

However observing the bound isolated does not provide new insights beyond Theorem 2 into the behaviour of the generalisation error and therefore we focus on the comparison between GNNs with and without residual connections.

For readability assume $\beta = \|\boldsymbol{X}\|_\infty$. Under this setup we can note that the generalisation error-bound increases with decreased alpha and in extension it

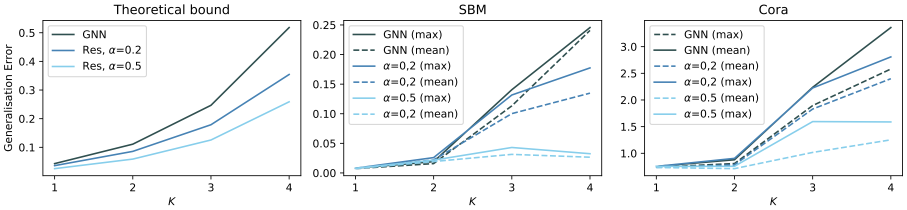

Fig. 2. *(left)* Theoretical bounds corresponding to Theorem 3. *(middle)* Influence of depth K under SBM. *(right)* Influence of depth K for *Cora*.

follows that the generalisation error-bound for a GNN with skip connection is lower then the one without. This implication is in line with the general notion that residual connections improve the performance of networks (Chen et al. 2020; Kipf et al. 2017). Our general intuition behind this behavior is that with increasing α, the network architecture is closer to the one of an one hidden layer network. Having good performance in shallow networks is something that is observed in our experiments as well as in previous work (e.g., (Kipf et al. 2017)). Therefore it appears that using the skip connection to obtain a deep network that resembles a shallow one leads to the performance increase.

B.2 Experiments on depth and Residual networks

The above observation suggests that including residual connections is beneficial with increasing depth which is consistent with the initial reason of introducing residual connections (Chen et al. 2020; Kipf et al. 2017). We further illustrate this in the context of the trend in (12). Similar to Section 3.2 we start by considering the vanilla GNN version and focus on the *influence of depth* where Figure 2 (left) illustrates Theorem 2, more specifically an exponential increase of K as shown in (8)–(**??**) (similar to Liao et al. (2021)). Empirically from Figure 2, (middle, right) we note that with increasing depth the generalisation error indeed increases for the first three layers significantly but then we observe a deviation from the theoretical bound. The rate of growth decreases, which is to be expected as the absolute values of $\mathcal{L}_u, \mathcal{L}_m$ are bound by construction. Future work with a focus on depth is necessary to refine this component of the bound. Extending the analysis of depth we now consider the *residual connections* as defined in (11). By (12) we can still observe the exponential dependency on K and therefore focus on two main aspects: i) Theoretically the generalisation error for the Resnet is upper bound by GNN, which empirically is observed for both the SBM as well as for Cora. ii) Focusing on the Resnets, Theorem 3 predicts an ordering in the generalisation error given by α which is again observed for both the SBM as well as for Cora. Therefore while there seems to be deviation in the exponential behaviour of K as given in Theorem 3, the ordering of the generalisation error-bound described by α is observed empirically. While this does not give us a complete picture we can note that the remarks on oversmoothing suggest that shallower networks are preferable and we again note that the VC dimension bound (3) does not provide any useful insights to the influence of depth.

C Proof Proposition 1 — Generalisation error bound for GNNs using VC-Dimension

Definition 2 (VC-Dimension) *Following Vapnik et al. (1971). Let* $\mathcal{H} \subseteq \{\pm 1\}^{\mathcal{X}}$ *be a binary function class and* $h \in \mathcal{H}$ *a function in this class. We define* $C = (x_1, \cdots x_m) \in \mathcal{X}^m$ *and say that* C *is shattered by* h *if for all assignments of labels to points in* C *there exists a parameterization of* h *such that* h *predicts all points in* C *without error. From there we define the* ***VC-dimension*** *of a non-empty hypothesis class* $\mathcal{H}$ *as the cardinality of the largest possible subset of* $\mathcal{X}$ *that can be shattered by* $\mathcal{H}$*. If* $\mathcal{H}$ *can shatter arbitrarily large sets, then* $\mathrm{VCdim}(\mathcal{H}) = \infty$.

C.1 Generalization using VC-Dimension under specific graph assumptions

For the hypothesis class over all **linear GNNs**, that is $\phi(x) := x$, with binary outputs, the VC Dimension is given by

$$\mathrm{VCdim}\big(\operatorname{sign} \circ \mathcal{H}_{\mathcal{G}}^{\text{linear}}\big) = \min\big\{d, \operatorname{rank}\big(\boldsymbol{S}\big), \min_{k \in [K-1]} \{d_k\}\big\}.$$

Similarly, the VC Dimension for the hypothesis class of GNNs with **ReLU non-linearities** and binary outputs, can be bounded as $\mathrm{VCdim}\big(\operatorname{sign} \circ \mathcal{H}_{\mathcal{G}}^{\text{ReLU}}\big) \leq \min\{\operatorname{rank}(\boldsymbol{S}), d_{K-1}\}$.

Using the above bounds, it follows that, for any $\delta \in (0,1)$, the generalisation error for any $h \in \operatorname{sign} \circ \mathcal{H}_{\mathcal{G}}$ satisfies, with probability $1 - \delta$,

$$\mathcal{L}_n(h) - \widehat{\mathcal{L}}_m(h) \leq \sqrt{\frac{8}{m}\left(\min\{\operatorname{rank}(\boldsymbol{S}), d_{K-1}\} \cdot \ln(em) + \ln\left(\frac{4}{\delta}\right)\right)}.$$

Proof. For this proof we will need the following know result on the VC-dimension of linearly independent points:

Theorem 4 (Burges (1998)) *Consider some set of* m *points in* $\mathbb{R}^n$*. Choose any one of the points as origin. Then the* m *points can be shattered by oriented hyperplanes if and only if the position vectors of the remaining points are linearly independent.*

For deriving $\mathrm{VCdim}\big(\operatorname{sign} \circ \mathcal{H}_{\mathcal{G}}^{\text{linear}}\big)$ we start with the VC-dimension of the final layer: $\mathrm{VCdim}(\mathcal{H}_{\boldsymbol{B}}^{\text{sign}})$ with

$$\mathcal{H}_{\boldsymbol{B}}^{\text{sign}} = \big\{h_{\boldsymbol{B}}^{\text{sign}}(x) := \operatorname{sign}(\boldsymbol{B}\boldsymbol{w}) \;\;:\;\; \boldsymbol{w} \in \mathbb{R}^m\big\}$$

over an arbitrary matrix $\boldsymbol{B} \in \mathbb{R}^{n \times m}$, where $\boldsymbol{B}$ is later substituted for the linear network. Let $\operatorname{rank}(\boldsymbol{B}) = r$ then we show that there is $c \subset [n], |c| = r$ s.t. $\forall\ b \in \{\pm 1\}^r$ and $h_{\boldsymbol{B}}^{\text{sign}}(c) = \{\pm 1\}^c$. Using SVD we decompose $\boldsymbol{B} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{V}^\top$ and define $\boldsymbol{z}_1^\top, \cdots, \boldsymbol{z}_m^\top \in \mathbb{R}^k$ as the rows of $\boldsymbol{U}$. Using this we rewrite:

$$\boldsymbol{B}\boldsymbol{w} = \begin{bmatrix} \boldsymbol{z}_1^\top \\ \vdots \\ \boldsymbol{z}_d^\top \end{bmatrix} \underbrace{\boldsymbol{\Lambda}\boldsymbol{V}^\top\boldsymbol{w}}_{=\boldsymbol{a} \in \mathbb{R}^d} = \begin{bmatrix} \boldsymbol{z}_1^\top \boldsymbol{a} \\ \vdots \\ \boldsymbol{z}_d^\top \boldsymbol{a} \end{bmatrix}$$

Rewrite $\mathcal{H}_{\boldsymbol{B}}^{\text{sign}}$ as $\mathcal{F}^{\text{sign}} = \left\{h_a(z) = \text{sign}(\boldsymbol{a}^\top \boldsymbol{z})\right\}$. Since $\mathcal{F}^{\text{sign}}$ lies in the class of all homogenious linear classifiers in r dimensions and from orthonormal condition on $\boldsymbol{z}$ it follows that $\text{span}\left(\{\boldsymbol{z}_1, \cdots \boldsymbol{z}_n\}\right) = \mathbb{R}^r$. Using this observation as well as results on the VC-dimension of linear independent pointsets Burges (1998) it follows that $\text{VCdim}(\mathcal{H}_{\boldsymbol{B}}^{\text{sign}}) = \text{VCdim}(\mathcal{F}^{\text{sign}}) = r$. Substituting $\boldsymbol{B}$ with the linear network and using that for two matrixes $\boldsymbol{B'}$ and $\boldsymbol{B}$: $\text{rank}(\boldsymbol{B'B}) = \min(\text{rank}(\boldsymbol{B'}), \text{rank}(\boldsymbol{B}))$ gives

$$\text{rank}\big(\boldsymbol{B}\big) := \text{rank}\big(\boldsymbol{S}\boldsymbol{H}^{(p)}\big) = \text{rank}\big(\boldsymbol{S}\cdots\boldsymbol{S}\boldsymbol{X}\boldsymbol{W}^{(1)}\cdots\boldsymbol{W}^{(p-1)}\big)$$

as the final result.

For extending to the non-linear setting we first note that we can not make a general statement on the rank of a matrix after applying a non-linearity. That is for some matrix $\boldsymbol{M}$ and non-linearity $\text{ReLU}(\cdot)$ we have no order relation between $\text{rank}(\boldsymbol{M})$ and $\text{rank}(\text{ReLU}(\boldsymbol{M}))$. This can be checked by a simple counterexample. Therefore the above presented proof does not extend to the hidden layer size but since the last layer is linear the dependency on $\boldsymbol{S}$ persists. We define the hypothesis class over all linear GNNs where all but the last activation function are linear $\phi_k(x) := x \ \forall k \in [K-1]$ and $\phi_p(x) := \text{sign}(x)$ as:

$$\mathcal{H}_{\mathcal{G}}^{\text{sign},\mathbb{I}} = \left\{h_{\mathcal{G}}^{\text{sign},\mathbb{I}}(\boldsymbol{X})\right\}$$

and recall that layer k has dimension d_k. Then the VC-Dimension is given by the minimum of the rank of the adjacency matrix, the dimension of the features and the minimum hidden layer size, that is,

$$\text{VCdim}\left(\mathcal{H}_{\mathcal{G}}^{\text{sign},\mathbb{I}}\right) = \min\left\{d, \text{rank}\big(\boldsymbol{S}\big), \min_{k\in[K-1]}\{d_k\}\right\}. \tag{13}$$

Therefore consider the hypothesis class GNNs with of non-linearities $\phi_k(x) := \text{ReLU}(x) \ \forall k \in [K-1]$ and $\phi_p(x) := \text{sign}(x)$:

$$\mathcal{H}_{\mathcal{G}}^{\text{sign},\text{ReLU}} = \left\{h_{\mathcal{G}}^{\text{sign},\text{ReLU}}(\boldsymbol{X})\right\}$$

and again compute the VC-Dimension, similar to the proof shown above, we can note that we lose information on the hidden layers (and therefore also on d) and the bound becomes

$$\text{VCdim}\left(\mathcal{H}_{\mathcal{G}}^{\text{sign},\text{ReLU}}\right) \le \min\left\{\text{rank}(\boldsymbol{S}), d_{p-1}\right\}, \tag{14}$$

that is, it still depends on the rank of $\boldsymbol{S}$ but only on the last hidden layer dimension.

Following defined we use the a standard result for generalisation e.g. in Shalev-Shwartz et al. (2014). For $\delta \in (0,1)$ any $h \in \mathcal{H}_{\mathcal{G}}$ satisfies

$$\mathcal{L}_n(h) - \widehat{\mathcal{L}}_m(h) \le \sqrt{\frac{8\Big(\text{VCdim}\left(\mathcal{H}_{\mathcal{G}}\right)\ln\left(\frac{em}{\text{VCdim}(\mathcal{H}_{\mathcal{G}})}\right) + \ln\left(\frac{4}{\delta}\right)\Big)}{m}} \tag{15}$$

with probability $1-\delta$.

Applying (13) and (14) to (15) gives the final bound. □

C.2 Additional notes on the remarks related to Proposition 1

Expected Rank of Erdös-Rényi graphs From Costello et al. (2008) we know the following result: Let c be a constant larger then $\frac{1}{2}$, then for any $\frac{c \ln n}{n} \leq p \leq \frac{1}{2}$ for a random $\mathcal{G}$ graph sampled from a Erdös-Rényi graph has $\mathrm{rank}(\boldsymbol{A}) \leq n - i(\mathcal{G})$ with probability $1 - \mathcal{O}\big((\ln \ln n)^{-\frac{1}{4}}\big)$, where $i(\mathcal{G})$ denotes the number of isolated vertices in $\mathcal{G}$.

In the same line we can additionally note that we get similar results (of the form that in expectation $\mathrm{rank}(\boldsymbol{A}) = n$) for more complex models like stochastic block models which we will discuss in further detail later, as for any matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ there are invertible matrices arbitrarily close to $\boldsymbol{A}$, under any norm for the $n \times n$ matrices. Motivated by those first findings we consider a different complexity measure, less reliant on combinatorial arguments, to get more insight into the role of graph information.

D Proof Theorem 1 — Generalization error bound for GNNs using TRC

Recall the definition of TRC as defined in El-Yaniv et al. (2009)[6]: Let $\mathcal{V} \subseteq \mathbb{R}^n$, $p \in [0, 0.5]$ and m the number of labeled points. Let $\boldsymbol{\sigma} = (\sigma_1, \dots, \sigma_n)^T$ be a vector of independent and identically distributed random variables, where σ_i takes value $+1$ or -1, each with probability p, and 0 with probability $1 - 2p$. The transductive Rademacher complexity (TRC) of $\mathcal{V}$ is defined as

$$\mathfrak{R}_{m,n}(\mathcal{V}) \triangleq \left(\frac{1}{m} + \frac{1}{n-m}\right) \cdot \mathop{\mathbb{E}}_{\sigma}\left[\sup_{\mathbf{v} \in \mathcal{V}} \boldsymbol{\sigma}^\top \mathbf{v}\right].$$

For this section we introduce the following notation: $Q \triangleq \left(\frac{1}{m} + \frac{1}{n-m}\right)$, which we later again substitute in the final expression.

To derive the TRC we start with the following propositions describing the recursive TRC for a GNN neuron that is applied $K-1$ times for all but the first layer.

Proposition 2 (Recursive TRC of one GNN neuron) *Consider* $g_{k+1} \triangleq \phi\left(\boldsymbol{b}_k + \boldsymbol{S} g_k\left(\boldsymbol{H}\right) \boldsymbol{W}_k\right)$, $k \in \{1, \cdots, K\}$. *Now we define the function class over one neuron as*

$$\mathcal{H}_{\mathcal{G}}^{\phi} \triangleq \left\{ h_{\mathcal{G}}^{\phi}(\boldsymbol{H}) = \phi\left(\boldsymbol{b}_i + \sum_{l}^{d_k} \boldsymbol{W}_{lj} \sum_{t}^{n} \boldsymbol{S}_{it} g(\boldsymbol{H})_{lj}\right) \;\middle|\; g \in \mathcal{F}, \|\boldsymbol{b}_i\|_1 \leq \beta \right\}$$

where $\mathcal{F}$ *is the class of* $\mathbb{R}^{n \times d_k} \to \mathbb{R}$, *including the zero function. Then with* $\boldsymbol{W}_{\cdot j} \triangleq \left[\boldsymbol{W}_{1j}, \cdots, \boldsymbol{W}_{d_k j}\right]^\top$*:*

$$\mathfrak{R}_{m,n}(\mathcal{H}_{\mathcal{G}}^{\phi}) \leq 2L_\phi\left(\beta Q(n) + \|\boldsymbol{S}\|_\infty \|\boldsymbol{W}_{\cdot j}\|_1 \mathfrak{R}_{m,n}(\mathcal{F})\right)$$

Proof. See section D.2 □

After the recursive application we end up with a formulation of all layers and a dependency on the TRC of the first layer. Therefore we then use the following proposition to finish the proof.

Proposition 3 (Bound on TRC, first layer) *Define the hypothesis class over the function of the first layer* g_0 *as:*

$$\mathcal{H}_{\mathcal{G}}^{\phi} \triangleq \left\{ h_{\mathcal{G}}^{\phi}(\boldsymbol{X}) = \phi\left(\boldsymbol{b} + \boldsymbol{S}\boldsymbol{X}\boldsymbol{W}_1\right) \;\middle|\; \|\boldsymbol{b}\|_1 \leq \beta \right\}$$

then the TRC is give by

$$\mathfrak{R}_{m,n}(\mathcal{H}_{\mathcal{G}}^{\phi}) \leq L_\phi\left(\beta Q(n) 2 + Q\|\boldsymbol{W}_1\|_\infty \|\boldsymbol{S}\boldsymbol{X}\|_{2\to\infty} \sqrt{\frac{2\log(n)}{d}}\right)$$

[6] Note that El-Yaniv et al. (2009) considered TRC in terms of u and m which we change to rewriting $u = n - m$ such that the expression is only in terms of the total number of nodes and the number of marked nodes.

Proof. See section D.3 □

Then by combining the above results we obtain Theorem 1 as follows: Consider $\mathcal{H}_{\mathcal{G}}^{\phi,\beta,\omega} \subseteq \mathcal{H}_{\mathcal{G}}^{\phi}$ such that the trainable parameters satisfy $\|\boldsymbol{b}_k\|_1 \leq \beta$ and $\|\boldsymbol{W}_k\|_\infty \leq \omega$ for every $k \in [K]$. The transductive Randemacher complexity (TRC) of the restricted hypothesis class is bounded as

$$\mathfrak{R}_{m,n}(\mathcal{H}_{\mathcal{G}}^{\phi,\beta,\omega}) \leq \frac{c_1 n^2}{m(n-m)} \left(\sum_{k=0}^{K-1} c_2^k \left\| \boldsymbol{S} \right\|_\infty^k \right) + c_3 c_2^K \left\| \boldsymbol{S} \right\|_\infty^K \left\| \boldsymbol{S}\boldsymbol{X} \right\|_{2\to\infty} \sqrt{\log(n)}\,,$$

where $c_1 \triangleq 2L_\phi\beta$, $c_2 \triangleq 2L_\phi\omega$, $c_3 \triangleq L_\phi\omega\sqrt{2/d}$ and L_ϕ is Lipschitz constant for activation ϕ.

D.1 TRC calculus

In the following we proof some preliminary lemmas for TRC that we will use in the later steps.

Lemma 1 (Scalar multiplication) *Let $A \subseteq \mathbb{R}^n$, a scalar $c \in \mathbb{R}$ and a vector $\boldsymbol{a}_0 \in \mathbb{R}^n$ then*

$$\mathfrak{R}_{m,n}\left(\{c\boldsymbol{a} + \boldsymbol{a}_0 : \boldsymbol{a} \in A\}\right) \leq |c|\mathfrak{R}_{m,n}(A)$$

Proof. Directly by construction. □

Lemma 2 (Addition) *Let $A \subseteq \mathbb{R}^n, B \subseteq \mathbb{R}^n$ then*

$$\mathfrak{R}_{m,n}(A+B) = \mathfrak{R}_{m,n}(A) + \mathfrak{R}_{m,n}(B)$$

Proof. By construction and linearity of expectation. □

Lemma 3 (Convex hull) *Let $A \subseteq \mathbb{R}^n$ and $A' = \left\{\sum_{j=1}^N \alpha_j \boldsymbol{a}^{(j)} \;\middle|\; N \in \mathbb{N},\ \forall j,\ \boldsymbol{a}^{(j)} \in A, \alpha_j \geq 0, \|\alpha\|_1 = 1\right\}$ then*

$$\mathfrak{R}_{m,n}(A) = \mathfrak{R}_{m,n}(A').$$

Proof. The proof follows similar to the one for inductive Rademacher complexity (e.g. Shalev-Shwartz et al. (2014)). We first note that for any vector $\boldsymbol{v}$ the following holds:

$$\sup_{\alpha \geq 0 : \|\alpha\|_1 = 1} \sum_{j=1}^N \alpha_j \boldsymbol{v}_j = \max_j \boldsymbol{v}_j$$

Then:

$$\begin{aligned}
\mathfrak{R}_{m,n}(A') &= Q\mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{\alpha\geq 0:\|\alpha\|_1=1}\sup_{\{\boldsymbol{a}^{(i)}\}_{i=1}^{N}}\sum_{i=1}^{n}\boldsymbol{\sigma}_i\sum_{j=1}^{N}\alpha_j\boldsymbol{a}_i^{(j)}\right]\\
&= Q\mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{\alpha\geq 0:\|\alpha\|_1=1}\sum_{j=1}^{N}\alpha_j\sup_{\boldsymbol{a}^{(j)}}\sum_{i=1}^{n}\boldsymbol{\sigma}_i\boldsymbol{a}_i^{(j)}\right]\\
&= Q\mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{\boldsymbol{a}\in A}\sum_{i=1}^{n}\boldsymbol{\sigma}_i\boldsymbol{a}_i\right]\\
&= \mathfrak{R}_{m,n}(A)
\end{aligned}$$

which concludes the proof. □

Lemma 4 (Contraction El-Yaniv et al. (2009)) *Let $A \subseteq \mathbb{R}^n$ be a set of vectors. Let $f(\,\cdot\,)$ and $g(\,\cdot\,)$ be real-value functions. Let $\boldsymbol{\sigma} = \{\sigma_i\}_{i=1}^n$ be Rademacher variables as defined in Definition 1. If for all $1 \leq i \leq n$ and any $\boldsymbol{a}, \boldsymbol{a}' \in A$, $|f(\boldsymbol{a}_i) - f(\boldsymbol{a}'_i)| \leq |g(\boldsymbol{a}_i) - g(\boldsymbol{a}'_i)|$ then*

$$\mathbb{E}_{\boldsymbol{\sigma}}\left[\sum_{i=1}^{n}\sigma_i f(\boldsymbol{a}_i)\right] \leq \mathbb{E}_{\boldsymbol{\sigma}}\left[\sum_{i=1}^{n}\sigma_i g(\boldsymbol{a}_i)\right]$$

Extending this to Lipschitz continues functions. Let $v(\,\cdot\,)$ be a L_v-Lipschitz continues function such that $|v(f(\boldsymbol{a}_i)) - v(f(\boldsymbol{a}'_i))| \leq \frac{1}{L_v}|f(\boldsymbol{a}_i) - f(\boldsymbol{a}'_i)|$. Now let the corresponding hypothesis classes be $\mathcal{F} \triangleq \{f(\cdot)\}, \mathcal{V} \triangleq \{v(f(\cdot))\}$ then

$$\mathfrak{R}_{m,n}(\mathcal{V}) \leq \frac{1}{L_v}\mathfrak{R}_{m,n}(\mathcal{H}) \tag{16}$$

Lemma 5 (Cardinality of finite sets) *Let $A = \{\boldsymbol{a}_1, \cdots, \boldsymbol{a}_n\}$ be a finite set of vectors in $\mathbb{R}^d$ and let $\overline{\boldsymbol{a}} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{a}_i$ then*

$$\mathfrak{R}_{m,n}(A) \leq \max_{\boldsymbol{a}\in A}\|\boldsymbol{a} - \overline{\boldsymbol{a}}\|_2\sqrt{\frac{2\log(n)}{d}}$$

Proof. The proof follows the general idea of the proof for *Massarts Lemma* (see e.g. Shalev-Shwartz et al. (2014)).

From Lemma 3 wlog. let $\overline{\boldsymbol{a}} = 0$. Let $\lambda > 0$ and $A' = \{\lambda \boldsymbol{a}_1, \cdots, \lambda \boldsymbol{a}_n\}$. Therefore

$$\begin{aligned}
\frac{1}{Q}\mathfrak{R}_{m,n}(A') &= \mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\max_{\boldsymbol{a}\in A'} \langle \boldsymbol{\sigma}, \boldsymbol{a}\rangle\right] \\
&= \mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\log\left(\max_{\boldsymbol{a}\in A'} \exp\left(\langle \boldsymbol{\sigma}, \boldsymbol{a}\rangle\right)\right)\right] \\
&\leq \mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\log\left(\sum_{\boldsymbol{a}\in A'} \exp\left(\langle \boldsymbol{\sigma}, \boldsymbol{a}\rangle\right)\right)\right] && \text{Jensen inequality} \\
&\leq \log\left(\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sum_{\boldsymbol{a}\in A'} \exp\left(\langle \boldsymbol{\sigma}, \boldsymbol{a}\rangle\right)\right]\right) && \sigma_i \text{ is i.i.d.} \\
&= \log\left(\sum_{\boldsymbol{a}\in A'} \prod_{i=1} \mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\left[\exp(\boldsymbol{\sigma}_i \boldsymbol{a}_i)\right]\right)
\end{aligned}$$

Bound $\mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\left[\exp(\boldsymbol{\sigma}_i \boldsymbol{a}_i)\right]$:

$$\begin{aligned}
\mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\left[\exp(\boldsymbol{\sigma}_i \boldsymbol{a}_i)\right] &= p\exp(1\boldsymbol{a}_i) + (1-2p)\exp(0\boldsymbol{a}_i) + p\exp(-1\boldsymbol{a}_i) \\
& && \text{by definition of } \boldsymbol{\sigma}_i \\
&= (1-2p) + p\sum_{i=0}^{\infty}\frac{(-\boldsymbol{a})^i + \boldsymbol{a}^i}{i!} \\
&\leq \frac{1}{2}\sum_{i=0}^{\infty}\frac{(-\boldsymbol{a})^i + a^i}{i!} && \text{as } p \leq \tfrac{1}{2}\text{. Equality for } p = \tfrac{1}{2}. \\
&= \frac{\exp(\boldsymbol{a}_i) + \exp(-\boldsymbol{a}_i)}{2} \\
&\leq \exp\left(\frac{\boldsymbol{a}_i^2}{2}\right)
\end{aligned}$$

Because

$$\frac{\exp(a) + \exp(-a)}{2} = \sum_{n=0}^{\infty}\frac{a^{2n}}{(2n)!} \leq \sum_{2^n n!}^{\infty} = \frac{a^{2n}}{2^n n!}\exp\left(\frac{a^2}{2}\right)$$

and $(2n)! \geq 2^n n!\ \forall n \geq 0$. Going back we now get:

$$\begin{aligned}
\frac{1}{Q}\mathfrak{R}_{m,n}(A') &\leq \log\left(\sum_{\boldsymbol{a}\in A'}\prod_{i=1}\mathop{\mathbb{E}}_{\boldsymbol{\sigma}_i}\left[\exp(\boldsymbol{\sigma}_i\boldsymbol{a}_i)\right]\right)\\
&\leq \log\left(\sum_{\boldsymbol{a}\in A'}\prod_{i=1}\exp\left(\frac{\boldsymbol{a}_i^2}{2}\right)\right)\\
&= \log\left(\sum_{\boldsymbol{a}\in A'}\exp\left(\frac{\|\boldsymbol{a}\|^2}{2}\right)\right)\\
&\leq \log\left(|\boldsymbol{A}'|\max_{\boldsymbol{a}\in A'}\exp\left(\frac{\|\boldsymbol{a}\|^2}{2}\right)\right)\\
&= \log\left(|\boldsymbol{A}'|\right) + \max_{\boldsymbol{a}\in A'}\left(\frac{\|\boldsymbol{a}\|^2}{2}\right)
\end{aligned}$$

By construction $\mathfrak{R}_{m,n}(A) = \frac{1}{\lambda}\mathfrak{R}_{m,n}(A')$ and therefore $\mathfrak{R}_{m,n} \leq \frac{1}{\lambda d}\left(\log(|A|) + \lambda^2 \max_{\boldsymbol{a}\in A'}\left(\frac{\|\boldsymbol{a}\|^2}{2}\right)\right)$. By setting $\lambda = \sqrt{\frac{2\log(|A|)}{\max_{\boldsymbol{a}\in A'}\|\boldsymbol{a}\|^2}}$ and rearranging:

$$\mathfrak{R}_{m,n}(A) \leq \max_{\boldsymbol{a}\in A}\|\boldsymbol{a} - \overline{\boldsymbol{a}}\|_2\sqrt{\frac{2\log(n)}{d}}$$

which concludes the proof. □

D.2 Recursive bound on the TRC of single neurons

We start from the general GNN setup as defined as follows: Consider a class of GNNs defined over K layers, with dimension of layer $k \in [K]$ being d_k and $\boldsymbol{S} \in \mathbb{R}^{n\times n}$ the diffusion operator. Let ϕ, ψ be L_ϕ, L_ψ-Lipschitz pointwise functions. Define:

$$\begin{aligned}
g_{k+1} &\triangleq \phi\left(\boldsymbol{b}_k + \boldsymbol{S}g_k\left(\boldsymbol{H}\right)\boldsymbol{W}_k\right),\\
g_0 &\triangleq \boldsymbol{X}
\end{aligned}$$

and the hypothesis class over all such functions as

$$\mathcal{H}_{\mathcal{G}}^{\phi,\psi} \triangleq \left\{h_{\mathcal{G}}^{\phi,\psi}(\boldsymbol{X}) = \psi\left(g_K \circ \cdots \circ g_0\right)\right\}.$$

From there we derive a recursive TRC bound depending on the previous layer.

Consider $g_{k+1} \triangleq \phi\left(\boldsymbol{b}_k + \boldsymbol{S}g_k\left(\boldsymbol{H}\right)\boldsymbol{W}_k\right)$, $k \in \{1, \cdots, K\}$. Now we define the function class over one neuron as

$$\mathcal{H}_{\mathcal{G}}^{\phi} \triangleq \left\{h_{\mathcal{G}}^{\phi}(\boldsymbol{H}) = \phi\left(\boldsymbol{b}_i + \sum_{l}^{d_k}\boldsymbol{W}_{lj}\sum_{t}^{n}\boldsymbol{S}_{it}g(\boldsymbol{H})_{lj}\right) \;\middle|\; g \in \mathcal{F}, \|\boldsymbol{b}\|_1 \leq \beta\right\}$$

where $\mathcal{F}$ is the class of $\mathbb{R}^{n\times d_k} \to \mathbb{R}$, including the zero function. Then with $\boldsymbol{W}_{\cdot j} \triangleq [\boldsymbol{W}_{1j}, \cdots, \boldsymbol{W}_{d_k j}]^\top$:

$$\mathfrak{R}_{m,n}(\mathcal{H}^{\phi}_{\mathcal{G}}) \leq 2L_\phi \left(\beta Q(n) + \|\boldsymbol{S}\|_\infty \|\boldsymbol{W}_{\cdot j}\|_1 \mathfrak{R}_{m,n}(\mathcal{F})\right)$$

Proof.

By Lemma 4 and Lemma 2 we get

$$\mathfrak{R}_{m,n}(\mathcal{H}^{\phi}_{\mathcal{G}}) \leq L_\phi \left(\mathfrak{R}_{m,n}(\mathcal{H}_{lin}) + \mathfrak{R}_{m,n}(\mathcal{H}_{bias})\right)$$

where

$$\begin{aligned}\mathcal{H}_{lin} &\triangleq \left\{ h_{lin}(\boldsymbol{H}) = \sum_{l}^{d_k} \boldsymbol{W}_{lj} \sum_{t}^{n} \boldsymbol{S}_{it} g(\boldsymbol{H})_{lj} \;\middle|\; g \in \mathcal{F}, \|\boldsymbol{W}_{\cdot j}\|_1 \leq \omega \right\}\\ \mathcal{H}_{bias} &\triangleq \{h_{bias}(\boldsymbol{H}) = \boldsymbol{b} \mid |b| \leq \beta\}\end{aligned}$$

with $\boldsymbol{W}_{\cdot j} \triangleq [\boldsymbol{W}_{1j}, \cdots, \boldsymbol{W}_{d_k j}]^\top$. Bounding terms individually.

Bound $\mathfrak{R}_{m,n}(\mathcal{H}_{lin})$

We start by rewriting the linear term. For readability $g_{lj} := g(\boldsymbol{H})_{lj}$

$$\begin{aligned}\boldsymbol{H}_{ij} &= \sum_{l}^{d_k} \boldsymbol{W}_{lj} \sum_{t}^{n} \boldsymbol{S}_{it} g_{lj}\\ &= \underbrace{\boldsymbol{W}_{1j}\boldsymbol{S}_{i1}g_{1j} + \cdots + \boldsymbol{W}_{1j}\boldsymbol{S}_{in}g_{1j}}_{\boldsymbol{W}_{1j}g_{1j}\left(\sum_t^n \boldsymbol{S}_{it}\right)} + \underbrace{\boldsymbol{W}_{2j}\boldsymbol{S}_{i1}g_{2j} + \cdots + \boldsymbol{W}_{2j}\boldsymbol{S}_{in}g_{2j}}_{\boldsymbol{W}_{2j}g_{2j}\left(\sum_t^n \boldsymbol{S}_{it}\right)} + \cdots\\ &\quad \text{with } \sum_{t}^{n} \boldsymbol{S}_{it} \leq \|\boldsymbol{S}\|_\infty\\ &\leq \|\boldsymbol{S}\|_\infty \left(\sum_{l}^{d_k} \boldsymbol{W}_{1j} g_{1j} \right)\end{aligned}$$

Now we define

$$\begin{aligned}\widetilde{\mathcal{H}}_{lin} &\triangleq \left\{ h_{lin}(\boldsymbol{H}) = \sum_{l}^{d_k} \boldsymbol{W}_{lj} g(\boldsymbol{H})_{lj} \;\middle|\; g \in \mathcal{F}, \|\boldsymbol{W}_{\cdot j}\|_1 \leq \omega \right\}\\ \widetilde{\mathcal{H}}'_{lin} &\triangleq \left\{ h_{lin}(\boldsymbol{H}) = \sum_{l}^{d_k} \boldsymbol{W}_{lj} g(\boldsymbol{H})_{lj} \;\middle|\; g \in \mathcal{F}, \|\boldsymbol{W}_{\cdot j}\|_1 = \omega \right\}\end{aligned}$$

and since $\|\boldsymbol{S}\|_\infty$ is constant we get by Lemma 1

$$\mathfrak{R}_{m,n}(\mathcal{H}_{lin}) \leq \|\boldsymbol{S}\|_\infty \mathfrak{R}_{m,n}(\widetilde{\mathcal{H}}_{lin}).$$

To further bound $\mathfrak{R}_{m,n}(\widetilde{\mathcal{H}}_{lin})$ we can a similar process then for standard deep neural networks with slight deviation on the indexing of the weight matrix.

Let Hull$\,(\cdot)$ be a convex hull. In the first step we show that

$$\mathfrak{R}_{m,n}(\widetilde{\mathcal{H}}_{lin}) = \omega\mathfrak{R}_{m,n}(\text{Hull}\,(\mathcal{F}-\mathcal{F}))$$

where $\mathcal{F}-\mathcal{F} \triangleq \{f-f',\ f \in \mathcal{F}, f' \in \mathcal{F}\}$. Note that the maximum over all function over $\boldsymbol{W}_{il}$ with constraint $\|\boldsymbol{W}_{\cdot j}\|_1 \leq \omega$ is achieved for $\|\boldsymbol{W}_{\cdot j}\|_1 = \omega$ then

$$\mathfrak{R}_{m,n}(\widetilde{\mathcal{H}}_{lin}) = \mathfrak{R}_{m,n}(\widetilde{\mathcal{H}}'_{lin})$$

Let $\mathbf{0}$ be the zero function. Then for $\|\boldsymbol{W}_{\cdot j}\|_1 = 1$:

$$\sum_l \boldsymbol{W}_{lj} g_{lj} = \sum_{l:\boldsymbol{W}_{lj}\geq 0} \boldsymbol{W}_{lj}(g_{lj}-\mathbf{0}) + \sum_{l:\boldsymbol{W}_{lj}<0} |\boldsymbol{W}_{lj}|(\mathbf{0}-g_{lj})$$

which is Hull$\,(\mathcal{F}-\mathcal{F})$. Combining the above results we get:

$$\begin{aligned}
\mathfrak{R}_{m,n}(\mathcal{H}_{lin}) &\leq \|\boldsymbol{S}\|_\infty \omega\mathfrak{R}_{m,n}\left(\widetilde{\mathcal{H}}_{lin}\right) \\
&= \|\boldsymbol{S}\|_\infty \omega\mathfrak{R}_{m,n}\left(\text{Hull}\,(\mathcal{F}-\mathcal{F})\right) \\
&= \|\boldsymbol{S}\|_\infty \omega\mathfrak{R}_{m,n}\left(\mathcal{F}-\mathcal{F}\right) \\
&= \|\boldsymbol{S}\|_\infty \omega\left(\mathfrak{R}_{m,n}\left(\mathcal{F}\right) + \mathfrak{R}_{m,n}\left(-\mathcal{F}\right)\right) && \text{Lemma 2} \\
&= 2\|\boldsymbol{S}\|_\infty \omega\mathfrak{R}_{m,n}\left(\mathcal{F}\right) && \text{Lemma 1}
\end{aligned}$$

which concludes this part of the proof.

Bound $\mathfrak{R}_{m,n}(\mathcal{H}_{bias})$

Start by writing out $\mathfrak{R}_{m,n}(\cdot)$

$$\begin{aligned}
\mathfrak{R}_{m,n}(\mathcal{H}_{bias}) &= Q\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sup_{\boldsymbol{b}:|\boldsymbol{b}|\leq\beta} \boldsymbol{b}\sum_{i=1}^{n}\boldsymbol{\sigma}_i\right] \\
&\leq Q\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sup_{\boldsymbol{b}:|\boldsymbol{b}|\leq\beta} |\boldsymbol{b}|\left|\sum_{i=1}^{n}\boldsymbol{\sigma}_i]\right|\right] \\
&= \beta Q\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\left|\sum_{i=1}^{n}\boldsymbol{\sigma}_i\right|\right] \\
&\leq \beta Q\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sum_{i=1}^{n}|\boldsymbol{\sigma}_i|\right] \\
&\leq \beta Q\sum_{i=1}^{n}\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[|\boldsymbol{\sigma}_i|\right] \\
&\leq \beta Q\sum_{i=1}^{n}2p \\
&\leq \beta Q(n)2p \\
&\leq \beta Q(n)2
\end{aligned}$$

which concludes this part of the proof. Combining the two bounds gives:

$$\mathfrak{R}_{m,n}(\mathcal{H}_{\mathcal{G}}^{\phi}) \leq 2L_{\phi}\left(\beta Q(n) + \|\boldsymbol{S}\|_{\infty}\|\boldsymbol{W}_{\cdot j}\|_1\mathfrak{R}_{m,n}(\mathcal{F})\right)$$

concluding the proof of Proposition 2. □

D.3 Bound on the TRC for the first layer

Define the hypothesis class over the function of the first layer g_0 as:

$$\mathcal{H}_{\mathcal{G}}^{\phi} \triangleq \left\{h_{\mathcal{G}}^{\phi}(\boldsymbol{X}) = \phi\left(\boldsymbol{b} + \boldsymbol{S}\boldsymbol{X}\boldsymbol{W}_1\right)\right\}$$

then the TRC is give by

$$\mathfrak{R}_{m,n}(\mathcal{H}_{\mathcal{G}}^{\phi}) \leq L_{\phi}\left(\|\boldsymbol{b}\|_1 Q(n)2 + Q\|\boldsymbol{W}_1\|_{\infty}\|\boldsymbol{S}\boldsymbol{X}\|_{2\to\infty}\sqrt{\frac{2\log(n)}{d}}\right)$$

Proof. Frist similar to Proposition 2 we use Lemma 2 and Lemma 4

$$\mathfrak{R}_{m,n}(\mathcal{H}_{\mathcal{G}}^{\phi}) \leq L_{\phi}\left(\mathfrak{R}_{m,n}(\mathcal{H}_{lin}) + \mathfrak{R}_{m,n}(\mathcal{H}_{bias})\right)$$

As before $\mathfrak{R}_{m,n}(\mathcal{H}_{bias}) \leq \beta Q(n)2p$. In this case we define the linear term as

$$\mathcal{H}_{lin} \triangleq \left\{h_{lin}(\boldsymbol{X}) = \boldsymbol{S}\boldsymbol{X}\boldsymbol{W}\right\}.$$

Bounding the TRC of $\mathcal{H}_{lin}$

$$\begin{aligned}\mathfrak{R}_{m,n}(\mathcal{H}_{lin}) &= Q\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sup_{\boldsymbol{W}:\|\boldsymbol{W}\|_\infty\leq\omega}\boldsymbol{\sigma}^\top\boldsymbol{S}\boldsymbol{X}\boldsymbol{W}\right]\\ &\leq Q\left\|\boldsymbol{W}\right\|_\infty\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\left\|\boldsymbol{\sigma}^\top\boldsymbol{S}\boldsymbol{X}\right\|_\infty\right]\end{aligned}$$

To bound $\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\left\|\boldsymbol{\sigma}^\top\boldsymbol{S}\boldsymbol{X}\right\|_\infty\right]$ we define $\boldsymbol{t}_i = (x_{1j},\dots,x_{nj})^\top$ and $T = \{t_1,\dots,t_n\}, T_- = \{-t_1,\dots,-t_n\}$. Therefore

$$\begin{aligned}\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\left\|\boldsymbol{\sigma}^\top\boldsymbol{S}\boldsymbol{X}\right\|_\infty\right] &\leq \mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\max_{t\in T}|\boldsymbol{\sigma}^\top\boldsymbol{S}\boldsymbol{t}|\right]\\ &= \mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\max_{t\in T\cup T_-}\boldsymbol{\sigma}^\top\boldsymbol{S}\boldsymbol{t}\right]\\ &\leq \max_{t\in T\cup T_-}\|\boldsymbol{S}\boldsymbol{t}\|_2\sqrt{\frac{2\log(n)}{d}} &&\text{Lemma 5}\\ &= \|\boldsymbol{S}\boldsymbol{t}\|_{2\to\infty}\sqrt{\frac{2\log(n)}{d}}\end{aligned}$$

Combining with the above results gives

$$\mathfrak{R}_{m,n}(\mathcal{H}_{lin}) \leq Q\left\|\boldsymbol{W}\right\|_\infty\|\boldsymbol{S}\boldsymbol{t}\|_{2\to\infty}\sqrt{\frac{2\log(n)}{d}}.$$

Taking the bound on the bias term into considerations gives the final bound and concludes the proof of Proposition 3. □

D.4 Additional notes on the remarks related to Theorem 1

Influence of the graph information: Empty and fully-connected graph. To be able to analyse the influence of the graph information we can note that the graph information comes into play through $\|\boldsymbol{S}\boldsymbol{X}\|_{2\to\infty}$. We can rewrite this expression as $\|\boldsymbol{S}\boldsymbol{X}\|_{2\to\infty} = \max_j\sqrt{\sum_i(\boldsymbol{S}\boldsymbol{X})^2_{ij}}$ and then by replacing $\boldsymbol{S}$ with the empty ($\boldsymbol{A} = K_\mathcal{G}$) and the complete graph ($\boldsymbol{A} = \mathbb{I}$) gives: $\|K_\mathcal{G}\boldsymbol{X}\|_{2\to\infty} = \max_j\frac{1}{\sqrt{n}}\sqrt{\left(\sum_k\boldsymbol{X}_{kj}\right)^2}$, and $\|\mathbb{I}\boldsymbol{X}\|_{2\to\infty} = \max_j\sqrt{\sum_k\boldsymbol{X}^2_{kj}}$ and since $\left(\sum_k\boldsymbol{X}_{kj}\right)^2 \leq n\left\|\boldsymbol{X}_{\cdot j}\right\|_2^2$ it follows that $\mathfrak{R}(\mathcal{H}^\phi_{K_\mathcal{G}}) \leq \mathfrak{R}(\mathcal{H}^\phi_\mathbb{I})$ which is consistent with the observation obtained from the VC-Dimension bound. In both cases the complexity measure of the fully connected graph is lower then the if we would not consider graph information.

Influence of the graph information: b-regular graph. Now consider a setup that incorporates a larger number of graphs. Assume $\boldsymbol{S} := \boldsymbol{D}^{-\frac{1}{2}}(\boldsymbol{A} + \mathbb{I})\boldsymbol{D}^{-\frac{1}{2}}$ and that we only consider the graph information (e.g. $\boldsymbol{X} = \mathbb{I}$), then for a b-regular graph (a graph where every vertex has degree b) we can write $\|\boldsymbol{S}\mathbb{I}\|_{2\to\infty} = \max_j \|\boldsymbol{S}_{\cdot j}\|_2 = \sqrt{\sum_{i\sim j} \frac{1}{\boldsymbol{D}_i \boldsymbol{D}_j}} = \frac{1}{\sqrt{b}} < 1$. Therefore adding graph information results in $\mathfrak{R}(\mathcal{H}_{\mathcal{G}}^{\phi}) \leq \mathfrak{R}(\mathcal{H}_{\mathbb{I}}^{\phi})$ and therefore the complexity resulting in not using graph information upper bounds the complexity that results if we consider graph information.

E Proof Theorem 3 — TRC for Residual GNNs

Recall the setup for residual connections as defined in the main paper where we can now write the layer wise propagation rule as

$$g_{k+1} \triangleq \phi\left((1-\alpha)\left(\boldsymbol{b}_k + \boldsymbol{S} g_k\left(\boldsymbol{H}\right)\boldsymbol{W}_k\right) + \alpha g_0\left(\boldsymbol{H}\right)\right), \qquad \text{with } \alpha \in (0,1).$$

We can now derive a generalization error bound similar to the one given in Theorem 1 for the Residual network. As most of the steps are the same we will only remark the main changes. Recall that for the vanilla case we considered

$$\mathfrak{R}_{m,n}(\mathcal{H}_{\mathcal{G}}^{\phi}) \leq L_\phi\left(\mathfrak{R}_{m,n}(\mathcal{H}_{lin}) + \mathfrak{R}_{m,n}(\mathcal{H}_{bias})\right)$$

and by Lemma 2 and Lemma 1 we obtain a similar bound for the Residual network as

$$\mathfrak{R}_{m,n}(\mathcal{H}_{\mathcal{G}}^{\phi}) \leq L_\phi\left((1-\alpha)\mathfrak{R}_{m,n}(\mathcal{H}_{lin}) + (1-\alpha)\mathfrak{R}_{m,n}(\mathcal{H}_{bias}) + \alpha\mathfrak{R}_{m,n}(\mathcal{H}_{\boldsymbol{X}})\right).$$

The bounds for $\mathfrak{R}_{m,n}(\mathcal{H}_{lin})$ and $\mathfrak{R}_{m,n}(\mathcal{H}_{bias})$ are as derived in section D. $\mathfrak{R}_{m,n}(\mathcal{H}_{\boldsymbol{X}})$ can be bound as

$$\mathfrak{R}_{m,n}(\mathcal{H}_{\boldsymbol{X}}) \leq 2Q\left\|\boldsymbol{X}\right\|_\infty n$$

Where the proof follows analogous to the one for the *bias term*, $\mathfrak{R}_{m,n}(\mathcal{H}_{bias}$.

Again with recursively applying the bounds for each layer and combining it with the bound on the first layer results in the full TRC bound. Consider a Residual network as defined in (11) and $\mathcal{H}_{\mathcal{G}}^{\phi,\beta,\omega} \subset \mathcal{H}_{\mathcal{G}}^{\phi}$ such that the trainable parameters satisfy $\left\|\boldsymbol{b}_k\right\|_1 \leq \beta$ and $\left\|\boldsymbol{W}_k\right\|_\infty \leq \omega$ for every $k \in [K]$. Then with $\alpha \in (0,1)$ and $c_1 \triangleq 2L_\phi\beta$, $c_2 \triangleq 2L_\phi\omega$, $c_3 \triangleq L_\phi\omega\sqrt{2/d}$ the TRC of the restricted class or Residual GNNs is bounded as

$$\begin{aligned}\mathfrak{R}_{m,n}(\mathcal{H}_{\mathcal{G}}^{\phi,\beta,\omega}) \;\leq\; & \frac{((1-\alpha)c_1 + \alpha 2L_\phi\left\|\boldsymbol{X}\right\|_\infty)n^2}{m(n-m)}\left(\sum_{k=0}^{K-1}(1-\alpha)c_2^k\left\|\boldsymbol{S}\right\|_\infty^k\right) \\ & + \alpha 2L_\phi\left\|\boldsymbol{X}\right\|_\infty + (1-\alpha)c_3c_2^K\left\|\boldsymbol{S}\right\|_\infty^K\left\|\boldsymbol{S}\boldsymbol{X}\right\|_{2\to\infty}\sqrt{\log(n)}\end{aligned}$$

F Proof Theorem 2 — Expected TRC for GNNs under SBM

F.1 Setup (recap from the main paper)

We assume that the node features are sampled latent true classes, given a $\boldsymbol{z} = (z_1, \dots, z_n) \in \{\pm 1\}^n$. The node features are sampled from a Gaussian mixture model (GMM), that is, feature for node-i is sampled as $\boldsymbol{x}_i \sim \mathcal{N}(z_i \boldsymbol{\mu}, \sigma^2 \mathbb{I})$ for some $\boldsymbol{\mu} \in \mathbb{R}^d$ and $\sigma \in (0, \infty)$. We express this in terms of $\boldsymbol{X}$ as

$$\boldsymbol{X} = \mathcal{X} + \boldsymbol{\epsilon} \ \in \ \mathbb{R}^{n \times d}, \qquad \text{where } \mathcal{X} = \boldsymbol{z}\boldsymbol{\mu}^\top \ \text{ and } \boldsymbol{\epsilon} = (\epsilon_{ij})_{i \in [n], j \in [d]} \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2).$$

We refer to above as $\boldsymbol{X} \sim$ 2GMM. On the other hand, we assume that graph has two latent communities, characterised by $\boldsymbol{y} \in \{\pm 1\}^n$. The graph is generated from a stochastic block model with two classes (2SBM), where edges (i, j) are added independently with probability $p \in (0, 1]$ if $y_i = y_j$, and with probability $q < [0, p)$ if $y_i \neq y_j$. In other words, we define the random adjacency $\boldsymbol{A} \sim$ 2SBM as a symmetric binary matrix with $\boldsymbol{A}_{ii} = 0$, and $(\boldsymbol{A}_{ij})_{i<j}$ indenpendent such that

$$\boldsymbol{A}_{ij} \sim \text{Bernoulli}(\mathcal{A}_{ij}), \qquad \text{where } \mathcal{A} = \frac{p+q}{2} \boldsymbol{1}\boldsymbol{1}^\top + \frac{p-q}{2} \boldsymbol{y}\boldsymbol{y}^\top - p\mathbb{I}.$$

The choice of two different latent classes $\boldsymbol{z}, \boldsymbol{y} \in \{\pm 1\}^n$ allows study of the case where the graph and feature information of do not align completely. We use $\mathit{\Gamma} = |\boldsymbol{y}^\top \boldsymbol{z}| \in [0, n]$ to quantify this alignment. Assuming $\boldsymbol{y}, \boldsymbol{z}$ are both balanced, that is, $\sum_i y_i = \sum_i z_i = 0$.

In addition the TRC is given by Theorem 1:

Consider $\mathcal{H}_{\mathcal{G}}^{\phi,\beta,\omega} \subseteq \mathcal{H}_{\mathcal{G}}^{\phi}$ such that the trainable parameters satisfy $\|\boldsymbol{b}_k\|_1 \leq \beta$ and $\|\boldsymbol{W}_k\|_\infty \leq \omega$ for every $k \in [K]$. The transductive Randemacher complexity (TRC) of the restricted hypothesis class is bounded as

$$\mathfrak{R}_{m,n}(\mathcal{H}_{\mathcal{G}}^{\phi,\beta,\omega}) \leq \frac{c_1 n^2}{m(n-m)} \left(\sum_{k=0}^{K-1} c_2^k \|\boldsymbol{S}\|_\infty^k \right) + c_3 c_2^K \|\boldsymbol{S}\|_\infty^K \|\boldsymbol{S}\boldsymbol{X}\|_{2\to\infty} \sqrt{\log(n)}\ ,$$

where $c_1 \triangleq 2L_\phi \beta$, $c_2 \triangleq 2L_\phi \omega$, $c_3 \triangleq L_\phi \omega \sqrt{2/d}$ and L_ϕ is Lipschitz constant for activation ϕ.

F.2 Main Proof

From the above bound we can note that to derive the TRC in expectation we have to compute $\mathbb{E}\left[\|\boldsymbol{S}\|_\infty^k\right]$ and $\mathbb{E}\left[\|\boldsymbol{S}\|_\infty^k \|\boldsymbol{S}\boldsymbol{X}\|_{2\to\infty}\right]$ where we can decompose the latter as follows

$$\begin{aligned}
\mathbb{E}\left[\|\boldsymbol{S}\|_\infty^k \|\boldsymbol{S}\boldsymbol{X}\|_{2\to\infty}\right] \leq & \mathbb{E}\left[\|\boldsymbol{S}\|_\infty^k \|\mathcal{S}\mathcal{X}\|_{2\to\infty}\right] \\
& + \mathbb{E}\left[\|\boldsymbol{S}\|_\infty^k \|(\boldsymbol{S}-\mathcal{S})\mathcal{X}\|_{2\to\infty}\right] + \mathbb{E}\left[\|\boldsymbol{S}\|_\infty^k \|\boldsymbol{S}(\boldsymbol{X}-\mathcal{X})\|_{2\to\infty}\right] \\
\leq & \|\mathcal{S}\mathcal{X}\|_{2\to\infty} \mathbb{E}\left[\|\boldsymbol{S}\|_\infty^k\right] \\
& + \sqrt{\mathbb{E}\left[\|\boldsymbol{S}\|_\infty^{2k}\right]}\sqrt{\mathbb{E}\left[\|(\boldsymbol{S}-\mathcal{S})\mathcal{X}\|_{2\to\infty}^2\right]} \\
& + \sqrt{\mathbb{E}\left[\|\boldsymbol{S}\|_\infty^{2k}\right]}\sqrt{\mathbb{E}\left[\|(\boldsymbol{X}-\mathcal{X})\boldsymbol{S}\|_{2\to\infty}^2\right]}
\end{aligned}$$

where the second inequality follows from noting that $\|\mathcal{S}\mathcal{X}\|_{2\to\infty}$ is deterministic and does not depend on the expectation and the decomposition of the last two terms follows from using Cauchy-Schwarz inequality.

Table F.2 gives an overview over the bounds on the different terms, where the individual entries are derived in section F.3.

tableOverview over different concentration bounds for *self loop* and *degree normalization*. Let $C = (1 + o(1))$

	Self Loop	Degree Normalized
F.3: $\|\mathcal{S}\mathcal{X}\|_{2\to\infty}$	$C\,\|\boldsymbol{\mu}\|_\infty\, n\left(1+\left(\frac{p-q}{2}\right)^2 \varGamma^2\right)$	$C\,\|\boldsymbol{\mu}\|_\infty \frac{\left(1+\left(\frac{p-q}{2}\right)^2\varGamma^2\right)}{\left(\frac{p+q}{2}\right)}$
F.3: $\mathbb{E}\left[\|(\boldsymbol{S}-\mathcal{S})\,\mathcal{X}\|_{2\to\infty}^2\right]$	$Cn^2p\,\|\boldsymbol{\mu}\|_\infty$	$C\frac{n\ln(n)}{1+(n-1)q}\,\|\boldsymbol{\mu}\|_\infty$
F.3: $\mathbb{E}\left[\|(\boldsymbol{X}-\mathcal{X})\,\boldsymbol{S}\|_{2\to\infty}^2\right]$	$Cn^2p\sigma^2(1+2\ln d)$	$C\frac{1}{q}$
F.3: $\mathbb{E}\left[\|\boldsymbol{S}\|_\infty^k\right]$	$(Cnp)^k$	$\left(C\frac{p}{q}\right)^{\frac{k}{2}}$

F.3 Concentration Bounds

Bound $\mathbb{E}\left[\|(\boldsymbol{S}-\boldsymbol{\mathcal{S}})\,\boldsymbol{\mathcal{X}}\|_{2\to\infty}\right]$ We first note that:

$$\begin{aligned}
\|(\boldsymbol{S}-\mathcal{S})\,\mathcal{X}\|_{2\to\infty} &= \left\|(\boldsymbol{S}-\mathcal{S})\,\boldsymbol{z}\boldsymbol{\mu}^\top\right\|_{2\to\infty} && \text{by definition of } \mathcal{X}\\
&= \max_j \left\|(\boldsymbol{S}-\mathcal{S})\,\boldsymbol{z}\boldsymbol{\mu}_j^\top\right\|_2 && \text{by definition of } \|\ \cdot\ \|_{2\to\infty}\\
&= \|(\boldsymbol{S}-\mathcal{S})\,\boldsymbol{z}\|_2\,\|\boldsymbol{\mu}\|_\infty && (17)
\end{aligned}$$

and we only have to compute the expectation of $\|(\boldsymbol{S}-\mathcal{S})\,\boldsymbol{z}\|_2$ as $\|\boldsymbol{\mu}\|_\infty$ is deterministic. Taking the expectation:

$$\begin{aligned}
\mathbb{E}\left[\|(\boldsymbol{S}-\mathcal{S})\,\boldsymbol{z}\|_2\right] &\le \sqrt{\mathbb{E}\left[\boldsymbol{z}^\top(\boldsymbol{S}-\mathcal{S})^\top(\boldsymbol{S}-\mathcal{S})\,\boldsymbol{z}\right]}\\
&= \left(\sum_{ij}\boldsymbol{z}_i\boldsymbol{z}_j\sum_k\mathbb{E}\left[(\boldsymbol{S}-\mathcal{S})_{ki}\,(\boldsymbol{S}-\mathcal{S})_{kj}\right]\right)^{\frac{1}{2}} \qquad (18)
\end{aligned}$$

where (18) follows from the fact that $\boldsymbol{z}$ is deterministic. From this expression we can now consider the self loop and degree normalized case for the diffusion operator.

Case 1: Self loop.

$\sum_k \mathbb{E}\left[(\boldsymbol{S}-\mathcal{S})_{ki}\,(\boldsymbol{S}-\mathcal{S})_{kj}\right]$ in (18) now becomes $\sum_k \mathbb{E}\left[(\boldsymbol{A}-\mathcal{A})_{ki}\,(\boldsymbol{A}-\mathcal{A})_{kj}\right]$ where we distinguish two cases:

$$\begin{aligned}
i \ne j \qquad &\Rightarrow \boldsymbol{A}_{ki} \text{ and } \boldsymbol{A}_{kj} \text{ are independent} \Rightarrow \mathbb{E}\left[(\boldsymbol{A}-\mathcal{A})_{ki}\,(\boldsymbol{A}-\mathcal{A})_{kj}\right] = 0\\
i = j \qquad &\Rightarrow \mathbb{E}\left[(\boldsymbol{A}-\mathcal{A})_{ki}\,(\boldsymbol{A}-\mathcal{A})_{ki}\right] = \mathrm{Var}(\boldsymbol{A}_{ki}) = \mathcal{A}_{ki}(1-\mathcal{A}_{ki})
\end{aligned}$$

Therefore (18) becomes

$$\begin{aligned}
\mathbb{E}\left[\left\|(\boldsymbol{S}-\mathcal{S})\,\boldsymbol{z}\right\|_2\right] &\leq \left(\sum_i \boldsymbol{z}_i^2 \sum_k \mathcal{A}_{ki}(1-\mathcal{A}_{ki})\right)^{\frac{1}{2}} \\
&= \left(\sum_{ik} \mathcal{A}_{ki}(1-\mathcal{A}_{ki})\right)^{\frac{1}{2}} && \because \boldsymbol{z}_i^2 = 1 \\
&\leq \left(\sum_{ik} \mathcal{A}_{ki}\right)^{\frac{1}{2}} \\
&\leq \left(n^2 \frac{p+q}{2}\right)^{\frac{1}{2}} \\
&= n\sqrt{\frac{p+q}{2}}
\end{aligned}$$

and giving us the final bound as using the above in (17):

$$\mathbb{E}\left[\left\|(\boldsymbol{S}-\mathcal{S})\,\mathcal{X}\right\|_{2\to\infty}\right] \leq n\sqrt{\frac{p+q}{2}}\,\|\boldsymbol{\mu}\|_\infty$$

Case 2: Degree normalized.

Note that for this section we initially considered an extension of the degree normalized model where the self loop is weighted by γ. For the final version however we set $\gamma = 1$.

As before first note that:

$$\begin{aligned}
\left\|(\boldsymbol{S}-\mathcal{S})\,\mathcal{X}\right\|_{2\to\infty} &= \left\|(\boldsymbol{S}-\mathcal{S})\,\boldsymbol{z}\boldsymbol{\mu}^\top\right\|_{2\to\infty} \\
&= \max_j \left\|(\boldsymbol{S}-\mathcal{S})\,\boldsymbol{z}\boldsymbol{\mu}_j^\top\right\|_2 \\
&= \left\|(\boldsymbol{S}-\mathcal{S})\,\boldsymbol{z}\right\|_2 \|\boldsymbol{\mu}\|_\infty \qquad (19)
\end{aligned}$$

and we only have to compute the expectation of $\left\|(\boldsymbol{S}-\mathcal{S})\,\boldsymbol{z}\right\|_2$ in (19). To bound this term we start by defining:

$$\begin{aligned}
\mathcal{S} &\triangleq (\mathcal{D}+\gamma\mathbb{I})^{-\frac{1}{2}}(\mathcal{A}+\gamma\mathbb{I})(\mathcal{D}+\gamma\mathbb{I})^{-\frac{1}{2}} \\
\boldsymbol{S} &\triangleq (\boldsymbol{D}+\gamma\mathbb{I})^{-\frac{1}{2}}(\boldsymbol{A}+\gamma\mathbb{I})(\boldsymbol{D}+\gamma\mathbb{I})^{-\frac{1}{2}} \\
\overline{\boldsymbol{S}} &\triangleq (\mathcal{D}+\gamma\mathbb{I})^{-\frac{1}{2}}(\boldsymbol{A}+\gamma\mathbb{I})(\mathcal{D}+\gamma\mathbb{I})^{-\frac{1}{2}}
\end{aligned}$$

such that we can write:

$$\left\|(\boldsymbol{S}-\mathcal{S})\,\boldsymbol{z}\right\|_2 \leq \left\|\left(\boldsymbol{S}-\overline{\boldsymbol{S}}\right)\boldsymbol{z}\right\|_2 + \left\|\left(\overline{\boldsymbol{S}}-\mathcal{S}\right)\boldsymbol{z}\right\|_2 \qquad (20)$$

and bound the two terms separately:

Bound first term in (20)**:** $\left\|\left(\overline{\boldsymbol{S}}-\mathcal{S}\right)\boldsymbol{z}\right\|_2$

First we note that:

$$\left\|\left(\overline{\boldsymbol{S}}-\mathcal{S}\right)\boldsymbol{z}\right\|_2 \leq \left\|(\mathcal{D}+\gamma\mathbb{I})^{-\frac{1}{2}}(\boldsymbol{A}-\mathcal{A})(\mathcal{D}+\gamma\mathbb{I})^{-\frac{1}{2}}\boldsymbol{z}\right\|_2$$

and therefore

$$\mathbb{E}\left[\left\|\left(\overline{\boldsymbol{S}}-\mathcal{S}\right)\boldsymbol{z}\right\|_2\right] \leq \left(\mathbb{E}\left[\boldsymbol{z}^\top(\mathcal{D}+\gamma\mathbb{I})^{-\frac{1}{2}}(\boldsymbol{A}-\mathcal{A})(\mathcal{D}+\gamma\mathbb{I})^{-1}(\boldsymbol{A}-\mathcal{A})(\mathcal{D}+\gamma\mathbb{I})^{-\frac{1}{2}}\boldsymbol{z}\right]\right)^{-\frac{1}{2}}$$

$$= \left(\sum_{i,j}\frac{\boldsymbol{z}_i\boldsymbol{z}_j}{\sqrt{(\gamma+\mathcal{D}_{ii})(\gamma+\mathcal{D}_{jj})}}\underbrace{\sum_{k\neq i,j}\frac{\mathbb{E}\left[(\boldsymbol{A}-\mathcal{A})_{ki}(\boldsymbol{A}-\mathcal{A})_{kj}\right]}{\gamma+\mathcal{D}_{kk}}}_{\text{term 2}}\right)^{-\frac{1}{2}} \tag{21}$$

$$\leq \left(\sum_i\frac{\boldsymbol{z}_i^2}{\gamma+\mathcal{D}_{ii}}\cdot\frac{\mathcal{D}_{ii}}{\gamma+(n-1)q}\right)^{-\frac{1}{2}} \tag{22}$$

$$\leq \left(\frac{n}{\gamma+(n-1)q}\right)^{-\frac{1}{2}} \qquad \because \boldsymbol{z}_i^2 = 1$$

Where the step form (21) to (22) follows by bounding (21), *term 2* as follows. For $i \neq j$ the expression is zero. Otherwise for $i = j$:

$$\begin{aligned}\sum_{k\neq i,j}\frac{\mathbb{E}\left[(\boldsymbol{A}-\mathcal{A})_{ki}(\boldsymbol{A}-\mathcal{A})_{kj}\right]}{\gamma+\mathcal{D}_{kk}} &= \sum_{k\neq i}\frac{\mathrm{Var}(\boldsymbol{A}_{ki})}{\gamma+\mathcal{D}_{kk}}\\ &= \sum_{k\neq i}\frac{\mathcal{A}_{ki}(1-\mathcal{A}_{ki})}{\gamma+\mathcal{D}_{kk}}\\ &\leq \sum_{k\neq i}\frac{\mathcal{A}_{ki}}{\gamma+(n-1)q} \qquad \because \mathcal{D}_{kk} \geq (n-1)q\\ &= \frac{\mathcal{D}_{ii}}{\gamma+(n-1)q}\end{aligned}$$

Therefore

$$\mathbb{E}\left[\left\|\left(\overline{\boldsymbol{S}}-\mathcal{S}\right)\boldsymbol{z}\right\|_2\right] \leq \sqrt{\frac{n}{\gamma+(n-1)q}}$$

Bound second term in (20)**:** $\left\|\left(\boldsymbol{S}-\overline{\boldsymbol{S}}\right)\boldsymbol{z}\right\|_2$

Let $\boldsymbol{B} \triangleq \boldsymbol{D} + \gamma\mathbb{I}$ and $\boldsymbol{C} \triangleq \mathcal{D} + \gamma\mathbb{I}$. We first consider the following decomposition:

$$\begin{aligned}
&\boldsymbol{B}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{B}^{-\frac{1}{2}} - \boldsymbol{C}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{C}^{-\frac{1}{2}} \\
&= \boldsymbol{B}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{B}^{-\frac{1}{2}} - \boldsymbol{B}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{B}^{-\frac{1}{2}}\boldsymbol{B}^{\frac{1}{2}}\boldsymbol{C}^{-\frac{1}{2}} + \boldsymbol{B}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{B}^{-\frac{1}{2}}\boldsymbol{B}^{\frac{1}{2}}\boldsymbol{C}^{-\frac{1}{2}} - \underbrace{\boldsymbol{C}^{-\frac{1}{2}}\boldsymbol{B}^{\frac{1}{2}}\boldsymbol{B}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{B}^{-\frac{1}{2}}\boldsymbol{B}^{\frac{1}{2}}\boldsymbol{C}^{-\frac{1}{2}}}_{\text{equal to } \boldsymbol{C}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{C}^{-\frac{1}{2}}} \\
&= \underbrace{\boldsymbol{B}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{B}^{-\frac{1}{2}}}_{\boldsymbol{S}}\left(\mathbb{I} - \boldsymbol{B}^{\frac{1}{2}}\boldsymbol{C}^{-\frac{1}{2}}\right) + \left(\mathbb{I} - \boldsymbol{C}^{-\frac{1}{2}}\boldsymbol{B}^{\frac{1}{2}}\right)\underbrace{\boldsymbol{B}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{B}^{-\frac{1}{2}}}_{S}\boldsymbol{B}^{\frac{1}{2}}\boldsymbol{C}^{-\frac{1}{2}} \qquad (23)
\end{aligned}$$

Using (23) we can bound the expectation of $\left\|\left(\boldsymbol{S} - \overline{\boldsymbol{S}}\right)\boldsymbol{z}\right\|_2$ as:

$$\begin{aligned}
&\mathbb{E}\left[\left\|\left(\boldsymbol{S} - \overline{\boldsymbol{S}}\right)\boldsymbol{z}\right\|_2\right] \\
=&\mathbb{E}\left[\left((\boldsymbol{D}+\gamma\mathbb{I})^{-\frac{1}{2}}(\boldsymbol{A}+\gamma\mathbb{I})(\boldsymbol{D}+\gamma\mathbb{I})^{-\frac{1}{2}} - (\mathcal{D}+\gamma\mathbb{I})^{-\frac{1}{2}}(\boldsymbol{A}+\gamma\mathbb{I})(\mathcal{D}+\gamma\mathbb{I})^{-\frac{1}{2}}\right)\boldsymbol{z}\right] \\
\leq&\mathbb{E}\left[\left\|\boldsymbol{S}\left(\mathbb{I} - (\boldsymbol{D}+\gamma\mathbb{I})^{\frac{1}{2}}(\mathcal{D}+\gamma\mathbb{I})^{-\frac{1}{2}}\right)\boldsymbol{z}\right\|_2\right] \qquad (24) \\
&+\mathbb{E}\left[\left\|\left(\mathbb{I} - (\boldsymbol{D}+\gamma\mathbb{I})^{\frac{1}{2}}(\mathcal{D}+\gamma\mathbb{I})^{-\frac{1}{2}}\right)\boldsymbol{S}(\boldsymbol{D}+\gamma\mathbb{I})^{\frac{1}{2}}(\mathcal{D}+\gamma\mathbb{I})^{-\frac{1}{2}}\boldsymbol{z}\right\|_2\right] \qquad (25)
\end{aligned}$$

Bound (24):

$$\begin{aligned}
\mathbb{E}\left[\left\|\boldsymbol{S}\left(\mathbb{I} - (\boldsymbol{D}+\gamma\mathbb{I})^{\frac{1}{2}}(\mathcal{D}+\gamma\mathbb{I})^{-\frac{1}{2}}\right)\boldsymbol{z}\right\|_2\right] &\leq \mathbb{E}\left[\|\boldsymbol{S}\|_2\left\|\left(\mathbb{I} - (\boldsymbol{D}+\gamma\mathbb{I})^{\frac{1}{2}}(\mathcal{D}+\gamma\mathbb{I})^{-\frac{1}{2}}\right)\boldsymbol{z}\right\|_2\right] \\
&\leq \sqrt{\sum_i \mathbb{E}\left[\left(1 - \sqrt{\frac{\boldsymbol{D}_{ii}+\gamma}{\mathcal{D}_{ii}+\gamma}}\right)^2 \boldsymbol{z}_i^2\right]} \\
&\qquad\qquad\qquad \because \|\boldsymbol{S}\|_2 \leq 1 \\
&\leq \sqrt{\sum_i \mathbb{E}\left[\left(1 - \sqrt{\frac{\boldsymbol{D}_{ii}+\gamma}{\mathcal{D}_{ii}+\gamma}}\right)^2\right]}
\end{aligned}$$

we therefore now need to compute $\sum_i \mathbb{E}\left[\left(1-\sqrt{\frac{\boldsymbol{D}_{ii}+\gamma}{\mathcal{D}_{ii}+\gamma}}\right)^2\right]$. Note that for $x \geq 0,\ |1-\sqrt{x}| \leq |1-x|$. Using this we write

$$\begin{aligned}
\sum_i \mathbb{E}\left[\left(1-\sqrt{\frac{\boldsymbol{D}_{ii}+\gamma}{\mathcal{D}_{ii}+\gamma}}\right)^2\right] &\leq \sum_i \mathbb{E}\left[\left(1-\frac{\boldsymbol{D}_{ii}+\gamma}{\mathcal{D}_{ii}+\gamma}\right)^2\right] \\
&= \sum_i 1-2+\frac{\mathbb{E}\left[(\boldsymbol{D}_{ii}+\gamma)^2\right]}{(\mathcal{D}_{ii}+\gamma)^2} \\
&= \sum_i -1+\frac{\mathbb{E}\left[(\gamma+\sum_{k\neq i}\boldsymbol{A}_{ik})^2\right]}{(\mathcal{D}_{ii}+\gamma)^2} \\
&= -n+\sum_i \frac{(\mathcal{D}_{ii}+\gamma)^2+\mathcal{D}_{ii}+\sum_{k\neq i}\boldsymbol{A}_{ik}^2}{(\mathcal{D}_{ii}+\gamma)^2} \\
&= \sum_i \frac{\sum_{k\neq i}\boldsymbol{A}_{ik}(1-\boldsymbol{A}_{ik})}{(\mathcal{D}_{ii}+\gamma)^2} \\
&\leq \sum_i \frac{1}{\mathcal{D}_{ii}+\gamma} \\
&\leq \frac{n}{\gamma+(n-1)q} \qquad (26)
\end{aligned}$$

Bound (25):

$$\begin{aligned}
&\mathbb{E}\left[\left\|\left(\mathbb{I}-(\boldsymbol{D}+\gamma\mathbb{I})^{\frac{1}{2}}(\mathcal{D}+\gamma\mathbb{I})^{-\frac{1}{2}}\right)\boldsymbol{S}(\boldsymbol{D}+\gamma\mathbb{I})^{\frac{1}{2}}(\mathcal{D}+\gamma\mathbb{I})^{-\frac{1}{2}}\boldsymbol{z}\right\|_2\right] \\
\leq&\mathbb{E}\left[\left\|\mathbb{I}-(\boldsymbol{D}+\gamma\mathbb{I})^{\frac{1}{2}}(\mathcal{D}+\gamma\mathbb{I})^{-\frac{1}{2}}\right\|_2\|\boldsymbol{S}\|_2\left\|(\boldsymbol{D}+\gamma\mathbb{I})^{\frac{1}{2}}(\mathcal{D}+\gamma\mathbb{I})^{-\frac{1}{2}}\boldsymbol{z}\right\|_2\right] \\
\leq&\mathbb{E}\left[\max_i\left(1-\sqrt{\frac{(\boldsymbol{D}+\gamma\mathbb{I})_{ii}}{(\mathcal{D}+\gamma\mathbb{I})_{ii}}}\right)\left\|(\boldsymbol{D}+\gamma\mathbb{I})^{\frac{1}{2}}(\mathcal{D}+\gamma\mathbb{I})^{-\frac{1}{2}}\boldsymbol{z}\right\|_2\right] \\
\leq&\left(\underbrace{\mathbb{E}\left[\max_i\left(1-\sqrt{\frac{\boldsymbol{D}_{ii}+\gamma}{\mathcal{D}_{ii}+\gamma}}\right)\right]}_{term1}\underbrace{\mathbb{E}\left[\left\|(\boldsymbol{D}+\gamma\mathbb{I})^{\frac{1}{2}}(\mathcal{D}+\gamma\mathbb{I})^{-\frac{1}{2}}\boldsymbol{z}\right\|^2\right]}_{term2}\right)^{\frac{1}{2}} \qquad (27)
\end{aligned}$$

where (27) follows from applying the Cauchy-Schwarz inequality. Then for (27) *term 2* we get:

$$\begin{aligned}
\mathbb{E}\left[\left\|(\boldsymbol{D}+\gamma\mathbb{I})^{\frac{1}{2}}(\mathcal{D}+\gamma\mathbb{I})^{-\frac{1}{2}}\boldsymbol{z}\right\|^2\right] &= \sum_i \mathbb{E}\left[\frac{\boldsymbol{D}_{ii}+\gamma}{\mathcal{D}_{ii}+\gamma}\boldsymbol{z}_i^2\right] \\
&= \sum_i \underbrace{\frac{\mathbb{E}\left[\boldsymbol{D}_{ii}+\gamma\right]}{\mathcal{D}_{ii}+\gamma}}_{=1} \\
&= n
\end{aligned}$$

(27) *term 1* we again note that for $x \geq 0$, $|1-\sqrt{x}| \leq |1-x|$. Using this we write:

$$\begin{aligned}
\mathbb{E}\left[\max_i\left(1-\sqrt{\frac{\boldsymbol{D}_{ii}+\gamma}{\mathcal{D}_{ii}+\gamma}}\right)^2\right] &\leq \mathbb{E}\left[\max_i\left(1-\frac{\boldsymbol{D}_{ii}+\gamma}{\mathcal{D}_{ii}+\gamma}\right)^2\right] \\
&\leq \frac{1}{s}\ln\left(\exp\left(\mathbb{E}\left[s\max_i\left(1-\frac{\boldsymbol{D}_{ii}+\gamma}{\mathcal{D}_{ii}+\gamma}\right)^2\right]\right)\right) \\
&\leq \frac{1}{s}\ln\left(\mathbb{E}\left[\exp\left(s\max_i\left(1-\frac{\boldsymbol{D}_{ii}+\gamma}{\mathcal{D}_{ii}+\gamma}\right)^2\right)\right]\right) \\
&= \frac{1}{s}\ln\left(\mathbb{E}\left[\max_i\left(\exp s\left(\left(1-\frac{\boldsymbol{D}_{ii}+\gamma}{\mathcal{D}_{ii}+\gamma}\right)^2\right)\right)\right]\right) \\
&\leq \frac{1}{s}\ln\left(\underbrace{\sum_i\mathbb{E}\left[\underbrace{\exp\left(s\underbrace{\left(1-\frac{\boldsymbol{D}_{ii}+\gamma}{\mathcal{D}_{ii}+\gamma}\right)^2}_{\boldsymbol{y}_i}\right)}_{term1}\right]}_{term2}\right)
\end{aligned} \tag{28}$$

Now to further bound (28) we first compute (28), term 1 as:

$$\begin{aligned}
\exp(s\boldsymbol{y}_i) &= 1 + s\boldsymbol{y}_i + \sum_{k\geq 2}\frac{(s\boldsymbol{y}_i)^k}{k!} \\
&= 1 + s\boldsymbol{y}_i + (s\boldsymbol{y}_i)\sum_{k\geq 2}\frac{(s\boldsymbol{y}_i)^{k-1}}{k!} \\
&= 1 + s\boldsymbol{y}_i + (s\boldsymbol{y}_i)\sum_{k\geq 0}\frac{(s\boldsymbol{y}_i)^{k}}{(k+1)k!} \\
&\leq 1 + s\boldsymbol{y}_i + (s\boldsymbol{y}_i)\exp(s\boldsymbol{y}_i) \\
&\leq 1 + (\exp(s)+1)s\boldsymbol{y}_i
\end{aligned}$$

Taking the expectation over the previous line, using linearity of expectation and the expression for $\sum_i \mathbb{E}[\boldsymbol{y}_i]$ from (26) it follows that for (28), term 2 we obtain

$$\begin{aligned}
\sum_i \mathbb{E}\left[\exp(s\boldsymbol{y}_i)\right] &\leq n + (\exp(s)+1)s\sum_i\mathbb{E}\left[\boldsymbol{y}_i\right] \\
&= n + (\exp(s)+1)s\frac{n}{\gamma+(n-1)q}
\end{aligned}$$

Going back to (28):

$$\begin{aligned}
(28) &\leq \frac{1}{s} \ln \left(n + (\exp(s) + 1) s \frac{n}{\gamma + (n-1)q} \right) && \forall s > 0 \\
&\leq \frac{1}{s} \ln \left(n + \exp(2s) \frac{n}{\gamma + (n-1)q} \right) && \text{Note: } s > 0 \Rightarrow \ln s \leq s - 1 \\
& && \Rightarrow (e^s + 1)s \leq e^{2s} \\
&\leq \frac{\ln(n)}{s} + \frac{1}{s} \ln \left(1 + \frac{\exp(2s)}{\gamma + (n-1)q} \right) && \text{Let } e^{2s} \geq \gamma + (n-1)q \\
&\leq \frac{\ln(n)}{s} + \frac{1}{s} \ln \left(\frac{2\exp(2s)}{\gamma + (n-1)q} \right) \\
&\leq \frac{\ln(n)}{s} + 2 + \frac{1}{s} \ln \left(\frac{2}{\gamma + (n-1)q} \right) && \text{Take } s := \gamma + (n-1)q \geq 2 \\
&\leq C \frac{\ln(n)}{\gamma + (n-1)q}
\end{aligned}$$

Finally combining the above results:

$$\begin{aligned}
\mathbb{E}\left[\left\| (\boldsymbol{S} - \overline{\boldsymbol{S}}) \boldsymbol{z} \right\|_2 \right] &\leq \sqrt{\frac{n}{\gamma + (n-1)q}} + \sqrt{n \frac{C \ln(n)}{\gamma + (n-1)q}} \\
&= C \sqrt{\frac{n \ln(n)}{\gamma + (n-1)q}}
\end{aligned}$$

and

$$\mathbb{E}\left[\| (\boldsymbol{S} - \mathcal{S}) \mathcal{X} \|_{2\to\infty} \right] \leq C \sqrt{\frac{n \ln n}{\gamma + (n-1)q}} \, \|\boldsymbol{\mu}\|_\infty$$

This concludes he bound of $\mathbb{E}\left[\| (\boldsymbol{S} - \mathcal{S}) \mathcal{X} \|_{2\to\infty} \right]$. □

Bound $\mathbb{E}\left[\| (\boldsymbol{X} - \boldsymbol{\mathcal{X}}) \boldsymbol{S} \|_{\mathbf{2\to\infty}} \right]$ We first note that

$$\begin{aligned}
\mathbb{E}\left[\| (\boldsymbol{X} - \mathcal{X}) \boldsymbol{S} \|_{2\to\infty} \right] &= \mathbb{E}\left[\max_{j \in [d]} \| \boldsymbol{S} \epsilon_{\cdot j} \|_2 \right] \\
&\leq \left(\mathbb{E}\left[\max_{j \in [d]} \| \boldsymbol{S} \epsilon_{\cdot j} \|_2^2 \right] \right)^{\frac{1}{2}}
\end{aligned}$$

Let $z \sim \mathcal{N}(0, \sigma^2\mathbb{I})$ then

$$\begin{aligned} \|\boldsymbol{Sz}\|_2^2 &= \boldsymbol{z}^\top \boldsymbol{S}^\top \boldsymbol{Sz} \\ &= \boldsymbol{zV\Lambda V}^\top \boldsymbol{z} && \text{Eigendecompsition} \\ &= \sum_{i=1}^{n} \lambda_i \boldsymbol{z}_i'^2 && \text{where } \boldsymbol{V}^\top \boldsymbol{z} = \boldsymbol{z}_i' \sim \mathcal{N}(0, \sigma^2\mathbb{I}) \\ &= \sum_{i=1;\lambda_i>0}^{n} \lambda_i \sigma^2 \boldsymbol{y}_i && \boldsymbol{y}_i, \cdots, \boldsymbol{y}_d \overset{iid}{\sim} \mathcal{X}^2 \end{aligned}$$

Where the first line follows from the eigendecomposition $\boldsymbol{S}^\top \boldsymbol{S} = \boldsymbol{V\Lambda V}^\top$. Therefore $\|\boldsymbol{Sz}\|_2^2$ is distributed as a generalised $\mathcal{X}^2$ with mean $\sigma\,\mathrm{Tr}(\boldsymbol{S}^\top \boldsymbol{S})$ and variance $2\sum \lambda_i \sigma^4 = 2\sigma^4 \left\|\boldsymbol{S}^\top \boldsymbol{S}\right\|_F^2$. Now define

$$\mathrm{MGF}_y(s) = \frac{1}{\exp\left(\frac{1}{2}\sum_{i:\lambda_i>0} \log(1 - 2s\lambda_i)\right)}$$

and consider $s \in \left(0, \frac{1}{2\lambda_{min}}\right)$ where λ_{min} is the smallest non-zero eigenvalue of $\boldsymbol{S}^\top \boldsymbol{S}$.

$$\begin{aligned} \exp\left(s\mathbb{E}\left[\max_j \boldsymbol{y}_j\right]\right) &\leq \mathbb{E}\left[\exp\left(s \max(\boldsymbol{y}_j)\right]\right) \\ &= \mathbb{E}\left[\max \exp\left(s\boldsymbol{y}_j\right]\right) \\ &\leq \sum_j \mathbb{E}\left[\exp\left(s\boldsymbol{y}_j\right)\right] \\ &= d \cdot \mathrm{MGF}_{\boldsymbol{y}}(s) \\ &= d \exp\left(-\frac{1}{2} \sum_{i:\lambda_i>0} \log(1 - 2s\lambda_i)\right) \end{aligned}$$

it follows that

$$\begin{aligned} \mathbb{E}\left[\max_j \boldsymbol{y}_j\right] &\leq \frac{\ln d}{s} - \frac{1}{2s} \sum_{i:\lambda_i>0} \underbrace{\log(1 - 2s\lambda_i)}_{\leq -2s\lambda_i} \\ &\leq \frac{\ln d}{s} + \underbrace{\sum_{i:\lambda_i>0} \lambda_i}_{\mathrm{Tr}(\boldsymbol{S}^\top \boldsymbol{S})} && \because \log(1+x) \leq x \ \forall x > -1 \\ &\leq 2\lambda_{min} \ln d + \mathrm{Tr}(\boldsymbol{S}^\top \boldsymbol{S}) \ \ \because s \in \left(0, \tfrac{1}{2\lambda_{min}}\right) && \text{and min for } s = \tfrac{1}{2\lambda_{min}} \end{aligned}$$

Using $\sigma_{min}(\boldsymbol{S}) \leq \|\boldsymbol{S}\|_2$ and $\|\boldsymbol{S}\|_F \leq k\,\|\boldsymbol{S}\|_2$ we can bound the last line as $\|\boldsymbol{S}\|_2^2\,(k + 2\ln d)$ in the low-rank setting. However since we consider $\boldsymbol{S}$ to be random this is not applicable (also see the remarks in the VC Dimension section). Therefore

$$\begin{aligned}2\lambda_{min}\ln d + \mathrm{Tr}(\boldsymbol{S}^\top \boldsymbol{S}) &= \sigma_{min}^2(\boldsymbol{S})\ln d + \|\boldsymbol{S}\|_F^2\\ &\leq \|\boldsymbol{S}\|_F^2(1+2\ln d)\end{aligned}$$

and taking the square root gives us the final result:

$$\mathbb{E}\left[\|(\boldsymbol{X}-\mathcal{X})\,\boldsymbol{S}\|_{2\to\infty}\right] \leq \sigma\,\|\boldsymbol{S}\|_F\sqrt{1+2\ln d}$$

Bound $\mathbb{E}\left[\|\boldsymbol{S}\|_F^2\right]$.

Case 1: Self loop.

We first note that $\|\boldsymbol{S}\|_F^2 = n +$ *number of edges* and therefore:

$$\begin{aligned}\mathbb{E}\left[\|\boldsymbol{S}\|_F^2\right] &\leq n + n^2 p\\ &= (1+o(1))n^2 p\end{aligned}$$

Therefore

$$\mathbb{E}\left[\|(\boldsymbol{X}-\mathcal{X})\,\boldsymbol{S}\|_{2\to\infty}^2\right] \leq (1+o(1))n^2 p\sigma^2(1+2\ln d)$$

Case 2: Degree normalized.

Note that we here overload the notation d such that we define the degree for node i as d_i and similar d_{min} is the minimum degree.

$$
\begin{aligned}
\mathbb{E}\left[\|\boldsymbol{S}\|_F^2\right] =& \mathbb{E}\left[\|\boldsymbol{S}\|_F^2 \middle| \left\{d_{min} > np - \sqrt{4cnp\ln n}\right\}\right] \mathbb{P}\left(d_{min} > np - \sqrt{4cnp\ln n}\right) \\
&+ \mathbb{E}\left[\|\boldsymbol{S}\|_F^2 \middle| \left\{d_{min} < np - \sqrt{4cnp\ln n}\right\}\right] \mathbb{P}\left(d_{min} < np - \sqrt{4cnp\ln n}\right) \\
\leq& \mathbb{E}\left[\|\boldsymbol{S}\|_F^2 \middle| \left\{d_{min} > np - \sqrt{4cnp\ln n}\right\}\right] \mathbb{P}\left(d_{min} > np - \sqrt{4cnp\ln n}\right) + \underbrace{n^2 \frac{1}{n^c}}_{=o(1)} \\
\leq& \sum_{i,j} \frac{\boldsymbol{A}_{ij} + \mathbb{I}\{i = j\}}{(d_i + 1)(d_j + 1)} \\
\leq& \frac{1}{d_{min} + 1} \sum_i \underbrace{\frac{\sum_j \boldsymbol{A}_{ij} + \mathbb{I}\{i = j\}}{d_i + 1}}_{=1} \\
\leq& \frac{n}{nq + 1 - \sqrt{4cnp\ln n}} \\
=& (1 + o(1))\frac{1}{q}
\end{aligned}
$$

Therefore

$$
\mathbb{E}\left[\|(\boldsymbol{X} - \mathcal{X})\,\boldsymbol{S}\|_{2\to\infty}^2\right] \leq (1 + o(1))\frac{\sigma^2(1 + 2\ln d)}{q}
$$

This concludes the bound of $\mathbb{E}\left[\|(\boldsymbol{X} - \mathcal{X})\,\boldsymbol{S}\|_{2\to\infty}^2\right]$. □

Bound $\mathbb{E}\left[\|\boldsymbol{S}\|_{\boldsymbol{\infty}}^{\boldsymbol{k}}\right]$. In general we can note that $\|\mathcal{S}\|_\infty^k = \max_{1\leq i\leq n}\left(\sum_{j=1}^n \mathcal{S}_{ij}\right)^k$

Case 1: Self loop.

We first define the degree for node i as

$$
d_i \sim \text{Bin}\left(\frac{n}{2} - 1, p\right) + \text{Bin}\left(\frac{n}{2}, q\right)
$$

then $\|\mathcal{S}\|_\infty = \max_{1\le i\le n}\left(\sum_{j=1}^n \mathcal{S}_{ij}\right) = 1 + \max_i d_i$ and assume $p > \frac{lnn}{n}$ and let $t = \sqrt{4np\ln n}$

$$\begin{aligned}\mathbb{P}\left(d_i - \mathbb{E}\left[d_i\right] > t\right) &\le \exp\left(\frac{-\frac{t^2}{2}}{np + \frac{t}{3}}\right) && \text{Bernstein inequality}\\ &\le \exp\left(\frac{-4cnp\ln n}{4np}\right)\\ &= \frac{1}{n}c\end{aligned}$$

and therefore

$$\begin{aligned}\mathbb{P}\left(\max_i d_i \ge np + \sqrt{4cnp\ln n}\right) &\le \frac{1}{n^{c-1}}\\ \mathbb{P}\left((1+\max_i d_i)^k \ge (1 + np + \sqrt{4cnp\ln n})^k\right) &\le \frac{1}{n}c\end{aligned}$$

and

$$\begin{aligned}\mathbb{E}\left[(1+\max_i d_i)^k\right] &\le (1 + np + \sqrt{4cnp\ln n})^k + \frac{1}{n^{c-i}}n^k\\ &= (1 + np + \sqrt{4cnp\ln n})^k + n^{k+1-c}\end{aligned}$$

For large n and $p \gg \frac{(lnn)^2}{n}$ take $c = \ln n$:

$$\mathbb{E}\left[\|\boldsymbol{S}\|_\infty^k\right] \le ((1+o(1))np)^k$$

Case 2: Degree normalized.

$$\begin{aligned}\|\boldsymbol{S}\|_\infty &= \max_i \sum_j \boldsymbol{S}_{ij}\\ &= \max_i \sum_j \frac{\boldsymbol{A}_{ij}}{\sqrt{d_i+1}\sqrt{d_j+1}}\\ &\le \max_i \frac{1}{\sqrt{d_{min}+1}}\frac{\sum_j \boldsymbol{A}_{ij}}{\sqrt{d_i+1}}\\ &= \max_i \sqrt{\frac{d_i+1}{d_{min}+1}}\\ &\le \sqrt{\frac{d_min+1}{d_{min}+1}}\end{aligned}$$

Similar to above we can now note that:

$$\begin{aligned}\mathbb{P}\left(\max_i d_i \ge np + \sqrt{4cnp\ln n}\right) &\le \frac{1}{n^c}\\ \mathbb{P}\left(\max_i d_i \le np + \sqrt{4cnp\ln n}\right) &\le \frac{1}{n^c}\end{aligned}$$

and it follows

$$\mathbb{P}\left(\sqrt{\frac{d_{max}+1}{d_{min}+1}} \geq \frac{np+\sqrt{4cnp\ln n}+1}{np-\sqrt{4cnp\ln n}+1}\right) \leq \frac{2}{n^c}$$

For large n and $p,q \gg \frac{(lnn)^2}{n}$:

$$\begin{aligned}\mathbb{E}\left[\|\boldsymbol{S}\|_\infty^k\right] &\leq \mathbb{E}\left[\left(\frac{d_{max}+1}{d_{min}+1}\right)^{\frac{k}{2}}\right] \\ &= \left((1+o(1))\frac{p}{q}\right)^{\frac{k}{2}}\end{aligned}$$

This concludes the bound of $\mathbb{E}\left[\|\boldsymbol{S}\|_\infty^k\right]$. □

Bound $\|\boldsymbol{\mathcal{S}\mathcal{X}}\|_{2\to\infty}$.

Case 1: Self loop.

$$\begin{aligned}\mathcal{S}\mathcal{X} &= (1-p)\boldsymbol{z}\boldsymbol{\mu}^\top - \frac{p-q}{2}\boldsymbol{y}\boldsymbol{y}^\top\boldsymbol{z}\boldsymbol{\mu}^\top \\ &= \left((1-p)\boldsymbol{z} - \left(\frac{p-q}{2}\boldsymbol{y}^\top\boldsymbol{z}\right)\boldsymbol{y}\right)\boldsymbol{\mu}^\top\end{aligned}$$

and

$$(\mathcal{S}\mathcal{X})_{ij} = \left((1-p)\boldsymbol{z}_i - \underbrace{\left(\frac{p-q}{2}\boldsymbol{y}^\top\boldsymbol{z}\right)}_{\triangleq\delta}\boldsymbol{y}_i\right)\boldsymbol{\mu}_j$$

Now using this to compute the two-infinity norm:

$$\begin{aligned}\|\mathcal{S}\mathcal{X}\|_{2\to\infty} &= \|\boldsymbol{\mu}\|_\infty\sqrt{\sum_i((1-p)\boldsymbol{z}_i-\delta\boldsymbol{y}_i)^2} \\ &= \|\boldsymbol{\mu}\|_\infty\sqrt{\sum_i(1-p)^2+\delta^2-2\delta(1-p)\boldsymbol{y}_i\boldsymbol{z}_i} \\ &= \|\boldsymbol{\mu}\|_\infty\left(n(1-p)^2+n(\boldsymbol{y}^\top\boldsymbol{z})^2\left(\frac{p-q}{2}\right)^2-2(\boldsymbol{y}^\top\boldsymbol{z})^2\frac{p-q}{2}(1-p)\right) \\ &= (1+o(1))\,\|\boldsymbol{\mu}\|_\infty\, n\left(1+\left(\frac{p-q}{2}\right)^2(\boldsymbol{y}^\top\boldsymbol{z})^2\right)\end{aligned}$$

Case 2: Degree normalized.

We note that the expected degree is $(1 + o(1))n\frac{p+q}{2}$ and therefore similar to above we obtain

$$\|\mathcal{S}\mathcal{X}\|_{2\to\infty} = (1 + o(1))\,\|\boldsymbol{\mu}\|_\infty \frac{\left(1 + \left(\frac{p-q}{2}\right)^2 (\boldsymbol{y}^\top \boldsymbol{z})^2\right)}{\left(\frac{p+q}{2}\right)}.$$

This concludes the bound of $\|\mathcal{S}\mathcal{X}\|_{2\to\infty}$. □

G Experimental Details

G.1 Data

SBM. For the SBM experiments we follow the description in the main paper: assume that the node features are sampled latent true classes, given a $\boldsymbol{z} = (z_1, \ldots, z_n) \in \{\pm 1\}^n$. The node features are sampled from a Gaussian mixture model (GMM), that is, feature for node-i is sampled as $\boldsymbol{x}_i \sim \mathcal{N}(z_i \boldsymbol{\mu}, \sigma^2 \mathbb{I})$ for some $\boldsymbol{\mu} \in \mathbb{R}^d$ and $\sigma \in (0, \infty)$. We express this in terms of $\boldsymbol{X}$ as

$$\boldsymbol{X} = \mathcal{X} + \boldsymbol{\epsilon} \ \in \ \mathbb{R}^{n \times d}, \qquad \text{where } \mathcal{X} = \boldsymbol{z}\boldsymbol{\mu}^\top \ \text{ and } \boldsymbol{\epsilon} = (\epsilon_{ij})_{i \in [n], j \in [d]} \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2).$$

We refer to above as $\boldsymbol{X} \sim$ 2GMM. On the other hand, we assume that graph has two latent communities, characterised by $\boldsymbol{y} \in \{\pm 1\}^n$. The graph is generated from a stochastic block model with two classes (2SBM), where edges (i, j) are added independently with probability $p \in (0, 1]$ if $y_i = y_j$, and with probability $q < [0, p)$ if $y_i \neq y_j$. In other words, we define the random adjacency $\boldsymbol{A} \sim$ 2SBM as a symmetric binary matrix with $\boldsymbol{A}_{ii} = 0$, and $(\boldsymbol{A}_{ij})_{i<j}$ indenpendent such that

$$\boldsymbol{A}_{ij} \sim \text{Bernoulli}(\mathcal{A}_{ij}), \qquad \text{where } \mathcal{A} = \frac{p+q}{2} \mathbf{1}\mathbf{1}^\top + \frac{p-q}{2} \boldsymbol{y}\boldsymbol{y}^\top - p\mathbb{I}.$$

The choice of two different latent classes $\boldsymbol{z}, \boldsymbol{y} \in \{\pm 1\}^n$ allows study of the case where the graph and feature information of do not align completely.

Therefore for to characterise the model we need to define: $p, q, n, \boldsymbol{z}, \boldsymbol{y}, \boldsymbol{\mu}, \sigma$

Cora. For the real world experiments we use the cora dataset Rossi et al. (2015)[7]. The dataset consists of 2708 machine learning papers and is split into seven classes: *Case_ Based, Genetic_ Algorithms, Neural_ Networks, Probabilistic_ Methods, Reinforcement_ Learning, Rule_ Learning, Theory.* The features are a bag of words of size 1433.

G.2 Experiments Section 3.2

SBM Setup and Data. We consider the synthetic data to be generated as defined in (6) and (7). We sample the SBM with the following parameters as default: $n = 500, d = 100, p = 0.2, q = 0.01, \Gamma = n, m = 100, u = 400$. $\boldsymbol{\mu}$ is sampled uniformly. The GNN is by default a one layer model $K = 1$ with hidden layer size $d_1 = 16$, ReLu activation, $\phi(\cdot) = \text{ReLU}(\cdot)$ and squared loss. Plotted is the error over the displayed change of parameters for epochs[8] between 50 and 1000 (over 50 intervals). We plot the results averaged over five random initialisation.

[7] Using the import from `https://github.com/tkipf/pygcn/tree/master/data/cora`

[8] A consideration of different epochs is important as the presented bounds do not take the optimization explicitly into consideration. As stated previously a future way to do so could be by analysing the behaviour of the the bounds on the parameters during optimization.

Change alignment. We consider the SBM[9] setting as defined above while now varying $\mathit{\Gamma} \in (0, n)$ over 10 steps and for easier readability plot $\frac{\mathit{\Gamma}}{n}$. The GNN is optimized using SGD with learning rate 0.001.

Change graph size. We consider the SBM setting as defined above with setting again $\frac{\mathit{\Gamma}}{n} = 1$ while now varying the graph size $n \in (200, 2000)$ over 10 steps while adjusting $\frac{m}{n}$ accordingly. The GNN is optimized using SGD with learning rate 0.01.

Change number of marked points. We consider the SBM setting as defined above with $\frac{\mathit{\Gamma}}{n} = 0.7, p = 0.2, q = 0.15$ while now varying the number of observe points such that $\frac{m}{n} \in (0.01, 0.05)$ over 10 steps. The GNN is optimized using SGD with learning rate 0.2.

Plot theoretical bound. Recall that for plotting the theoretical bound we can only plot the trend of the bound as the absolute value is out of the $(0, 1)$ range. This problem is inherent to the bound given in El-Yaniv et al. (2009) that we base our TRC bounds on, as the slack terms can already exceeds 1 and therefore further research on general TRC generalisation gaps is necessary to characterise the absolute gap between theory and experiments. More specifically we scale *SBM, change alignment* and *SBM, change graph size* by a factor of 25 and *SBM, change number of marked points* by a factor of 30. Again as noted in the main paper we fix the bounds on the on the learnable parameters for plotting the theoretical bounds. From samples we observe that $\beta, \omega \approx 0.1$ and therefore consider this for the plots. A more detailed analysis of this will be necessary in future research to investigate how the change of those bounds changes the generalisation error bound.

Cora Setup and Data. We now consider the *Cora* dataset with $n = 2708$ and $\frac{m}{n} = 0.1$. The GNN follows the setup of the SBM with the difference that we now consider a multi-class problem. Therefore a *negative log likelihood loss* is considered. In addition we consider the Adam optimizer Kingma et al. (2015) with learning rate 0.01.

Change alignment. We simulate a change in the feature structure by adding noise to the feature vector as $\boldsymbol{X} + \epsilon$ where $\epsilon_{i\cdot}$ is *i.i.d.* distributed $\mathcal{N}(0, \sigma_{\text{Feat}}^2 \mathbb{I})$ and again observe a similar behaviour to the SBM. We vary $\sigma_{\text{Feat}} \in (0, 0.1)$ over 10 steps.

Cora, change graph size. To change the graph size we sample 10 subgraphs of size $n \in (1354, 2708)$.

Change number of marked points. For varying the number of observe points we consider $\frac{m}{n} \in (0.05, 0.3)$ over 10 steps.

[9] Remark on change in training and SBM setting: Since we are interested in upper bounds we observe that under some settings the trends are more clear then in others. For example for some learning rate the change might be less obvious then for the reported one.

G.3 Experiments Section B.2 (Residual connections)

Setup and Data. We consider the same general setup as above (section G.2). We now change the parameter K. For implementing residual connections we slightly deviate from (11) by considering the residual connection to be to the first layer instead of the features directly. This change follows Chen et al. (2020) where the residual connection was proposed as otherwise the size of the hidden layer would be fixed to n. For the experiments we consider $d_i = 16 \ \forall i \in [K]$.

Change depth. For both datasets we now changed the depth for $K \in [4]$ and two different residual connections with $\alpha \in \{0.2, 0.5\}$.

G.4 Implementation

For the implementation of the GNN we use official code of Kipf et al. (2017)[10] as a foundation that is provided under an *MIT License*.

Experiments are ran on a MacBook Pro (16-inch, 2019), processor 2,3 GHz 8-Core Intel Core i9, memory 32 GB 2667 MHz DDR4.

[10] https://github.com/tkipf/pyGNN