Graph Semi-Supervised Learning for Point Classification on Data Manifolds

Caio F. Deberaldini Netto
Applied Mathematics and Statistics Department
Johns Hopkins University
Baltimore, MD 21218
cnetto1@jhu.edu

Zhiyang Wang
Halıcıoğlu Data Science Institute
University of California San Diego
San Diego, CA 92093
zhw135@ucsd.edu

Luana Ruiz
Applied Mathematics and Statistics Department
Johns Hopkins University
Baltimore, MD 21218
lrubini1@jhu.edu

Abstract

We propose a graph semi-supervised learning framework for classification tasks on data manifolds. Motivated by the manifold hypothesis, we model data as points sampled from a low-dimensional manifold $\mathcal{M} \subset \mathbb{R}^F$. The manifold is approximated in an unsupervised manner using a variational autoencoder (VAE), where the trained encoder maps data to embeddings that represent their coordinates in $\mathbb{R}^F$. A geometric graph is constructed with Gaussian-weighted edges inversely proportional to distances in the embedding space, transforming the point classification problem into a semi-supervised node classification task on the graph. This task is solved using a graph neural network (GNN). Our main contribution is a theoretical analysis of the statistical generalization properties of this data-to-manifold-to-graph pipeline. We show that, under uniform sampling from $\mathcal{M}$, the generalization gap of the semi-supervised task diminishes with increasing graph size, up to the GNN training error. Leveraging a training procedure that resamples a slightly larger graph at regular intervals during training, we then show that the generalization gap can be reduced even further, vanishing asymptotically. Finally, we validate our findings with numerical experiments on image classification benchmarks, demonstrating the empirical effectiveness of our approach.

1 Introduction

Graph neural networks (GNNs) have achieved strong results on graph-structured data in diverse areas, including molecular biology [31], network science [30], and natural language processing [60, 62]. Among their many applications, GNNs are most effective in semi-supervised scenarios, where a single graph is given with node features, but only a subset of nodes are labeled, and the goal is to use the labeled subset to infer labels for the rest.

GNNs' performance in this setting stems from properties such as permutation equivariance and stability, allowing patterns learned on certain substructures to generalize across the graph, even under small deformations [59, 37, 45, 3, 28, 27]. Of special significance is *transferability*, the ability to retain predictive capacity across different graphs from the same family, which is crucial when supervision is partial [54, 34, 41]

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: New Perspectives in Advancing Graph Machine Learning.

Transferability has been especially studied in *geometric graphs*, where nodes represent manifold samples and edges denote proximity. GNNs have shown strong transfer across such graphs in tasks like point cloud classification and planning [13, 66, 17]. Furthermore, many high-dimensional datasets are believed to concentrate around low-dimensional manifolds, per the manifold hypothesis [7, 74, 64]. Given this, a key question arises:

*Can we exploit the **intrinsic geometry of data**, combined with **GNN transferability**, to boost prediction on high-dimensional data?*

This is the focus of our work. For image classification, we construct a geometric graph from data and apply graph-based semi-supervised learning, leveraging manifold structures so GNNs can better capture data distributions and improve predictive accuracy.

2 Related work

Graph neural networks. GNNs are deep convolutional architectures designed for graphs [61]. Each layer performs a graph convolution, an extension generalizing classical convolutions for graphs, followed by a node-wise nonlinearity [21, 59, 37, 20, 14]. Key theoretical properties such as invariance, stability, and locality stem from graph convolutions [58, 28], which, like image convolutions, operate by shift-and-sum via local node-to-neighbor exchanges [63, 29, 21]. Many GNNs are also commonly described through local aggregation functions, equivalent to first-order convolutional filters [72, 33, 31, 59].

GNN transferability. Transferability is often studied using graphons, which model limits of dense graphs [44, 1, 23, 11, 12, 43]. Recent work established graphon convolutions and their convergence, leading to asymptotic and non-asymptotic results for transferring GNNs between graphs sampled from the same graphon with bounded error [53, 54, 56, 55, 57, 16].

However, graphons lack explicit geometry, unless additional structure is imposed on the node sample space. This gap is partially addressed by considering graphs sampled from general topological spaces [13, 41], though these works treat the sampling operator generically, without specifying topology. In contrast, we focus on graphs sampled from a submanifold embedded in Euclidean space: by assigning the uniform measure to the manifold and sampling points uniformly, we construct graphs based on these points' distances in ambient space.

The manifold hypothesis. The manifold hypothesis posits that high-dimensional data lies near lower-dimensional manifolds [7], which has motivated advances in sample complexity theory [46, 25], dimensionality reduction [4, 18, 52, 19, 2, 24], and manifold-regularized learning [5, 15, 47]. Unlike classical approaches, we use learned embeddings as graph signals based on the manifold, allowing GNNs to fully exploit geometric structure for improved generalization.

3 Background

3.1 Manifold hypothesis and geometric graph approximation

Under the manifold hypothesis, high-dimensional data can be represented as points u of a d-dimensional submanifold $\mathcal{M}$ embedded in some Euclidean space $\mathbb{R}^F$, i.e., $u \in \mathcal{M}$ and $\mathcal{M} \subset \mathbb{R}^F$. Since $\mathcal{M}$ is an embedded submanifold, u can be expressed in ambient space coordinates via the map $\mathcal{X} : \mathcal{M} \to \mathbb{R}^F$.

In general, we do not know the manifold $\mathcal{M}$, so in order to estimate it we first obtain embeddings $x \in \mathbb{R}^F$ from the data—through some dimensionality reduction (e.g., PCA) or learning (e.g., self-supervised) technique—and assume $x = \mathcal{X}(u)$. The manifold can then be approximated using a geometric graph. Explicitly, let x_i and x_j denote the embeddings associated with samples i and j. These samples are seen as nodes of an undirected graph G, and they are connected by an edge with weight

$$w_{ij} = \begin{cases} \exp -\frac{\|x_i - x_j\|^2}{2\sigma^2} & \text{if } i \neq j \\ 0 & \text{if } i = j. \end{cases} \tag{1}$$

Given n samples, we write the graph adjacency matrix $A_n \in \mathbb{R}^{n\times n}$ entry-wise as $[A_n]_{ij} = w_{ij}$, and the graph Laplacian as $L_n = \mathrm{diag}(A_n\mathbf{1}) - A_n$. As we discuss in the following L_n provides arbitrarily good approximations of the Laplace-Beltrami (LB) operator of $\mathcal{M}$ as $n \to \infty$.

3.2 The Laplace-Beltrami operator and graph Laplacian convergence

Submanifolds of Euclidean space are locally Euclidean, meaning that in a neighborhood of any given point $u \in \mathcal{M}$, the manifold can be approximated by an Euclidean space via its tangent space.

The tangent space of $\mathcal{M}$ at a point $u \in \mathcal{M}$ consists of all tangent vectors at u. A vector $v \in \mathbb{R}^F$ is considered a tangent vector of $\mathcal{M}$ at u if there exists a smooth curve γ such that $\gamma(0) = u$ and $\dot\gamma(0) = v$. That is, tangent vectors correspond to the derivatives of curves $\gamma : \mathbb{R} \to \mathcal{M}$. The tangent space at u, denoted $T_u\mathcal{M}$, is therefore defined as $T_u\mathcal{M} = \{\dot\gamma(0) \mid \text{smooth } \gamma : \mathbb{R} \to \mathcal{M}\,, \gamma(0) = u\}$ [51]. The union of all tangent spaces across the manifold $\mathcal{M}$ forms the tangent bundle $T\mathcal{M}$.

With this notion of tangent space, we can define gradients of functions defined on $\mathcal{M}$. Consider for instance the map $\mathcal{X} : \mathcal{M} \to \mathbb{R}^F$, which satisfies $\mathcal{X} \in C^\infty(\mathcal{M})$. The gradient $\nabla\mathcal{X} \in T\mathcal{M}$ is a vector field satisfying $\langle \nabla\mathcal{X}(u), v\rangle = \frac{\mathrm{d}}{\mathrm{d}t}\big|_{t=0}(\mathcal{X}\circ\gamma)(t)$ for any tangent vector $v \in T_u\mathcal{M}$ and any smooth curve γ such that $\gamma(0) = u$ and $\dot\gamma(0) = v$ [49]. In the opposite direction, given a smooth vector field $V \in T\mathcal{M}$ and an orthonormal basis $e_1, \dots, e_D$ of $T_u\mathcal{M}$, the divergence $\nabla\cdot V \in \mathcal{C}^\infty(\mathcal{M})$ is defined as $\nabla\cdot V = \sum_{i=1}^D \langle \partial_i V, e_i\rangle$.

By composing the gradient and divergence operators, we obtain the Laplace-Beltrami (LB) operator $\mathcal{L} : \mathcal{C}^\infty(\mathcal{M}) \to \mathcal{C}^\infty(\mathcal{M})$, given by [8]

$$\mathcal{L}\mathcal{X} = -\nabla\cdot(\nabla\mathcal{X})\,. \tag{2}$$

When $\mathcal{M}$ is compact, the operator $\mathcal{L}$ has a discrete, real, and positive spectrum, with eigenvalues λ_i and eigenfunctions ϕ_i, $i = 1, 2, \dots$ (arranged in increasing order of eigenvalues w.l.o.g.).

Convergence of L_n to $\mathcal{L}$. To relate L_n with the Laplace-Beltrami operator $\mathcal{L}$ of $\mathcal{M}$, one can define the continuous extension $\mathcal{L}_n$ of L_n operating on $X \in C^\infty(\mathcal{M})$ as [6]

$$\mathcal{L}_n\mathcal{X}(u) = \mathcal{X}(u)\frac{1}{n}\sum_{i=1}^n e^{-\frac{\|u-u_i\|^2}{2\sigma_n^2}} - \frac{1}{n}\sum_{i=1}^n \mathcal{X}(u_i)e^{-\frac{\|u-u_i\|^2}{2\sigma_n^2}}\,. \tag{3}$$

By carefully choosing parameters $\{\sigma_n\}$, it can be shown that, for $\mathcal{X} \in C^\infty(\mathcal{M})$,

$$\lim_{n\to\infty} \frac{1}{\sigma_n^{2m+2}}\mathcal{L}_n\mathcal{X}(u) = C_{\mathcal{M}}\mathcal{L}\mathcal{X}(u) \tag{4}$$

where $C_{\mathcal{M}}$ is a constant independent of n. Explicitly, the Laplacian of geometric graphs constructed from embeddings x_i as in (1) (which are equal to $\mathcal{X}(u_i)$) converges point-wise to the LB operator of the underlying manifold.

3.3 Graph semi-supervised learning

Let $G = (\mathcal{V}, \mathcal{E})$, $|\mathcal{V}| = n$, be a graph with vertex set $\mathcal{V}$ and edge set $\mathcal{E} \subseteq \mathcal{V}\times\mathcal{V}$. Let $X \in \mathbb{R}^{n\times F}$ be node attributes or features associated with the nodes of G; i.e., each node $i \in \mathcal{V}$ is associated with a F-dimensional signal. Suppose we want to use the information in X to assign each node to one of C classes represented by a label vector $y \in \{1, \dots, C\}^n$.

The graph semi-supervised approach to this task consists of sampling a training node subset $\mathcal{T} \subset \mathcal{V}$ and solving the following optimization problem:

$$\min_{h\in\mathcal{H}} R_{\mathcal{T}}(h) = \min_{h\in\mathcal{H}} l(y, h(X, G); \mathcal{T}) := \min_{h\in\mathcal{H}} \tilde l(M_{\mathcal{T}}y, M_{\mathcal{T}}h(X, G)) \tag{5}$$

where $\mathcal{H}$ is a hypothesis class, $\tilde l$ is a loss function (e.g., the 2-norm), and $M_{\mathcal{T}} \in \{0,1\}^{|\mathcal{T}|\times n}$ is a matrix acting as the training mask, i.e., each row has *exactly* one non-zero entry equals, and each column has *at most* one non-zero entry. We call l the semi-supervised loss. Note that, though the loss is only calculated at nodes $i \in \mathcal{T}$, the signal information X across all the nodes in G is used to compute $h(X, G)$.

Ultimately, we want h to generalize well to the unseen nodes $\mathcal{V}\setminus\mathcal{T}$. This ability is measured by the generalization gap

$$GA(h) = |R_{\mathcal{V}\setminus\mathcal{T}}(h) - R_{\mathcal{T}}(h)|. \tag{6}$$

3.4 Graph Neural Networks (GNNs) and GNN convergence

GNNs are neural network (NN) architectures tailored to graphs. They have multiple layers, each consisting of a linear map followed by a nonlinear activation function, and each operation is adapted to respect the sparsity pattern of the graph. In practice, this restriction is met by parametrizing the linear map of the NN layer by a graph matrix representation, typically the adjacency matrix or Laplacian. Here, we consider the graph Laplacian $L \in \mathbb{R}^{n \times n}$. The ℓth GNN layer is defined as [59]

$$X_\ell = \rho(h(X_{\ell-1}, L)) = \rho\left(\sum_{k=0}^{K-1} L^k X_{\ell-1} W_{\ell k} \right) \tag{7}$$

where $X_\ell \in \mathbb{R}^{n \times F_\ell}$ and $X_{\ell-1} \in \mathbb{R}^{n \times F_{\ell-1}}$ are the *embeddings* at layers ℓ and $\ell - 1$, and $W_{\ell k} \in \mathbb{R}^{F_{\ell-1} \times F_\ell}$ are learnable parameters. The function $\rho : \mathbb{R} \to \mathbb{R}$ is a nonlinear function such as the ReLU or sigmoid, which acts independently on each entry as $[\rho(X)]_{ij} = \rho([X]_{ij})$.

For an $\mathscr{L}$-layer GNN, the GNN output is $Y = X_{\mathscr{L}}$ and, given input data X, $X_0 = X$. For a more compact description, we will represent this GNN as a map $Y = \Phi_{\mathcal{W}}(X, L)$ parametrized by the learnable weights $\mathcal{W} = \{W_{\ell k}\}_{\ell,k}$ at all layers.

Convergence to MNNs. A Manifold Neural Network (MNN) layer is defined pointwise at $u \in \mathcal{M}$ as [67, 68]

$$\mathcal{X}_\ell(u) = \rho\left(\sum_{k=0}^{K-1} \left(e^{-k\mathcal{L}} \mathcal{X}_{\ell-1}\right)(u) W_{\ell k} \right) \tag{8}$$

with $\mathcal{X}_\ell : \mathcal{M} \to \mathbb{R}^{F_\ell}$, $W_{\ell k} \in \mathbb{R}^{F_{\ell-1} \times F_\ell}$, and ρ nonlinear and entry-wise. Once again for compactness, given input $\mathcal{X}_0 = \mathcal{X}$ we represent the whole MNN as a map $\mathcal{Y} = \Phi_{\mathcal{W}}(\mathcal{X}, \mathcal{L})$.

The following result motivates seeing point classification on manifolds as graph semi-supervised learning, and is the cornerstone of the theoretical generalization results in the next section.

Proposition 3.1 ([68], simplified)**.** *Let $\Phi_{\mathcal{W}}$ be an MNN on the d-dimensional manifold $\mathcal{M}$. Let $\{u_1, \dots, u_n\}$ be a set of points sampled uniformly from $\mathcal{M}$ and L_n the corresponding geometric graph Laplacian. Define the map $\mathcal{P}_n : \mathcal{X} \mapsto X_n$:*

$$[X_n]_{ij} = [(\mathcal{P}_n \mathcal{X})(u_i)]_j = [\mathcal{X}(u_i)]_j. \tag{9}$$

Suppose Assumptions B.1–B.3 (stated in Section 4) hold. Then, with probability at least $1 - \delta$,

$$\|\Phi_{\mathcal{W}}(\mathcal{P}_n \mathcal{X}, L_n) - \mathcal{P}_n \Phi_{\mathcal{W}}(\mathcal{X}, \mathcal{L})\| = \mathcal{O}\left(\sqrt[d+4]{\frac{\log 1/\delta}{n}} \right). \tag{10}$$

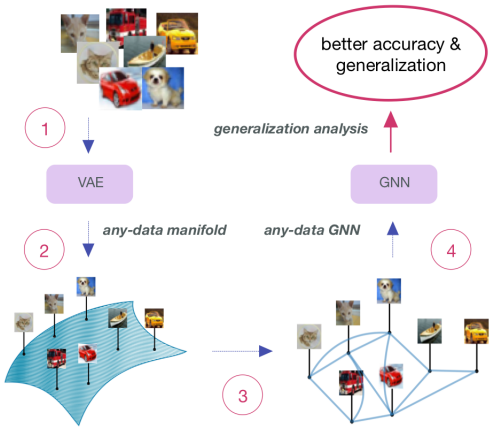
In words, on geometric graphs sampled from a manifold, a GNN with weights $\mathcal{W}$ converges to an MNN with the same set of weights.

4 Classification as graph semi-supervised learning

Consider a standard classification task in which the goal is to assign data $\mathrm{X} \in \mathcal{S}_\mathrm{X}$ (the sample space $\mathcal{S}_\mathrm{X}$ is arbitrary) to one of C classes using labels $\mathrm{y} \in \{1, \dots, C\}$. Given labeled data $\{\mathrm{X_m}, \mathrm{y}_m\}_{m=1}^M$, the classical supervised learning approach consists of selecting a training set $\mathcal{T} \subset \{1, \dots, M\}$; minimizing some loss over $\mathcal{T}$; and computing the classification accuracy on the test set $\{1, \dots, M\} \backslash \mathcal{T}$ to evaluate the ability of the model to generalize.

Leveraging the manifold hypothesis, this problem can be parametrized in a different way. The data X_m are high-dimensional feature vectors, but under the manifold hypothesis, they admit lower-dimensional representations as points $u_m \in \mathcal{M}$ with $\mathcal{M}$ a d-dimensional embedded submanifold of $\mathbb{R}^F$. Suppose we know the map $\psi : \mathcal{S}_\mathrm{X} \to \mathcal{M}$ that achieves such lower-dimensional representations, and also the map $\mathcal{X} : \mathcal{M} \to \mathbb{R}^F$ allowing to write $u \in \mathcal{M}$ in ambient space coordinates as $\mathcal{X}(u) \in \mathbb{R}^F$. Then we can represent $\mathrm{X_m} \in \mathcal{S}_\mathrm{X}$ as $x_m = \mathcal{X}(\psi(\mathrm{X_m})) \in \mathbb{R}^F$.

As discussed in Section 3.1, the embeddings x_m, when learned, can be used to approximate the manifold $\mathcal{M}$ via a geometric graph G where each sample m is a node and each edge has weight $w_{mm'} = \exp\left(-\|x_m - x_{m'}\|^2 / 2\sigma^2\right)$ for $m \neq m'$ [cf. (1)]. Here, we will instead see the graph G as the support of the graph semi-supervised learning problem from Section 3.3 parametrized by a GNN.



(a)

Setup

- The manifold $\mathcal{M}$ is a d-dimensional embedded submanifold of $\mathbb{R}^F$.
- We randomly sample n points from $\mathcal{M}$, forming the n-node geometric graph G_n with Laplacian L_n.
- The loss $\tilde{l}$ in (5) is the mean ℓ_2 norm.
- Data $X_n \in \mathbb{R}^{n\times F}$ is defined as in (11), with rows $[X_n]_{i:} \in \mathbb{R}^F$.
- GNN sees all of X_n at training, but the loss is computed only on a training subset $\mathcal{T}$ of size p.
- The test set is $\{1,\dots,n\}\setminus\mathcal{T}$, with size q, so $p+q=n$.

(b)

Figure 1: (a) Framework schematic. We start by constructing VAE embeddings (1), computing their pairwise distances to form manifolds (2), and sampling graphs from the manifolds (3). GNNs are trained on these graphs to leverage geometric information for image classification (4). (b) Setup for Theorems 4.2–4.4 and Corollary 4.5.

Specifically, on the graph G define the node attribute matrix $X \in \mathbb{R}^{n\times F}$, where

$$[X]_{i:} = x_i, \tag{11}$$

i.e., row i stores the embedding vector corresponding to node i. Define also the label vector $y \in \{1,\dots,C\}^n$ where $[y]_m = \mathrm{y}_m$. The goal is to solve the minimization problem in (5) over hypothesis class $\mathcal{H} = \{\Phi_{\mathcal{W}}(X,L)$ s.t. $\mathcal{W} = \{W_{\ell k}\}_{\ell,k}, W_{\ell k} \in \mathbb{R}^{F_{\ell-1}\times F_\ell}\}$ where $\Phi_{\mathcal{W}}$ is the GNN composed by layers (7) and L is the graph Laplacian.

4.1 Generalization

The rationale for reformulating standard point classification as semi-supervised learning on a graph is to exploit the geometry in the data to improve predictive performance. We first demonstrate this theoretically by showing that the generalization gap of graph semi-supervised learning on geometric graphs sampled from a manifold decreases asymptotically with the graph size. Due to space constraints, we point to Figure 1b (Appendix **??**) for the setup definition.

Lemma 4.1. *Suppose Assumptions B.1–B.3 hold. With probability at least $1-\delta$, for any GNN $\Phi_{\mathcal{W}}$ as in **Setup**, we have*

$$|\tilde{l}(\mathcal{Y},\Phi_{\mathcal{W}}(\mathcal{X},\mathcal{L})) - \tilde{l}(M_{\mathcal{T}}Y_n, M_{\mathcal{T}}\Phi_{\mathcal{W}}(X_n,L_n))| = \mathcal{O}\left(\frac{1}{i_c} + \sqrt[d+4]{\frac{\log 1/\delta}{n}}\right)$$

where $M_{\mathcal{T}}$ is the training mask [cf. (5)]. Proof and omitted assumptions are provided in Appendix B.2.

Theorem 4.2 (An unsatisfactory generalization bound)**.** *Under **Setup**, suppose the minimum of the optimization problem in (5) is achieved by $\Phi_{\mathcal{W}^*_G}$, i.e., by the GNN with weights $\mathcal{W}^*_G$, and that Assumptions B.1–B.3 hold. Let $p > q$. With probability at least $1-\delta$,*

$$GA(\Phi_{\mathcal{W}^*_G}) = \mathcal{O}\left(\frac{1}{i_c} + \sqrt[d+4]{\frac{\log 1/\delta}{n}} + \frac{p-q}{pq}\tilde{l}(\mathcal{Y},\Phi_{\mathcal{W}^*_G}(\mathcal{X},\mathcal{L}))\right). \tag{12}$$

The proof and omitted assumptions are provided in Appendix B.

The generalization gap is upper-bounded by three terms: (1) a term involving the convolutional filter bandwidth c, which remains small for high enough bandwidths; (2) a term reflecting the convergence

of GNNs on graph sequences approaching a manifold, which vanishes as n grows [cf. Prop. 3.1]; and (3) a term depending on the training and test set sizes p, q, and the loss of the GNN $\Phi_{\mathcal{W}_G^*}$ over the full manifold $\mathcal{M}$.

The third term is notable for its $(p-q)/pq$ factor, revealing the influence of imbalance between training and test sizes on the generalization gap. For typical proportions $p = \nu n, q = (1-\nu)n$, the third term in (30) is $\mathcal{O}(n^{-1}\tilde{l}(\mathcal{Y}, \Phi_{\mathcal{W}_G^*}(\mathcal{X}, \mathcal{L}))$ unless the split is balanced ($\nu = 0.5$), in which case this contribution disappears.

If $\nu > 0.5$, whether this term dominates depends on $\Phi_{\mathcal{W}_G^*}$'s loss over $\mathcal{M}$. This is undesirable for two reasons: the loss requires access to the entire manifold (i.e., the test set), and since $\Phi_{\mathcal{W}_G^*}$ is trained on G_n, it may not minimize loss on $\mathcal{M}$. The next lemma presents a potential improvement to this bound.

Specifically, we can chain Theorem 4.2 and Lemma B.5 once more to derive an upper bound on the generalization gap that no longer depends on the loss on the entire manifold, but rather on the minimum semi-supervised training loss on the graph G_n:

$$l_G^* = \tilde{l}(M_{\mathcal{T}} Y_n, M_{\mathcal{T}} \Phi_{\mathcal{W}_G^*}(X_n, L_n)). \tag{13}$$

Theorem 4.3 (A satisfactory generalization bound)**.** *Under **Setup**, suppose the minimum of the optimization problem in* (5) *is achieved by $\Phi_{\mathcal{W}_G^*}$, i.e., by the GNN with weights $\mathcal{W}_G^*$, and that Assumptions B.1–B.3 hold. Let $p > q$. With probability at least $1-\delta$,*

$$GA(\Phi_{\mathcal{W}_G^*}) = \mathcal{O}\left(\frac{p}{qi_c} + \sqrt[d+4]{\frac{\log 1/\delta}{n}} + \frac{p-q}{pq} l_G^* \right). \tag{14}$$

The proof and omitted assumptions are provided in Appendix C.

The generalization bound in this theorem is more satisfactory, as now the term depending on the loss realized by the GNN can be controlled through optimization over the training set $\mathcal{T}$. However, this comes at the cost of an increase in the constant term from $1/i_c$ in Theorem (4.2) to $p/(qi_c)$ in Theorem 4.3. In modern machine learning, one typically has significantly more training samples p than test samples q. Hence, this increase might be non-negligible in practice.

4.2 Learning on graphs of increasing size

In this section we discuss an alternative GNN training algorithm inspired by [16] allowing to directly minimize $\tilde{l}(\mathcal{Y}, \Phi_{\mathcal{W}}(\mathcal{X}, \mathcal{L}))$, the loss on the manifold, and as such to curb the increase in the generalization gap observed in Theorem 4.3.

The algorithm is rather simple. Instead of fixing the graph G_n during the entire training process, we instead start from an n_0-node graph G_{n_0} and, after Δt gradient updates over this graph, resample a graph G_{n_1} with $n_1 = n_0 + \Delta n$ from $\mathcal{M}$. We proceed to do Δt gradient updates over G_{n_1}, then resample G_{n_2} and repeat. Explicitly, the kth iterate is given by

$$\mathcal{W}_{k+1} = \mathcal{W}_k - \eta_k \nabla_{\mathcal{W}} l(Y_{n_t}, \Phi(X_{n_t}, L_{n_t})). \tag{15}$$

with $t = \lfloor k/\Delta t \rfloor$.

Under mild assumptions, it can be shown that the GNN obtained by solving problem (5) on this graph sequence minimizes the empirical risk on the manifold $\mathcal{M}$.

Theorem 4.4. *Under **Setup**, let $\Phi_{\mathcal{W}}$ be a GNN learned with iterates* (15)*. If at each step k the number of nodes n_t is such that*

$$\mathbb{E}[\|\nabla_{\mathcal{W}} \tilde{l}(\mathcal{Y}, \Phi_{\mathcal{W}_k}(\mathcal{X}, \mathcal{L})) - \nabla_{\mathcal{W}} l(Y_{n_t}, \Phi_{\mathcal{W}_k}(X_{n_t}, L_{n_t}))\|] < \|\nabla_{\mathcal{W}} \tilde{l}(\mathcal{Y}, \Phi_{\mathcal{W}_k}(\mathcal{X}, \mathcal{L}))\| - \epsilon \tag{16}$$

then after at most $k^* = \mathcal{O}(1/\epsilon^2)$ iterations $\Phi_{\mathcal{W}_{G_{n_t}}^*} = \Phi_{\mathcal{W}_{k^*}}$ is within an ϵ-neighborhood of the solution of the empirical risk minimization problem on $\mathcal{M}$. The proof and omitted assumptions are provided in Appendix D.

This result is of independent interest, as it prescribes an algorithm for achieving approximate solutions of risk minimization problems on manifolds by solving them on sequences of geometric graphs. In our specific context, it further allows one to obtain GNNs with improved generalization gap. This is done by combining Theorems 4.2 and 4.4 in the following corollary.

Corollary 4.5 (A better generalization bound)**.** *Let* $l^*_{\mathcal{M}} = \min_{\mathcal{W}} \tilde{l}(\mathcal{Y}, \Phi_{\mathcal{W}}(\mathcal{X}, \mathcal{L}))$*. Under* ***Setup****, let* $\Phi_{\mathcal{W}^*_{G_{n_t}}}$ *be the GNN learned on a sequence of graphs as in Theorem 4.4. With probability at least* $1-\delta$*,*

$$GA(\Phi_{\mathcal{W}_{G^*_{n_t}}}) = \mathcal{O}\left(\frac{1}{i_c} + \sqrt[d+4]{\frac{\log 1/\delta}{n}} + \frac{p-q}{pq}(l^*_{\mathcal{M}} + \epsilon)\right). \tag{17}$$

This approach leads to a smaller generalization gap than the one in Theorem 4.3, and more practical than the one in Theorem 4.2, as it guarantees close to minimum loss on the manifold.

5 Experiments

To evaluate our theoretical findings, we conducted experiments on several benchmarks. Specifically, we tested our method on MNIST, FMNIST, CIFAR10, FER2013, CelebA and PathMNIST benchmarks [40, 71, 38, 32, 42, 73]. However, due to the lack of space, we point the reader to Appendix A for all the experimental details, the results, and the following discussions.

6 Conclusions

We proposed a semi-supervised image classification method that builds a geometric graph from VAE embeddings and applies GNNs, drawing on the manifold hypothesis. Our approach enables generalization analysis and shows that the generalization gap shrinks with more data, which we confirm experimentally. The model outperforms baselines on all datasets. Limitations include reliance on VAE embedding quality, sensitivity to graph construction parameters, and computational overhead from a two-stage pipeline, which may hinder scalability.

References

[1] M. Avella-Medina, F. Parise, M. Schaub, and S. Segarra. Centrality measures for graphons: Accounting for uncertainty in networks. *IEEE Trans. Netw. Sci. Eng.*, 7(1):520–537, 2018. 2

[2] M. Balasubramanian and E. L. Schwartz. The Isomap algorithm and topological stability. *Science*, 295(5552):7–7, 2002. 2

[3] Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Caglar Gulcehre, Francis Song, Andrew Ballard, Justin Gilmer, George Dahl, Ashish Vaswani, Kelsey Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matt Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018. 1

[4] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003. 2

[5] M. Belkin and P. Niyogi. Convergence of Laplacian eigenmaps. 19, 2006. 2

[6] M. Belkin and P. Niyogi. Towards a theoretical foundation for Laplacian-based manifold methods. *Journal of Computer and System Sciences*, 74(8):1289–1308, 2008. 3

[7] Y. Bengio, A. Courville, and P. Vincent. Representation Learning: A Review and New Perspectives. 35(8):1798–1828, 2013. 2

[8] P. Bérard. *Spectral geometry: direct and inverse problems*, volume 1207. Springer, 2006. 3

[9] Erik Bernhardsson. Annoy: Approximate nearest neighbors in c++/python. https://github.com/spotify/annoy, 2018. Accessed: 2025-05-12. 12

[10] Dimitri P. Bertsekas and John N. Tsitsiklis. Gradient convergence in gradient methods with errors. *SIAM Journal on Optimization*, 10(3):627–642, 2000. 24

[11] C. Borgs and J. Chayes. Graphons: A nonparametric method to model, estimate, and design algorithms for massive networks. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, pages 665–672, 2017. 2

[12] C. Borgs, J. T. Chayes, L. Lovász, V. T. Sós, and K. Vesztergombi. Convergent sequences of dense graphs I: Subgraph frequencies, metric properties and testing. *Adv. Math.*, 219(6):1801–1851, 2008. 2

[13] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. Geometric deep learning: Going beyond Euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017. 2

[14] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. Spectral networks and deep locally connected networks on graphs. 2014. 2

[15] J. Calder, K. Miller, and A. L. Bertozzi. Novel batch active learning approach and its application to synthetic aperture radar datasets. In *Proc. of SPIE Vol*, volume 12520, pages 125200B–1, 2023. 2

[16] J. Cervino, L. Ruiz, and A. Ribeiro. Learning by transference: Training graph neural networks on growing graphs. 71:233–247, 2023. 2, 6, 19, 20, 23, 24, 25

[17] Juan Cervino, Luiz FO Chamon, Benjamin David Haeffele, Rene Vidal, and Alejandro Ribeiro. Learning globally smooth functions on manifolds. In *International Conference on Machine Learning*, pages 3815–3854. PMLR, 2023. 2

[18] R. R. Coifman and S. Lafon. Diffusion maps. *Applied and computational harmonic analysis*, 21(1):5–30, 2006. 2

[19] R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *Proceedings of the National Academy of Sciences*, 102(21):7426–7431, 2005. 2

[20] M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. Barcelona, Spain, 5-10 December 2016. NIPS Foundation. 2

[21] J. Du, J. Shi, S. Kar, and J. M. F. Moura. On graph convolution for graph CNNs. In *2018*, pages 239–243, Lausanne, Switzerland, 4-6 June 2018. IEEE. 2

[22] Vijay Prakash Dwivedi, Chaitanya K Joshi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking Graph Neural Networks. *Journal of Machine Learning Research*, 24(43):1–48, 2023. 14

[23] J. Eldridge, M. Belkin, and Y. Wang. Graphons, mergeons, and so on! 29, 2016. 2

[24] E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. 35(11):2765–2781, 2013. 2

[25] C. Fefferman, S. Mitter, and H. Narayanan. Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049, 2016. 2

[26] M. Fey and J. E. Lenssen. Fast graph representation learning with PyTorch Geometric. *arXiv:1903.02428 [cs.LG]*, 2019. 14

[27] F. Gama, J. Bruna, and A. Ribeiro. Stability of graph scattering transforms. In *33rd*, Vancouver, BC, 8-14 December 2019. NeuriPS Foundation. 1

[28] F. Gama, J. Bruna, and A. Ribeiro. Stability properties of graph neural networks. 68:5680–5695, 2020. 1, 2

[29] F. Gama, A. G. Marques, G. Leus, and A. Ribeiro. Convolutional neural network architectures for signals supported on graphs. 67:1034–1049, 2018. 2

[30] Chen Gao, Yu Zheng, Nian Li, Yinfeng Li, Yingrong Qin, Jinghua Piao, Yuhan Quan, Jianxin Chang, Depeng Jin, Xiangnan He, and Yong Li. A survey of graph neural networks for recommender systems: Challenges, methods, and directions. *ACM Trans. Recomm. Syst.*, 1(1), March 2023. 1

[31] J. Gilmer, S. Schoenholz, P. Riley, O. Vinyals, and G. Dahl. Neural message passing for quantum chemistry. pages 1263–1272. PMLR, 2017. 1, 2

[32] Ian J Goodfellow, Dumitru Erhan, Pierre Luc Carrier, Aaron Courville, Mehdi Mirza, Ben Hamner, Will Cukierski, Yichuan Tang, David Thaler, Dong-Hyun Lee, et al. Challenges in representation learning: A report on three machine learning contests. In *Neural information processing: 20th international conference, ICONIP 2013, daegu, korea, november 3-7, 2013. Proceedings, Part III 20*, pages 117–124. Springer, 2013. 7, 12

[33] W. Hamilton, Z. Ying, and J. Leskovec. Inductive representation learning on large graphs. 30, 2017. 2, 14

[34] N. Keriven, A. Bietti, and S. Vaiter. Convergence and stability of graph convolutional networks on large random graphs. In *34th*, volume 33, pages 21512–21523. NeurIPS Foundation, 2020. 1

[35] D. P. Kingma and J. L. Ba. ADAM: A method for stochastic optimization. In *3rd*, San Diego, CA, 7-9 May 2015. Assoc. Comput. Linguistics.

[36] Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. *Proceedings of the International Conference on Learning Representations*, 2014. 12

[37] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *5th*, Toulon, France, 24-26 April 2017. Assoc. Comput. Linguistics. 1, 2

[38] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. 7, 12

[39] M. J. Kusner, B. Paige, and J. M. Hernández-Lobato. Grammar variational autoencoder. pages 1945–1954. PMLR, 2017. 12

[40] Yann LeCun, Corinna Cortes, and Christopher JC Burges. The mnist database of handwritten digits. *http://yann.lecun.com/exdb/mnist*, 1998. 7, 12

[41] R. Levie, W. Huang, L. Bucci, M. Bronstein, and G. Kutyniok. Transferability of spectral graph convolutional neural networks. 22(272):1–59, 2021. 1, 2

[42] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015. 7, 12, 14

[43] L. Lovász. *Large Networks and Graph Limits*, volume 60. American Mathematical Society, 2012. 2

[44] A. Magner, M. Baranwal, and A. O. Hero. The power of graph convolutional networks to distinguish random graph models. In *2020 IEEE International Symposium on Information Theory (ISIT)*, pages 2664–2669. IEEE, 2020. 2

[45] Haggai Maron, Heli Ben-Hamu, Nadav Shamir, and Yaron Lipman. Invariant and Equivariant Graph Networks. *International Conference on Learning Representations*, 2019. 1

[46] H. Narayanan and S. Mitter. Sample complexity of testing the manifold hypothesis. 23, 2010. 2

[47] Partha Niyogi. Manifold regularization and semi-supervised learning: Some theoretical analyses. 14(5), 2013. 2

[48] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035, 2019. 14

[49] P. Petersen. Riemannian geometry. *Graduate Texts in Mathematics/Springer-Verlarg*, 2006. 3

[50] Y. Pu, Z. Gan, R. Henao, X. Yuan, C. Li, A. Stevens, and L. Carin. Variational autoencoder for deep learning of images, labels and captions. 29, 2016. 12

[51] J. Robbin and D. Salamon. *Introduction to differential geometry*. Springer Nature, 2022. 3

[52] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000. 2

[53] L. Ruiz, L. F. O. Chamon, and A. Ribeiro. The Graphon Fourier Transform. In *45th*, pages 5660–5664, Barcelona, Spain (Virtual), 4-8 May 2020. IEEE. 2

[54] L. Ruiz, L. F. O. Chamon, and A. Ribeiro. Graphon neural networks and the transferability of graph neural networks. In *34th*, Vancouver, BC (Virtual), 6-12 December 2020. NeurIPS Foundation. 1, 2

[55] L. Ruiz, L. F. O. Chamon, and A. Ribeiro. Graphon filters: Signal processing in very large graphs. In *28th*, pages 1050–1054, Amsterdam, The Netherlands (Virtual), 18-22 January 2021. IEEE. 2

[56] L. Ruiz, L. F. O. Chamon, and A. Ribeiro. Graphon signal processing. 69:4961–4976, 2021. 2

[57] L. Ruiz, L. F. O. Chamon, and A. Ribeiro. Transferability properties of graph neural networks. 2023. 2

[58] L. Ruiz, F. Gama, A. G. Marques, and A. Ribeiro. Invariance-preserving localized activation functions for graph neural networks. 68:127–141, 2020. 2

[59] L. Ruiz, F. Gama, and A. Ribeiro. Graph neural networks: Architectures, stability and transferability. *Proc. IEEE*, 109(5):660–682, 2021. 1, 2, 4

[60] Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. *Advances in neural information processing systems*, 30, 2017. 1

[61] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The graph neural network model. 20(1):61–80, 2008. 2

[62] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *The semantic web: 15th international conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, proceedings 15*, pages 593–607. Springer, 2018. 1

[63] S. Segarra, A. G. Marques, and A. Ribeiro. Optimal graph-filter design and applications to distributed linear network operators. 65:4117–4131, August 2017. 2

[64] David W. Sroczynski, Or Yair, Ronen Talmon, and Ioannis G. Kevrekidis. Data-driven evolution equation reconstruction for parameter-dependent nonlinear dynamical systems. *Israel Journal of Chemistry*, 58(6-7):711–723, 2018. 2

[65] Ulrike Von Luxburg, Mikhail Belkin, and Olivier Bousquet. Consistency of spectral clustering. *The Annals of Statistics*, pages 555–586, 2008. 16, 22, 23

[66] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (TOG)*, 38(5):146:1–146:12, 2019. 2

[67] Z. Wang, L. Ruiz, and A. Ribeiro. Stability of neural networks on Riemannian manifolds. In *29th*, Dublin, Ireland (Virtual), 23-27 August 2021. IEEE. 4

[68] Zhiyang Wang, Juan Cervino, and Alejandro Ribeiro. A Manifold Perspective on the Statistical Generalization of Graph Neural Networks. *arXiv preprint arXiv:2406.05225*, 2024. 4

[69] Zhiyang Wang, Juan Cervino, and Alejandro Ribeiro. Generalization of Geometric Graph Neural Networks. *arXiv preprint arXiv:2409.05191*, 2024. 15

[70] Zhiyang Wang, Juan Cervino, and Alejandro Ribeiro. Generalization of Geometric Graph Neural Networks. *Asilomar Conference on Signals, Systems, and Computers*, 2024. 17, 22

[71] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 7, 12

[72] K. Xu, W. Hu, J. Leskovec, and S. Jegelka. How powerful are graph neural networks? In *7th*, pages 1–17, New Orleans, LA, 6-9 May 2019. Assoc. Comput. Linguistics. 2

[73] Jiancheng Yang, Rui Shi, and Bingbing Ni. Medmnist classification decathlon: A lightweight automl benchmark for medical image analysis. In *IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pages 191–195, 2021. 7, 12, 14

[74] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A. Efros. Generative visual manipulation on the natural image manifold. In *European Conference on Computer Vision (ECCV)*, pages 597–613. Springer, 2016. 2

A Experimental results and details

Experimental setup. We conduct experiments on MNIST, FMNIST, CIFAR10, FER2013, CelebA and PathMNIST benchmarks [40, 71, 38, 32, 42, 73]. Since these are image datasets, we first have to define a way to extract meaningful graphs from this setting. A natural approach is to first construct embeddings that represent each image, usually of a lower dimension than the input (image) space, and take advantage of the geometry of such a lower-dimensional manifold with graphs.

In this work, we make use of autoencoders to build representative embeddings. Since we want to preserve the images' translational invariances/equivariances, we set our encoder/decoder networks to be Convolutional Neural Networks (CNNs). In addition, to account for implicit invariances/equivariances in the data, which might not be captured by explicit symmetries incorporated in the model's architecture, we propose to use Variational Autoencoders (VAEs) [36] to learn the latent space. Since VAEs learn a Gaussian approximation of the embeddings' distribution in the latent space, they add more structure to the low-dimensional manifold, which makes it smoother than deterministic AE counterparts, as seen in previous works [50, 39].

Given a set of images $\{\mathrm{X}_m\}_{m=1}^M$ from the ambient space $\mathcal{S}_\mathrm{X}$, the encoder $f_{\text{enc}}\colon \mathcal{S}_\mathrm{X} \to \mathbb{R}^F$ reduces the data to a F-dimensional embedding $z_m = f_{\text{enc}}(\mathrm{X}_m)$, while the decoder $f_{\text{dec}}\colon \mathbb{R}^F \to \mathcal{S}_\mathrm{X}$ maps the embedding back to the original space $\hat{\mathrm{X}}_m = f_{\text{dec}}(z_m)$. In our setting, our embedding is defined as the posterior distribution's estimated mean. For MNIST, CelebA, and PathMNIST, we found that the best latent space has size $F = 128$, for FMNIST $F = 256$, for CIFAR10 $F = 1024$, and FER2013 $F = 64$.

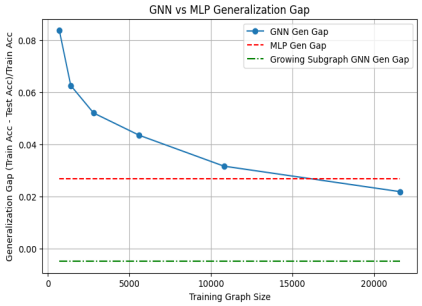
Having access to the embeddings z_m, we can approximate the image manifold with a graph by computing the pairwise distance between image embeddings following the steps from Section 3.1, and then process this graph using a GNN to predict the image labels via semi-supervised node (image) classification. Concretely, given a dataset consisting of pairs $\{z_m, \mathrm{y}_m\}_{m=1}^M$, where $\mathrm{y}_m \in \{1, \dots, C\}$ is the class label for image m, we construct a graph G by considering the image embeddings (z_m) to be nodes and computing their pairwise edge weights with a Gaussian kernel. However, since computing pairwise distances between all embeddings in the dataset would be impractical, in practice, we use an approximate nearest neighbor (ANN) algorithm to construct a 100-nearest neighbor graph. Specifically, we apply a tree-based ANN method [9] to find neighbors efficiently and then assign edges with Gaussian weights $w_{ij} = \exp(-\frac{|z_i - z_j|^2}{2\sigma^2})$.

Experimental results. We present our empirical results under three perspectives: (i) adherence to the theoretical results, (ii) effectiveness of our model, measured in terms of image classification accuracy on the test set of standard splits, and (iii) robustness and flexibility of our method. It's worth noting that all experiment details are provided in Appendix A.

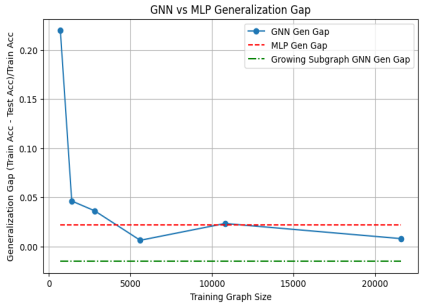
For (i), as shown in Figure 2 GNNs trained on fixed subgraphs (blue) exhibit large generalization gaps for small training graph sizes, but the gap decreases steadily with more nodes, eventually outperforming the MLP baseline. This behavior is consistent with the prediction of Theorem 4.3. GNNs trained on sequences of growing subgraphs (green) achieve the smallest generalization gap across all datasets, in agreement with Corollary 4.5, and consistently outperform both fixed-graph GNNs and MLPs.

For (ii), as shown in Table 1 our GNN achieves the highest test accuracy across all four datasets when trained on the full data graph. On MNIST, it reaches perfect accuracy, as expected given the simplicity of the task. On FMNIST and FER2013, our model outperforms all compared methods by a notable margin. While the MLP performs slightly better than kNN on CIFAR10, our GNN method surpasses both, achieving the best accuracy with substantially reduced overfitting as reflected by the smaller gap between train and test performance.

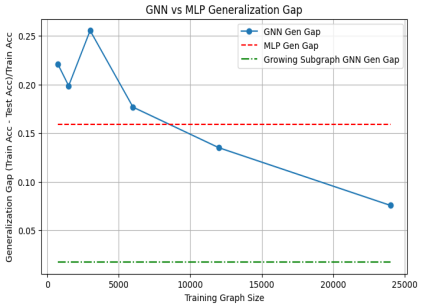
Finally, for (iii), we included two sets of experiments to showcase the robustness of our framework, which also complements the support on both theory and practical effectiveness. In the first set, we applied it to two large-scale datasets: (i) CelebA [42], a diverse dataset of celebrities' faces in different poses, backgrounds, and attributes, and (ii) PathMNIST [73], a dataset for histopathology detection – medical image analysis. It is worth noting that, since CelebA has multiple labels for each image, we selected two attributes, i.e., smiling and gender, and framed the tasks as separate binary classifications. The resulting datasets were coined CelebA-Smiling and CelebA-Gender.



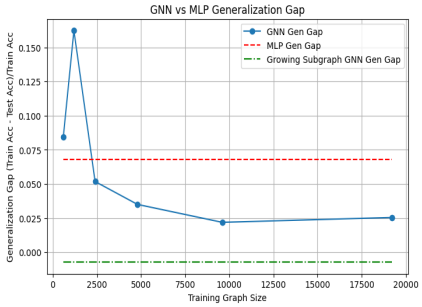
(a) MNIST



(b) FMNIST

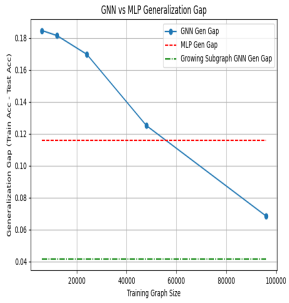


(c) CIFAR10

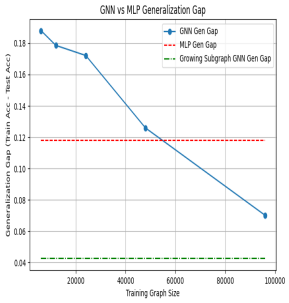


(d) FER2013

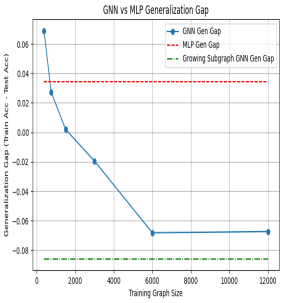
Figure 2: Generalization gap relative to training accuracy for (a) MNIST, (b) FMNIST, (c) CIFAR10, (d) FER2013. We compare an MLP trained on the VAE embeddings of the full dataset (red); GNNs **fully** trained on subgraphs of the full data graph with size given by the x-axis (blue, Thm. 4.3); and a GNN learned on this sequence of subgraphs, one per epoch (green, Cor. 4.5). The generalization gap decreases with graph size (blue), and is substantially smaller when training on growing subgraphs (green), in line with our theoretical predictions.



(a) CelebA-Smiling



(b) CelebA-Gender



(c) PathMNIST

Figure 3: Generalization gap relative to training accuracy for (a) CelebA-Smiling, (b) CelebA-Gender, (c) PathMNIST. We compare an MLP trained on the VAE embeddings of the full dataset (red); GNNs **fully** trained on subgraphs of the full data graph with size given by the x-axis (blue, Thm. 4.3); and a GNN learned on this sequence of subgraphs, one per epoch (green, Cor. 4.5). The generalization gap decreases with graph size (blue) and is substantially smaller when training on growing subgraphs (green), in line with our theoretical predictions.

The second set of experiments aimed to assess the importance of the invariances and equivariances captured by our proposed CNNVAE embeddings. To this end, we generated alternative embeddings using PCA across all datasets, and then trained and evaluated a GNN using these representations.

As seen in Figure 3 and Table 2, our method still outperforms the baselines throughout all the new datasets, though the graph is larger than the ones we previously tested (more than twice the size.) These results not only align with our theoretical claims – i.e., strong performance and a shrinking generalization gap as graph size increases – but also highlight the robustness of our approach.

Table 1: Accuracy on the **full dataset/graph**. Our method outperforms compared methods on every dataset, achieving the highest test accuracy and smallest generalization gap.

Model	MNIST		FMNIST		CIFAR10		FER2013	
	Test	Train	Test	Train	Test	Train	Test	Train
GCN (superpixel graph) [22]	90.12	96.46	–	–	54.14	70.16	–	–
kNN	96.31	96.92	83.76	86.40	40.93	43.65	36.58	58.81
MLP	97.40	100.00	84.35	86.53	54.29	66.49	42.40	50.05
GNN (ours)	**100.00**	100.00	**84.46**	85.28	**61.83**	63.18	**48.38**	47.97

Table 2: Accuracy on the **full dataset/graph**. Our method outperforms compared methods on CelebA-Smiling, CelebA-Gender, and PathMNIST, achieving the highest test accuracy and smallest generalization gap. Given the size of the graphs for the first two datasets ($> 162k$ images/nodes), we didn't have time to finish assessing the training accuracy for the kNN model.

Model	CelebA-Smiling		CelebA-Gender		PathMNIST	
	Test	Train	Test	Train	Test	Train
GCN (superpixel graph) [22]	–	–	–	–	–	–
kNN	70.15	(*timeout*)	79.09	(*timeout*)	60.67	72.92
MLP	81.33	93.92	81.38	93.05	66.16	70.16
GNN (ours)	**87.58**	90.37	**87.51**	90.32	**72.95**	66.46

Furthermore, when comparing GNNs trained on VAE versus PCA-based embeddings (Table 3), our method maintained superior performance across all datasets. This suggests that preserving translational and data distribution-related invariances/equivariances leads to a more structured latent space, thus enabling the GNN to share geometric information among images more effectively.

Experiments were conducted with two different settings, depending on the memory complexity related to the size of the data manifold and its dimension. Specifically, for smaller graphs, i.e., MNIST, FMNIST, and FER2013 datasets, we used a server with 2x NVIDIA GeForce RTX 4090 (24GB) GPU, 128GB of RAM, and a CPU AMD Ryzen Threadripper PRO 5955WX 16-Cores. For medium-to-large ones, i.e., CIFAR10, CelebA [42], and PathMNIST [73] we experimented using a server with 2x NVIDIA RTX 6000 Ada Generation (48GB) GPU, 500GB of RAM, and a CPU AMD EPYC 7453 28-Core Processor. Both servers used Ubuntu 22.04.4 LTS as a Linux distro.

We used the original split for each one of the datasets. For each experiment, which is directly related to the number of sampled nodes, we performed 4 runs and presented the mean. It's worth noting that, to make the comparisons fairer, especially with the SLIC-based GNN ([22]), we trained our models under a fixed computational budget of less than $100k$ parameters.

The model used is a 1-layer GNN with SAGEConv [33] for the generalization gap analysis presented in Figures 2a-2d and 3a-3c, and the results showed in Table 1, 2 and 3. We used PyTorch [48] and, more specifically, PyTorch Geometric (PyG) framework [26] for the models.

To obtain the best embedding representation, we use Weights & Biases (W&B) to fine-tune the VAE's hyperparameters. We optimize the number of layers in the CNN encoder/decoder and the latent space dimension. The CNN has three convolutional layer blocks, each with 1–5 layers, ensuring encoder-decoder symmetry. The latent dimension is chosen from 32, 64, 128, 256, 364, 512, 1024. A grid search sweeps through the cartesian product of these configurations. The best parameters vary by dataset: MNIST, FMNIST and PathMNIST perform best with [3, 3, 1] convolutional blocks and latent dimensions of 128, 256, and 128, respectively. CIFAR10 requires [4, 5, 2] blocks and a 1024-dimensional latent space, and FER2013 and CelebA [42] need [3, 3, 3] blocks, 64 and 128 latent sizes, respectively. We set the KL divergence weight to balance regularization and reconstruction.

Tables 4 and 5 summarize the hyperparameters used in our experiments for all datasets. We used Adam optimizer with the dataset's respective parameters.

We expect to release the code of our project in the near future.

Table 3: Accuracy on the **full dataset/graph**. VAE-based embeddings provided a more structured latent space, which translates to our method outperforming one that was trained on embeddings generated using PCA. Across all datasets – MNIST, FMNIST, CIFAR10, FER2013, CelebA-Smiling, CelebA-Gender, and PathMNIST – a GNN trained on the VAE embeddings achieved the highest test accuracy.

Model	MNIST		FMNIST		CIFAR10		FER2013		CelebA-Smiling		CelebA-Gender		PathMNIST	
	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train
GNN (PCA)	55.10	53.36	66.23	66.14	27.25	26.82	27.40	26.94	60.72	61.99	74.01	73.52	72.24	65.20
GNN (VAE)	**100.00**	100.00	**84.46**	85.25	**61.83**	63.18	**48.38**	47.97	**87.58**	90.37	**87.51**	90.32	**72.95**	66.46

Table 4: VAE training hyperparameters for each dataset.

Parameter	**MNIST**	**FMNIST**	**CIFAR10**	**FER2013**	**CelebA**	**PathMNIST**
Batch size	64	64	64	64	64	64
Learning rate	0.0001	0.0001	0.0001	0.0003	0.0003	0.0001
Number of epochs	50	50	50	50	50	50
Latent size	128	256	1024	64	128	128
Num. of layers	[3, 3, 1]	[3, 3, 1]	[4, 5, 2]	[3, 3, 3]	[3, 3, 3]	[3, 3, 1]

Table 5: GNN training hyperparameters for each dataset.

Parameter	**MNIST**	**FMNIST**	**CIFAR10**	**FER2013**	**CelebA**	**PathMNIST**
Batch size	256	256	256	256	256	256
Learning rate	0.01	0.01	0.01	0.01	0.01	0.01
Kernel width	4.0	0.8	5.0	4.0	3.5	5.0
Hidden dimension	128	128	128	128	128	128
Num. of layers	1	1	1	1	1	1

B Proof of Theorem 4.2

B.1 Assumptions

First, let us state a few assumptions that will be used in the following proofs.

Assumption B.1. *The convolutional maps in $\Phi_{\mathcal{W}}$ are locally Lipschitz continuous on $\mathcal{M}$ and have norm at most 1.*

Assumption B.2. *The convolutions in all layers of $\Phi_{\mathcal{W}}$ are low-pass filters with bandwidth c, i.e., if $\mathcal{Y}$ is the output of a convolution, $\langle \mathcal{Y}, \phi_i \rangle = 0$ for $\lambda_i > c$, and $i_c = \arg\min_i (\lambda_i - c)\mathbf{1}(\lambda_i \geq c)$.*

Assumption B.3. *The nonlinear function ρ and its first-order derivative ρ' have Lipschitz constant 1 and $\rho(0) = 0$, i.e., the function is normalized Lipschitz continuous.*

B.2 Lemmas

Furthermore, we need the following lemma adapted from [69].

Lemma B.4. *Let $\mathcal{M} \subset \mathbb{R}^F$ be a manifold equipped with a Laplace-Beltrami (LB) operator $\mathcal{L}$, as defined in* (2)*, a self-adjoint operator, whose eigenpairs are $\{\lambda_i, \phi_i\}_{i=1}^{\infty}$. Moreover, let $f, g \in L^2(\mathcal{M})$ be manifold signals over $\mathcal{M}$, and $\mathcal{P}_n$ the sampling operator used to sample manifold signals. Therefore, we have that:*

$$|\|\mathcal{P}_n f\| - \|f\|_{\mathcal{M}}| = \mathcal{O}\left(\sqrt[4]{\frac{\log(1/\delta)}{n}}\right). \tag{18}$$

Proof. The inner product between these signals is defined as

$$\langle f, g \rangle_{\mathcal{M}} = \int_{\mathcal{M}} f(x) g(x) d\mu(x), \tag{19}$$

where $d\mu(x)$ is the volume element of $\mathcal{M}$ w.r.t. its measure μ. Hence, one can define the norm of such a signal as

$$\|f\|_{\mathcal{M}}^2 = \langle f, f \rangle_{\mathcal{M}}. \tag{20}$$

Given that we have $\{X_1, \dots, X_N\}$ randomly sampled points from $\mathcal{M}$, by Theorem 19 in [65] we have that

$$|\langle \mathcal{P}_N f, \phi_i \rangle - \langle f, \phi_i \rangle_{\mathcal{M}}| = \mathcal{O}\left(\sqrt{\frac{\log(1/\delta)}{N}}\right). \tag{21}$$

The above implies that

$$|\|\mathcal{P}_n f\|^2 - \|f\|^2_{\mathcal{M}}| = \mathcal{O}\left(\sqrt{\frac{\log(1/\delta)}{n}}\right), \tag{22}$$

which further implies that

$$|\|\mathcal{P}_n f\| - \|f\|_{\mathcal{M}}| \approx \mathcal{O}\left(\sqrt[4]{\frac{\log(1/\delta)}{n}}\right). \tag{23}$$

□

Lemma B.5. *Suppose Assumptions B.1–B.3 hold. With probability at least* $1-\delta$*, for any GNN* $\Phi_{\mathcal{W}}$ *as in* ***Setup****, we have*

$$\begin{aligned} &|\tilde{l}(\mathcal{Y}, \Phi_{\mathcal{W}}(\mathcal{X}, \mathcal{L})) - \tilde{l}(M_{\mathcal{T}} Y_n, M_{\mathcal{T}} \Phi_{\mathcal{W}}(X_n, L_n))| \\ &\qquad = \mathcal{O}\left(\frac{1}{i_c} + \sqrt[d+4]{\frac{\log 1/\delta}{n}}\right) \end{aligned} \tag{24}$$

where $M_{\mathcal{T}}$ *is the training mask [cf.* (5)*].*

Proof. We first write the difference between the loss function of the GNN and the MNN trained on the same set of parameters for the semi-supervised setting:

$$\begin{aligned} &\left|\tilde{l}(M_{\mathcal{T}} Y_n, M_{\mathcal{T}} \Phi_{\mathcal{W}}(X_n, L_n)) - \tilde{l}(\mathcal{Y}, \Phi_{\mathcal{W}}(\mathcal{X}, \mathcal{L}))\right| \\ &= \frac{1}{p}\,|\|M_{\mathcal{T}} \Phi_{\mathcal{W}}(X_n, L_n) - M_{\mathcal{T}} Y_n\|_2 - \|\Phi_{\mathcal{W}}(\mathcal{X}, \mathcal{L}) - \mathcal{Y}\|_{\mathcal{M}}| \\ &= \frac{1}{p}|\|M_{\mathcal{T}} \Phi_{\mathcal{W}}(X_n, L_n) - M_{\mathcal{T}} Y_n \\ &\quad + (M_{\mathcal{T}} \mathcal{P}_N \Phi_{\mathcal{W}}(\mathcal{X}, \mathcal{L}) - M_{\mathcal{T}} \mathcal{P}_N \Phi_{\mathcal{W}}(\mathcal{X}, \mathcal{L}))\|_2 \\ &\quad - \|\Phi_{\mathcal{W}}(\mathcal{X}, \mathcal{L}) - \mathcal{Y}\|_{\mathcal{M}}| \\ &\leq \frac{1}{p}|\|M_{\mathcal{T}} \Phi_{\mathcal{W}}(X_n, L_n) - M_{\mathcal{T}} \mathcal{P}_N \Phi_{\mathcal{W}}(\mathcal{X}, \mathcal{L})\|_2 \\ &\quad + \|M_{\mathcal{T}} \mathcal{P}_N \Phi_{\mathcal{W}}(\mathcal{X}, \mathcal{L}) - M_{\mathcal{T}} \mathcal{P}_N \mathcal{Y}\|_2 \\ &\quad - \|\Phi_{\mathcal{W}}(\mathcal{X}, \mathcal{L}) - \mathcal{Y}\|_{\mathcal{M}}|. \end{aligned} \tag{25}$$

In (25) we used the fact that $Y_n = \mathcal{P}_N \mathcal{Y}$. Now, since the training mask has unitary norm, i.e., $\|M_{\mathcal{T}}\| = 1$ we have that:

$$\begin{aligned} &\frac{1}{p}|\|M_{\mathcal{T}} \Phi_{\mathcal{W}}(X_n, L_n) - M_{\mathcal{T}} \mathcal{P}_N \Phi_{\mathcal{W}}(\mathcal{X}, \mathcal{L})\|_2 \\ &\qquad + \|M_{\mathcal{T}} \mathcal{P}_N \Phi_{\mathcal{W}}(\mathcal{X}, \mathcal{L}) - M_{\mathcal{T}} \mathcal{P}_N \mathcal{Y}\|_2 \\ &\qquad - \|\Phi_{\mathcal{W}}(\mathcal{X}, \mathcal{L}) - \mathcal{Y}\|_{\mathcal{M}}| \\ &\leq \underbrace{\frac{1}{p}|\|M_{\mathcal{T}} \Phi_{\mathcal{W}}(X_n, L_n) - M_{\mathcal{T}} \mathcal{P}_N \Phi_{\mathcal{W}}(\mathcal{X}, \mathcal{L})\|_2|}_{①} \\ &+ \frac{1}{p}|\|\mathcal{P}_N \Phi_{\mathcal{W}}(\mathcal{X}, \mathcal{L}) - \mathcal{P}_N \mathcal{Y}\|_2 - \|\Phi_{\mathcal{W}}(\mathcal{X}, \mathcal{L}) - \mathcal{Y}\|_{\mathcal{M}}|. \end{aligned} \tag{26}$$

By lemma B.4, the second term on (26) has order $\mathcal{O}((\log(1/\delta)/N)^{1/4}$. Therefore, our proof boils down to finding an upper bound to the term (1) above:

$$\begin{aligned}
&\frac{1}{p}|\|M_{\mathcal{T}}\Phi_{\mathcal{W}}(X_n, L_n) - M_{\mathcal{T}}\mathcal{P}_N\Phi_{\mathcal{W}}(\mathcal{X}, \mathcal{L})\|_2| \\
&= \frac{1}{p}\left[\sum_{i\in\mathcal{T}}(\Phi_{\mathcal{W}}(X_n, L_n))_i - \Phi_{\mathcal{W}}(\mathcal{X}, \mathcal{L})(x_i))^2\right]^{1/2} \\
&\leq \frac{1}{p}\sum_{i\in\mathcal{T}}|(\Phi_{\mathcal{W}}(X_n, L_n))_i - \Phi_{\mathcal{W}}(\mathcal{X}, \mathcal{L})(x_i)| \qquad (27) \\
&\leq \frac{1}{p}\cdot p\cdot|\rho((h(L_n)X_n)_i) - \rho(h(\mathcal{L})\mathcal{X}(x_i))|, \qquad (28)
\end{aligned}$$

where in the first inequality (27) we used the fact that, for $v \in \mathbb{R}^F$, $\sum_i |v_i| \geq (\sum_i {v_i}^2)^{1/2}$, whilst in the second (28), we take the largest absolute difference between the GNN and MNN. Finally, given that the nonlinear functions ρ are normalized Lipschitz continuous, we have the following bound:

$$\begin{aligned}
&\frac{1}{p}\cdot p\cdot|\rho((h(L_n)X_n)_i) - \rho(h(\mathcal{L})\mathcal{X}(x_i))| \\
&\leq |(h(L_n)X_n)_i - h(\mathcal{L})\mathcal{X}(x_i)| \\
&= |(h(L_n)\mathcal{P}_n\mathcal{X})_i - (\mathcal{P}_n h(\mathcal{L})\mathcal{X})_i| \\
&= |[h(L_n)\mathcal{P}_n\mathcal{X} - \mathcal{P}_n h(\mathcal{L})\mathcal{X}]_i| \\
&\leq \|h(L_n)\mathcal{P}_n\mathcal{X} - \mathcal{P}_n h(\mathcal{L})\mathcal{X}\|_2 \\
&\leq (C_1 + C_2)\left(\frac{\log(\frac{C_1}{\delta})}{p}\right)^{\frac{1}{d+4}} + C_3\sqrt{\frac{\log(\frac{1}{\delta})}{p}} + \frac{C_4}{i_c}, \qquad (29)
\end{aligned}$$

$C_1 = C_{\mathcal{M},1}\frac{\pi^2}{6}\|\mathcal{X}\|_{\mathcal{M}}, C_2 = C_{\mathcal{M},2}\frac{\pi^2}{6}, C_3 = \frac{\pi^2}{6}, C_4 = \|\mathcal{X}\|_{\mathcal{M}}$, where $C_{\mathcal{M},1}$ and $C_{\mathcal{M},2}$ are constants that depend on the dimension d and the volume of the manifold.

The last step in (29) is an adaptation of the argument used in [70] to prove the bound for the difference between the graph and manifold filters (Equation (51), [70]). □

Theorem B.6. *Under **Setup**, suppose the minimum of the optimization problem in* (5) *is achieved by $\Phi_{\mathcal{W}_G^*}$, i.e., by the GNN with weights $\mathcal{W}_G^*$, and that Assumptions B.1–B.3 hold. Let $p > q$. With probability at least $1-\delta$,*

$$GA(\Phi_{\mathcal{W}_G^*}) = \mathcal{O}\left(\frac{1}{i_c} + \sqrt[d+4]{\frac{\log 1/\delta}{n}} + \frac{p-q}{pq}\tilde{l}(\mathcal{Y}, \Phi_{\mathcal{W}_G^*}(\mathcal{X}, \mathcal{L}))\right). \qquad (30)$$

Proof. Let $R_{\mathcal{T}}(\mathcal{W}_G^*) = \frac{1}{p}\tilde{l}(M_{\mathcal{T}}Y_n, M_{\mathcal{T}}\Phi_{\mathcal{W}_G^*}(X_n, L_n))$ and $R_{\mathcal{V}\setminus\mathcal{T}}(\mathcal{W}_G^*) = \frac{1}{q}\tilde{l}(M_{\mathcal{V}\setminus\mathcal{T}}Y_n, M_{\mathcal{V}\setminus\mathcal{T}}\Phi_{\mathcal{W}_G^*}(X_n, L_n))$ be the training and test error, respectively. Taking the L_2 loss as our loss function, we have that

$$\begin{aligned}
R_{\mathcal{T}}(\mathcal{W}_G^*) &= \frac{1}{p}\|M_{\mathcal{T}}\Phi_{\mathcal{W}_G^*}(X_n, L_n) - M_{\mathcal{T}}Y_n\|_2 \\
&= \frac{1}{p}\left[\sum_{i\in\mathcal{T}}(\Phi_{\mathcal{W}_G^*}(X_n, L_n))_i - \Phi_{\mathcal{W}_G^*}(\mathcal{X}, \mathcal{L})(x_i))^2\right]^{1/2}, \qquad (31) \\
R_{\mathcal{V}\setminus\mathcal{T}}(\mathcal{W}_G^*) &= \frac{1}{q}\|M_{\mathcal{V}\setminus\mathcal{T}}\Phi_{\mathcal{W}_G^*}(X_n, L_n) - M_{\mathcal{V}\setminus\mathcal{T}}Y_n\|_2 \\
&= \frac{1}{q}\left[\sum_{i\in\mathcal{V}\setminus\mathcal{T}}(\Phi_{\mathcal{W}_G^*}(X_n, L_n))_i - \Phi_{\mathcal{W}_G^*}(\mathcal{X}, \mathcal{L})(x_i))^2\right]^{1/2}. \qquad (32)
\end{aligned}$$

Under the transductive learning setting, the generalization gap $GA(\Phi_{\mathcal{W}_G^*}) = \left|R_{\mathcal{V}\setminus\mathcal{T}}(\mathcal{W}_G^*) - R_{\mathcal{T}}(\mathcal{W}_G^*)\right|$ is bounded as follows

$$
\begin{aligned}
&GA(\Phi_{\mathcal{W}_G^*}) \\
&= \Bigg| \frac{1}{q}\tilde{l}(M_{\mathcal{V}\setminus\mathcal{T}}Y_n, M_{\mathcal{V}\setminus\mathcal{T}}\Phi_{\mathcal{W}_G^*}(X_n, L_n)) \\
&\quad - \frac{1}{p}\tilde{l}(M_{\mathcal{T}}Y_n, M_{\mathcal{T}}\Phi_{\mathcal{W}_G^*}(X_n, L_n)\Bigg| \underset{(\pm)\frac{(p+q)}{pq}\tilde{l}(\mathcal{Y}, \Phi_{\mathcal{W}_G^*}(\mathcal{X}, \mathcal{L}))}{\leq} \\
&\Bigg|\left(\frac{1}{q}\tilde{l}(M_{\mathcal{V}\setminus\mathcal{T}}Y_n, M_{\mathcal{V}\setminus\mathcal{T}}\Phi_{\mathcal{W}_G^*}(X_n, L_n)) - \frac{1}{q}\tilde{l}(\mathcal{Y}, \Phi_{\mathcal{W}_G^*}(\mathcal{X}, \mathcal{L}))\right) \\
&+ \left(\frac{1}{p}\tilde{l}(\mathcal{Y}, \Phi_{\mathcal{W}_G^*}(\mathcal{X}, \mathcal{L})) - \frac{1}{p}\tilde{l}(M_{\mathcal{T}}Y_n, M_{\mathcal{T}}\Phi_{\mathcal{W}_G^*}(X_n, L_n)\right) \\
&+ \left(\frac{(p-q)}{pq}\tilde{l}(\mathcal{Y}, \Phi_{\mathcal{W}_G^*}(\mathcal{X}, \mathcal{L}))\right)\Bigg| \qquad (33)
\end{aligned}
$$

$$
\begin{aligned}
&\leq \underbrace{\frac{1}{q}\left|\tilde{l}(M_{\mathcal{V}\setminus\mathcal{T}}Y_n, M_{\mathcal{V}\setminus\mathcal{T}}\Phi_{\mathcal{W}_G^*}(X_n, L_n)) - \tilde{l}(\mathcal{Y}, \Phi_{\mathcal{W}_G^*}(\mathcal{X}, \mathcal{L}))\right|}_{(2)} \\
&+ \underbrace{\frac{1}{p}\left|\tilde{l}(\mathcal{Y}, \Phi_{\mathcal{W}_G^*}(\mathcal{X}, \mathcal{L})) - \tilde{l}(M_{\mathcal{T}}Y_n, M_{\mathcal{T}}\Phi_{\mathcal{W}_G^*}(X_n, L_n)\right|}_{(3)} \\
&+ \tfrac{(p-q)}{pq}|\tilde{l}(\mathcal{Y}, \Phi_{\mathcal{W}_G^*}(\mathcal{X}, \mathcal{L}))|. \qquad (34)
\end{aligned}
$$

That completes the proof since the previous lemma provides the bounds for (2) and (3). □

C Proof of Theorem 4.3

Lemma B.5 can be applied one more time to the last term of Theorem 4.2 so that we can get a bound that will depend on an observable object, i.e., the loss of the GNN. Specifically, we have

Theorem C.1. *Under **Setup**, suppose the minimum of the optimization problem in* (5) *is achieved by $\Phi_{\mathcal{W}_G^*}$, i.e., by the GNN with weights $\mathcal{W}_G^*$, and that Assumptions B.1–B.3 hold. Let $p > q$. With probability at least $1-\delta$,*

$$
GA(\Phi_{\mathcal{W}_G^*}) = \mathcal{O}\left(\frac{p}{qi_c} + \sqrt[d+4]{\frac{\log 1/\delta}{n}} + \frac{p-q}{pq}l_G^*\right). \qquad (35)
$$

Proof.

$$
\begin{aligned}
&\tfrac{(p-q)}{pq}|\tilde{l}(\mathcal{Y}, \Phi_{\mathcal{W}_G^*}(\mathcal{X}, \mathcal{L}))| \underset{\pm\tilde{l}(M_{\mathcal{T}}Y_n, M_{\mathcal{T}}\Phi_{\mathcal{W}_G^*}(X_n, L_n))}{=} \\
&\tfrac{(p-q)}{pq}|\tilde{l}(\mathcal{Y}, \Phi_{\mathcal{W}_G^*}(\mathcal{X}, \mathcal{L})) - \tilde{l}(M_{\mathcal{T}}Y_n, M_{\mathcal{T}}\Phi_{\mathcal{W}_G^*}(X_n, L_n)) \\
&\qquad + \tilde{l}(M_{\mathcal{T}}Y_n, M_{\mathcal{T}}\Phi_{\mathcal{W}_G^*}(X_n, L_n))| = \\
&\underbrace{\tfrac{(p-q)}{pq}|\tilde{l}(\mathcal{Y}, \Phi_{\mathcal{W}_G^*}(\mathcal{X}, \mathcal{L})) - \tilde{l}(M_{\mathcal{T}}Y_n, M_{\mathcal{T}}\Phi_{\mathcal{W}_G^*}(X_n, L_n))|}_{(3),\ \textbf{mult. by factor } \frac{(p-q)}{pq}} \\
&\qquad + \tfrac{(p-q)}{pq}|\tilde{l}(M_{\mathcal{T}}Y_n, M_{\mathcal{T}}\Phi_{\mathcal{W}_G^*}(X_n, L_n))|. \qquad (36)
\end{aligned}
$$

Finally, recapping the definition for the minimum semi-supervised training loss on the graph G as

$$l_G^* = \tilde{l}(M_{\mathcal{T}} Y_n, M_{\mathcal{T}} \Phi_{\mathcal{W}_G^*}(X_n, L_n)) \tag{37}$$

with some additional algebraic manipulation of the constant factors, we achieve the bound. □

D Proof of Theorem 4.4

D.1 Assumptions

We start by stating necessary assumptions.

Assumption D.1. *The convolutional maps in $\Phi_{\mathcal{W}}$ are locally Lipschitz on $\mathcal{M}$ and have norm at most* 1.

Assumption D.2. *The nonlinear function ρ and its first-order derivative ρ' have Lipschitz constant* 1. *Also, $\rho(0) = 0$.*

Assumption D.3. *The convolutions in all layers of $\Phi_{\mathcal{W}}$ are low-pass filters with bandwidth c. I.e., if $\mathcal{Y}$ is the output of a convolution, $\langle \mathcal{Y}, \phi_i \rangle = 0$ for $\lambda_i > c$, and $i_c = \arg\min_i (\lambda_i - c)\mathbf{1}(\lambda_i \geq c)$.*

Assumption D.4. *The sampling operator $\mathcal{P}_n$ has unitary norm.*

Assumption D.5. *Let $\tilde{\mathbf{l}} \in \mathbb{R}^n$ be such that $[\tilde{\mathbf{l}}]_i = n^{-1}\tilde{l}([Y]_i, [Y']_i)$ where $\tilde{l}$ is a standard loss function with Lipschitz constant* 1. *The semi-supervised loss function l is defined as $l(Y, Y') = n|\mathcal{T}|^{-1}(M_{\mathcal{T}}\tilde{\mathbf{l}})^T\mathbf{1}$ where $M_{\mathcal{T}} \in \{0,1\}^{|\mathcal{T}|\times n}$ is the training mask. Since $\sigma_{max}(M_{\mathcal{T}}) = 1$, l has Lipschitz constant $n/|\mathcal{T}|$, which is equal to ν^{-1} when $|\mathcal{T}| = \nu n$.*

D.1.1 Lemmas

We will also need the following lemmas adapted from [16].

Lemma D.6. *Let $\Phi_{\mathcal{W}}(\mathcal{X}, \mathcal{L})$ be an MNN with $F_\ell = F$ for $1 \leq \ell \leq \mathscr{L} - 1$ and $F_{\mathscr{L}} = 1$. Let $\Phi(X_n, L_n)$ be a GNN with same weights $\mathcal{W}$ on a geometric graph G_n sampled uniformly from $\mathcal{M}$ as in* (1)*. Under Assumptions D.1-D.4, with probability $1 - \delta$ it holds that*

$$\begin{aligned}
&\|\mathcal{P}_n \nabla_{\mathcal{W}} \Phi_{\mathcal{W}}(\mathcal{X}, \mathcal{L}) - \nabla_{\mathcal{W}} \Phi_{\mathcal{W}}(X_n, L_n)\| \\
&\leq 2\sqrt{(\mathscr{L}-1)KF^2 + KF}\,\mathscr{L}^3 F^{3\mathscr{L}-3}\left(C_1'\varepsilon + C_2'\sqrt{\frac{\log 1/\delta}{n}}\right) \qquad (38)\\
&\leq 2\sqrt{2(\mathscr{L}-1)K}\,\mathscr{L}^3 F^{3\mathscr{L}-2}\left(C_1'\varepsilon + C_2'\sqrt{\frac{\log 1/\delta}{n}}\right). \qquad (39)
\end{aligned}$$

Proof. We will first show that the gradient with respect to any arbitrary element $[W_{\ell k}]_{fg} \in \mathbb{R}$ of $\mathcal{W}$ can be uniformly bounded. Note that the maximum is attained if $\ell = \ell^\dagger = 1$. Without loss of generality, assuming $\ell^\dagger > \ell - 1$ and $\omega = [W_{\ell^\dagger k}]_{fg} \in \mathbb{R}$, we can begin by using the output of the MNN to write

$$\begin{aligned}
&\|\mathcal{P}_n \nabla_\omega \Phi(\mathcal{X}, \mathcal{L}) - \nabla_\omega \Phi(X_n, L_n)\| \\
&\qquad \leq \|\nabla_\omega \mathcal{P}_n \Phi(\mathcal{X}, \mathcal{L}) - \nabla_\omega \Phi(X_n, L_n)\| \\
&\qquad = \|\nabla_\omega [\mathcal{P}_n \mathcal{X}_{\mathscr{L}}]_f - \nabla_\omega [X_{n\mathscr{L}}]_f\| \\
&\qquad = \Bigg\| \nabla_\omega \rho\Bigg(\sum_{g=1}^{F_{\mathscr{L}-1}} \sum_{k=0}^{K-1} \mathcal{P}_n e^{-k\mathcal{L}} [\mathcal{X}_{\mathscr{L}-1}]_g [W_{\mathscr{L}k}]_{fg} \Bigg) \\
&\qquad\qquad - \nabla_\omega \rho\Bigg(\sum_{g=1}^{F_{\mathscr{L}-1}} \sum_{k=0}^{K-1} L_n^k [X_{n\mathscr{L}-1}]_g [W_{\mathscr{L}k}]_{fg} \Bigg) \Bigg\| \qquad (40)
\end{aligned}$$

where we have dropped the subscript $\mathcal{W}$ from Φ for simplicity.

Applying the chain rule and using the triangle inequality, we get

$$
\begin{aligned}
&\|\nabla_\omega[\mathcal{P}_n\mathcal{X}_{\mathscr{L}}]_f - \nabla_\omega[X_{n\mathscr{L}}]_f\| \\
&\leq \Bigg\| \Bigg(\nabla\rho\Bigg(\sum_{g=1}^{F_{\mathscr{L}-1}} \sum_{k=0}^{K-1} \mathcal{P}_n e^{-k\mathcal{L}} [\mathcal{X}_{\mathscr{L}-1}]_g [W_{\mathscr{L}k}]_{fg} \Bigg) \\
&\qquad - \nabla\rho\Bigg(\sum_{g=1}^{F_{\mathscr{L}-1}} \sum_{k=0}^{K-1} L_n^k [X_{n\mathscr{L}-1}]_g [W_{\mathscr{L}k}]_{fg} \Bigg) \Bigg) \\
&\qquad \times \mathcal{P}_n \nabla_\omega \Bigg(\sum_{g=1}^{F_{\mathscr{L}-1}} \sum_{k=0}^{K-1} e^{-k\mathcal{L}} [\mathcal{X}_{\mathscr{L}-1}]_g [W_{\mathscr{L}k}]_{fg} \Bigg) \Bigg\| \\
&+ \Bigg\| \nabla\rho\Bigg(\sum_{g=1}^{F_{\mathscr{L}-1}} \sum_{k=0}^{K-1} L_n^k [X_{n\mathscr{L}-1}]_g [W_{\mathscr{L}k}]_{fg} \Bigg) \\
&\qquad \times \Bigg(\nabla_\omega \sum_{g=1}^{F_{\mathscr{L}-1}} \sum_{k=0}^{K-1} \mathcal{P}_n e^{-k\mathcal{L}} [\mathcal{X}_{\mathscr{L}-1}]_g [W_{\mathscr{L}k}]_{fg} \\
&\qquad\qquad - \nabla_\omega \sum_{g=1}^{F_{\mathscr{L}-1}} \sum_{k=0}^{K-1} L_n^k [X_{n\mathscr{L}-1}]_g [W_{\mathscr{L}k}]_{fg} \Bigg) \Bigg\|. \qquad (41)
\end{aligned}
$$

Next, we use Cauchy-Schwarz inequality, Assumptions D.2 and D.4, and [16][Lemma 2, adapted to MNNs] to bound the terms corresponding to the gradient of the nonlinearity ρ and the norm of the MNN respectively. Explicitly,

$$
\begin{aligned}
&\|\nabla_\omega[\mathcal{P}_n\mathcal{X}_{\mathscr{L}}]_f - \nabla_\omega[X_{n\mathscr{L}}]_f\| \\
&\leq \Bigg\| \sum_{g=1}^{F_{\mathscr{L}-1}} \sum_{k=0}^{K-1} \mathcal{P}_n e^{-k\mathcal{L}} [\mathcal{X}_{\mathscr{L}-1}]_g [W_{\mathscr{L}k}]_{fg} \\
&\qquad - \sum_{g=1}^{F_{\mathscr{L}-1}} \sum_{k=0}^{K-1} L_n^k [X_{n\mathscr{L}-1}]_g [W_{\mathscr{L}k}]_{fg} \Bigg\| F^{\mathscr{L}-1} \|\mathcal{X}\| \\
&+ \Bigg\| \sum_{g=1}^{F_{\mathscr{L}-1}} \nabla_\omega \sum_{k=0}^{K-1} \mathcal{P}_n \Bigg(e^{-k\mathcal{L}} [\mathcal{X}_{\mathscr{L}-1}]_g [W_{\mathscr{L}k}]_{fg} \\
&\qquad - L_n^k [X_{n\mathscr{L}-1}]_g [W_{\mathscr{L}k}]_{fg}) \Bigg) \Bigg\| \qquad (42)
\end{aligned}
$$

Applying the triangle inequality to the second term, we get

$$
\begin{aligned}
&\|\nabla_\omega[\mathcal{P}_n\mathcal{X}_{\mathscr{L}}]_f - \nabla_\omega[X_{n\mathscr{L}}]_f\| \\
&\leq \Bigg\| \sum_{g=1}^{F_{\mathscr{L}-1}} \sum_{k=0}^{K-1} \mathcal{P}_n e^{-k\mathcal{L}}[\mathcal{X}_{\mathscr{L}-1}]_g [W_{\mathscr{L}k}]_{fg} \\
&\qquad - \sum_{g=1}^{F_{\mathscr{L}-1}} \sum_{k=0}^{K-1} L_n^k [X_{n\mathscr{L}-1}]_g [W_{\mathscr{L}k}]_{fg} \Bigg\| F^{\mathscr{L}-1} \|\mathcal{X}\| \\
&+ \Bigg\| \sum_{g=1}^{F_{\mathscr{L}-1}} \nabla_\omega \sum_{k=0}^{K-1} \Bigg(\mathcal{P}_n e^{-k\mathcal{L}}[W_{\mathscr{L}k}]_{fg} \\
&\qquad - L_n^k \mathcal{P}_n [W_{\mathscr{L}k}]_{fg} \Bigg) [\mathcal{X}_{\mathscr{L}-1}]_g \Bigg\| \\
&+ \sum_{g=1}^{F_{\mathscr{L}-1}} \Bigg\| \nabla_\omega \sum_{k=0}^{K-1} L_n^k \Bigg([\mathcal{P}_n \mathcal{X}_{\mathscr{L}-1}]_g - [X_{n\mathscr{L}-1}]_g \Bigg) [W_{\mathscr{L}k}]_{fg}) \Bigg\|. \qquad (43)
\end{aligned}
$$

Now note that as we consider the case in which $\ell_\dagger < \mathscr{L} - 1$, using the Cauchy-Schwarz inequality we can use the same bound for the first and second terms on the right hand side of (43). Also note that, by Assumption D.1, the filters are non-expansive, which allows us to write

$$
\begin{aligned}
&\|\nabla_\omega[\mathcal{P}_n\mathcal{X}_{\mathscr{L}}]_f - \nabla_\omega[X_{n\mathscr{L}}]_f\| \\
&\leq \Bigg\| \sum_{g=1}^{F_{\mathscr{L}-1}} \sum_{k=0}^{K-1} \mathcal{P}_n e^{-k\mathcal{L}}[\mathcal{X}_{\mathscr{L}-1}]_g [W_{\mathscr{L}k}]_{fg} \\
&\qquad - \sum_{g=1}^{F_{\mathscr{L}-1}} \sum_{k=0}^{K-1} L_n^k [X_{n\mathscr{L}-1}]_g [W_{\mathscr{L}k}]_{fg} \Bigg\| F^{\mathscr{L}-1} \|\mathcal{X}\| \\
&+ \Bigg\| \sum_{g=1}^{F_{\mathscr{L}-1}} \sum_{k=0}^{K-1} \mathcal{P}_n e^{-k\mathcal{L}}[W_{\mathscr{L}k}]_{fg} - L_n^k \mathcal{P}_n [W_{\mathscr{L}k}]_{fg} \Bigg\| F^{\mathscr{L}-1} \|\mathcal{X}\| \\
&+ \sum_{g=1}^{F_{\mathscr{L}-1}} \Bigg\| \nabla_\omega \Bigg([\mathcal{P}_n \mathcal{X}_{\mathscr{L}-1}]_g - [X_{n\mathscr{L}-1}]_g \Bigg) \Bigg\|. \qquad (44)
\end{aligned}
$$

The only term that remains to bound has the exact same bound derived in (40), but on the previous layer $\mathscr{L} - 2$. Hence, by applying the same steps $\mathscr{L} - 2$ times, we can obtain a bound for any element ω of tensor $\mathcal{H}$.

$$
\begin{aligned}
&\|\nabla_\omega[\mathcal{P}_n\mathcal{X}_{\mathscr{L}}]_f - \nabla_\omega[X_{n\mathscr{L}}]_f\| \\
&\leq \mathscr{L} F^{\mathscr{L}-2} \Bigg\| \sum_{g=1}^{F_{\mathscr{L}-1}} \sum_{k=0}^{K-1} \mathcal{P}_n e^{-k\mathcal{L}}[\mathcal{X}_{\mathscr{L}-1}]_g [W_{\mathscr{L}k}]_{fg} \\
&\qquad\qquad - \sum_{g=1}^{F_{\mathscr{L}-1}} \sum_{k=0}^{K-1} L_n^k [X_{n\mathscr{L}-1}]_g [W_{\mathscr{L}k}]_{fg} \Bigg\| F^{\mathscr{L}-1} \|\mathcal{X}\| \\
&+ \mathscr{L} F^{\mathscr{L}-2} \Bigg\| \sum_{g=1}^{F_{\mathscr{L}-1}} \sum_{k=0}^{K-1} \mathcal{P}_n e^{-k\mathcal{L}}[W_{\mathscr{L}k}]_{fg} \\
&\qquad\qquad - L_n^k \mathcal{P}_n [W_{\mathscr{L}k}]_{fg} \Bigg\| F^{\mathscr{L}-1} \|\mathcal{X}\| \\
&+ \sum_{g=1}^{F_{\mathscr{L}-1}} \Bigg\| \nabla_\omega \Bigg([\mathcal{P}_n \mathcal{X}_1]_g - [X_{n1}]_g \Bigg) \Bigg\|. \qquad (45)
\end{aligned}
$$

Note that the two first terms on the right hand side can be upper bounded by Prop. 3.1. For the third term, the derivative of a convolutional filter at coefficient $k^\dagger = i$ is itself a convolutional filter with coefficients $[w_i]_{fg}$. The values of $[w_i]_{fg}$ are 1 if $j = i$ and 0 otherwise. Additionally, this new filter also verifies Assumption D.1, as $\mathcal{X}$ is bandlimited. Denote this filter Φ_w. Considering that $\ell^\dagger = 1$, and using [70][Prop. 2], [65][Thm. 19] , together with the fact that $\mathcal{X}$ is bandlimited and the triangle inequality, with probability $1 - \delta$ we have

$$\begin{aligned}
&\left\| \Phi_w(X_n, L_n) - \mathcal{P}_n \Phi_w(\mathcal{X}, \mathcal{L}) \right\| \\
&\leq \|\lambda_c - \lambda_{cn}\| \|\mathcal{X}\| + \|X_n - \mathcal{P}_n \mathcal{X}\| \qquad (46) \\
&\leq \sqrt{F} C_{\mathcal{M},1} \lambda_c \varepsilon + \sqrt{F} C_3 \sqrt{\frac{\log 1/\delta}{n}} \qquad (47)
\end{aligned}$$

where we have assumed each feature in $\mathcal{X}$ has unit norm at most. Now, substituting the third term in (45) for (46), and using Prop. 3.1 for the first two terms, with probability $1 - \delta$, we have

$$\begin{aligned}
&\|\nabla_\omega [\mathcal{P}_n \mathcal{X}_{\mathscr{L}}]_f - \nabla_\omega [X_{n\mathscr{L}}]_f\| \\
&\leq 2\mathscr{L}^3 F^{3\mathscr{L}-3} \left(C_1 \varepsilon + C_2 \sqrt{\frac{\log 1/\delta}{n}} \right) \\
&+ F\sqrt{F} C_{\mathcal{M},1} \lambda_c \varepsilon + F\sqrt{F} C_3 \sqrt{\frac{\log 1/\delta}{n}} \qquad (48)
\end{aligned}$$

To achieve the final result, note that the set $\mathcal{W}$ has $(\mathscr{L} - 1)KF^2 + KF$ elements, and each element is upper bounded by (48). □

Lemma D.7. *Let $\Phi_{\mathcal{W}}(\mathcal{X}, \mathcal{L})$ be an MNN with $F_\ell = F$ for $0 \leq \ell \leq \mathscr{L} - 1$ and $F_{\mathscr{L}} = 1$, and. Let $\Phi_{\mathcal{W}}(X_n, L_n)$ be a GNN with same weights $\mathcal{W}$ on a geometric graph G_n sampled uniformly from $\mathcal{M}$ as in* (1)*. Under Assumptions D.1–D.5, with probability $1 - \delta$ it holds that*

$$\begin{aligned}
&\|\nabla_{\mathcal{W}} l(\mathcal{Y}, \Phi_{\mathcal{W}}(\mathcal{X}, \mathcal{L})) - \nabla_{\mathcal{W}} l(Y_n, \Phi(X_n, L_n))\| \\
&\leq 2\nu^{-1} \sqrt{(\mathscr{L} - 1)KF^2 + KF}\, \mathscr{L}^3 F^{3\mathscr{L}-3} \left(C_1'' \varepsilon \right. \\
&\quad \left. + C_2'' \sqrt{\frac{\log 1/\delta}{n}} \right) \qquad (49) \\
&\leq 2\nu^{-1} \sqrt{2(\mathscr{L} - 1)K}\, \mathscr{L}^3 F^{3\mathscr{L}-2} \left(C_1'' \varepsilon + C_2'' \sqrt{\frac{\log 1/\delta}{n}} \right). \qquad (50)
\end{aligned}$$

Proof. In order to analyze the norm of the gradient with respect to the tensor $\mathcal{H}$, we start by taking the derivative with respect to a single element of the tensor, ω, as in the proof of the previous lemma. Also as before, we drop subscript $\mathcal{W}$ in Φ. Using the chain rule to compute the gradient of the loss function l, we get

$$\begin{aligned}
&\|\nabla_\omega (l(\mathcal{P}_n \mathcal{Y}, \mathcal{P}_n \Phi(\mathcal{X}, \mathcal{L})) - l(Y_n, \Phi(X_n, L_n)))\| \\
&= \|\nabla l(\mathcal{P}_n \mathcal{Y}, \mathcal{P}_n \Phi(\mathcal{X}, \mathcal{L})) \nabla_\omega \mathcal{P}_n \Phi(\mathcal{X}, \mathcal{L}) \\
&\quad - \nabla l(Y_n, \Phi(X_n, L_n)) \nabla_\omega \Phi(X_n, L_n)\| \qquad (51)
\end{aligned}$$

and by the Cauchy-Schwarz and the triangle inequalities, it holds

$$\begin{aligned}
&\|\nabla_\omega (l(\mathcal{P}_n \mathcal{Y}, \mathcal{P}_n \Phi(\mathcal{X}, \mathcal{L})) - l(Y_n, \Phi(X_n, L_n)))\| \\
&\leq \|\nabla l(\mathcal{P}_n \mathcal{Y}, \mathcal{P}_n \Phi(X, \mathcal{L})) \\
&\quad - \nabla l(Y_n, \Phi(X_n, L_n))\| \|\nabla_\omega \mathcal{P}_n \Phi(\mathcal{X}, \mathcal{L})\| \\
&+ \|\nabla l(Y_n, \Phi(X_n, L_n))\| \|\nabla_\omega \mathcal{P}_n \Phi(\mathcal{X}, \mathcal{L}) - \nabla_\omega \Phi(X_n, L_n)\| \qquad (52)
\end{aligned}$$

By the triangle inequality and Assumption D.5, it follows

$$\begin{aligned}
&\|\nabla_\omega(l(\mathcal{P}_n\mathcal{Y},\mathcal{P}_n\Phi(\mathcal{X},\mathcal{L}))-(Y_n,\Phi(X_n,L_n)))\| \\
&\leq \|\nabla l(\mathcal{P}_n\mathcal{Y},\mathcal{P}_n\Phi(\mathcal{X},\mathcal{L}))-\nabla l(\mathcal{P}_n\mathcal{Y},\Phi(X_n,L_n))\| \\
&\quad\times\|\nabla_\omega\mathcal{P}_n\Phi(\mathcal{X},\mathcal{L})\|\|\nabla l(Y_n,\Phi(X_n,L_n)) \\
&\qquad-\nabla\ell(\mathcal{P}_nY,\Phi(X_n,L_n))\| \\
&\quad\times\|\nabla_\omega\mathcal{P}_n\Phi(\mathcal{X},\mathcal{L})\|+\|\nabla_\omega(\mathcal{P}_n\Phi(\mathcal{X},\mathcal{L})-\Phi(X_n,L_n))\| \qquad (53)\\
&\leq \nu^{-1}\Bigg(\|Y_n-\mathcal{P}_n\mathcal{Y}\| \\
&\qquad\qquad+\|\Phi(X_n,L_n)-\mathcal{P}_n\Phi(\mathcal{X},\mathcal{L})\|\Bigg)\|\nabla_\omega\mathcal{P}_n\Phi(\mathcal{X},\mathcal{L})\| \\
&\quad+\|\nabla_\omega(\mathcal{P}_n\Phi(\mathcal{X},\mathcal{L})-\Phi(X_n,L_n))\|. \qquad (54)
\end{aligned}$$

Next, we can use [16][Lemma 2, adapted to MNNs], Prop. 3.1, Lemma D.6, and [65][Thm. 19] to obtain

$$\begin{aligned}
&\|\nabla_\omega(l(\mathcal{P}_n\mathcal{Y},\mathcal{P}_n\Phi(\mathcal{X},\mathcal{L}))-l(Y_n,\Phi(X_n,L_n)))\| \\
&\leq \nu^{-1}\Bigg(C_5\sqrt{\frac{\log 1/\delta}{n}} \\
&\qquad\qquad+\mathscr{L}F^{\mathscr{L}-2}\Bigg(C_1\varepsilon+C_2\sqrt{\frac{\log 1/\delta}{n}}\Bigg)\Bigg)F^{\mathscr{L}-1}\sqrt{F} \\
&\quad+2\nu^{-1}\mathscr{L}^3F^{3\mathscr{L}-3}\Bigg(C_1'\varepsilon+C_2'\sqrt{\frac{\log 1/\delta}{n}}\Bigg) \qquad (55)
\end{aligned}$$

where we also assume $\|\mathcal{X}\|\leq\sqrt{F}$.

Finally, when $\tilde{l}$ is the 2-norm we can use [65][Thm. 19] to show:

$$\begin{aligned}
&\|\nabla_\omega(l(\mathcal{Y},\Phi(\mathcal{X},\mathcal{L}))-l(Y_n,\Phi(X_n,L_n)))\| \\
&\leq \|\nabla_\omega(l(\mathcal{Y},\Phi(\mathcal{X},\mathcal{L}))-l(\mathcal{P}_n\mathcal{Y},\mathcal{P}_n\Phi(\mathcal{X},\mathcal{L})))\| \\
&\quad+\|\nabla_\omega(l(\mathcal{P}_n\mathcal{Y},\mathcal{P}_n\Phi(\mathcal{X},\mathcal{L}))-l(Y_n,\Phi(X_n,L_n)))\| \qquad (56)\\
&\leq \nu^{-1}\Bigg(\tilde{C}_5\sqrt{\frac{\log 1/\delta}{n}} \\
&\qquad\qquad+\mathscr{L}F^{\mathscr{L}-2}\Bigg(C_1\varepsilon+\tilde{C}_2\sqrt{\frac{\log 1/\delta}{n}}\Bigg)\Bigg)F^{\mathscr{L}-1}\sqrt{F} \\
&\quad+2\nu^{-1}\mathscr{L}^3F^{3\mathscr{L}-3}\Bigg(C_1'\varepsilon+\tilde{C}_2'\sqrt{\frac{\log 1/\delta}{n}}\Bigg). \qquad (57)
\end{aligned}$$

Noting that tensor $\mathcal{W}$ has $(\mathscr{L}-1)KF^2+KF$ elements, and that each individual term can be bounded by (55), we arrive at the desired result. □

Consider the ERM problem in (5) and let $\Phi_{\mathcal{W}}(\mathcal{X},\mathcal{L})$ be an $\mathscr{L}$-layer MNN with $F_\ell=F$ for $0\leq\ell\leq L-1$ and $F_{\mathscr{L}}=1$. Let $\Phi(X_n,L_n)$ be a GNN with same weights $\mathcal{W}$ on a geometric graph G_n sampled uniformly from $\mathcal{M}$ as in (1). Under Assumptions D.1–D.5, it holds

$$\begin{aligned}
&\mathbb{E}[\|\nabla_{\mathcal{W}}l(\mathcal{Y},\Phi_{\mathcal{W}}(\mathcal{X},\mathcal{L}))-\nabla_{\mathcal{W}}l(Y_n,\Phi_{\mathcal{W}}(X_n,L_n))\|] \\
&=\mathcal{O}\Bigg(\gamma\Bigg(\varepsilon+\sqrt{\frac{\log n}{n}}\Bigg)\Bigg) \qquad (58)
\end{aligned}$$

where γ is a constant that depends on the number of layers L, features F, and filter taps K of the GNN.

Proof. To start, consider the event A_n such that

$$\|\nabla_{\mathcal{W}} l(\mathcal{P}_n\mathcal{Y}, \mathcal{P}_n\Phi(\mathcal{X},\mathcal{L})) - \nabla_{\mathcal{W}} l(Y_n, \Phi(X_n, L_n))\| \\ \leq 2\sqrt{2(\mathscr{L}-1)K}\mathscr{L}^3 F^{3\mathscr{L}-2}\left(C_1''\varepsilon + C_2''\sqrt{\frac{\log 1/\delta}{n}}\right) \tag{59}$$

where we have dropped the subscript $\mathcal{W}$ where it is clear from context. Taking the disjoint events A_n and A_n^c, and denoting the indicator function $\mathbf{1}(\cdot)$, we split the expectation as

$$\mathbb{E}[\|\nabla_{\mathcal{W}} l(\mathcal{Y}, \Phi(\mathcal{X},\mathcal{L})) - \nabla_{\mathcal{W}} l(Y_n, \Phi(X_n, L_n))\|] \\ = \mathbb{E}[\|\nabla_{\mathcal{W}}(l(\mathcal{Y}, \Phi(\mathcal{X},\mathcal{L})) - (Y_n, \Phi(X_n, L_n)))\|\mathbf{1}(A_n)] \\ + \mathbb{E}[\|\nabla_{\mathcal{W}} l(\mathcal{Y}, \Phi(\mathcal{X},\mathcal{L})) - \nabla_{\mathcal{W}} l(Y_n, \Phi(X_n, L_n))\|\mathbf{1}(A_n^c)] \tag{60}$$

We can then bound the term corresponding to A_n^c using the chain rule, the Cauchy-Schwarz inequality, Assumption D.5, and [16][Lemma 2, adapted to MNNs] as follows

$$\|\nabla_{\mathcal{W}} l(\mathcal{Y}, \Phi(\mathcal{X},\mathcal{L})) - \nabla_{\mathcal{W}} l(Y_n, \Phi(X_n, L_n))\| \\ \leq \|\nabla_{\mathcal{W}} l(\mathcal{Y}, \Phi(\mathcal{X}\mathcal{L}))\| + \|\nabla_{\mathcal{W}} l(Y_n, , L_n))\| \tag{61}$$

$$\leq \|\nabla l(\mathcal{Y}, \Phi(\mathcal{X}\mathcal{L}))\|\|\nabla_{\mathcal{W}}\Phi(\mathcal{X},\mathcal{L})\| \\ + \|\nabla l(Y_n, \Phi(X_n, L_n))\|\|\nabla_{\mathcal{W}}\Phi(X_n, L_n)\| \tag{62}$$

$$\leq \|\nabla_{\mathcal{W}}\Phi(\mathcal{X},\mathcal{L})\| + \|\nabla_{\mathcal{W}}\Phi(X_n, L_n)\| \tag{63}$$

$$\leq 2F^{\mathscr{L}}\sqrt{(\mathscr{L}-1)KF + K}. \tag{64}$$

Going back to (60), we can substitute the bound obtained in (64), take $P(A_n) = 1-\delta$, and use Lemma D.7 to get

$$\mathbb{E}[\|\nabla_{\mathcal{W}} l(\mathcal{Y}, \Phi(\mathcal{X},\mathcal{L})) - \nabla_{\mathcal{W}} l(Y_n, \Phi(X_n, L_n))\|] \\ \leq \delta 2F^{\mathscr{L}}\sqrt{(\mathscr{L}-1)KF + K} \\ + (1-\delta)2\nu^{-1}\sqrt{2(\mathscr{L}-1)K}\mathscr{L}^3 F^{3\mathscr{L}-2}\Bigg(C_1''\varepsilon \\ + C_2''\sqrt{\frac{\log 1/\delta}{n}}\Bigg). \tag{65}$$

Setting $\delta = \frac{1}{\sqrt{n}}$ completes the proof. □

Assumption D.8. *The MNN $\Phi_{\mathcal{W}}(\mathcal{X},\mathcal{L})$ is α-Lipschitz, and its gradient $\nabla_{\mathcal{W}}\Phi_{\mathcal{W}}(\mathcal{X},\mathcal{L})$ is β-Lipschitz, with respect to the parameters $\mathcal{W}$.*

Lemma D.9. *Consider the ERM problem in* (5) *and let $\Phi_{\mathcal{W}}(\mathcal{X},\mathcal{L})$ be an $\mathscr{L}$-layer MNN with $F_\ell = F$ for $0 \leq \ell \leq L-1$ and $F_{\mathscr{L}} = 1$. Fix $\epsilon > 0$ and step size $\eta < \theta^{-1}$, with $\theta = \alpha + \beta F\sqrt{2K(\mathscr{L}-1)}$. Let $\Phi(X_n, L_n)$ be a GNN with same weights $\mathcal{W}$ on a geometric graph G_n sampled uniformly from $\mathcal{M}$ as in* (1)*. Consider the iterates generated by* (15)*. Under Assumptions D.1–D.8, if at step k of epoch e the number of nodes $n(e)$ verifies*

$$\mathbb{E}[\|\nabla_{\mathcal{W}} l(\mathcal{Y}, \Phi_{\mathcal{W}_k}(\mathcal{X},\mathcal{L})) - \nabla_{\mathcal{W}} l(Y_{n(e)}, \Phi_{\mathcal{W}_k}(X_{n(e)}, L_{n(e)}))\|] \\ \leq \|\nabla_{\mathcal{W}} l(\mathcal{Y}, \Phi_{\mathcal{W}_k}(\mathcal{X},\mathcal{L}))\| \tag{66}$$

then the iterate generated by graph learning step (15) *satisfies*

$$\mathbb{E}[l(\mathcal{Y}, \Phi_{\mathcal{W}_{k+1}}(\mathcal{X},\mathcal{L}))] \leq l(\mathcal{Y}, \Phi_{\mathcal{W}_k}(\mathcal{X},\mathcal{L})). \tag{67}$$

Proof. To start, we do as in [10], i.e., we define a continuous function $g(\epsilon)$ that at $\epsilon = 1$ takes the value of the loss function on $\mathcal{M}$ at iteration $k+1$, and at $\epsilon = 0$, the value at iteration k. Explicitly,

$$g(\epsilon) = l(\mathcal{Y}, \Phi_{\mathcal{W}_k - \epsilon\eta_k\nabla_{\mathcal{W}} l(Y_n, \Phi_{\mathcal{W}_k}(X_n, L_n))}(\mathcal{X},\mathcal{L})). \tag{68}$$

Function $g(\epsilon)$ is evaluated on the manifold data $\mathcal{Y}, \mathcal{X}, \mathcal{L}$, but the steps are controlled by the graph data Y_n, X_n, L_n. Applying the chain rule, the derivative of $g(\epsilon)$ with respect to ϵ can be written as

$$\frac{\partial g(\epsilon)}{\partial \epsilon} = -\eta_k \nabla_{\mathcal{W}} l(Y_n, \Phi_{\mathcal{W}_k}(X_n, L_n)) \\ \times \nabla_{\mathcal{W}} l(\mathcal{Y}, \Phi_{\mathcal{W}_k - \epsilon\eta_k\nabla_{\mathcal{W}} l(Y_n, \Phi_{\mathcal{W}_k}(X_n, L_n))}(\mathcal{X},\mathcal{L})). \tag{69}$$

Between iterations $k+1$ and k, the difference in the loss function l can be written as the difference between $g(1)$ and $g(0)$,

$$g(1) - g(0) = l(\mathcal{Y}, \Phi_{\mathcal{W}_{k+1}}(\mathcal{X}, \mathcal{L})) - l(\mathcal{Y}, \Phi_{\mathcal{W}_k}(\mathcal{X}, \mathcal{L})). \tag{70}$$

Integrating the derivative of $g(\epsilon)$ in $[0, 1]$, we get

$$\begin{aligned}
&l(\mathcal{Y}, \Phi_{\mathcal{W}_{k+1}}(\mathcal{X}, \mathcal{L})) - l(\mathcal{Y}, \Phi_{\mathcal{W}_k}(\mathcal{X}, \mathcal{L})) = g(1) - g(0) \\
&= \int_0^1 \frac{\partial g(\epsilon)}{\partial \epsilon} d\epsilon \\
&= -\int_0^1 \eta_k \nabla_{\mathcal{W}} l(Y_n, \Phi_{\mathcal{W}_k}(X_n, L_n)) \\
&\qquad \times \nabla_{\mathcal{W}} l(\mathcal{Y}, \Phi_{\mathcal{W}_k - \epsilon\eta_k \nabla_{\mathcal{W}} l(Y_n, \Phi_{\mathcal{W}_k}(X_n, L_n))}(\mathcal{X}, \mathcal{L})) d\epsilon.
\end{aligned} \tag{71}$$

Now note that the last term of the integral does not depend on ϵ. Thus, we can proceed by adding and subtracting $\nabla_{\mathcal{H}} l(Y, \Phi(\mathcal{H}_k, \mathcal{L}, X))$ inside the integral to get

$$\begin{aligned}
&l(\mathcal{Y}, \Phi_{\mathcal{W}_{k+1}}(\mathcal{X}, \mathcal{L})) - l(Y, \Phi(X; \mathcal{H}_k, \mathcal{L})) \\
&= -\eta_k \nabla_{\mathcal{W}} l(Y_n, \Phi_{\mathcal{W}_k}(X_n, L_n)) \\
&\times \int_0^1 \nabla l(\mathcal{Y}, \Phi_{\mathcal{W}_k - \epsilon\eta_k \nabla l(Y_n, \Phi_{\mathcal{W}_k}(X_n, L_n))}(\mathcal{X}, \mathcal{L})) \\
&\quad + \nabla_{\mathcal{W}} l(\mathcal{Y}, \Phi_{\mathcal{W}_k}(\mathcal{X}, \mathcal{L})) - \nabla_{\mathcal{W}} l(\mathcal{Y}, \Phi_{\mathcal{W}_k}(\mathcal{X}, \mathcal{L})) d\epsilon \\
&= -\eta_k \nabla_{\mathcal{W}} l(Y_n, \Phi_{\mathcal{W}_k}(X_n, L_n)) \nabla_{\mathcal{W}} l(\mathcal{Y}, \Phi_{\mathcal{W}_k}(\mathcal{X}, \mathcal{L})) \\
&- \eta_k \nabla_{\mathcal{W}} l(Y_n, \Phi_{\mathcal{W}_k}(X_n, L_n)) \\
&\quad \times \int_0^1 \nabla_{\mathcal{W}} l(\mathcal{Y}, \Phi_{\mathcal{W}_k - \epsilon\eta_k \nabla l(Y_n, \Phi_{\mathcal{W}_k}(X_n, L_n))}(\mathcal{X}, \mathcal{L})) \\
&\qquad - \nabla l(\mathcal{Y}, \Phi_{\mathcal{W}_k}(\mathcal{X}, \mathcal{L})) d\epsilon.
\end{aligned} \tag{72}$$

Next, we can apply the Cauchy-Schwarz inequality to the last term of (72) and take the norm of the integral (which is smaller that the integral of the norm), to obtain

$$\begin{aligned}
&l(\mathcal{Y}, \Phi_{\mathcal{W}_{k+1}}(\mathcal{X}, \mathcal{L})) - l(\mathcal{Y}, \Phi_{\mathcal{W}_k}(\mathcal{X}, \mathcal{L})) \\
&\leq -\eta_k \nabla_{\mathcal{W}} l(Y_n, \Phi_{\mathcal{W}_k}(X_n, L_n)) \nabla_{\mathcal{W}} l(\mathcal{Y}, \Phi_{\mathcal{W}_k}(\mathcal{X}, \mathcal{L})) \\
&+ \eta_k \|\nabla_{\mathcal{W}} l(Y_n, \Phi_{\mathcal{W}_k}(X_n, L_n))\| \\
&\quad \times \int_0^1 \|\nabla_{\mathcal{W}} l(\mathcal{Y}, \Phi_{\mathcal{W}_k}(\mathcal{X}, \mathcal{L})) \\
&\qquad - \nabla l(\mathcal{Y}, \Phi_{\mathcal{W}_k - \epsilon\eta_k \nabla l(Y_n, \Phi_{\mathcal{W}_k}(X_n, L_n))}(\mathcal{X}, \mathcal{L}))\| d\epsilon.
\end{aligned} \tag{73}$$

By [16][Lemma 6, adapted to MNNs], we use θ to write

$$\begin{aligned}
&l(\mathcal{Y}, \Phi_{\mathcal{W}_{k+1}}(\mathcal{X}, \mathcal{L})) - l(\mathcal{Y}, \Phi_{\mathcal{W}_k}(\mathcal{X}, \mathcal{L})) \\
&\leq -\eta_k \nabla_{\mathcal{W}} l(Y_n, \Phi_{\mathcal{W}_k}(X_n, Ln)) \nabla_{\mathcal{W}} l(\mathcal{Y}, \Phi_{\mathcal{W}_k}(\mathcal{X}, \mathcal{L})) \\
&+ \theta\eta_k \|\nabla_{\mathcal{W}} l(Y_n, \Phi_{\mathcal{W}_k}(X_n, L_n))\| \\
&\quad \times \int_0^1 \left\| \eta_k \nabla_{\mathcal{W}} l(Y_n, \Phi_{\mathcal{W}_n}(X_n, L_n)) \right\| \epsilon d\epsilon
\end{aligned} \tag{74}$$

$$\begin{aligned}
&\leq -\eta_k \nabla_{\mathcal{W}} l(Y_n, \Phi_{\mathcal{W}_k}(X_n, L_n)) \nabla_{\mathcal{W}} l(\mathcal{Y}, \Phi_{\mathcal{W}_k}(\mathcal{X}, \mathcal{L})) \\
&\quad + \frac{\eta_k^2 \theta}{2} \|\nabla_{\mathcal{W}} l(Y_n, \Phi_{\mathcal{W}_k}(X_n, L_n))\|^2.
\end{aligned} \tag{75}$$

Factoring out η_k, we get

$$\begin{aligned} & l(\mathcal{Y}, \Phi_{\mathcal{W}_{k+1}}(\mathcal{X}, \mathcal{L})) - l(\mathcal{Y}, \Phi_{\mathcal{W}_k}(\mathcal{X}, \mathcal{L})) \\ & \leq -\frac{\eta_k}{2}\Big(- \|\nabla_{\mathcal{W}} l(Y_n, \Phi_{\mathcal{W}_k}(X_n, L_n))\|^2 \\ & \quad + 2\nabla_{\mathcal{W}} l(Y_n, \Phi_{\mathcal{W}_k}(X_n, L_n))^T \nabla_{\mathcal{W}} l(\mathcal{Y}, \Phi_{\mathcal{W}_k}(\mathcal{X}, \mathcal{L}))\Big) \\ & + \frac{\eta_k^2\theta - \eta_k}{2}\|\nabla_{\mathcal{W}} l(Y_n, \Phi_{\mathcal{W}_k}(X_n, L_n))\|^2. \end{aligned} \tag{76}$$

Given that the norm is induced by the vector inner product in Euclidean space, for any two vectors A, B, $||A - B||^2 - ||B||^2 = ||A||^2 - 2\langle A, B\rangle$. Hence,

$$\begin{aligned} & l(\mathcal{Y}, \Phi_{\mathcal{W}_{k+1}}(\mathcal{X}, \mathcal{L})) - l(\mathcal{Y}, \Phi_{\mathcal{W}_k}(\mathcal{X}, \mathcal{L})) \\ & \leq \frac{-\eta_k}{2}\big(\|\nabla_{\mathcal{W}} l(\mathcal{Y}, \Phi_{\mathcal{W}_k}(\mathcal{X}, \mathcal{L}))\|^2 \\ & \quad - ||\nabla_{\mathcal{W}} l(Y_n, \Phi_{\mathcal{W}_k}(X_n, L_n)) - \nabla_{\mathcal{W}} l(\mathcal{Y}, \Phi_{\mathcal{W}_k}(\mathcal{X}, \mathcal{L}))||^2\big) \\ & + \frac{\eta_k^2\theta - \eta_k}{2}\|\nabla_{\mathcal{W}} l(Y_n, \Phi_{\mathcal{W}_k}(X_n, L_n))\|^2. \end{aligned} \tag{77}$$

Considering the first term on the right hand side, we know that the norm of the expected difference between the gradients is bounded by Prop. D.1.1. Given that norms are positive, the inequality still holds when the elements are squared (if $a > b, a \in \mathbb{R}_+, b \in \mathbb{R}_+$, then $a^2 > b^2$). Considering the second term on the right hand side, we impose the condition that $\eta_k < \frac{1}{\theta}$, which makes this term negative. Taking the expected value over all the nodes completes the proof. □

Theorem D.10. *Under **Setup**, let $\Phi_{\mathcal{W}}$ be a GNN learned with iterates* (15)*. If at each step k the number of nodes n_t is such that*

$$\begin{aligned} \mathbb{E}[\|\nabla_{\mathcal{W}} \tilde{l}(\mathcal{Y}, \Phi_{\mathcal{W}_k}(\mathcal{X}, \mathcal{L})) - \nabla_{\mathcal{W}} l(Y_{n_t}, \Phi_{\mathcal{W}_k}(X_{n_t}, L_{n_t}))\|] \\ < \|\nabla_{\mathcal{W}} \tilde{l}(\mathcal{Y}, \Phi_{\mathcal{W}_k}(\mathcal{X}, \mathcal{L}))\| - \epsilon \end{aligned} \tag{78}$$

then after at most $k^* = \mathcal{O}(1/\epsilon^2)$ iterations $\Phi_{\mathcal{W}^*_{G_{n_t}}} = \Phi_{\mathcal{W}_{k^*}}$ is within an ϵ-neighborhood of the solution of the empirical risk minimization problem on $\mathcal{M}$.

Proof. For every $\epsilon > 0$, we define the stopping time k^* as

$$k^* := \min_{k \geq 0}\{\mathbb{E}[\|\nabla_{\mathcal{W}} \tilde{l}(Y, \Phi_{\mathcal{W}_k}(\mathcal{X}, \mathcal{L}))\|] \leq \gamma\varepsilon + \epsilon\}. \tag{79}$$

Given the final iterates at $k = k^*$ and the initial values at $k = 0$, we can express the expected difference between the loss $\tilde{l}$ as the summation over the difference of iterates,

$$\begin{aligned} & \mathbb{E}[\tilde{l}(Y, \Phi_{\mathcal{W}_0}(\mathcal{X}, \mathcal{L})) - \tilde{l}(\mathcal{Y}, \Phi_{\mathcal{W}_{k^*}}(\mathcal{X}, \mathcal{L}))] \\ & = \mathbb{E}\left[\sum_{k=1}^{k^*} \tilde{l}(\mathcal{Y}, \Phi(\mathcal{X}; \mathcal{H}_{k-1}, \mathcal{L})) - \tilde{l}(\mathcal{Y}, \Phi(\mathcal{X}; \mathcal{H}_k, \mathcal{L})\right]. \end{aligned} \tag{80}$$

Taking the expected value with respect to the final iterate $k = k^*$, we get

$$\begin{aligned} & \mathbb{E}\Big[\tilde{l}(\mathcal{Y}, \Phi_{\mathcal{W}_0}(\mathcal{X}, \mathcal{L})) - \tilde{l}(\mathcal{Y}, \Phi_{\mathcal{W}_{k^*}}(\mathcal{X}, \mathcal{L}))\Big] \\ & = \mathop{\mathbb{E}}_{k^*}\left[\mathbb{E}\Big[\sum_{k=1}^{k^*} \tilde{l}(\mathcal{Y}, \Phi_{\mathcal{W}_{k-1}}(\mathcal{X}, \mathcal{L})) - \tilde{l}(\mathcal{Y}, \Phi_{\mathcal{W}_k}(\mathcal{X}, \mathcal{L})\Big]\right] \\ & = \sum_{t=0}^{\infty} \mathbb{E}\Big[\sum_{k=1}^{t} \tilde{l}(\mathcal{Y}, \Phi_{\mathcal{W}_{k-1}}(\mathcal{X}, \mathcal{L})) \\ & \qquad - \tilde{l}(\mathcal{Y}, \Phi_{\mathcal{W}_k}(\mathcal{X}, \mathcal{L})\Big] P(k^* = t). \end{aligned} \tag{81}$$

Lemma D.9 applied to any $k \leq k^*$ verifies

$$\mathbb{E}\left[\tilde{l}(\mathcal{Y}, \Phi_{\mathcal{W}_{k-1}}(\mathcal{X}, \mathcal{L})) - \tilde{l}(\mathcal{Y}, \Phi_{\mathcal{W}_k}(\mathcal{X}, \mathcal{L}))\right] \geq \eta\gamma\epsilon^2. \tag{82}$$

Coming back to (81), we get

$$\begin{aligned}&\mathbb{E}\left[\tilde{l}(\mathcal{Y}, \Phi(\mathcal{X}; \mathcal{H}_0, \mathcal{L})) - \tilde{l}(\mathcal{Y}, \Phi_{\mathcal{W}_{k^*}}(\mathcal{X}, \mathcal{L}))\right] \\ &\geq \eta\gamma\epsilon^2 \sum_{t=0}^{\infty} tP(k^* = t) = \eta\gamma\epsilon^2 \mathbb{E}[k^*].\end{aligned} \tag{83}$$

Since the loss function $\tilde{l}$ is non-negative,

$$\frac{\mathbb{E}\left[\tilde{l}(\mathcal{Y}, \Phi_{\mathcal{W}_0}(\mathcal{X}, \mathcal{L}))\right]}{\eta\gamma\epsilon^2} \geq \mathbb{E}[k^*], \tag{84}$$

from which we conclude that $k^* = \mathcal{O}(1/\epsilon^2)$. □

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Theoretical and empirical contributions stated in abstract are detailed in Section 1 (Introduction). Theoretical contributions are presented as stated in Section 4. Empirical contributions are presented as stated in Section 5.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: Limitations are discussed in the conclusions.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The most relevant assumptions are provided in the main body. Secondary assumptions can be found in the supplement, and their locations are clearly referenced in the main body.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All experiment details are provided in the supplement.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: While the code is not yet released (due to anonymity concerns), we hope to include it in the camera-ready.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All experiment details are provided in the supplement.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Experiments are held on standard train-test splits for all datasets, so we do not average over multiple runs. If the reviewers feel this is needed due to randomness in initialization, we will include error bars in the rebuttal.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All experiment details are provided in the supplement.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We have read and abide by the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: We do not anticipate any specific societal impacts beyond the general impacts of theoretical ML research.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We use standard benchmarks (MNIST, FMNIST, CIFAR10, FER2013) and appropriately cite the original papers.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: The paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.