# Improved Image Classification with Manifold Neural Networks

Caio F. Deberaldini Netto*, Zhiyang Wang[†] and Luana Ruiz*

*Abstract*—**Graph Neural Networks (GNNs) have gained popularity in various learning tasks, with successful applications in fields like molecular biology, transportation systems, and electrical grids. These fields naturally use graph data, benefiting from GNNs' message-passing framework. However, the potential of GNNs in more general data representations, especially in the image domain, remains underexplored. Leveraging the manifold hypothesis, which posits that high-dimensional data lies in a low-dimensional manifold, we explore GNNs' potential in this context. We construct an image manifold using variational autoencoders, then sample the manifold to generate graphs where each node is an image. This approach reduces data dimensionality while preserving geometric information. We then train a GNN to predict node labels corresponding to the image labels in the classification task, and leverage convergence of GNNs to manifold neural networks to analyze GNN generalization. Experiments on MNIST and CIFAR10 datasets demonstrate that GNNs generalize effectively to unseen graphs, achieving competitive accuracy in classification tasks.**

*Index Terms*—**graph neural networks, manifold neural networks, variational autoencoders, generalization**

## I. Introduction

The manifold hypothesis posits that high-dimensional data such as images lie on or near a low-dimensional manifold embedded within a high-dimensional ambient space. This assumption is widely used in machine learning to explain why certain algorithms can generalize well despite the high dimensionality of the input data [1]. In the machine learning community, for instance, dimensionality reduction and manifold learning are research fields where the manifold hypothesis is applied with great success to reconstruct the low-dimensional geometrical structure of (sub-)manifolds from data [2]–[4].

Despite the success of these approaches, and the rise of geometric deep learning techniques such as graph and group-invariant neural networks [5]–[10], the manifold structure underlying data without explicit geometry remains underexplored in deep learning. Inspired by recently introduced manifold neural networks (MNNs) [11], [12] and convergence results [13] demonstrating that GNNs on geometric graphs sampled from them converge to MNNs, we propose a novel framework for image classification using GNNs.

Our first contribution is a method to build the manifold from image data. We do so by leveraging variational autoencoders

(VAEs) [14] which, unlike deterministic autoencoders, can produce meaningful image representations along a smooth and structured embedding space. After learning VAE image embeddings in an unsupervised manner, the graph is constructed by computing Gaussian kernel distances between embeddings, which are used as edge weights.

Our second contribution is a machine learning pipeline wherein images are seen as discrete points from the image manifold connected through a geometric graph, and embeddings are seen as a signal on this graph. Given this signal, we then train a GNN to predict node labels corresponding to the image labels in the classification task.

We validate our framework theoretically by proving that, on geometric graphs sampled from a manifold, GNNs have bounded generalization gap and, further, that this bound decreases with the graph size. This result is also verified empirically via numerical experiments on the MNIST and CIFAR10 datasets. Our numerical results show that GNNs achieve better generalization than a multilayer perceptron (MLP) trained on individual VAE embeddings, and that our method outperforms another GNN-based method in which graphs are built by interpreting image pixels as graph nodes [15].

## II. Background

Before diving into our main contribution and method, we introduce preliminary definitions relating to graphs, graph neural networks, and manifold neural networks.

### A. Graph Signals, Graph Convolutions, Graph Neural Networks

A *graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ is defined as a triplet composed by a set of nodes $\mathcal{V}$, where $N = |\mathcal{V}|$ is the number of nodes, a set of edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, where $(i,j) \in \mathcal{E}$ if nodes $i$ and $j$ are connected, and a function $\mathcal{W} \colon \mathcal{E} \to \mathbb{R}$ that attributes weights to edges.

**Graph signals.** Here, graphs are endowed with signals. More precisely, graph signals are defined as vectors $\mathbf{x} \in \mathbb{R}^N$, where $x_i$ corresponds to the value of the signal at node $i$.

Given an undirected graph $\mathcal{G}$, the *graph shift operator* (GSO) $\mathbf{S} \in \mathbb{R}^{N \times N}$ is defined as a symmetric matrix s.t. $\mathbf{S}_{i,j} \neq 0$ for every $(i,j) \in \mathcal{E}$, and $\mathbf{S}_{i,j} = 0$ otherwise. The GSO operates on graph signals as $\mathbf{S}\mathbf{x}$ and, intuitively, it propagates/diffuses the graph signal through the nodes by aggregating the information of each node's neighborhood. Common examples of GSOs are the graph adjacency $\mathbf{A}$,

$$\begin{cases} \mathbf{A}_{i,j} = 1, \text{ if } (i,j) \in \mathcal{E} \\ \mathbf{A}_{i,j} = 0 \text{ otherwise,} \end{cases}$$

* Department of Applied Mathematics and Statistics, Mathematical Institute for Data Science (MINDS), Data Science and Artificial Intelligence Institute (DSAI), Johns Hopkins University, Baltimore, USA. E-mail: {cnetto1, lrubini1}@jh.edu

[†] Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, USA. E-mail: zhiyangw@seas.upenn.edu

the graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{A}$, with $\mathbf{D}_{i,j} = \mathbf{A}_{i,:}\mathbb{1}_N, i = j$, and $\mathbf{D}_{i,j} = 0$ otherwise, and their normalized versions [16]. In this work, the chosen graph shift operator is the graph Laplacian, $\mathbf{S} = \mathbf{L}$.

**Graph convolutional filters.** Given a graph signal $\mathbf{x}$ and a GSO $\mathbf{S}$, the graph convolutional filter $\mathbf{h} : \mathbb{R}^N \to \mathbb{R}^N$ is defined as

$$\mathbf{h}(\mathbf{S})\mathbf{x} = \sum_{k=0}^{K-1} h_k \mathbf{S}^k \mathbf{x}, \tag{1}$$

i.e., it is a polynomial function of the GSO parameterized by coefficients $\{h_k\}_{k=0}^{K-1}$ and operating on graph signals [17], [18].

**Graph neural networks.** One can define a *graph neural network* as a stack of layers, each consisting of graph convolutional filters followed by point-wise non-linear transformations $\sigma : \mathbb{R} \to \mathbb{R}$. Precisely, we define the $l$th layer of a GNN as

$$\mathbf{X}^l = \sigma \left( \sum_{k=0}^{K-1} \mathbf{S}^k \mathbf{X}^{l-1} \mathbf{H}_k^l \right), \tag{2}$$

where $\mathbf{X}^{l-1} \in \mathbb{R}^{N \times d_{l-1}}$ is the layer input and $\mathbf{X}^l \in \mathbb{R}^{N \times d_l}$ is the layer output with $d_{l-1}$ input features and $d_l$ output features respectively, and $\mathbf{H}_k^l \in \mathbb{R}^{d_{l-1} \times d_l}$ the filter coefficient matrix of the $l$th layer, which is learned.

For succinctness, throughout this paper the notation used for a GNN will be that of a function $\Phi(\mathbf{X}; \mathcal{H}, \mathbf{S})$, where $\mathcal{H} = \{\mathbf{H}_k^l\}_{l,k}$ is the set of graph filter coefficients at all layers.

### B. Manifold Signals, Manifold Convolutions, Manifold Neural Networks

Let $\mathcal{M}$ be an $m$-dimensional, compact, and smooth submanifold embedded in $\mathbb{R}^D$ with an induced uniform measure. More formally, $\mathcal{M}$ is an $m$-dimensional smooth submanifold of $\mathbb{R}^D$ if and only if every point $u \in \mathcal{M}$ has an open neighborhood $U \subset \mathbb{R}^D$ that can be mapped to some open subset $\Omega \subset \mathbb{R}^m$ via a smooth map [19]. This is sometimes called the intrinsic definition of the manifold $\mathcal{M}$.

Submanifolds of Euclidean space are locally Euclidean, in the sense that, in the vicinity of any point $u \in \mathcal{M}$, the manifold and associated signals admit an Euclidean approximation via the so-called tangent space. The tangent space of $\mathcal{M}$ at a point $u \in \mathcal{M}$ is the collection of tangent vectors at $u$. A vector $\mathbf{v} \in \mathbb{R}^D$ is a tangent vector of $\mathcal{M}$ at $u$ if there exists a smooth curve $\gamma$ such that $\gamma(0) = u$ and $\dot{\gamma}(0) = \mathbf{v}$. In other words, a tangent vector can be seen as the derivative of a curve $\gamma : \mathbb{R} \to \mathcal{M}$. The tangent space at point $u$, denoted $T_u\mathcal{M}$, is then [19]

$$T_u\mathcal{M} = \{\dot{\gamma}(0) \mid \text{smooth } \gamma : \mathbb{R} \to \mathcal{M} , \gamma(0) = u\}.$$

The collection of all tangent spaces at all points of the manifold $\mathcal{M}$ is denoted $T\mathcal{M}$ and called the tangent bundle.

**Manifold signals.** A manifold signal can be defined as a function over $\mathcal{M}$, i.e., $f : \mathcal{M} \to \mathbb{R}$. We restrict our attention to $L^2$ functions over the manifold, i.e., $f \in L^2(\mathcal{M})$.

Given smooth $f \in L^2(\mathcal{M})$, the gradient $\nabla f \in T\mathcal{M}$ is the vector field satisfying $\langle \nabla f(u), \mathbf{v} \rangle = \frac{\mathrm{d}}{\mathrm{d}t}|_{t=0}\gamma(t)$ for any tangent vector $\mathbf{v} \in T_u\mathcal{M}$ and any smooth curve $\gamma$ such that

$\gamma(0) = u$ and $\dot{\gamma}(0) = \mathbf{v}$ [20]. Conversely, given a smooth vector field $F \in T\mathcal{M}$ and an orthonormal basis $\mathbf{e}_1, \ldots, \mathbf{e}_D$ of $T_u\mathcal{M}$, we define the divergence $\Delta F \in \mathcal{C}^\infty(\mathcal{M})$ as $\Delta F = \sum_{i=1}^{D} \langle \partial_i F, \mathbf{e}_i \rangle$.

The composition of the gradient and divergence operators yields the Laplace-Beltrami (LB) operator $\mathcal{L} : L^2(\mathcal{M}) \to L^2(\mathcal{M})$, defined as [21]

$$\mathcal{L}f = -\Delta(\nabla f). \tag{3}$$

This operator appears in mathematical models of various physical phenomena, including wave propagation, heat diffusion, and the movement of quantum particles. Here, we are particularly interested on its role in the heat equation, which allows defining a *manifold shift operator* (MSO) $e^{-\mathcal{L}}f$ diffusing the signal information $f$ through the manifold $\mathcal{M}$ analogously to the GSO [12].

**Manifold convolutional filters.** Henceforth, we can define *manifold convolutional filters* as follows

$$g = \mathbf{h}(\mathcal{L})f = \sum_{k=0}^{K-1} h_k e^{-k\mathcal{L}}f, \tag{4}$$

which, similarly to graph filters, are polynomial functions of the MSO parameterized by coefficients $\{h_k\}_{k=0}^{K-1}$, and applied to the manifold signal $f$ [22].

**Manifold neural networks.** At last, a *manifold neural network* (MNN) can be defined as a stack of layers each consisting of manifold convolutional filters followed by point-wise non-linear transformations. Formally, the $l$th layer of an MNN is given by

$$f^l(x) = \sigma \left( \sum_{k=0}^{K-1} e^{-k\mathcal{L}} f^{l-1}(x) \mathbf{H}_k^l \right), \tag{5}$$

where $f^{l-1} : \mathcal{M} \to \mathbb{R}^{d_{l-1}}$ is the layer input and $f^l : \mathcal{M} \to \mathbb{R}^{d_l}$ the layer output with $d_{l-1}$ input features and $d_l$ output features respectively, and $\mathbf{H}_k^l \in \mathbb{R}^{d_{l-1} \times d_l}$ are the learnable coefficients. Similar to the notation used for GNNs, in the following we write the MNN as a function $\Phi(f; \mathcal{H}, \mathcal{L})$, where $\mathcal{H}$ groups the manifold filter coefficients at all layers.

### III. EXPLOITING IMAGE MANIFOLDS AND GENERALIZATION OF GNNS VIA MNNS

The manifold hypothesis posits that high-dimensional data lie on or near a low-dimensional manifold embedded within a high-dimensional ambient space ($m \ll D$). This assumption is widely used in machine learning to explain why certain algorithms can generalize well despite the high dimensionality of the input data. Specifically, we assume that that is the case for image data [23]–[27]. Therefore, we need to define how we build and access that manifold, and how the incorporation of the manifold in the deep learning model affects its performance and generalization.

## A. Image manifolds

Given a set of images $\{X_i\}_{i=1}^N$ sampled i.i.d. uniformly from an image space $\mathcal{X}$, a natural approach to approximate their underlying manifold is to embed these images onto a lower dimensional space using machine learning techniques. A well-established architecture for learning data embeddings is the autoencoder, a customizable model consisting of an encoder and a decoder block [28]. The encoder $f_{\text{enc}} : \mathcal{X} \to \mathbb{R}^m$ reduces the data to a latent embedding $\mathbf{z}_i = f_{\text{enc}}(X_i)$ of specified size $m$, and the decoder $f_{\text{dec}} : \mathbb{R}^m \to \mathcal{X}$ takes this embedding and maps it back to original, ambient space, as $\tilde{X}_i = f_{\text{dec}}(\mathbf{z}_i)$. The functions $f_{\text{enc}}, f_{\text{dec}}$ are learned by minimizing the distance between $\tilde{X}_i$ and $X_i$,

$$\min_{f_{\text{enc}}, f_{\text{dec}}} \sum_{i=1}^N \| f_{\text{dec}}(f_{\text{enc}}(X_i)) - X_i \|_2^2 \tag{6}$$

The encoder and decoder are typically deep networks tailored to the type of the data $X_i$ and the associated invariances—in the case of images, Convolutional Neural Networks (CNNs) for translation invariance or equivariance. However, the data might also have invariances that are not known beforehand and hence not accounted for by the model used to parametrize the encoder and decoder. In these cases, autoencoders will often fail to map approximate invariants to close locations in embedding space, leading to poor approximations of the underlying manifold.

Therefore, we propose to first learn that latent space by embedding the images from its original domain into the manifold using Variational Autoencoders (VAEs) [14]. VAEs differ from deterministic autoencoders in that instead of learning deterministic embeddings, they learn a Gaussian approximation $q(\mathbf{z}|X) = \mathcal{N}(\mathbf{z}|\mu_z(X), \mathbf{\Sigma}_z(X))$ of the distribution $p(\mathbf{z}|X)$[1]. Intuitively, this probabilistic framework contributes to a smoother embedding space, which is indeed observed empirically [30], [31]. Further, the assumption of a Gaussian prior adds structure to the embedding space, and in practice it can be seen that the embedding dimensions are correlated with invariants of the data—provided the encoder and decoder are parametrized to preserve them—and other relevant features [32], [33]. Inspired by [27], we train a CNNVAE, a VAE with a CNN as encoder/decoder, in an unsupervised way.

## B. GNNs for image classification

Given the learned embeddings $\mathbf{z}_i$, we can compute pairwise distances between them and construct a graph that can be processed by a GNN. Specifically, given a labeled dataset $\{\mathbf{z}_i, y_i\}_{i=1}^N$ of embedded images, where $y_i \in \{1, \ldots, C\}$ is the class label of image $i$, every sampled image is considered a node in the graph $\mathcal{G}$. Given two images $i$ and $j$, the edge weight $\mathcal{W}(i, j)$ is given by the Gaussian kernel distance

$$\mathcal{W}(i, j) = \exp\left( -\frac{\|\mathbf{z}_i - \mathbf{z}_j\|_2^2}{\sigma^2} \right), \tag{7}$$

---

[1]For brevity, we omit details about the encoder and decoder parametrization and the loss function for the VAE, but a comprehensive introduction can be found in [29].

where $\sigma$ is the kernel width, which controls the neighborhood considered when propagating the nodes' information.

We further define the $m$-dimensional graph signal $\mathbf{Z} \in \mathbb{R}^{N \times m}$ supported on $\mathcal{G}$, where the $i$th row of $\mathbf{Z}$ corresponds to the $i$th embedding $\mathbf{z}_i$, and the scalar graph signal $\mathbf{y} \in \mathbb{R}^N$, where the $i$th entry corresponds to the $i$th image label $y_i$. These signals are then used to learn a GNN $\Phi_{\mathcal{H}}$ which approximates $\mathbf{y}$ as $\tilde{\mathbf{y}} = \Phi(\mathbf{Z}; \mathcal{H}, \mathbf{S})$ [cf. (2)], where $\mathbf{S}$ is the Laplacian of graph (7).

## C. Generalization of GNNs

Let $\mathcal{G}$ be a graph with $N$ nodes sampled i.i.d. uniformly from the $m$-dimensional image manifold $\mathcal{M}$, such that each node represents an image and, therefore, is endowed with a graph signal $\mathbf{z}_i \in \mathbb{R}^m$, which is the image's embedding. In addition, consider $\mathbf{Z} \in \mathbb{R}^{N \times m}$ to be the node feature matrix, i.e., the graph signal matrix supported on $\mathcal{G}$, and $\mathbf{y} \in \mathbb{R}^N$ the column-vector of the classes that each node belongs to.

Therefore, if we suppose that the manifold hypothesis holds for that scenario, we can take advantage of the results that relate GNNs' output and MNNs' output trained on that manifold. Specifically, if we have a GNN $\Phi(\mathbf{Z}; \mathcal{H}, \mathbf{S})$ trained to predict each node/image class $\{y_i\}_{i=1}^N$, then Proposition 1 and Corollary 2 in [11] show that the GNN's output converges, respectively, in *probability* and *expectation* to the MNN's output $\Phi(f; \mathcal{H}, \mathcal{L})$, under mild assumptions.

More formally, given a positive and Lipschitz continuous loss function, $\ell(\Phi(\mathbf{Z}; \mathcal{H}, \mathbf{S}), \mathbf{y})$, the training of the GNN seeks to minimize the empirical risk defined as

$$P_E^* = \min_{\mathcal{H}} R_E(\mathcal{H}) = \ell(\Phi(\mathbf{Z}; \mathcal{H}, \mathbf{S}), \mathbf{y}). \tag{8}$$

However, the GNN goal is to minimize the statistical risk

$$P_S^* = \min_{\mathcal{H}} R_S(\mathcal{H}) = \mathbb{E}_{\mathbf{Z} \sim \mu^N}[\ell(\Phi(\mathbf{Z}; \mathcal{H}, \mathbf{S}), \mathbf{y})]. \tag{9}$$

Then, as proposed by [11], the generalization gap (GA), defined as $GA = P_S^* - P_E^*$, of the GNN can be bounded as follows

**Theorem 1:** [11, Theorem 1] Suppose we have a GNN trained on $(\mathbf{Z}, \mathcal{G})$ with $N$ nodes sampled i.i.d. uniformly over a $m$-dimensional $\mathcal{M}$, the generalization gap GA is bounded in probability at least $1 - \delta$ satisfying that,

$$GA = \mathcal{O}\left( \left( \frac{\log \frac{N}{\delta}}{N} \right)^{\frac{1}{m+4}} + \left( \frac{\log N}{N} \right)^{\frac{1}{m+4}} \right).$$

*Proof*: See appendix.

This generalization gap depends on the size of the sampled graph $N$, i.e. the number of labeled images, as well as the underlying manifold dimension $m$. We observe that a GNN trained on a set of sampled images from the underlying image manifold can generalize to unseen graphs derived from the same image manifold. With these unseen graphs constructed from previously unlabeled image embeddings, the generalization capability demonstrates the GNN's ability to generalize and make predictions on new images.

| Model | MNIST | | CIFAR10 | |
|---|---|---|---|---|
| | Test Acc. | Train Acc. | Test Acc. | Train Acc. |
| GCN [15] | $90.12 \pm 0.15$ | $96.46 \pm 1.02$ | $54.14 \pm 0.40$ | $70.16 \pm 3.43$ |
| GCN$_{[5]}$ (Ours) | $95.43 \pm 0.11$ | $95.93 \pm 0.07$ | $57.61 \pm 0.19$ | $86.26 \pm 0.22$ |

## IV. EXPERIMENTAL RESULTS

To assess the validity of our method, we show that the generalization bound presented in (III-C) holds for standard image classification benchmarks. Specifically, we use MNIST and CIFAR10 [34], [35]. The former is a dataset of images of handwritten digits in greyscale, comprising $60,000$ training samples and $10,000$ test samples. Our expectation with this dataset was that our models would have an almost perfect performance, and we could have a sanity check. On the other hand, the latter is a dataset of RGB images of 10 different objects, comprising $50,000$ training samples and $10,000$ test samples.

For both datasets, we first preprocess the data by training the CNNVAE. This first task is to encode the images into a latent space, and then reconstruct them. From experiments, the best latent space for MNIST has size $m = 128$, while for CIFAR10, $m = 4096$.

After that, with the embedded images, we use a GNN to process the sampled manifold data and predict the class of the graph nodes. Precisely, for each embedded image we uniformly sample $N - 1$ images in the image's set (train or test), forming a graph with $N$ nodes. As previously stated, the graph signal or node feature matrix corresponds to the images' embeddings.



Fig. 2. Accuracy difference between train and test set for an increasing number of sampled nodes for CIFAR10 dataset. The generalization gap (GA) decreases as the number of nodes increases.

Empirical results are shown in Table I for both datasets. We compare our model, a GCN [36], with the results obtained by the same GNN model but using the SLIC superpixel technique to build the graph structure from images presented in [15]. Since our focus so far is not to produce state-of-the-art results, we didn't fine-tune any hyperparameter for the models we implemented. On the contrary, we used the same principles used by the authors in [15] when defining our neural networks' hyperparameters. For instance, we used a small hidden size representation and a small number of layers, such that the model had between 100k-500k parameters.

Our method had better results than that proposed in [15], as seen in Table I, and our GNN was able to reduce the gap between seen and unseen data, as seen in Figures 1 and 2, a result we expected seeing empirically, given the theoretical result showed in Theorem 1.

Results were obtained after training 10 GCN models with 10 different seeds for 300 epochs. For the latter experiment we did the same for each number of sampled nodes in $\{5, 10, 20, 25, 50\}$, and we also showed the generalization gap for a vanilla MLP trained on the same set of embeddings.

## V. CONCLUSIONS

In this work, we introduced a novel framework for image classification that exploits the manifold hypothesis by creating a geometric graph from image data embedded using a VAE. By treating these embeddings as graph signals, we applied GNNs to the classification tasks, achieving better generalization. Theoretical analysis confirmed that GNNs trained on graphs sampled from a manifold have a bounded generalization gap, decreasing as graph size grows. Experiments on MNIST and CIFAR10 showed our method's superiority over MLPs and pixel-based GNNs, opening new possibilities for deep learning with manifold representations, particularly in scenarios where the underlying geometry of data is not explicitly known.
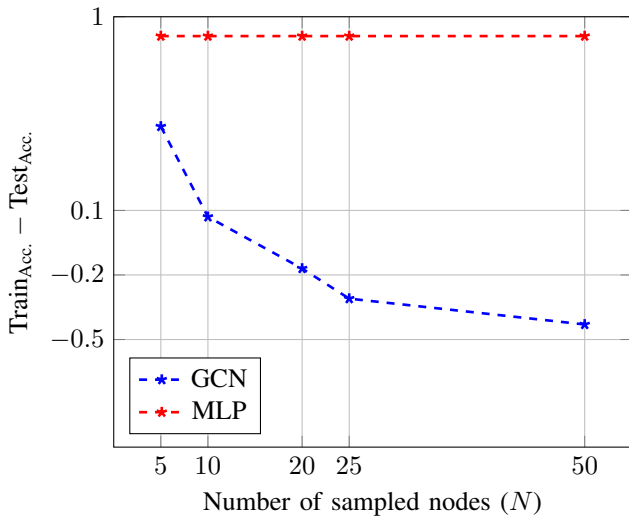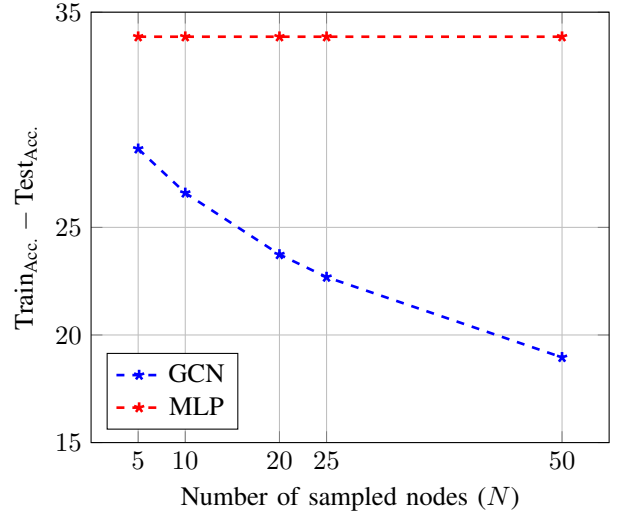


Fig. 1. Accuracy difference between train and test set for an increasing number of sampled nodes for MNIST dataset. The generalization gap (GA) decreases as the number of nodes increases.

# REFERENCES

[1] Yoshua Bengio, Aaron Courville, and Pascal Vincent, "Representation Learning: A Review and New Perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013. cit. on p. 1.

[2] Mikhail Belkin and Partha Niyogi, "Laplacian Eigenmaps for Dimensionality Reduction and Data Representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003. cit. on p. 1.

[3] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Grossberger, "UMAP: Uniform Manifold Approximation and Projection," *The Journal of Open Source Software*, vol. 3, no. 29, pp. 861, 2018. cit. on p. 1.

[4] Koshi Watanabe, Keisuke Maeda, Takahiro Ogawa, and Miki Haseyama, "SpectralMAP: Approximating Data Manifold With Spectral Decomposition," *IEEE Access*, vol. 11, pp. 31530–31540, 2023. cit. on p. 1.

[5] Taco Cohen and Max Welling, "Group Equivariant Convolutional Networks," in *International Conference on Machine Learning*. PMLR, 2016, pp. 2990–2999. cit. on p. 1.

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

[7] Brandon Anderson, Truong Son Hy, and Risi Kondor, "Cormorant: Covariant Molecular Neural Networks," *Advances in Neural Information Processing Systems*, vol. 32, 2019. cit. on p. 1.

[8] Soledad Villar, David W Hogg, Kate Storey-Fisher, Weichi Yao, and Ben Blum-Smith, "Scalars are universal: Equivariant machine learning, structured like classical physics," *Advances in Neural Information Processing Systems*, vol. 34, pp. 28848–28863, 2021. cit. on p. 1.

[9] Mario Geiger and Tess Smidt, "e3nn: Euclidean Neural Networks," *arXiv preprint arXiv:2207.09453*, 2022. cit. on p. 1.

[10] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst, "Geometric deep learning: going beyond euclidean data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017. cit. on p. 1.

[11] Zhiyang Wang, Juan Cervino, and Alejandro Ribeiro, "Generalization of Geometric Graph Neural Networks," *arXiv preprint arXiv:2409.05191*, 2024. cit. on pp. 1, 3, and 6.

[12] Zhiyang Wang, Luana Ruiz, and Alejandro Ribeiro, "Stability to deformations of manifold filters and manifold neural networks," *IEEE Transactions on Signal Processing*, 2024. cit. on pp. 1 and 2.

[13] Zhiyang Wang, Luana Ruiz, and Alejandro Ribeiro, "Geometric graph filters and neural networks: Limit properties and discriminability trade-offs," *IEEE Transactions on Signal Processing*, 2024. cit. on p. 1.

[14] Diederik P Kingma and Max Welling, "Auto-Encoding Variational Bayes," *Proceedings of the International Conference on Learning Representations*, 2014. cit. on pp. 1 and 3.

[15] 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*, vol. 24, no. 43, pp. 1–48, 2023. cit. on pp. 1 and 4.

[16] Antonio Ortega, Pascal Frossard, Jelena Kovačević, José M. F. Moura, and Pierre Vandergheynst, "Graph Signal Processing: Overview, Challenges, and Applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018. cit. on p. 2.

[17] S. Segarra, A. G. Marques, and A. Ribeiro, "Optimal graph-filter design and applications to distributed linear network operators," *IEEE Trans. Signal Process.*, vol. 65, pp. 4117–4131, Aug. 2017. cit. on p. 2.

[18] Luana Ruiz, Fernando Gama, and Alejandro Ribeiro, "Graph Neural Networks: Architectures, Stability, and Transferability," *Proceedings of the IEEE*, vol. 109, no. 5, pp. 660–682, 2021. cit. on p. 2.

[19] J. Robbin and D. Salamon, *Introduction to differential geometry*, Springer Nature, 2022. cit. on p. 2.

[20] P. Petersen, "Riemannian geometry," *Graduate Texts in Mathematics/Springer-Verlarg*, 2006. cit. on p. 2.

[21] P. Bérard, *Spectral geometry: direct and inverse problems*, vol. 1207, Springer, 2006. cit. on p. 2.

[22] Zhiyang Wang, Juan Cervino, and Alejandro Ribeiro, "A Manifold Perspective on the Statistical Generalization of Graph Neural Networks," *arXiv preprint arXiv:2406.05225*, 2024. cit. on p. 2.

[23] Gabriel Peyré, "Manifold models for signals and images," *Computer vision and image understanding*, vol. 113, no. 2, pp. 249–260, 2009. cit. on p. 2.

[24] Stanley Osher, Zuoqiang Shi, and Wei Zhu, "Low dimensional manifold model for image processing," *SIAM Journal on Imaging Sciences*, vol. 10, no. 4, pp. 1669–1690, 2017. cit. on p. 2.

[25] Johann Brehmer and Kyle Cranmer, "Flows for simultaneous manifold learning and density estimation," *Advances in neural information processing systems*, vol. 33, pp. 442–453, 2020. cit. on p. 2.

[26] Michael Arbel, Liang Zhou, and Arthur Gretton, "Generalized Energy Based Models," in *International Conference on Learning Representations*, 2021. cit. on p. 2.

[27] Kevin Miller, Jack Mauro, Jason Setiadi, Xoaquin Baca, Zhan Shi, Jeff Calder, and Andrea L Bertozzi, "Graph-based Active Learning for Semi-supervised Classification of SAR Data," in *Algorithms for Synthetic Aperture Radar Imagery XXIX*. SPIE, 2022, vol. 12095, pp. 126–139. cit. on pp. 2 and 3.

[28] Yasi Wang, Hongxun Yao, and Sicheng Zhao, "Auto-encoder based dimensionality reduction," *Neurocomputing*, vol. 184, pp. 232–242, 2016. cit. on p. 3.

[29] Carl Doersch, "Tutorial on variational autoencoders," *arXiv preprint arXiv:1606.05908*, 2016. cit. on p. 3.

[30] Matt J Kusner, Brooks Paige, and José Miguel Hernández-Lobato, "Grammar variational autoencoder," in *International conference on machine learning*. PMLR, 2017, pp. 1945–1954. cit. on p. 3.

[31] Nat Dilokthanakul, Pedro AM Mediano, Marta Garnelo, Matthew CH Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan, "Deep unsupervised clustering with gaussian mixture variational autoencoders," *arXiv preprint arXiv:1611.02648*, 2016. cit. on p. 3.

[32] Yunchen Pu, Zhe Gan, Ricardo Henao, Xin Yuan, Chunyuan Li, Andrew Stevens, and Lawrence Carin, "Variational autoencoder for deep learning of images, labels and captions," *Advances in neural information processing systems*, vol. 29, 2016. cit. on p. 3.

[33] Hanjun Dai, Yingtao Tian, Bo Dai, Steven Skiena, and Le Song, "Syntax-directed variational autoencoder for structured data," *arXiv preprint arXiv:1802.08786*, 2018. cit. on p. 3.

[34] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. cit. on p. 4.

[35] Alex Krizhevsky, Geoffrey Hinton, et al., "Learning multiple layers of features from tiny images," 2009. cit. on p. 4.

[36] Thomas N. Kipf and Max Welling, "Semi-Supervised Classification with Graph Convolutional Networks," in *International Conference on Learning Representations (ICLR)*, 2017. cit. on p. 4.

[37] Ulrike Von Luxburg, Mikhail Belkin, and Olivier Bousquet, "Consistency of spectral clustering," *The Annals of Statistics*, pp. 555–586, 2008. cit. on p. 6.

*A. Proof of Theorem 1*

To prove the generalization bound for the GNN, we first need to bring up two results that relate the output of a GNN $\Phi(\mathbf{Z}; \mathcal{H}, \mathbf{S})$ and an MNN $\Phi(f; \mathcal{H}, \mathcal{L})$.

**Definition 1:** A manifold signal $f$ is $\lambda_M$-bandlimited if for all eigenpairs $\{\lambda_i, \phi_i\}_{i=1}^{\infty}$ of the Laplace-Beltrami operator[2] $\mathcal{L}$ when $\lambda_i > \lambda_M$, we have $\langle f, \phi_i \rangle = 0$.

**Definition 2:** A filter is a low-pass filter if its frequency response satisfies

$$|\hat{h}(a)| = \mathcal{O}(a^{-d}).$$

**Definition 3:** A nonlinear activation function $\sigma(\cdot)$ is normalized Lipschitz-continuous if it satisfies

$$|\sigma(x) - \sigma(y)| \le |x - y|, \text{ with } \sigma(0) = 0.$$

**Proposition 1:** [11, Proposition 1] Let $\mathcal{M} \subset \mathbb{R}^D$ be an embedded manifold with Laplace-Beltrami operator $\mathcal{L}$ and a $\lambda_M$-bandlimited manifold signal $f$. Consider a pair of graph and graph signal $(\mathcal{G}, \mathbf{Z})$ with $N$ nodes sampled i.i.d. uniformly over $\mathcal{M}$. The graph Laplacian $\mathbf{L}$ is calculated with (7). Let $\Phi(\cdot; \mathcal{H}, \mathcal{L})$ be a single layer MNN on $\mathcal{M}$ (5) with single input and output features. Let $\Phi(\cdot; \mathcal{H}, \mathbf{L})$ be the GNN with the same architecture applied to the graph $\mathcal{G}$. Then, with the filters as low-pass and nonlinearities as normalized Lipschitz continuous, it holds in probability at least $1 - \delta$ that

$$\|\Phi(\mathbf{P}_N f; \mathcal{H}, \mathbf{L}) - \mathbf{P}_N \Phi(f; \mathcal{H}, \mathcal{L})\|_2 \le$$
$$C_1 \left( \frac{\log \frac{C_1 N}{\delta}}{N} \right)^{\frac{1}{m+4}} + C_2 \left( \frac{\log \frac{C_2 N}{\delta}}{N} \right)^{\frac{1}{m+4}} \theta_M^{-1}$$
$$+ C_3 \sqrt{\frac{\log(1/\delta)}{N}} + C_4 M^{-1},$$

where $\mathbf{P}_N \colon L^2(\mathcal{M}) \to L^2(\mathcal{V}(\mathcal{G}))$ is an uniform sampling operator, $C_1, C_2, C_3$ and $C_4$ are constants and $\theta_M = \min_{i=1,2,\ldots,M} |\lambda_i - \lambda_{i+1}|$.

**Corollary 1:** [11, Corrolary 2] The above difference bound between GNN and MNN also holds in expectation, since each node in $\mathcal{G}$ is sampled i.i.d. uniformly over $\mathcal{M}$

$$\mathbb{E}\big[\|\Phi(\mathbf{P}_N f; \mathcal{H}, \mathbf{L}) - \mathbf{P}_N \Phi(f; \mathcal{H}, \mathcal{L})\|_2\big] \le$$
$$C' N^{\frac{1}{m+4}} + C'' N^{-1/2} + C''' \left( \frac{\log N}{N} \right)^{\frac{1}{m+4}} + \bar{M} e^{-N/C} \sqrt{N},$$

where $C', C''$ and $C'''$ are constants, and $\bar{M} = 2\|\mathbf{P}_N f\|$.

Considering those results, suppose $\mathbf{H}_E \in \arg\min_{\mathcal{H}} R_E(\mathcal{H})$. Then, we have

[2]The LB operator $\mathcal{L}$ is self-adjoint and positive semidefinite (PSD). Hence, it admits a spectral decomposition.

$$GA \le R_S(\mathbf{H}_E) - R_E(\mathbf{H}_E)$$
$$= \mathbb{E}_{\mathbf{Z} \sim \mu^N}[\ell(\Phi(\mathbf{Z}; \mathbf{H}_E, \mathbf{L}), \mathbf{y})] - \ell(\Phi(\mathbf{Z}; \mathbf{H}_E, \mathbf{L}), \mathbf{y}). \quad (10)$$

Adding and subtracting the term $\ell(\Phi(f; \mathbf{H}_E, \mathcal{L}), g)$ we have the following

$$GA \le (\mathbb{E}_{\mathbf{Z} \sim \mu^N}[\ell(\Phi(\mathbf{Z}; \mathbf{H}_E, \mathbf{L}), \mathbf{y})] - \ell(\Phi(f; \mathbf{H}_E, \mathcal{L}), g))$$
$$+ (\ell(\Phi(f; \mathbf{H}_E, \mathcal{L}), g) - \ell(\Phi(\mathbf{Z}; \mathbf{H}_E, \mathbf{L}), \mathbf{y})). \quad (11)$$

Taking the absolute value of the above inequality and applying the triangle inequality, we have

$$GA \le \underbrace{\left| \mathbb{E}_{\mathbf{Z} \sim \mu^N}\big[\ell(\Phi(\mathbf{Z}; \mathbf{H}_E, \mathbf{L}), \mathbf{y})\big] - \ell(\Phi(f; \mathbf{H}_E, \mathcal{L}), g) \right|}_{①}$$
$$+ \underbrace{\left| \ell(\Phi(f; \mathbf{H}_E, \mathcal{L}), g) - \ell(\Phi(\mathbf{Z}; \mathbf{H}_E, \mathbf{L}), \mathbf{y}) \right|}_{②}. \quad (12)$$

Here, the loss function is assumed to be the $L_2$ loss. Therefore, the term ① in (12) can be written as

$$\left| \mathbb{E}_{\mathbf{Z}}\big[\ell(\Phi(\mathbf{Z}; \mathbf{H}_E, \mathbf{L}), \mathbf{y})\big] - \ell(\Phi(f; \mathbf{H}_E, \mathcal{L}), g) \right|$$
$$= \left| \mathbb{E}_{\mathbf{Z}}\big[\|\Phi(\mathbf{Z}; \mathbf{H}_E, \mathbf{L}) - \mathbf{y}\|\big] - \|\Phi(f; \mathbf{H}_E, \mathcal{L}) - g\|_{\mathcal{M}} \right| \quad (13)$$

Now, by subtracting and adding the term $\mathbb{E}_{\mathbf{Z}}[\mathbf{P}_N \Phi(f; \mathbf{H}_E, \mathcal{L})]$ inside the expectation above, and remembering the fact that $\mathbf{y} = \mathbf{P}_N g$, the expectation term from (13) becomes the following

$$\left| \mathbb{E}_{\mathbf{Z}}\big[\|\Phi(\mathbf{Z}; \mathbf{H}_E, \mathbf{L}) - \mathbf{y}\|\big] - \|\Phi(f; \mathbf{H}_E, \mathcal{L}) - g\|_{\mathcal{M}} \right|$$
$$\le \left| \mathbb{E}_{\mathbf{Z}}\big[\|\Phi(\mathbf{Z}; \mathbf{H}_E, \mathbf{L}) - \mathbf{P}_N \Phi(f; \mathbf{H}_E, \mathcal{L})\|\big] + \right.$$
$$\left. \mathbb{E}_{\mathbf{Z}}\big[\|\mathbf{P}_N \Phi(f; \mathbf{H}_E, \mathcal{L}) - \mathbf{P}_N g\|\big] - \|\Phi(f; \mathbf{H}_E, \mathcal{L}) - g\|_{\mathcal{M}} \right|$$

$$\le \left| \mathbb{E}_{\mathbf{Z}}\big[\|\Phi(\mathbf{Z}; \mathbf{H}_E, \mathbf{L}) - \mathbf{P}_N \Phi(f; \mathbf{H}_E, \mathcal{L})\|\big] \right| +$$
$$\left| \mathbb{E}_{\mathbf{Z}}\big[\|\mathbf{P}_N \Phi(f; \mathbf{H}_E, \mathcal{L}) - \mathbf{P}_N g\|\big] - \|\Phi(f; \mathbf{H}_E, \mathcal{L}) - g\|_{\mathcal{M}} \right|$$
$$(14)$$

The first term of equation (14) is bounded above using Corollary 1. For the second term, we need to use a derivation of Theorem 19 in [37]. More specifically, given that the nodes in $\mathcal{G}$ were sampled i.i.d. from $\mathcal{M}$, then

$$|\langle \mathbf{P}_N f, \phi_i \rangle - \langle f, \phi_i \rangle| = \mathcal{O}\left( \sqrt{\frac{\log 1/\delta}{N}} \right), \qquad (15)$$

for $\langle \cdot, \cdot \rangle$ being the inner product in $L^2$. This implies that $\left| \|\mathbf{P}_N f\|^2 - \|f\|_{\mathcal{M}}^2 \right| = \mathcal{O}\left( \sqrt{\frac{\log 1/\delta}{N}} \right)$, which indicates that $\|\mathbf{P}_N f\| = \|f\|_{\mathcal{M}} + \mathcal{O}\left( \frac{\log 1/\delta}{N} \right)^{\frac{1}{4}}$. Therefore, we have that

$$\mathbb{P}\Bigg( \Big| \|\mathbf{P}_N \Phi(f; \mathbf{H}_E, \mathcal{L}) - \mathbf{P}_N g\| - \|\Phi(f; \mathbf{H}_E, \mathcal{L}) - g\|_{\mathcal{M}} \Big|$$
$$\leq \mathcal{O}\left( \frac{\log 1/\delta}{N} \right)^{\frac{1}{4}} \Bigg) \geq 1 - \delta. \quad (16)$$

An expectation value can also be devised based on the probability bound, similarly to the result in Corollary 1, and then we can bound the second term of (14) as

$$\mathbb{E}\Big[ \|\mathbf{P}_N \Phi(f; \mathbf{H}_E, \mathcal{L}) - \mathbf{P}_N g\| - \|\Phi(f; \mathbf{H}_E, \mathcal{L}) - g\|_{\mathcal{M}} \Big]$$
$$\leq C N^{-\frac{1}{4}} + \mathcal{O}(e^{-N/C} \sqrt{N}). \quad (17)$$

Now, assuming, again, that the loss function is the $L_2$ loss, we can rewrite the term ② in (12) as

$$\left| \ell(\Phi(f; \mathbf{H}_E, \mathcal{L}), g) - \ell(\Phi(\mathbf{Z}; \mathbf{H}_E, \mathbf{L}), \mathbf{y}) \right|$$
$$= \Big| \|(\Phi(f; \mathbf{H}_E, \mathcal{L}) - g)\| - \|(\Phi(\mathbf{Z}; \mathbf{H}_E, \mathbf{L}) - \mathbf{y})\| \Big|. \quad (18)$$

Adding and subtracting an intermediate term $\mathbf{P}_N \Phi(f; \mathbf{H}_E, \mathcal{L})$, and applying the triangle inequality we have the following with probability at least $1 - \delta$

$$\Big| \|\Phi(\mathbf{Z}; \mathbf{H}_E, \mathbf{L}) - \mathbf{P}_N g\| - \|\Phi(f; \mathbf{H}_E, \mathcal{L}) - g\| \Big|$$
$$= \Big| \|\Phi(\mathbf{Z}; \mathbf{H}_E, \mathbf{L}) - \mathbf{P}_N \Phi(f; \mathbf{H}_E, \mathcal{L})\| +$$
$$\|\mathbf{P}_N \Phi(f; \mathbf{H}_E, \mathcal{L}) - \mathbf{P}_N g)\| \Big| - \|(\Phi(f; \mathbf{H}_E, \mathcal{L}) - g)\| \Big|$$

$$= \Big| \|\Phi(\mathbf{Z}; \mathbf{H}_E, \mathbf{L}) - \mathbf{P}_N \Phi(f; \mathbf{H}_E, \mathcal{L})\| \Big| +$$
$$\Big| \|\mathbf{P}_N \Phi(f; \mathbf{H}_E, \mathcal{L}) - \mathbf{P}_N g)\| \Big| - \|(\Phi(f; \mathbf{H}_E, \mathcal{L}) - g)\| \Big|. \quad (19)$$

The first term in (19) is bounded by Proposition 1, while the second term is bounded by (16). Taking the leading orders from those bounds, we conclude that

$$GA = \mathcal{O}\left( \left( \frac{\log \frac{N}{\delta}}{N} \right)^{\frac{1}{m+4}} + \left( \frac{\log N}{N} \right)^{\frac{1}{m+4}} \right). \quad (20)$$

$\square$