Gaussian Rank-Based Neighborhood Degree for Graph Neural Networks in Image Classification

Rafael Mendonça Duarte, Jean Roberto Ponciano, Lucas Pascotti Valem

[1]Institute of Mathematics and Computer Science (ICMC)
University of São Paulo (USP)
São Carlos – SP – Brazil

rmduarte@usp.br, jeanponciano@icmc.usp.br, lucas@icmc.usp.br

Abstract. *The exponential growth of data has intensified the gap between the availability of unlabeled data and the high cost of manual annotation. Graph Neural Networks (GNNs) have emerged as a promising solution, as they exploit relational structures and learn from both labeled and unlabeled data, performing semi-supervised learning. A crucial component of many of these models is degree-based normalization, which influences message propagation but typically assumes uniform importance among neighboring nodes. In image classification, graphs are usually constructed from feature similarity, where treating all neighbors equally may overlook important variations in relevance. Motivated by this gap, we propose GRaNDe (Gaussian Rank-based Neighborhood Degree). This novel degree measure integrates neighborhood ranking with Gaussian distance weighting to better capture node importance. Experiments on five public image classification datasets show accuracy improvements and competitive or favorable results compared to state-of-the-art methods. The code and supplementary materials are publicly available at* `grande.lucasvalem.com`.

1. Introduction

Advances in sensing, data acquisition, and storage technologies have dramatically expanded the volume of data available, and this growth continues to accelerate over time [Amini et al. 2025]. While the amount of available data has been increasing exponentially, the process of labeling this data has not followed the same trend. Data annotation is both time-consuming and costly, which limits the availability of labeled samples in many real-world applications [Amini et al. 2025]. As a result, semi-supervised learning strategies have gained increasing attention, as they aim to exploit labeled and unlabeled data during the learning process [Nass et al. 2021].

In this context, graph-based representations offer a natural way to encode sample relationships by representing samples as vertices and their relationships as edges, enabling the modeling of complex dependencies and structural patterns beyond pairwise feature comparisons. Information can then be iteratively aggregated from local neighborhoods, enabling the capture of both local and global patterns in the data. Such representations have been exploited in different scenarios. For example, preference graphs have been employed for diversity-aware ranking fusion in image retrieval [Figuerêdo et al. 2023], and heterogeneous graph embeddings combining local attributes and neighborhood information have been proposed to enhance node representations [Angonese and Galante 2026].

Among graph learning approaches, Graph Neural Networks (GNNs) have emerged as particularly promising, as they are able to effectively exploit the relational structure of data and propagate label information from a small set of labeled instances to a large number of unlabeled ones [Jiang and Bai 2024]. To fully exploit these properties, GNNs provide a principled framework that leverages graph-structured data by iteratively aggregating neighborhood information, enabling the model to capture both local and global structural patterns in a unified and scalable manner.

Originally, GNNs were developed for data that is naturally represented as graphs, such as citation networks, social networks, biological interaction graphs, and molecular structures, where entities and their relationships are explicitly defined [Jiang and Bai 2024]. In recent years, their applicability has increasingly expanded to multimedia data, particularly for visual data such as images, which can be transformed into graphs by modeling patches or feature descriptors as vertices and defining edges based on similarity, spatial proximity, or semantic relations. This representation allows GNNs to incorporate contextual structure beyond pairwise feature comparisons, benefiting tasks such as image classification, especially when labeled data is scarce.

Most GNNs, especially Graph Convolutional Networks (GCNs), are sensitive to the underlying graph topology [Khemani et al. 2024]. The way information is propagated across vertices plays a central role in the stability and effectiveness of these models [Liang et al. 2025]. In this context, degree-based normalization is a fundamental component of many GNN architectures, such as Approximate Personalized Propagation of Neural Predictions (APPNP) [Klicpera et al. 2019] and Simple Graph Convolution (SGC) [Wu et al. 2019], as it regulates how much influence each node and its neighbors exert during the message-passing process.

Given the importance of degree-based normalization in controlling message propagation, several studies have explored alternative approaches of node importance to replace or complement the standard degree measure, including eigenvector-based centrality, PageRank variants, betweenness, closeness, and diffusion-based scores [Chen et al. 2022]. However, most of these investigations focus on domains where the graph structure is readily available, such as social networks [Khoushehgir et al. 2024], citation graphs, biological interaction networks, and molecular graphs [Yu et al. 2025]. In contrast, their application to image datasets remains limited, as graphs derived from visual data are typically constructed from feature similarity, where the interpretation of node importance differs. As a result, despite their potential, alternative centrality-based normalization strategies remain relatively underexplored in GNNs for image classification.

Motivated by this gap, this paper proposes the **G**aussian **Ra**nk-based **N**eighborhood **De**gree (**GRaNDe**), which incorporates neighborhood ranking and Gaussian weighting over distances to define node importance beyond simple connectivity. Unlike the standard degree, by applying a Gaussian function to ranked distances, the proposed measure emphasizes closer and more relevant neighbors while smoothly attenuating the influence of more distant ones. Also, GRaNDe can be easily integrated into existing GNN architectures as a direct replacement for degree-based normalization, improving information propagation and, consequently, effectiveness in image classification. The **main contributions** are summarized as follows:

- We propose GRaNDe, a novel weighted degree function to replace the standard

degree centrality in GNNs. Unlike most existing approaches, which treat all neighbors of a node equally, GRaNDe incorporates distance information into the node degree through a Gaussian function and can be easily integrated into existing GNN models.
- Distances are updated with a Gaussian kernel during training, making GRaNDe adaptive by adjusting neighborhood importance as node representations evolve.
- Experiments on five public image classification datasets reveal that GRaNDe often improves classification accuracy across different settings, both on SGC and APPNP models.
- The proposed approach allows APPNP to sustain accuracy improvements as the number of neurons increases, effectively mitigating the performance saturation commonly observed in baseline approaches.
- A comparison with recent baselines shows that GRaNDe often achieves superior or competitive results.

The remainder of this paper is organized as follows. Section 2 details the main steps of our proposed approach, introduces GRaNDe, and explains how it is integrated into GNN models. Section 3 presents the experimental evaluation. Finally, Section 4 concludes the paper.

2. Proposed Approach

Our proposed approach is composed of three main steps, illustrated in Figure 1, with GNN training incorporating GRaNDe as the central contribution. The following subsections detail each step depicted in the figure (A, B, and C).

2.1. Feature Extraction

Since images are inherently high-dimensional, feature extraction is employed to obtain compact and discriminative representations that facilitate the separation of samples in the feature space. Traditionally, features were handcrafted by modeling visual characteristics such as color, shape, and texture. However, these descriptors are often limited in their ability to capture complex and high-level semantic patterns.

Current approaches rely on deep learning feature extractors, which generate more discriminative representations through transfer learning from models pretrained on large-scale datasets. In this work, we consider both Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs), extracting image embeddings from the final fully connected layer in CNNs and from the CLS token representation in ViTs.

An image dataset can be defined as $\mathcal{C} = \{x_1, x_2, \ldots, x_n\}$, where each image x_i is represented by a feature vector $\mathbf{x}_i \in \mathbb{R}^d$. The collection of all feature vectors is denoted by $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \subset \mathbb{R}^d$. These feature vectors can be arranged into a matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$, where each row corresponds to a feature vector $\mathbf{x}_i$ associated with an image x_i.

2.2. Graph Construction

Beyond feature representations, the input graph is also a key component of the GNN learning process. Firstly, we employ a BallTree algorithm to compute the Euclidean distances between feature vectors. By ordering these distances in ascending order, we obtain a ranked list of neighbors for each image. The ranked lists reflect the proximity between samples in the feature space, where closer neighbors are more similar and therefore more reliable for graph construction.

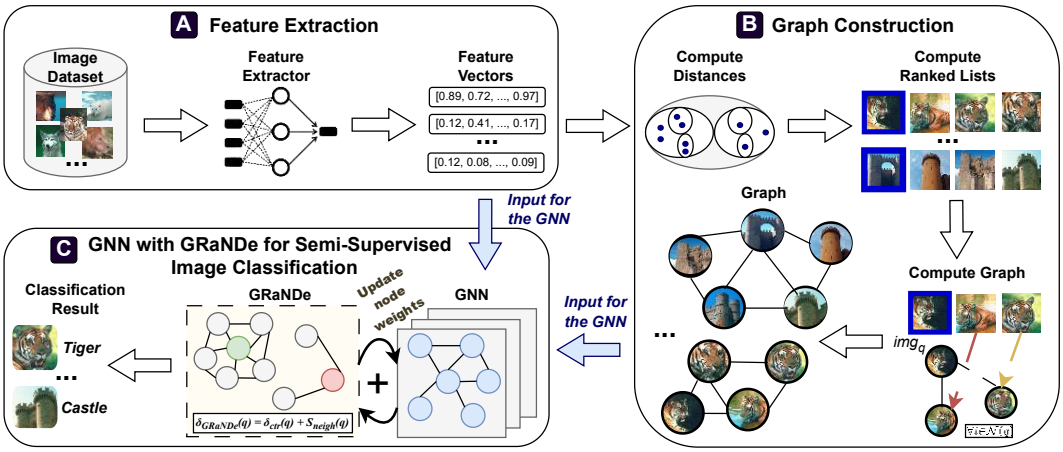

Figure 1. Main steps of our GNN workflow with GRaNDe for semi-supervised image classification.

In this study, we consider undirected graphs $G = (V, E)$ built from a dataset of n images, where each node represents a single image and the node set is defined as $V = \mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$. In particular, we adopt the Reciprocal k-Nearest Neighbor Graph (Rec. kNN) commonly used in the literature [Yang et al. 2025, Valem et al. 2023] that relies on neighborhood information derived from ranked lists.

An edge between two images is created only if they appear in each other's k nearest neighbor sets, a constraint that naturally provides a sparse graph, keeping only mutual edges:

$$E = \{(i, j) \mid j \in \mathcal{N}(\mathbf{x}_i, k) \ \wedge \ i \in \mathcal{N}(\mathbf{x}_j, k)\}. \tag{1}$$

2.3. GNN with GRaNDe for Image Classification

Once the data are organized as a graph, learning can be formulated as the problem of propagating and refining information across related samples. GNNs are designed to learn new node representations by jointly modeling node features and the underlying graph structure through neighborhood aggregation. These representations can be directly employed for classification at the node, edge, or graph level. In this work, we focus on node classification, where each node corresponds to an image.

Given the feature matrix $\mathbf{X}$ obtained in Step A and the graph G constructed in Step B, the learning stage consists of training a GNN to propagate information over the graph and produce predictions. This work considers diffusion-based GNNs, since they offer a simple and efficient formulation for feature propagation while preserving the structural information encoded in the graph. In these models, propagation is typically implemented through iterative multiplication by a normalized adjacency matrix $\hat{\mathbf{A}}$, where normalization is computed from the node degree. In this step, we employ the proposed GRaNDe to guide the training process and perform node classification.

Subsection (1) describes the computation of the normalized adjacency matrix $\hat{\mathbf{A}}$. Subsections (2) and (3) present the two diffusion-based GNN models considered in this work and detail how $\hat{\mathbf{A}}$ is incorporated into their formulations. Finally, Subsection (4) introduces the proposed GRaNDe.

2.3.1. Computing the Normalized Adjacency Matrix $\hat{\mathbf{A}}$

Given an undirected graph $G = (V, E)$ with $n = |V|$ nodes, its structure can be represented by an adjacency matrix $\mathbf{A} \in \{0, 1\}^{n \times n}$, where $\mathbf{A}_{ij} = 1$ if there exists an edge between nodes v_i and v_j, and $\mathbf{A}_{ij} = 0$ otherwise. To allow each node to retain its own information during the propagation process, self-loops are added to the graph, producing the augmented adjacency matrix $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$, where $\mathbf{I} \in \mathbb{R}^{n \times n}$ denotes the identity matrix.

The corresponding degree matrix $\tilde{\mathbf{D}} \in \mathbb{R}^{n \times n}$ is defined as a diagonal matrix whose entries are given by $\tilde{\mathbf{D}}_{ii} = \delta(i)$, where $\delta(i)$ denotes the node degree function, originally defined as the degree centrality δ_{ctr}. This degree is computed as:

$$\delta_{\text{ctr}}(i) = |\mathcal{N}(i)|, \tag{2}$$

where $\mathcal{N}(i)$ denotes the neighborhood of node v_i, including self-loops. Therefore, $|\mathcal{N}(i)|$ is the number of neighbors.

The normalized adjacency matrix used for diffusion is then computed as $\hat{\mathbf{A}} = \tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}}$. This normalization rescales message propagation according to node connectivity, attenuating the influence of highly connected nodes while amplifying that of sparsely connected ones. As a result, $\hat{\mathbf{A}}$ defines a diffusion operator that enables stable and efficient information propagation across the graph, while preserving its structural properties. Repeated multiplication by $\hat{\mathbf{A}}$ corresponds to performing multi-step diffusion over the graph, which constitutes the core mechanism of diffusion-based GNN models adopted in this work.

2.3.2. Simple Graph Convolution (SGC)

SGC [Wu et al. 2019] is a widely adopted GCN variant known for its lightweight architecture and computational efficiency, while keeping effective results in many cases.

Given the input feature matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$, SGC first applies a linear transformation to the node features:

$$\mathbf{H}^{(0)} = \mathbf{X}\mathbf{W}, \tag{3}$$

where $\mathbf{W} \in \mathbb{R}^{d \times c}$ is a learnable weight matrix and c denotes the number of classes.

The transformed features are then propagated through the graph by performing K successive diffusion steps using the normalized adjacency matrix $\hat{\mathbf{A}}$:

$$\mathbf{Z} = \hat{\mathbf{A}}^K \mathbf{H}^{(0)}. \tag{4}$$

The resulting representations $\mathbf{Z}$ are used for classification.

2.3.3. Approximate Personalized Propagation of Neural Predictions (APPNP)

APPNP [Klicpera et al. 2019] is a diffusion-based GNN model inspired by the Personalized PageRank formulation. Unlike SGC, APPNP decouples feature transformation from propagation and employs a teleport mechanism to preserve the original node representations during diffusion, alleviating the over-smoothing effect commonly observed in deep propagation.

Given the input feature matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$, APPNP first applies a linear transformation: $\mathbf{H}^{(0)} = \mathbf{X}\mathbf{W}$, where $\mathbf{W} \in \mathbb{R}^{d \times c}$ is a learnable weight matrix and c denotes the number of classes. The transformed features are then propagated through the graph using an iterative diffusion process defined as:

$$\mathbf{H}^{(k)} = (1-\alpha)\hat{\mathbf{A}}\mathbf{H}^{(k-1)} + \alpha\mathbf{H}^{(0)}, \quad k = 1, \ldots, K, \tag{5}$$

where $\alpha \in (0, 1)$ is the teleport probability that controls the trade-off between propagating information from neighboring nodes and retaining the original node representations.

After K propagation steps, the final node representations are given by $\mathbf{Z} = \mathbf{H}^{(K)}$. By repeatedly diffusing predictions while injecting the initial representations at each step, APPNP implements an approximation of Personalized PageRank over the graph. This mechanism enables effective long-range information propagation while maintaining stability and preserving discriminative node features.

2.3.4. Gaussian Rank-Based Neighborhood Degree (GRaNDe)

The proposed degree function is used as a direct replacement for the standard degree centrality in both SGC and APPNP models, so that the node degree $\delta(i)$ is no longer defined as δ_{ctr}, but as the proposed δ_{GRaNDe}. GRaNDe assigns a score to each node q, which is used to modulate message amplification during GNN propagation, with lower values leading to a stronger influence. GRaNDe is defined as:

$$\delta_{\text{GRaNDe}}(q) = \delta_{\text{ctr}}(q) + s_{\text{neigh}}(q), \tag{6}$$

where $\delta_{\text{ctr}}(q)$ denotes the degree centrality of node q, and $s_{\text{neigh}}(q)$ is a neighborhood penalty term.

By using $\delta_{\text{ctr}}(q)$ as part of $\delta_{\text{GRaNDe}}(q)$, we preserve the original intuition behind degree normalization, that nodes with many connections should have their messages attenuated to avoid dominating the propagation process and causing excessive smoothing of representations. In contrast, nodes with few connections are assigned smaller degree values, which amplifies their messages and allows them to exert a stronger influence during information diffusion.

On the other hand, $s_{\text{neigh}}(q)$ acts as a penalty term that measures the average dissimilarity between an image q and its neighbors. Specifically, it computes the mean of the inverse Gaussian similarities derived from the distances between q and the samples in its neighborhood, where the averaging factor corresponds to the neighborhood size (i.e., $\delta_{\text{ctr}}(q) = |\mathcal{N}(q)|$). Consequently, larger values of $s_{\text{neigh}}(q)$ indicate neighborhoods whose samples are farther from q on average, while smaller values correspond to closer neighborhoods. Therefore, we define the neighborhood penalty score as:

$$s_{\text{neigh}}(q) = \frac{1}{\delta_{\text{ctr}}(q)} \sum_{i \in \mathcal{N}(q)} \left[f_g(q, i, \sigma) \right]^{-1}, \tag{7}$$

where the function $f_g(q, i, \sigma)$ corresponds to a Gaussian radial basis function (RBF) applied to the Euclidean distance between nodes q and i.

The distances used by f_g are computed in the feature space that is progressively updated during training. As a consequence, the neighborhood penalty term $s_{\text{neigh}}(q)$ is recomputed at each training epoch using the updated feature representations $\mathbf{H}^{(0)}$. This allows GRaNDe to operate as a dynamic degree function, adapting to the evolving geometry of the feature space learned. Let $\mathbf{h}_q$ and $\mathbf{h}_i$ denote the feature representations of nodes q and i, respectively. The pairwise distance between nodes is computed using the Euclidean (ℓ_2) norm, $\rho(q,i) = \|\mathbf{h}_q - \mathbf{h}_i\|_2$. All distances are min-max normalized, since the Gaussian kernel is sensitive to the input range, and the resulting value is denoted by $\rho'(q,i) \in [0,1]$. The Gaussian kernel function is defined as:

$$f_g(q,i,\sigma) = \exp\left(-\frac{\rho'(q,i)^2}{\sigma}\right), \tag{8}$$

where the parameter $\sigma > 0$ is a positive scaling parameter that controls the rate at which the influence of a node i decays as its distance to q increases. The more similar the nodes, the higher the value. For notational simplicity, we also write the kernel as $f_g(\rho')$.

Please note that the inverse term in Equation 7 is intentionally used so that nodes with distant neighbors receive higher penalties, and vice versa. Moreover, division by zero is not an issue, as the Gaussian RBF is strictly positive for any finite distance, and the degree δ_{ctr} is always greater than or equal to one for any connected node, due to the presence of a self-loop at each node.

Figure 2 illustrates examples of how GRaNDe is computed. In example 1, the central node (in red) is connected to farther neighbors, leading to smaller Gaussian similarity values, which results in a higher neighborhood penalty and a larger GRaNDe value. In example 2, the node is surrounded by closer neighbors, providing higher similarity scores, reducing the penalty term, and producing a smaller GRaNDe value.

3. Experimental Evaluation

3.1. Experimental Setup

The evaluation was conducted on five publicly available image datasets, as follows:

- **Flowers17** [Nilsback and Zisserman 2006]: contains 1,360 images evenly distributed across 17 flower categories. Each class includes 80 images captured under diverse backgrounds and viewpoints, resulting in moderate intra-class variability.
- **Corel5k** [Liu and Yang 2013]: consists of 5,000 natural images spanning 50 semantic categories, including animals, landscapes, and indoor and outdoor scenes.
- **Pets** [Parkhi et al. 2012]: comprises 7,390 images from 37 pet classes, with significant variability in pose, illumination, scale, and background.
- **CUB200** [Wah et al. 2011]: includes 11,788 images of 200 bird species, providing detailed annotations commonly used for fine-grained recognition and localization tasks.
- **Dogs** [Khosla et al. 2011]: contains 20,580 images of 120 dog breeds, presenting substantial intra-class variation and inter-class similarity, which makes it a challenging benchmark for fine-grained image classification.

For every dataset, experiments were conducted using four deep learning models pretrained on the ImageNet dataset:

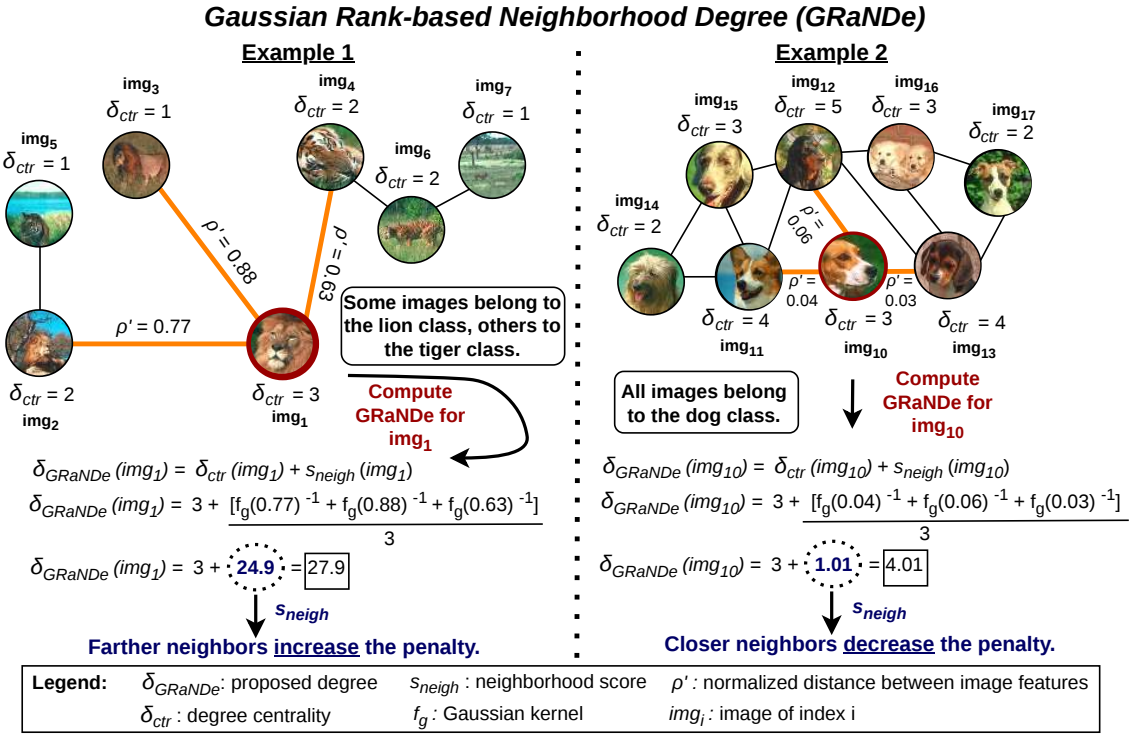

Figure 2. Illustration that exemplifies how GRaNDe is computed in two scenarios: *(i)* a node with farther, sparsely connected neighbors and *(ii)* a node with closer, densely connected neighbors. Images are from the Corel5k dataset. Self-loops are omitted for visualization purposes.

- **ResNet152** [He et al. 2016]: a deep residual CNN. We used the 2048-dimensional output from the final convolutional stage.
- **SENet154** [Hu et al. 2018]: a ResNet variant with channel-wise attention. Features were extracted from the final SE block (2048 dimensions).
- **DPNet92** [Chen et al. 2017]: a dual-path CNN that combines residual feature reuse with dense feature exploration. We used the global pooled output from the final stage (2688 dimensions).
- **ViT-B16** [Dosovitskiy et al. 2021]: a transformer model operating on patch sequences. We used the 768-dimensional class token from the last encoder layer.

Regarding the setup, we considered both SGC and APPNP models. In all experiments, we considered a reciprocal kNN graph with $k = 40$. We employed a learning rate of 10^{-3} for all datasets, except for CUB200, where a learning rate of 10^{-2} was used. For the GNN hyperparameters, we followed the default settings provided by the PyTorch Geometric repository[1], using 256 neurons as the hidden dimension for APPNP. All GNNs were trained for 200 epochs in each execution using the Adam optimizer. All results are reported using a 10-fold protocol, in which each fold is used once as the training set while the remaining folds are used for testing, following a setup with 10% of the data for training and 90% for testing with the objective of evaluating the proposed method in a semi-supervised setting with limited labeled data. Each reported result corresponds to the mean accuracy over 5 executions of this 10-fold procedure, and the standard devia-

[1] www.github.com/pyg-team/pytorch_geometric

tion is reported to show accuracy variability. These settings were chosen to ensure a fair comparison, consistent with the settings used by other methods.

3.2. Analysis of the Sigma (σ) Parameter

The parameter σ plays a crucial role in the Gaussian function f_g, as it controls the sensitivity of the kernel to distance variations, determining how quickly similarity decays as samples move farther apart in the feature space. Figure 3 illustrates the effect of the σ parameter on classification accuracy for different feature extractors using both SGC and APPNP models on the Flowers dataset. The shaded area indicates the standard deviation over multiple runs. Overall, $\sigma = 0.2$ achieved the highest accuracy in most configurations. Therefore, we set $\sigma = 0.2$ as the default parameter.

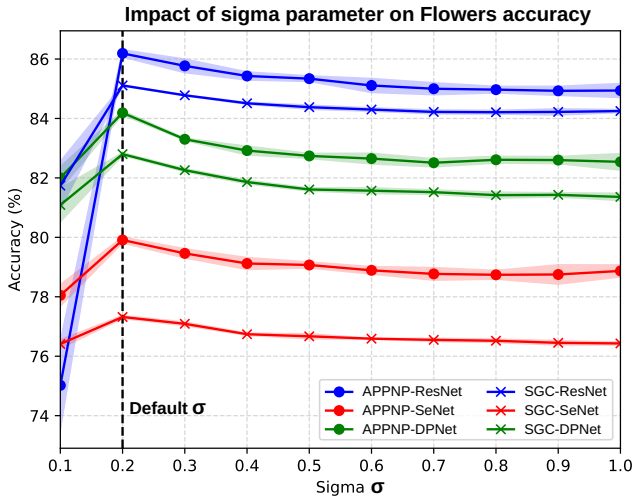

Figure 3. Impact of the σ parameter on classification accuracy for Flowers.

3.3. Semi-Supervised Image Classification

We conducted experiments on five datasets using four different feature extractors, replacing the standard degree centrality δ_{ctr} with the proposed GRaNDe (δ_{GRaNDe}), and compared the accuracy. Table 1 reports the results obtained with standard degree centrality and with GRaNDe. For our method, we report results using the default value $\sigma = 0.2$, as well as the σ that achieved the highest accuracy for each setting. The parameter σ was evaluated over the range $[0.1, 1.0]$ with a step size of 0.1. Relative gains are computed by comparing the results with and without GRaNDe.

Overall, GRaNDe often improves accuracy in both GNNs across all evaluated datasets and features. Notice that the highest gains are observed for APPNP, particularly on the CUB200 dataset, with DPNet (+15.98%) and SENet (+14.97%). This is probably because APPNP performs multiple propagation steps, which progressively reinforce the influence of GRaNDe throughout the diffusion process. In contrast, SGC relies on a single linear diffusion step, limiting the accumulation of such effects.

3.4. Comparison with Other Approaches

Since APPNP achieved higher relative gains in most evaluated configurations, we further extended its evaluation by comparing it against recent baselines using different numbers of neurons. We considered both Manifold-GCN (MGCN) [Valem et al. 2023] and

Table 1. Accuracy (%) gains for APPNP and SGC models when GRaNDe is employed. Results are reported for the default and best sigma. For each row, the best result is highlighted in bold. The best result for each dataset and GNN is highlighted in blue. Gains are calculated as the relative improvement of GRaNDe's best accuracy over the Degree Centrality baseline.

GNN	Dataset	Features	Accuracy with Degree Centrality	Accuracy with GRaNDe (ours)			Relative Gain
				Default $\sigma = 0.2$	Best σ	σ	
APPNP	Flowers17	ResNet	84.43 ± 0.088	**86.19 ± 0.121**	86.19 ± 0.121	0.2	+2.09%
		SENet	77.88 ± 0.190	**79.91 ± 0.097**	79.91 ± 0.097	0.2	+2.61%
		DPNet	82.00 ± 0.153	**84.19 ± 0.070**	84.19 ± 0.070	0.2	+2.67%
		ViT-B16	97.14 ± 0.019	**98.18 ± 0.048**	98.18 ± 0.048	0.2	+1.08%
	Corel5k	ResNet	92.65 ± 0.073	**93.25 ± 0.040**	93.25 ± 0.040	0.2	+0.65%
		SENet	92.63 ± 0.066	**93.42 ± 0.050**	93.42 ± 0.050	0.2	+0.85%
		DPNet	90.35 ± 0.079	**91.75 ± 0.040**	91.75 ± 0.040	0.2	+1.56%
		ViT-B16	94.24 ± 0.171	89.01 ± 0.304	**94.64 ± 0.096**	0.3	+0.42%
	CUB200	ResNet	49.49 ± 0.960	**54.58 ± 0.238**	54.58 ± 0.238	0.2	+10.27%
		SENet	38.08 ± 0.065	**43.78 ± 0.178**	43.78 ± 0.178	0.2	+14.97%
		DPNet	39.94 ± 0.507	41.69 ± 0.690	**46.32 ± 0.257**	0.3	+15.98%
		ViT-B16	71.56 ± 0.230	65.23 ± 0.607	**74.41 ± 0.222**	0.4	+3.99%
	Pets	ResNet	90.06 ± 0.051	**91.20 ± 0.042**	91.20 ± 0.042	0.2	+1.26%
		SENet	89.69 ± 0.051	**90.56 ± 0.037**	90.56 ± 0.037	0.2	+0.97%
		DPNet	91.13 ± 0.068	**91.78 ± 0.041**	91.78 ± 0.041	0.2	+0.72%
		ViT-B16	87.83 ± 0.036	86.37 ± 0.168	**88.50 ± 0.039**	0.3	+0.76%
	Dogs	ResNet	88.46 ± 0.039	89.12 ± 0.023	**89.25 ± 0.023**	0.3	+0.90%
		SENet	94.13 ± 0.014	**94.31 ± 0.013**	94.31 ± 0.013	0.2	+0.19%
		DPNet	92.65 ± 0.050	91.68 ± 0.180	**93.05 ± 0.019**	0.1	+0.43%
		ViT-B16	91.00 ± 0.012	76.83 ± 0.136	**91.34 ± 0.011**	0.4	+0.38%
SGC	Flowers17	ResNet	83.94 ± 0.041	**85.11 ± 0.021**	85.11 ± 0.021	0.2	+1.39%
		SENet	76.17 ± 0.051	**77.32 ± 0.048**	77.32 ± 0.048	0.2	+1.51%
		DPNet	81.30 ± 0.095	**82.80 ± 0.046**	82.80 ± 0.046	0.2	+1.84%
		ViT-B16	96.92 ± 0.034	97.38 ± 0.024	**97.76 ± 0.014**	0.1	+0.86%
	Corel5k	ResNet	91.98 ± 0.006	**92.55 ± 0.008**	92.55 ± 0.008	0.2	+0.62%
		SENet	92.15 ± 0.014	92.18 ± 0.035	**92.76 ± 0.012**	0.1	+0.66%
		DPNet	89.78 ± 0.042	**91.83 ± 0.043**	91.83 ± 0.043	0.2	+2.28%
		ViT-B16	95.47 ± 0.057	**95.77 ± 0.015**	95.77 ± 0.015	0.2	+0.31%
	CUB200	ResNet	53.79 ± 0.019	43.58 ± 0.614	**54.45 ± 0.027**	0.4	+1.23%
		SENet	40.32 ± 0.011	41.29 ± 0.007	**41.79 ± 0.013**	0.3	+3.64%
		DPNet	52.48 ± 0.578	52.46 ± 0.042	**52.69 ± 0.052**	0.7	+0.40%
		ViT-B16	78.35 ± 0.030	77.92 ± 0.020	**78.58 ± 0.013**	0.4	+0.30%
	Pets	ResNet	89.85 ± 0.016	90.45 ± 0.019	**90.79 ± 0.004**	0.1	+1.04%
		SENet	89.65 ± 0.017	89.87 ± 0.014	**89.98 ± 0.017**	0.1	+0.37%
		DPNet	91.08 ± 0.034	**91.29 ± 0.024**	91.29 ± 0.024	0.2	+0.23%
		ViT-B16	88.22 ± 0.032	**88.34 ± 0.023**	88.34 ± 0.023	0.2	+0.14%
	Dogs	ResNet	88.55 ± 0.014	**89.03 ± 0.010**	89.03 ± 0.010	0.2	+0.54%
		SENet	94.32 ± 0.008	94.25 ± 0.007	**94.35 ± 0.007**	0.1	+0.04%
		DPNet	93.10 ± 0.009	93.00 ± 0.010	**93.13 ± 0.009**	0.3	+0.03%
		ViT-B16	91.76 ± 0.010	**91.91 ± 0.009**	91.91 ± 0.009	0.2	+0.16%

Density-Guided Correlation Graph (DGCG) [Brito and Valem 2025] under the same experimental settings and hyperparameters by using the publicly available implementations.

Both baselines use GNNs for semi-supervised image classification. Manifold-GCN leverages manifold learning to build graphs that capture the intrinsic geometric structure of the feature space for semi-supervised classification, while DGCG builds graphs using rank correlation measures and an adaptive thresholding strategy guided by graph density.

Table 2 reports a comparison of our approach using APPNP against competing methods under different hidden layer sizes (64, 128, and 256 neurons). For a fixed dataset and feature extractor, GRaNDe achieves the best accuracy across most configurations. Moreover, while the baseline methods do not consistently benefit from larger hidden layers in all cases, GRaNDe shows accuracy improvements in most cases when the number

of neurons is increased.

Table 2. Impact of the number of neurons (64, 128, 256) on APPNP accuracy (%) for GRaNDe and baselines. For each dataset and feature extractor, the best result is highlighted in bold.

Dataset	Feature	Num. of Neurons	MGCN	DGCG	GRaNDe (ours)	σ
Flowers	ResNet	64	85.41	85.08	84.11	0.2
		128	85.38	85.06	85.81	
		256	85.35	85.14	**86.19**	
	SENet	64	78.82	77.40	79.90	0.2
		128	78.72	77.03	**80.10**	
		256	78.59	76.79	79.91	
	ViT-B16	64	97.43	97.68	96.62	0.2
		128	97.34	97.79	97.32	
		256	97.31	97.86	**98.18**	
Corel5k	ResNet	64	92.82	90.68	83.33	0.2
		128	92.72	91.10	91.74	
		256	92.54	91.27	**93.25**	
	SENet	64	91.92	90.46	80.80	0.2
		128	91.79	90.75	90.61	
		256	91.73	90.86	**93.42**	
	ViT-B16	64	95.13	90.23	92.79	0.3
		128	95.21	92.28	93.93	
		256	**95.22**	92.84	94.64	
CUB200	ResNet	64	48.44	51.54	51.77	0.2
		128	51.29	53.66	54.53	
		256	51.92	53.99	**54.58**	
	SENet	64	39.15	38.20	41.93	0.2
		128	38.84	40.16	43.28	
		256	38.64	40.79	**43.78**	
	ViT-B16	64	73.35	61.31	72.55	0.4
		128	76.12	69.00	74.24	
		256	**76.98**	72.87	74.41	

We also conducted a broader comparison against a diverse set of image classification approaches, as reported in Table 3. In this setting, the comparison is not limited to graph neural network-based approaches, but also includes a wider range of traditional and recent classification methods evaluated under the same experimental protocol. All methods were run using identical train/test splits and feature extractors to ensure a fair comparison. In this table, for our approach as well as DGCG and MGCN, we report the best accuracy achieved across both SGC and APPNP. Note that CoMatch [Li et al. 2021] operates directly on input images rather than extracted features and therefore provides the same accuracy regardless of the feature extractor. We also include the graph-based semi-supervised method GNN-LDS [Franceschi et al. 2019], the Graph Attention Network (GAT) [Veličković et al. 2018], as well as WSEF [Presotto et al. 2021], which leverages weakly supervised embeddings. Additionally, we included two commonly adopted semi-supervised strategies as baselines: label spreading (LS), which propagates labels to unlabeled samples before classification, and pseudolabeling (PL), a self-training approach that iteratively assigns pseudo-labels based on model predictions. The OPF classifier [Amorim et al. 2014] is also considered in both its standard and label-propagation variants. The abbreviation ML-perc denotes multi-layer perceptrons. When a reference is not provided, the method corresponds to a traditional approach from the *sklearn* library. Overall, GRaNDe achieved the best results in the majority of cases.

4. Conclusion

In this work, we introduced GRaNDe, a novel degree function for weighting nodes during GNN training. Many traditional diffusion-based GNNs rely on degree-based normal-

Table 3. Accuracy (%) for semi-supervised image classification using the same features, with only 10% training data. For each method, we report the best accuracy. Our results are in green, and the best per row is in bold.

Dataset	Features	CoMatch	SVM	OPF	ML-Perc.	PL+SGD	LS+kNN	LS+SVM	LS+OPF	LS+ML-Perc.	GNN-LDS	WSEF	GAT	MGCN	DGCG	GRaNDe (ours)
Flowers	ResNet	82.55	80.54	71.77	78.88	82.69	73.49	73.53	72.66	73.03	79.32	85.12	83.89	85.88	85.68	**86.19**
	SENet	**82.55**	73.30	64.00	72.62	76.87	58.05	59.84	59.25	59.39	73.69	76.16	76.74	78.82	78.42	80.10
	ViT-B16	82.55	96.75	96.50	92.59	96.84	95.74	94.49	94.22	95.13	96.66	97.82	96.99	97.43	98.09	**98.18**
Core15k	ResNet	85.70	88.73	83.56	87.10	89.76	83.98	83.26	82.32	82.53	88.94	91.68	92.02	93.08	91.48	**93.25**
	SENet	85.70	85.89	81.33	86.90	89.85	72.16	72.79	72.20	72.24	89.95	89.74	92.23	92.79	90.88	**93.42**
	ViT-B16	85.70	91.92	90.02	74.41	89.07	89.63	87.59	86.14	87.68	88.56	94.00	93.64	95.57	95.29	**95.77**
CUB200	ResNet	38.29	48.84	38.59	32.24	21.67	36.99	38.70	39.28	39.68	37.78	52.17	45.46	52.85	54.43	**54.58**
	SENet	38.29	35.32	30.94	32.15	20.96	20.00	24.82	25.38	25.72	—	36.49	33.02	40.31	41.00	**43.78**
	ViT-B16	38.29	75.61	73.27	12.02	30.19	66.15	66.81	66.68	62.81	52.42	78.64	78.03	79.27	**81.23**	78.58

ization, attenuating messages from highly connected nodes and amplifying those from sparsely connected ones, regardless of neighborhood quality. In contrast, GRaNDe extends this formulation by explicitly incorporating a neighborhood penalty score. Nodes whose neighbors are farther apart in the feature space receive higher penalties, while nodes with closer neighbors are penalized less. In addition, this penalty is computed using a Gaussian kernel applied to distances in the feature space that are updated at each propagation step, allowing GRaNDe to adapt to the node representations learned during training.

Experimental results on semi-supervised image classification tasks revealed that replacing standard degree centrality with GRaNDe improved the effectiveness in most evaluated settings, achieving results that are superior or comparable to the state-of-the-art. As future work, we intend to evaluate GRaNDe in a broader range of scenarios, including citation networks, textual data, and other multimodal settings such as video and audio. We also plan to investigate the integration of GRaNDe with alternative graph construction strategies and propagation mechanisms, as well as its applicability to different GNN architectures beyond diffusion-based models. We also intend to investigate automatic and adaptive strategies for selecting the Gaussian parameter σ to better accommodate varying data distributions.

Acknowledgments

We thank the São Paulo Research Foundation – FAPESP (grant #2025/10602-5) and the University of São Paulo (PRPI Ordinance No. 1032, "*Apoio aos Novos Docentes*") for financial support.

AI Usage Declaration

The authors used AI-based language models (Claude and Gemini) exclusively for reviewing and improving the clarity of text written by the authors.

References

Amini, M.-R., Feofanov, V., Pauletto, L., Hadjadj, L., Émilie Devijver, and Maximov, Y. (2025). Self-training: A survey. *Neurocomputing*, 616:128904.

Amorim, W. P., Falcão, A. X., and d. Carvalho, M. H. (2014). Semi-supervised pattern classification using optimum-path forest. In *27th SIBGRAPI Conference on Graphics, Patterns and Images*, pages 111–118.

Angonese, S. F. and Galante, R. (2026). Evaluating heterogeneous node embedding compositions using diversity metrics. *Journal of Information and Data Management*, 17(1):17–25.

Brito, G. M. and Valem, L. P. (2025). Density-guided rank correlation graphs for graph convolutional networks in image classification. In *2025 38th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 1–6.

Chen, Y., Li, J., Xiao, H., Jin, X., Yan, S., and Feng, J. (2017). Dual path networks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4470–4478. Curran Associates Inc.

Chen, Y., Tang, X., Qi, X., Li, C.-G., and Xiao, R. (2022). Learning graph normalization for graph neural networks. *Neurocomputing*, 493:613–625.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

Figuerêdo, J., Maia, A., and Calumby, R. (2023). A novel graph-based diversity-aware rank fusion method applied to image metasearch. In *Anais do XXXVIII Simpósio Brasileiro de Bancos de Dados*, pages 324–329, Porto Alegre, RS, Brasil. SBC.

Franceschi, L., Niepert, M., Pontil, M., and He, X. (2019). Learning discrete structures for graph neural networks. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1972–1982. PMLR.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Hu, J., Shen, L., and Sun, G. (2018). Squeeze-and-excitation networks. In *2018 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Jiang, W. and Bai, Y. (2024). Sgcl: Semi-supervised graph contrastive learning with confidence propagation algorithm for node classification. *Knowledge-Based Systems*, 301:112271.

Khemani, B., Patil, S., Kotecha, K., and Tanwar, S. (2024). A review of graph neural networks: concepts, architectures, techniques, challenges, datasets, applications, and future directions. *Journal of Big Data*, 11(1):18.

Khosla, A., Jayadevaprakash, N., Yao, B., and Fei-Fei, L. (2011). Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO.

Khoushehgir, F., Noshad, Z., Noshad, M., and Sulaimany, S. (2024). Npi-wgnn: A weighted graph neural network leveraging centrality measures and high-order common neighbor similarity for accurate ncrna–protein interaction prediction. *Analytics*, 3(4):476–492.

Klicpera, J., Bojchevski, A., and Günnemann, S. (2019). Combining neural networks with personalized pagerank for classification on graphs. In *International Conference on Learning Representations*.

Li, J., Xiong, C., and Hoi, S. C. (2021). Comatch: Semi-supervised learning with contrastive graph regularization. In *International Conference on Computer Vision (ICCV)*, pages 9475–9484.

Liang, B., Chen, S., Gui, L., WANG, H., Yu, Y., Xu, R., and Wong, K.-F. (2025). Centrality-guided pre-training for graph. In Yue, Y., Garg, A., Peng, N., Sha, F., and Yu, R., editors, *International Conference on Learning Representations*, volume 2025, pages 64863–64888.

Liu, G.-H. and Yang, J.-Y. (2013). Content-based image retrieval using color difference histogram. *Pattern Recognition*, 46(1):188 – 198.

Nass, C., Díaz, A. O., and Baldo, F. (2021). Ssl-vfc4.5: An approach to adapt very fast c4.5 classification algorithm to deal with semi-supervised learning. In *Anais do XXXVI Simpósio Brasileiro de Bancos de Dados*, pages 13–24, Porto Alegre, RS, Brasil. SBC.

Nilsback, M.-E. and Zisserman, A. (2006). A visual vocabulary for flower classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1447–1454.

Parkhi, O. M., Vedaldi, A., Zisserman, A., and Jawahar, C. V. (2012). Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Presotto, J. G. C., Valem, L. P., de Sá, N. G., Pedronette, D. C. G., and Papa, J. P. (2021). Weakly supervised learning through rank-based contextual measures. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 5752–5759.

Valem, L. P., Guimarães Pedronette, D. C., and Latecki, L. J. (2023). Graph convolutional networks based on manifold learning for semi-supervised image classification. *Computer Vision and Image Understanding*, 227:103618.

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

Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. (2011). The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology.

Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., and Weinberger, K. (2019). Simplifying graph convolutional networks. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6861–6871. PMLR.

Yang, J., Li, H., Du, B., and Ye, M. (2025). Cheb-gr: Rethinking k-nearest neighbor search in re-ranking for person re-identification. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19261–19270.

Yu, J., Wu, Z., Lu, J., Wang, T., and Wang, H. (2025). A centrality-based graph learning framework. In Kwok, J., editor, *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, pages 3588–3596. International Joint Conferences on Artificial Intelligence Organization. Main Track.