

Density-Guided Rank Correlation Graphs for Graph Convolutional Networks in Image Classification

Gabriel Maia Brito* and Lucas Pascotti Valem†

*São Carlos Institute of Physics (IFSC), University of São Paulo (USP), São Carlos, Brazil

†Institute of Mathematics and Computer Sciences (ICMC), University of São Paulo (USP), São Carlos, Brazil

Emails: gabrielmaiaab@usp.br, lucas@icmc.usp.br

Abstract—Graph Convolutional Networks (GCNs) have shown promising results in semi-supervised learning tasks, yet their effectiveness is highly dependent on the quality of the input graph. In image classification scenarios, graph construction remains a challenging step due to the general lack of inherent structural relationships between images. In this work, we propose a novel method to build input graphs for GCNs based on rank correlation measures between image similarity rankings. Each image is represented as a node, and edges are established according to the correlation between its ranked list of neighbors and those of other images. We introduce a density-guided strategy for automatically selecting the correlation threshold that controls the sparsity of the graph. Experiments conducted on three image classification datasets using three feature extractors and three GCN architectures show that the proposed correlation-based graphs outperform standard kNN and reciprocal kNN graphs in most cases, especially when used with the Simplified Graph Convolution (GCN-SGC) model. Our method surpasses several traditional and recent baselines, including techniques based on manifold learning and label propagation, while relying solely on contextual similarity through rank correlation without any post-processing refinement. The proposed approach’s source code and documentation are publicly available at corgcn.lucasvalem.com.

I. INTRODUCTION

Graph Convolutional Networks (GCNs) have established themselves as promising approaches for semi-supervised learning tasks, standing out for their ability to exploit structural relationships in data across various applications [1]. Even with the emergence of more recent approaches, such as *Graph Transformers*, traditional GCNs still achieve competitive effectiveness, often comparable to the state of the art in many scenarios [2]. However, these models are sensitive to graph structure, so effective graph modeling is crucial for successful learning.

Despite their potential, the application of GCNs to image datasets remains limited compared to other domains, mainly due to the challenges involved in graph modeling [1]. Improper graph construction can hinder the GCN’s ability to propagate relevant information between nodes, negatively impacting the model’s effectiveness. In many cases, the adopted strategy builds a separate graph for each image and then compares those graphs. Although this approach allows the representation of complex visual features, it imposes a high computational cost, especially in the pairwise graph comparison stages.

This work aims to investigate the use of contextual information, which involves comparing elements and their relationships with their respective neighbors in the similarity space, going beyond strictly pairwise comparisons [3], [4]. To this end, each image is modeled as a vertex in a graph, and edges are established based on contextual similarity measures that consider both the direct similarity between two vertices and the distribution of their neighbors. In the literature, there are works that construct kNN neighborhood graphs, which connect

each vertex to its k nearest neighbors according to a metric (e.g., Euclidean distance) in the embedding space [5].

The main goal of this research is to model graphs not in the traditional Euclidean space, but using ranking-based contextual similarity measures derived from rank correlation [3]. In this way, the edges established between vertices reflect not only pointwise similarity but also the information contained in their neighborhoods. The hypothesis is that this approach will improve the effectiveness of the results compared to existing approaches in image classification.

The main contributions of this work are as follows:

- We propose the construction of a graph in which edges are defined based on the rank correlation between images, enabling the incorporation of contextual information beyond simple pairwise similarities. Although correlation-based graphs have been explored in other works, particularly for image retrieval [6], our contribution lies in adapting and refining this structure as input to a GCN.
- The definition of a threshold, which varies according to different datasets, is a challenging aspect of correlation graphs. This work proposes an algorithm and evaluates the use of the density coefficient to automatically choose the appropriate threshold without user intervention, yielding significantly sparser graphs than kNN and baselines.
- A generalized Jaccard correlation function is introduced, which yields several variants, including JaccardMax and JaccardMean. While JaccardMax has demonstrated significant results in image retrieval tasks [6], to the best of our knowledge, this is the first work to evaluate JaccardMedian in image classification.
- A comprehensive experimental evaluation was conducted on multiple datasets, considering different visual feature extractors and semi-supervised learning scenarios. The results reveal that our approach outperforms several traditional and recent methods, validating the effectiveness of the proposed approach.

II. PROPOSED APPROACH

The proposed approach follows a series of sequential steps, as depicted in Figure 1, with graph modeling serving as a central component of the process (step 3), being the central contribution. The method unfolds as follows:

- 1. Feature Extraction:** Features are extracted for each image in the dataset using pre-trained deep neural networks.
- 2. Compute Ranked Lists:** Ranked lists of most similar images are generated for each sample based on feature similarity.
- 3. Compute Density-Guided Correlation Graph:** Rank correlation is computed between images based on the similarity of their ranked lists. A graph is constructed using our density-guided correlation thresholding approach.

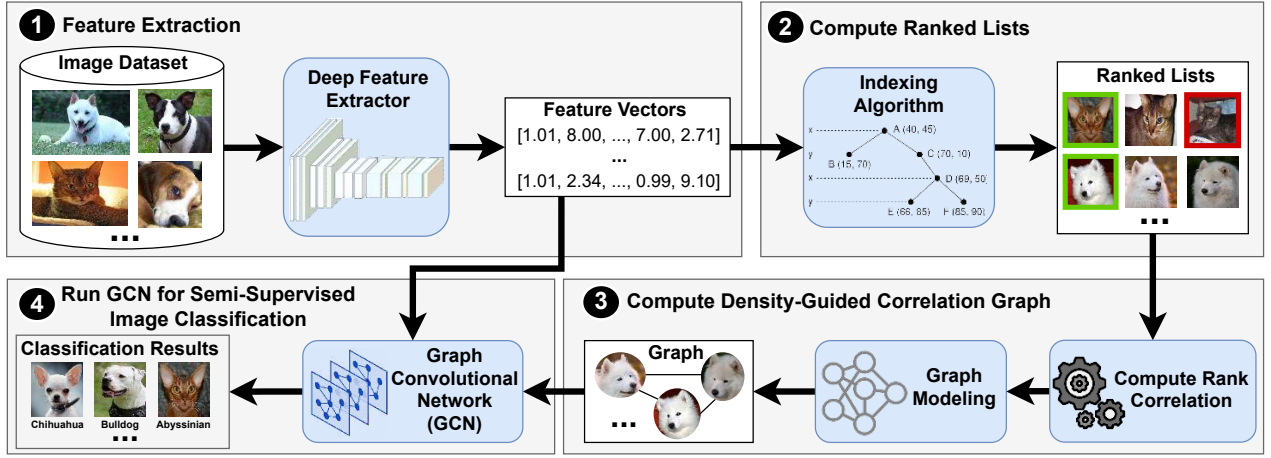


Fig. 1: Workflow of the steps that compose the proposed approach.

4. Run GCN for Semi-Supervised Image Classification:

A Graph Convolutional Network (GCN) is trained over the constructed graph and the original features. Final class predictions are obtained via the GCN in a semi-supervised classification setting.

The remainder of this section is organized as follows. Section II-A presents the formal rank model, which provides the foundation for computing the correlation measures. Section II-B details the rank correlation measures considered. Section II-C outlines the graph modeling approaches, covering both baseline methods and our proposed density-guided rank correlation graph construction.

A. Formal Rank Model

This work builds upon concepts from image retrieval, which serve as the foundation for defining the correlation measures. The image retrieval task consists of identifying and returning the most similar images from a collection $\mathcal{C}$ in response to a query image x_q . This process typically relies on features extracted from each image, which encode visual content into a high-dimensional space. Formally, a feature descriptor is defined as a function $f: \mathcal{C} \rightarrow \mathbb{R}^d$ that maps an image to a d -dimensional feature vector. For an image $x_i \in \mathcal{C}$, its representation is given by $\mathbf{x}_i = f(x_i) = [x_{i1}, x_{i2}, \dots, x_{id}]$, where $x_{ij} \in \mathbb{R}$ denotes the j -th feature component.

Thus, an image dataset can be represented as $\mathcal{C} = \{x_1, x_2, \dots, x_n\}$, where each image x_i is described by a feature vector $\mathbf{x}_i \in \mathbb{R}^d$. The resulting set of feature vectors is denoted by $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\} \subset \mathbb{R}^d$. The similarity between two images is defined by the distance between their feature vectors. Let $\rho: \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^+$ be a distance function, typically Euclidean. The distance between images x_i and x_j is given by $\rho(\mathbf{x}_i, \mathbf{x}_j)$, where $\mathbf{x}_i \in \mathbb{R}^d$ is the feature vector of x_i .

The k -nearest neighbors of a query image x_q form the set $\mathcal{N}(x_q, k)$, containing the k closest elements from $\mathcal{X}$ according to ρ . The ranked list τ_q is a permutation of $\mathcal{X}$ sorted by increasing distance to x_q , where $\tau_q(x_i) < \tau_q(x_j)$ implies $\rho(x_q, \mathbf{x}_i) \leq \rho(x_q, \mathbf{x}_j)$.

B. Rank Correlation Measures

To quantify the similarity between ranked lists, we adopt correlation measures that evaluate the degree of agreement

among top- k elements. Given two ranked lists τ_i and τ_j , and a neighborhood depth k , a measure returns a value in the range $[0, 1]$, where higher values indicate stronger agreement.

1) *Rank-Biased Overlap (RBO)*: For computing the correlation between ranked lists, the Rank-Biased Overlap (RBO) [3] measure is used. This measure considers the overlap between top- k lists at increasing depths. The weight of the overlap is calculated based on probabilities defined at each depth. It can be formally defined as follows:

$$\lambda(\tau_i, \tau_j, k, \mu) = (1 - \mu) \sum_{d=1}^k \mu^{d-1} \times \frac{|\mathcal{N}(x_i, k) \cap \mathcal{N}(x_j, k)|}{d},$$

where $\mathcal{N}(x_i, k)$ denotes the top- k neighborhood for image x_i and μ is a constant ($\mu = 0.9$ was used in this research).

2) *Jaccard Correlations*: Inspired by the original Jaccard index, different variants have been proposed to detect strong similarity indications at different depths of ranked lists. In this work, we define the generalized Jaccard function as:

$$JacFn(\tau_i, \tau_j, k) = \text{Agg} \left(\left\{ \frac{|\mathcal{N}(x_i, k_d) \cap \mathcal{N}(x_j, k_d)|}{|\mathcal{N}(x_i, k_d) \cup \mathcal{N}(x_j, k_d)|} \right\}_{k_d=1}^k \right).$$

Here, $\text{Agg}(\cdot)$ denotes an aggregation function over the Jaccard similarities computed at different depths $k_d \in [1, k]$. This function can be instantiated as the *maximum* (JacMax), *mean* (JacMean, usually known as JaccardK), or *median* (JacMed), depending on the desired behavior.

C. Graph Modeling

In addition to feature vectors, the input graph of a GCN is fundamental to its learning process [7]. GCNs were originally proposed for datasets with an available graph or structured data. Despite recent advances, GCN-based approaches are still underexplored in image scenarios [8], [9] when compared to other domains. The main challenge lies in that, since the graph is not available in such scenarios, it must be constructed for these datasets. Most approaches build the graph using the kNN or reciprocal kNN strategy [5], [10], [11].

In this work, for all graphs $G = (V, E)$ constructed over a dataset of n images, such that $V = \mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, the difference lies in how the edge set E is defined.

1) *k*-Nearest Neighbor (*k*NN) Graph: Edges are formed by connecting each image to its *k* most similar neighbors:

$$E_{knn} = \{(i, j) \mid j \in \mathcal{N}(\mathbf{x}_i, k)\}.$$

2) *Reciprocal k*-Nearest Neighbor Graph: Edges are included only between pairs of images that are mutually among each other's *k* nearest neighbors:

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

This symmetric condition yields a sparser graph that retains only reciprocal relationships.

3) *Density-Guided Correlation Graph (ours)*: We propose to use rank correlation measures to build graphs provided as input to GCNs. We model the graph considering each vertex as representing an image in the dataset, with edges constructed between them if the correlation of their ranked lists exceeds a given threshold. Figure 2 illustrates a graph in which each vertex corresponds to an image from the collection, and edges are defined based on the correlation between ranked lists. Two example images, highlighted in blue, illustrate the process, showing a correlation of 0.82 between them. We hypothesize that visually similar images (shown in green) tend to exhibit higher correlation values, while images from different flower species (shown in red) are expected to have lower correlations.

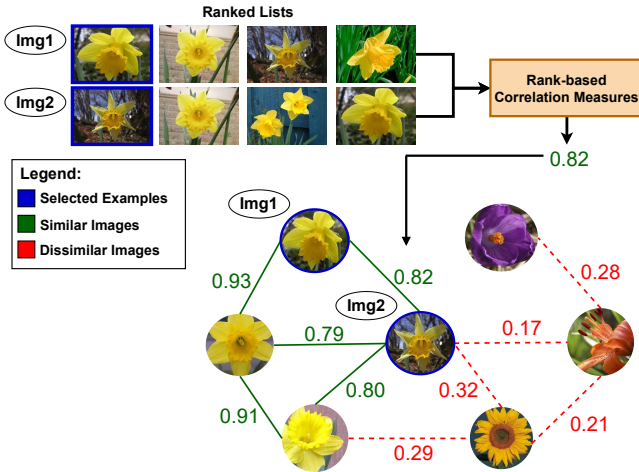


Fig. 2: Illustration of the correlation graph computed for the Flowers17 [12] dataset, where edge weights are determined by the correlation between ranked lists. It is expected that more similar images exhibit higher correlation (in green), while less similar images show lower correlation (in red).

The correlation graph $G_{corr} = (V, E_{corr})$ is defined by

$$E_{corr} = \{(i, j) \mid j \in \mathcal{N}(\mathbf{x}_i, L) \wedge c(\tau_i, \tau_j, k) > th\}.$$

For each image x_i we first identify its L nearest neighbors $\mathcal{N}_i$. Then, for each neighbor $x_j \in \mathcal{N}_i$, we compute the rank correlation $c(\tau_i, \tau_j, k)$ between their respective ranked lists up to k neighborhood depth. Whenever $c(\tau_i, \tau_j, k)$ exceeds the threshold th , the directed edge (i, j) is added to E_{corr} .

To automatically define the graph threshold th , we propose a density-based threshold approach. Given a dataset of size $n = |\mathcal{C}|$ and L nearest neighbors per node, the graph density is defined as $\delta = (|E_{corr}|)/(n \times L)$, where $|E_{corr}|$ denotes the number of edges in the correlation graph and $(n \times L)$

is the maximum number of edges possible. To standardize the computation of δ , we set $L = 200$ throughout the experiments.

Algorithm 1 performs a bisection search on the correlation threshold th in the interval $[0, 1]$ so that the resulting graph density δ lies within a user specified range $[\delta_{low}, \delta_{high}]$. Starting with bounds $low = 0$, $high = 1$ and $th = 0.5$, each iteration calls `CorrelationGraph` to construct $G_{corr} = (V, E_{corr})$. This function inspects each node's L nearest neighbors and adds a directed edge (i, j) whenever the rank correlation $c(\tau_i, \tau_j, k)$ exceeds th . After computing the new density, if $\delta < \delta_{low}$ the algorithm lowers the high bound to th , and if $\delta > \delta_{high}$ it raises the low bound to th and repeats until the density falls within the target interval or a maximum number of iterations is reached. In this study, the bisection search was limited to $I_{max} = 5$.

Algorithm 1: Density-Guided Rank Correlation Graph

Input : Image features $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, number of neighbors L , correlation depth k , density interval $[\delta_{low}, \delta_{high}]$, max iterations I_{max} .
Output: Graph $G_{corr} = (V, E_{corr})$ and threshold th
Initialization: $low \leftarrow 0$, $high \leftarrow 1$, $th \leftarrow 0.5$, $i \leftarrow 0$;
repeat
 $G_{corr} \leftarrow \text{CorrelationGraph}(\mathcal{X}, L, k, th)$;
 $\delta(G_{corr}) \leftarrow \frac{|E_{corr}|}{n \times L}$;
 if $\delta(G_{corr}) < \delta_{low}$ **then**
 $high \leftarrow th$;
 else if $\delta(G_{corr}) > \delta_{high}$ **then**
 $low \leftarrow th$;
 $th \leftarrow \frac{low + high}{2}$;
 $i \leftarrow i + 1$;
until $\delta_{low} \leq \delta(G_{corr}) \leq \delta_{high}$ **or** $i \geq I_{max}$;
return G_{corr}, th

III. EXPERIMENTAL EVALUATION

A. Datasets and Features

This work evaluated the results on three publicly available image datasets, described as follows:

- **Flowers17** [12]: consists of 1360 images evenly distributed across 17 flower categories. Each class contains 80 images captured under varying backgrounds and viewpoints, offering moderate intra-class variability.
- **Pets** [13]: comprises 7409 images from 37 pet classes, including high variability in pose, lighting, and background.
- **CUB200** [14]: contains 11788 images of 200 bird species. The dataset provides class labels for fine-grained recognition and localization studies.

For each dataset, we employed three deep learning models pretrained on the ImageNet dataset:

- **ResNet152** [15]: a deep residual CNN. We used the 2048-dimensional output from the final convolutional stage.
- **SENet154** [16]: a ResNet variant with channel-wise attention. Features were extracted from the final SE block (2048 dimensions).
- **ViT-Base** [17]: a transformer model operating on patch sequences. We used the 768-dimensional class token from the last encoder layer.

B. GCN Models and Setup

The following GCN models were evaluated:

- **SGC** [18]: Simplified Graph Convolution (SGC) removes intermediate nonlinearities and collapses multiple weight matrices into a single linear transformation applied after k -step adjacency propagation.
- **APPNP** [19]: Approximate Personalized Propagation of Neural Predictions (APPNP) decouples prediction and propagation by first generating node embeddings with a multilayer perceptron and then refining them via personalized PageRank propagation.
- **ARMA** [20]: employs AutoRegressive Moving Average (ARMA) filters to model graph convolution as a recursive propagation process, enabling deep filtering with stable frequency responses through stacked ARMA layers.

The GCN models were trained for 200 epochs with a learning rate of 10^{-3} across all datasets, except for CUB200, where a learning rate of 10^{-2} was used. We adopted the default hyperparameters provided by the PyTorch Geometric repository (https://github.com/pyg-team/pytorch_geometric). In all cases, we set $k = 40$ for computing the correlation measures and for the depth of kNN graphs. For our method, we evaluated $L = 200$ and $L = 40$, where L sets the edge selection scope and k defines the rank correlation depth.

All experiments followed a 10-fold cross-validation protocol, in which each fold was used once as the training set and the remaining nine as the test set. This setup ensures that every subset is used for training in at least one execution. Notably, this configuration trains the model on only 10% of the data while testing on the remaining 90%, thus creating a particularly challenging scenario with limited labeled data and few labeled samples per class. For all reported results, we present the mean accuracy and the sample standard deviation computed across folds. Each result corresponds to the average over 5 runs of the 10-fold protocol. We do not report the standard deviation across executions, as it consistently remained below 0.5 in all experiments.

C. Analysis of Graph Density Interval

To compute the correlation graph based on graph density, we first conducted an experiment to determine the most appropriate density interval. Figure 3 presents a surface approximation derived from multiple executions. The analysis was performed for both (a) *JaccardMax* and (b) *JaccardMedian* using the GCN-SGC model on the Flowers dataset with ResNet features. The SGC model was selected for this evaluation due to its high sensitivity to the graph structure [18]. The Flowers dataset was chosen for its relatively small size.

Each red point in the figure represents a distinct execution using a correlation threshold varying from 0.1 to 0.9 (nine values in total). For each execution, the corresponding graph density (δ) and classification accuracy are reported. The surface provides a visual approximation that highlights the relationship between graph density and other parameters. The plot indicates that the highest accuracy is obtained with lower-density graphs. Based on this observation, we selected the density interval $[\delta_{\text{low}}, \delta_{\text{high}}] = [0.03, 0.04]$ for all subsequent experiments. With the current setup, our graphs are at least $5 \times$ sparser than kNN graphs, as detailed in the supplementary material. Sparse graphs are generally advantageous for GCNs as they help mitigate over-smoothing, reduce noise propagation, and preserve the local structure of the data, thereby enabling more effective feature aggregation [10].

D. Semi-Supervised Classification Results

A comprehensive experimental evaluation was conducted on three datasets, employing three distinct feature extractors and three GCN models. As baselines, we considered both kNN and reciprocal kNN graphs and compared their classification accuracy against our approach, which leverages density-guided correlation graphs for semi-supervised image classification. The evaluation considered four rank correlation measures: RBO, JaccardK, JaccardMedian, and JaccardMax.

Tables I and II present the results for two L values. The setting $L = 200$ provides a broader neighborhood for edge selection, while $L = 40 = k$ allows assessing the impact of aligning neighborhood size with the correlation depth. We observe that graphs constructed with rank correlation and guided by a density-based threshold consistently outperform both kNN and reciprocal kNN baselines across almost all dataset–feature–model combinations. This holds for both settings of the graph size parameter L , indicating that our adaptive threshold selection is robust to the choice of L .

The SGC model benefits most from the proposed graphs. For instance, on Flowers17 with ViT features, the Jaccard-Median graph achieves 98.08% accuracy, a 1.22% relative gain over the best reciprocal kNN graph. On CUB200 with ResNet features and APPNP, accuracy increases from 48.39% (reciprocal kNN) to 52.52%, corresponding to an 8.53% relative gain. Even ARMA, which is generally less sensitive to graph structure, exhibits improvements in many scenarios.

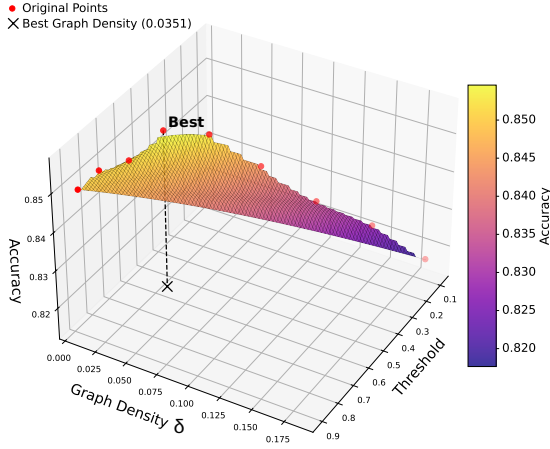
E. Comparison with Other Approaches

Table III reports the best results achieved by our proposed approach across Tables I and II, focusing on the Flowers and CUB200 datasets (the smallest and largest, respectively) for each feature extractor. These results are compared against both traditional and recent methods under the same train/test splits for the semi-supervised image classification task. Note that CoMatch operates on input images instead of extracted features; therefore, the accuracy remains the same across feature extractors. We include two widely adopted semi-supervised learning strategies as baselines: a label-spreading (LS) algorithm that propagates labels to unlabeled samples before classification, and pseudolabel (PL), a self-training approach that iteratively assigns pseudo-labels based on model predictions. The abbreviations SL-prec. and ML-prec. refer to single-layer and multi-layer perceptrons, respectively. Using ViT features, our approach outperformed all baselines. For other features, it achieved the highest accuracy on CUB200 and remains highly competitive on Flowers. All baselines are cited accordingly; when a reference is not provided, it refers to a traditional method used from the *scikit-learn* library.

IV. CONCLUSION

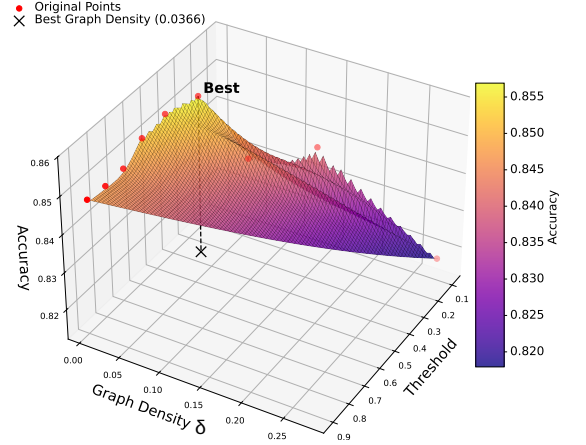
In this work, we introduced rank correlation graphs as input to GCNs to enhance semi-supervised image classification. By leveraging graph density, our method enables the automatic selection of the correlation threshold. Experimental results, conducted on three image classification datasets using three feature extractors and three different GNN architectures, reveal that correlation-based graphs outperformed both kNN and reciprocal kNN graphs in most cases, especially when used with the SGC model, achieving results superior to most existing approaches. In contrast to other methods, our approach does not rely on manifold learning or post-processing steps to refine the graph structure.

Impact of Correlation Thresholding on Graph Density and Accuracy



(a) JaccardMax

Impact of Correlation Thresholding on Graph Density and Accuracy



(b) JaccardMedian

Fig. 3: Surface plots showing the impact of correlation thresholding on graph density and classification accuracy using the GCN-SGC model with ResNet features on the Flowers dataset.

As future work, we aim to investigate different GCN architectures. We also intend to explore strategies for combining different graph constructions, such as reciprocal and correlation-based graphs, to better leverage their complementary properties. Another promising direction is using manifold learning as a post-processing step to refine the latent feature space and improve the proposed correlation graph structures.

ACKNOWLEDGMENT

We thank the São Paulo Research Foundation – FAPESP (grant #2025/10602-5), the Institute of Mathematics and Computer Sciences (ICMC), the São Carlos Institute of Physics (IFSC), the Research and Innovation Committee (CPqI-ICMC), and the University of São Paulo (PRPI Ordinance No. 1032, “*Apoio aos Novos Docentes*”) for financial support.

REFERENCES

- [1] B. Zhang, C. Fan, S. Liu, K. Huang, X. Zhao, J. Huang, and Z. Liu, “The Expressive Power of Graph Neural Networks: A Survey,” *IEEE Transactions on Knowledge & Data Engineering*, vol. 37, no. 03, pp. 1455–1474, Mar. 2025.
- [2] Y. Luo, L. Shi, and X.-M. Wu, “Classic GNNs are strong baselines: Reassessing GNNs for node classification,” in *The 38th Conf. on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.
- [3] W. Webber, A. Moffat, and J. Zobel, “A similarity measure for indefinite rankings,” *ACM Transactions on Information Systems*, vol. 28, no. 4, pp. 20:1–20:38, 2010.
- [4] J. G. C. Presotto, L. P. Valem, N. G. de Sá, D. C. G. Pedronette, and J. P. Papa, “Weakly supervised learning through rank-based contextual measures,” in *2020 25th International Conference on Pattern Recognition (ICPR)*, 2021, pp. 5752–5759.
- [5] L. P. Valem, D. C. G. Pedronette, and L. J. Latecki, “Graph convolutional networks based on manifold learning for semi-supervised image classification,” *Computer Vision and Image Understanding*, vol. 227, p. 103618, 2023.
- [6] V. Pereira-Ferrero, T. Lewis, L. Valem, L. Ferrero, D. Pedronette, and L. Latecki, “Unsupervised affinity learning based on manifold analysis for image retrieval: A survey,” *Computer Science Review*, vol. 53, p. 100657, 2024.
- [7] W. Ju, S. Yi, Y. Wang, Z. Xiao, Z. Mao, H. Li, Y. Gu, Y. Qin, N. Yin, S. Wang, X. Liu, X. Luo, P. S. Yu, and M. Zhang, “A survey of graph neural networks in real world: Imbalance, noise, privacy and ood challenges,” *arXiv preprint arXiv:2403.04468*, 2024.
- [8] J. Rodrigues and J. Carbonera, “Graph convolutional networks for image classification: Comparing approaches for building graphs from images,” in *Proceedings of the 26th International Conference on Enterprise Information Systems - Volume 1: ICEIS, INSTICC*. SciTePress, 2024, pp. 437–446.
- [9] G. Zhang, J. Pan, Z. Zhang, H. Zhang, C. Xing, B. Sun, and M. Li, “Hybrid graph convolutional network for semi-supervised retinal image classification,” *IEEE Access*, vol. 9, pp. 35 778–35 789, 2021.
- [10] H. Senior, G. Slabaugh, S. Yuan, and L. Rossi, “Graph neural networks in vision-language image understanding: a survey,” *The Visual Computer*, vol. 41, pp. 491–516, 03 2024.
- [11] A. Sadasivan, K. Gananathan, J. D. Pal Nesamony Rose Mary, and S. Balasubramanian, “A systematic survey of graph convolutional networks for artificial intelligence applications,” *WIREs Data Mining and Knowledge Discovery*, vol. 15, no. 2, p. e70012, 2025, e70012 DMKD-00790.R1.
- [12] M.-E. Nilsback and A. Zisserman, “A visual vocabulary for flower classification,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2006, pp. 1447–1454.
- [13] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. V. Jawahar, “Cats and dogs,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.
- [14] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, “The Caltech-UCSD Birds-200-2011 Dataset,” California Institute of Technology, Tech. Rep. CNS-TR-2011-001, 2011.
- [15] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
- [16] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *2018 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [17] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale,” in *International Conference on Learning Representations*, 2021.
- [18] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger, “Simplifying graph convolutional networks,” in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 6861–6871.
- [19] J. Klicpera, A. Bojchevski, and S. Günnemann, “Combining neural networks with personalized pagerank for classification on graphs,” in *International Conference on Learning Representations*, 2019.
- [20] F. M. Bianchi, D. Grattarola, L. Livi, and C. Alippi, “Graph neural networks with convolutional arma filters,” *IEEE TPAMI*, pp. 1–1, 2021.
- [21] J. Li, C. Xiong, and S. C. Hoi, “Comatch: Semi-supervised learning with contrastive graph regularization,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 9475–9484.
- [22] W. P. Amorim, A. X. Falcão, and M. H. d. Carvalho, “Semi-supervised pattern classification using optimum-path forest,” in *27th SIBGRAPI Conference on Graphics, Patterns and Images*, 2014, pp. 111–118.
- [23] L. Franceschi, M. Niepert, M. Pontil, and X. He, “Learning discrete structures for graph neural networks,” in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 1972–1982.

TABLE I: **Accuracy (%) results for $L = 200$.** For each row, the best-performing correlation graph is highlighted in bold. The overall best result for each dataset and feature extractor is marked with a gray background and blue font. Relative gains are reported for both kNN and reciprocal kNN graphs, with positive gains shown in green and negative gains in red.

Dataset	Feature Extractor	GCN Model	Baseline Graphs		Correlation Graphs (ours)				Relative Gains Over	
			kNN	Rec. kNN	RBO	JaccardK	JaccardMed	JaccardMax	kNN	Rec. kNN
Flowers17	ResNet	SGC	79.68 $\pm$ 1.08	83.98 $\pm$ 1.77	84.33 $\pm$ 2.37	85.41 $\pm$ 1.30	85.68 $\pm$ 1.46	85.55 $\pm$ 2.16	+7.53%	+2.02%
		APNP	77.16 $\pm$ 1.53	83.13 $\pm$ 1.43	84.83 $\pm$ 1.49	85.13 $\pm$ 1.12	85.29 $\pm$ 1.14	82.85 $\pm$ 1.48	+10.53%	+2.60%
		ARMA	78.56 $\pm$ 1.40	83.24 $\pm$ 1.50	83.65 $\pm$ 2.01	82.08 $\pm$ 2.00	82.07 $\pm$ 1.80	81.59 $\pm$ 2.30	+6.47%	+0.49%
	SeNet	SGC	72.93 $\pm$ 1.78	76.17 $\pm$ 2.21	78.42 $\pm$ 1.96	76.20 $\pm$ 2.54	76.08 $\pm$ 2.45	77.15 $\pm$ 3.03	+7.52%	+2.95%
		APNP	70.80 $\pm$ 1.82	76.98 $\pm$ 1.33	78.35 $\pm$ 1.88	77.79 $\pm$ 2.00	77.59 $\pm$ 1.64	75.97 $\pm$ 2.73	+10.66%	+1.77%
		ARMA	72.81 $\pm$ 1.79	76.89 $\pm$ 1.63	77.39 $\pm$ 1.55	75.54 $\pm$ 2.47	75.59 $\pm$ 2.14	76.33 $\pm$ 2.43	+6.29%	+0.65%
	ViT	SGC	92.85 $\pm$ 1.22	96.90 $\pm$ 0.51	98.01 $\pm$ 0.31	97.99 $\pm$ 0.41	98.08 $\pm$ 0.36	98.06 $\pm$ 0.44	+5.63%	+1.22%
		APNP	90.14 $\pm$ 1.71	96.95 $\pm$ 0.56	97.85 $\pm$ 0.51	97.77 $\pm$ 0.47	97.82 $\pm$ 0.50	96.96 $\pm$ 0.60	+8.55%	+0.92%
		ARMA	91.10 $\pm$ 0.50	96.73 $\pm$ 0.40	96.76 $\pm$ 0.78	95.76 $\pm$ 0.71	95.08 $\pm$ 0.89	94.86 $\pm$ 1.26	+6.21%	+0.03%
Pets	ResNet	SGC	87.79 $\pm$ 0.90	89.50 $\pm$ 0.68	90.32 $\pm$ 0.67	90.04 $\pm$ 1.27	89.99 $\pm$ 1.16	89.96 $\pm$ 1.17	+2.88%	+0.92%
		APNP	87.99 $\pm$ 0.75	90.50 $\pm$ 0.74	90.50 $\pm$ 0.58	90.62 $\pm$ 0.77	90.62 $\pm$ 0.80	90.65 $\pm$ 0.77	+3.02%	+0.17%
		ARMA	87.69 $\pm$ 0.79	89.44 $\pm$ 0.79	89.36 $\pm$ 0.84	88.06 $\pm$ 0.99	87.99 $\pm$ 0.96	88.18 $\pm$ 1.11	+1.90%	-0.09%
	SeNet	SGC	85.55 $\pm$ 0.87	89.05 $\pm$ 0.76	89.67 $\pm$ 0.48	89.59 $\pm$ 0.59	89.53 $\pm$ 0.66	89.69 $\pm$ 0.69	+4.84%	+0.72%
		APNP	84.94 $\pm$ 0.66	90.11 $\pm$ 0.50	89.80 $\pm$ 0.54	90.08 $\pm$ 0.58	90.05 $\pm$ 0.51	90.17 $\pm$ 0.57	+6.16%	+0.07%
		ARMA	86.51 $\pm$ 0.48	89.55 $\pm$ 0.50	89.47 $\pm$ 0.53	89.30 $\pm$ 0.55	89.06 $\pm$ 0.58	89.20 $\pm$ 0.65	+3.42%	-0.09%
	ViT	SGC	85.16 $\pm$ 1.04	88.28 $\pm$ 0.94	89.48 $\pm$ 0.76	89.84 $\pm$ 0.72	89.70 $\pm$ 0.72	89.75 $\pm$ 0.72	+5.50%	+1.77%
		APNP	77.85 $\pm$ 0.84	86.98 $\pm$ 1.02	88.80 $\pm$ 1.14	88.56 $\pm$ 1.33	88.22 $\pm$ 1.32	88.41 $\pm$ 1.28	+14.06%	+2.09%
		ARMA	80.01 $\pm$ 1.25	85.01 $\pm$ 1.01	83.87 $\pm$ 1.42	80.40 $\pm$ 1.39	78.21 $\pm$ 1.59	78.40 $\pm$ 1.03	+4.82%	-1.34%
CUB200	ResNet	SGC	47.55 $\pm$ 0.34	53.71 $\pm$ 0.36	53.43 $\pm$ 0.65	54.43 $\pm$ 0.65	53.95 $\pm$ 0.70	53.52 $\pm$ 0.52	+14.47%	+1.34%
		APNP	30.73 $\pm$ 1.40	48.39 $\pm$ 0.41	52.52 $\pm$ 0.40	51.42 $\pm$ 0.59	50.25 $\pm$ 0.55	51.46 $\pm$ 0.42	+70.90%	+8.53%
		ARMA	38.55 $\pm$ 0.64	44.27 $\pm$ 0.49	46.39 $\pm$ 0.54	41.04 $\pm$ 0.59	38.67 $\pm$ 0.85	42.21 $\pm$ 0.88	+20.34%	+4.79%
	SeNet	SGC	36.48 $\pm$ 0.68	40.31 $\pm$ 0.25	39.77 $\pm$ 0.48	39.51 $\pm$ 0.60	38.95 $\pm$ 0.45	41.00 $\pm$ 0.50	+12.39%	+1.71%
		APNP	29.90 $\pm$ 0.48	38.11 $\pm$ 0.34	39.06 $\pm$ 0.30	38.49 $\pm$ 0.30	38.37 $\pm$ 0.31	39.12 $\pm$ 0.28	+30.81%	+2.65%
		ARMA	32.86 $\pm$ 0.54	34.22 $\pm$ 0.30	34.25 $\pm$ 0.26	31.40 $\pm$ 0.32	30.66 $\pm$ 0.35	30.82 $\pm$ 0.52	+4.23%	+0.09%
	ViT	SGC	74.23 $\pm$ 0.38	78.03 $\pm$ 0.42	81.23 $\pm$ 0.55	80.87 $\pm$ 0.58	80.80 $\pm$ 0.51	80.76 $\pm$ 0.46	+9.43%	+4.10%
		APNP	55.84 $\pm$ 1.38	68.48 $\pm$ 0.46	72.33 $\pm$ 0.77	63.63 $\pm$ 1.12	64.00 $\pm$ 1.02	61.75 $\pm$ 1.39	+29.53%	+5.62%
		ARMA	59.71 $\pm$ 0.85	64.90 $\pm$ 0.72	60.19 $\pm$ 0.73	45.28 $\pm$ 1.89	46.74 $\pm$ 1.45	42.71 $\pm$ 2.51	+0.80%	-7.26%

TABLE II: **Accuracy (%) results for $L = 40$.** For each row, the best-performing correlation graph is highlighted in bold. The overall best result for each dataset and feature extractor is marked with a gray background and blue font. Relative gains are reported for both kNN and reciprocal kNN graphs, with positive gains shown in green and negative gains in red.

Dataset	Feature Extractor	GCN Model	Baseline Graphs		Correlation Graphs (ours)				Relative Gains Over	
			kNN	Rec. kNN	RBO	JaccardK	JaccardMed	JaccardMax	kNN	Rec. kNN
Flowers17	ResNet	SGC	79.68 $\pm$ 1.08	83.98 $\pm$ 1.77	84.41 $\pm$ 2.49	85.28 $\pm$ 1.51	85.59 $\pm$ 1.43	85.40 $\pm$ 2.18	+7.42%	+1.92%
		APNP	77.16 $\pm$ 1.53	83.13 $\pm$ 1.43	84.13 $\pm$ 1.92	84.97 $\pm$ 1.13	85.13 $\pm$ 1.25	82.66 $\pm$ 1.79	+10.33%	+2.41%
		ARMA	78.56 $\pm$ 1.40	83.24 $\pm$ 1.50	83.50 $\pm$ 2.11	82.10 $\pm$ 1.94	82.11 $\pm$ 2.10	81.48 $\pm$ 2.02	+6.29%	+0.31%
	SeNet	SGC	72.93 $\pm$ 1.78	76.17 $\pm$ 2.21	77.96 $\pm$ 2.45	70.07 $\pm$ 2.96	76.19 $\pm$ 2.93	76.39 $\pm$ 2.96	+6.89%	+2.35%
		APNP	70.80 $\pm$ 1.82	76.98 $\pm$ 1.33	78.21 $\pm$ 2.35	78.04 $\pm$ 2.34	77.93 $\pm$ 2.00	75.67 $\pm$ 2.56	+10.47%	+1.59%
		ARMA	72.81 $\pm$ 1.79	76.89 $\pm$ 1.63	77.45 $\pm$ 1.93	75.52 $\pm$ 2.62	75.80 $\pm$ 2.49	76.27 $\pm$ 2.22	+6.37%	+0.73%
	ViT	SGC	92.85 $\pm$ 1.22	96.90 $\pm$ 0.51	98.08 $\pm$ 0.35	98.01 $\pm$ 0.41	98.09 $\pm$ 0.36	98.07 $\pm$ 0.45	+5.64%	+1.23%
		APNP	90.14 $\pm$ 1.71	96.95 $\pm$ 0.56	97.81 $\pm$ 0.56	97.82 $\pm$ 0.44	97.80 $\pm$ 0.50	96.93 $\pm$ 0.69	+8.52%	+0.90%
		ARMA	91.10 $\pm$ 0.50	96.73 $\pm$ 0.40	96.80 $\pm$ 1.02	95.33 $\pm$ 1.55	94.57 $\pm$ 2.01	94.91 $\pm$ 1.20	+6.26%	+0.07%
Pets	ResNet	SGC	87.79 $\pm$ 0.90	89.50 $\pm$ 0.68	90.16 $\pm$ 0.71	89.97 $\pm$ 0.96	90.07 $\pm$ 1.16	90.07 $\pm$ 1.13	+2.69%	+0.73%
		APNP	87.99 $\pm$ 0.75	90.50 $\pm$ 0.74	90.24 $\pm$ 0.71	90.44 $\pm$ 0.70	90.47 $\pm$ 0.77	90.47 $\pm$ 0.76	+2.82%	-0.03%
		ARMA	87.69 $\pm$ 0.79	89.44 $\pm$ 0.79	88.79 $\pm$ 1.02	88.54 $\pm$ 0.96	87.90 $\pm$ 1.32	88.38 $\pm$ 0.91	+1.25%	-0.73%
	SeNet	SGC	85.55 $\pm$ 0.87	89.05 $\pm$ 0.76	89.31 $\pm$ 0.60	89.45 $\pm$ 0.71	89.28 $\pm$ 0.79	89.35 $\pm$ 0.87	+4.56%	+0.45%
		APNP	84.94 $\pm$ 0.66	90.11 $\pm$ 0.50	89.29 $\pm$ 0.60	89.89 $\pm$ 0.65	89.80 $\pm$ 0.64	89.87 $\pm$ 0.55	+5.83%	-0.24%
		ARMA	86.51 $\pm$ 0.48	89.55 $\pm$ 0.50	89.07 $\pm$ 0.59	89.30 $\pm$ 0.60	89.17 $\pm$ 0.61	89.25 $\pm$ 0.57	+3.23%	-0.28%
	ViT	SGC	85.16 $\pm$ 1.04	88.28 $\pm$ 0.94	89.35 $\pm$ 0.88	89.78 $\pm$ 0.74	89.72 $\pm$ 0.65	89.68 $\pm$ 0.80	+5.43%	+1.70%
		APNP	77.85 $\pm$ 0.84	86.98 $\pm$ 1.02	88.19 $\pm$ 1.35	88.41 $\pm$ 1.32	88.23 $\pm$ 1.33	88.24 $\pm$ 1.31	+13.56%	+1.64%
		ARMA	80.01 $\pm$ 1.25	85.01 $\pm$ 1.01	83.31 $\pm$ 1.34	79.75 $\pm$ 1.41	79.21 $\pm$ 1.37	79.88 $\pm$ 1.24	+4.12%	-1.99%
CUB200	ResNet	SGC	47.55 $\pm$ 0.34	53.71 $\pm$ 0.36	51.66 $\pm$ 0.66	53.87 $\pm$ 0.61	53.50 $\pm$ 0.54	53.43 $\pm$ 0.47	+13.29%	+0.30%
		APNP	30.73 $\pm$ 1.40	48.39 $\pm$ 0.41	51.66 $\pm$ 0.33	51.24 $\pm$ 0.49	50.41 $\pm$ 0.50	51.09 $\pm$ 0.38	+68.11%	+6.76%
		ARMA	38.55 $\pm$ 0.64	44.27 $\pm$ 0.49	44.51 $\pm$ 0.67	42.92 $\pm$ 0.66	40.91 $\pm$ 0.84	42.58 $\pm$ 0.57	+15.46%	+0.54%
	SeNet	SGC	36.48 $\pm$ 0.68	40.31 $\pm$ 0.25	39.23 $\pm$ 0.53	39.86 $\pm$ 0.66	38.94 $\pm$ 0.54	40.32 $\pm$ 0.40	+10.53%	+0.03%
		APNP	29.90 $\pm$ 0.48	38.11 $\pm$ 0.34	39.28 $\pm$ 0.40	39.03 $\pm$ 0.44	38.47 $\pm$ 0.34	38.98 $\pm$ 0.45	+31.37%	+3.07%
		ARMA	32.86 $\pm$ 0.54	34.22 $\pm$ 0.30	34.30 $\pm$ 0.42	31.52 $\pm$ 0.42	31.78 $\pm$ 0.39	31.34 $\pm$ 0.51	+4.38%	+0.23%
	ViT	SGC	74.23 $\pm$ 0.38	78.03 $\pm$ 0.42	81.08 $\pm$ 0.62	80.86 $\pm$ 0.58	80.77 $\pm$ 0.55	80.75 $\pm$ 0.45	+9.23%	+3.90%
		APNP	55.84 $\pm$ 1.38	68.48 $\pm$ 0.46	70.34 $\pm$ 0.50	62.87 $\pm$ 1.31	64.15 $\pm$ 0.88	61.62 $\pm$ 0.88	+25.97%	+2.72%
		ARMA	59.71 $\pm$ 0.85	64.90 $\pm$ 0.72	57.02 $\pm$ 1.10	44.80 $\pm$ 1.17	46.24 $\pm$ 2.26	42.90 $\pm$ 1.71	-4.50%	-12.14%

TABLE III: Accuracy comparison (%) with other approaches from the literature using the same features, considering scenarios with only 10% of the training data. Our results are shown in blue, and the best accuracy per line is highlighted in bold.

												LS+	LS+	LS+	GNN-				
Features	Dataset	CoMatch [21]	kNN	SVM	OPF [22]	SL- Perc.	ML- Perc.	PL+ SGD	LS+ kNN	LS+ SVM	OPF [22]	SL- Perc.	ML- Perc.	LDS [23]	WSEF [4]	MGCN [5]	Ours		
ResNet	Flowers	82.55	63.67	80.54	71.77	75.44	78.88	82.69	73.49	73.53	72.66	72.34	73.03	79.32	85.12	85.88	85.68		
	CUB200	38.29	36.67	48.84	38.59	39.91	32.24	21.67	36.99	38.70	39.28	39.21	39.68	37.78	52.17	52.85	54.43		
SENet	Flowers	82.55	48.71	73.30	64.00	71.84	72.62	76.87	58.05	59.84	59.25	59.27	59.39	—	76.16	78.82	78.42		
	CUB200	38.29	22.23	35.32	30.94	36.39	32.15	20.96	20.00	24.82	25.38	25.41	25.72	37.78	36.49	40.31	41.00		
ViT-B16	Flowers	82.55	91.91	96.75	96.50	75.79	92.59	96.84	95.74	94.49	94.22	93.71	95.13	96.66	97.82	97.43	98.09		
	CUB200	38.29	56.62	75.61	73.27	70.84	12.02	30.19	66.15	66.81	66.68	65.45	62.81	52.42	78.64	79.27	81.23		