FROM GRAPH LOCAL EMBEDDING TO DEEP METRIC LEARNING

Anonymous authors
Paper under double-blind review

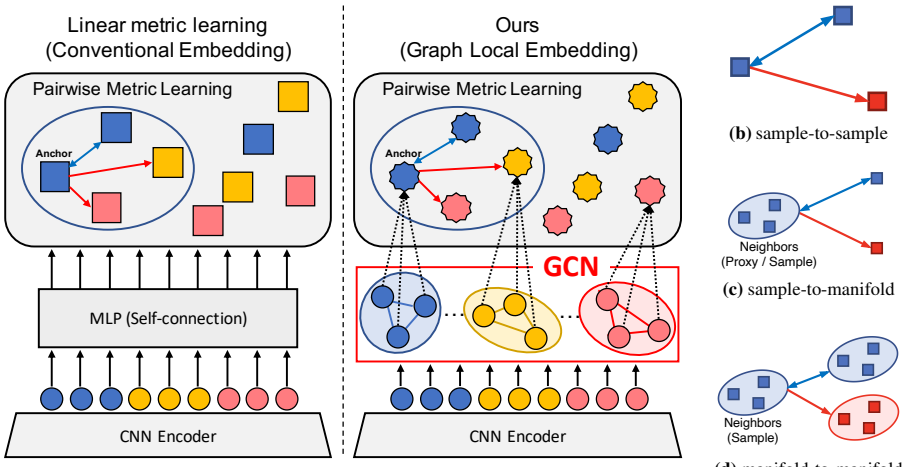
ABSTRACT

Deep metric learning continues to play a crucial role in many computer vision applications, while its various mining and weighting strategies have been extensively investigated. Techniques based on pairwise learning often use excessive random sampling and end up in slow convergence and model degradation. Further, neural network approaches mostly employ MLP layers for metric learning. The tactic can indeed be thought of as graph convolutions with only self-connections, indicating that local neighborhood relationships are neglected. We comprehensively identify the missing neighborhood relationships issue of conventional embedding and propose a novel approach, termed as Graph Local Embedding (GLE), to deep metric learning. Our method explores the local relationships and draws on the graph convolution networks to construct a discriminative mapping for embedding learning. The strategy can enhance metric learning by exploring the manifold-to-manifold relationships. By focusing on an essential variety of neighboring relations within GLE, burdens of redundant pairs can be substantially eased, and the context of each encoded data is greatly enriched. We demonstrate in the experiments that coupling GLE with existing metric learning techniques can yield impressive performance gains on popular benchmark datasets for fine-grained retrieval.

1 INTRODUCTION

Deep metric learning has become an active research topic and is widely applied in many areas, such as object/face recognition, few-shot learning, and image retrieval tasks. It mainly focuses on learning an embedding to encode data points of the same class to stay together and those of different classes to be far away. This is typically achieved by enforcing a loss function to promote intra-class compactness and inter-class separability effectively. In general, metric learning can be built on two kinds of scenarios, *classwise* and *pairwise*. The former aims to approximate the center of each class to realize the objective (Aziere & Todorovic, 2019; Deng et al., 2019; Kim et al., 2020; Liu et al., 2017; Movshovitz-Attias et al., 2017; Wang et al., 2018). However, it often calls for a rigid training procedure, not easily generalized to, *e.g.*, unseen classes. The latter divides training samples into pair or triplet relations and considers a metric function to build up feature discrimination (Schroff et al., 2015; Sohn, 2016; Song et al., 2016; Wah et al., 2011; Wang et al., 2019b;a; Manmatha et al., 2017; Zheng et al., 2019). Nevertheless, due to training with random sampling, it inevitably causes the resulting formulation involving too many redundant pairs, rather than a good number of informative samples, and consequently leads to slow convergence and model degradation, which could substantially limit the targeted performance improvement.

Among the abovementioned techniques, those that are based on the pairwise scheme often learn the embedding via a single multi-layer perceptron (MLP) layer, as shown in the left column (Conventional Embedding) of Figure 1a. Indeed, existing approaches essentially converge to designing various mining and weighting strategies to explore the informative samples for embedding learning. They rely heavily on the effectiveness of the mining and weighting schemes to model targeted pairs, *e.g.*, to properly group similar/positive pairs or dissimilar/negative pairs. The optimization then proceeds to promote compactness for those similar pairs and separation for dissimilar pairs. Such approaches mainly focus on optimizing the inter-sample, *i.e.*, *sample-to-sample*, correlations based on the neighborhood structure of the resulting embedding, as shown in Figure 1b. It implicitly im-



(a) Conventional MLP Embedding vs. Graph Local Embedding (GLE)

Figure 1: (a) Comparison between conventional embedding and our Graph Local Embedding (GLE). Our method considers the essential local neighboring relations to effectively construct the embedding via pairwise metric learning. Different colors and shapes reflect class labels and embedding spaces, respectively. (b–d) Three types of neighbor relationships in metric learning. (b) The *sample-to-sample* relationships correspond to the linear metric learning. (c) The *sample-to-manifold* relationships correspond to the message-passing based embedding. The proposed GLE aims to encode the local manifold for each sample to explore the (d) *manifold-to-manifold* relationships.

plies that the *neighbor relationships* among samples in the induced feature spaces of convolutional neural networks (CNNs) are mostly neglected during embedding learning.

Motivated by that neighbor relationships are crucial in manifold learning to construct a discriminative embedding, a new avenue of recent research efforts on metric learning has explored the mechanism of *message passing*, which exchanges pivotal information among neighboring nodes to unify a robust representation. Group loss in Elezi et al. (2020) takes a class-prior matrix and adopts hand-crafted rules to iteratively refine the class information, while MPN by Seidenschwarz et al. (2021) exploits the Transformer design to learn the attention scores. Zhu et al. (2020) instead propose ProxyGML that employs multiple intra-class proxies to construct local manifolds and takes the label propagation scheme to transfer the node information. The design principle of these approaches is essentially class-driven and the encoded representations are learned to correlate their class center—that is, the underlying constructed graph is used to impose *sample-to-manifold* relationships for metric learning, as illustrated in Figure 1c.

Deviating from adopting the MLP based sample-to-sample and the message-passing based sample-to-manifold mechanisms, our method, termed as *graph local embedding* (GLE), tackles the missing consideration of neighborhood relationships in conventional CNN-based embeddings by exploring the association between the MLP and the graph convolutional network (GCN) layer. Specifically, we point out that the MLP of metric learning can be cast in terms of graph convolution with only *self-connections*, indicating that local neighborhood relationships are neglected during the embedding construction. Such self-connections suggest that the pairwise deep metric learning techniques concern only the *sample-to-sample* relationships in learning the targeted embedding and may suffer the burden of redundant pairs. We instead model the local manifold of each data point and then construct a robust GCN representation to integrate the local neighborhood relationships, as shown in the right column (Graph Local Embedding) of Figure 1a. More specifically, by performing graph convolutions over a sparse graph, we can encode the local neighborhood relationship of each sample into the GCN feature. Hence, the succeeding metric learning can explore the relationships of inter-neighborhood, *i.e.*, *manifold-to-manifold*, as shown in Figure 1d. Once we impose the metric function to promote the feature discrimination, not only the inter-sample dependency will be considered, but also their neighbors will be jointly optimized to satisfy the imposed criteria.

Our main contributions can be characterized as follows: (1) We pinpoint the inefficiency of the conventional MLP of metric learning as a naïve graph convolution with only self-connections, essentially disregarding the neighborhood relations. (2) We introduce a novel metric learning framework that constructs a robust GCN embedding, effectively accounting for local manifold relations. (3) Our method achieves comparable results or significant performance gains over other SOTA pairwise metric learning techniques in testing with popular fine-grained image-retrieval benchmark datasets.

2 Related Work

2.1 Pairwise Metric Learning

The goal of metric learning is to establish a discriminative embedding that projects the original feature representations of the data to a new space with high intra-class compactness and inter-class separability. Here, we discuss some classical pairwise metric learning techniques. Chopra et al. (2005) introduce *contrastive loss* for learning an embedding with Siamese networks to promote the discrimination. The formulation forms several positive and negative pairs where the positive pairs are encouraged to be closer to each other, and the negative pairs to be farther away at least beyond a specified margin. Schroff et al. (2015) considers a *triplet loss* whose triplet relation indicates the anchor, positive, and negative samples, in embedding learning. It works by minimizing the distance between the anchor to the positive sample and simultaneously maximizing the distance between the anchor to the negative one. However, the design principle of the above-mentioned losses tend to cause inefficient learning due to excessively redundant pairs. This concern prompts recent studies to emphasize developing mining and weighting schemes to resolve this issue. The margin based loss in Manmatha et al. (2017) designs a distance weighted sampling scheme to seek informative samples and to impose a margin penalty to separate inter-class samples. Both *N-pair loss* (Sohn, 2016) and *lifted structure loss* (Song et al., 2016) consider mining harder negative samples to speed up convergence. On the other hand, *ranked List Loss* in Wang et al. (2019a) exploits all sample pairs to construct a comprehensive structure for metric learning, while *MS loss* by Wang et al. (2019b) instead extensively evaluates the type of similarity pairs and designs a principled approach to mining and weighting informative pairs. Departing from mining informative samples, Sun et al. (2020) propose *circle loss* that achieves self-paced weighting, via investigating the disparity between an optimal solution and the sample itself, to dynamically adjust the gradient of each sample, further resulting in flexible optimization. Some other studies (Duan et al., 2018; Mao et al., 2019) consider adversarial attacks to improve the feature discrimination further. The DAMLRRM by Xu et al. (2019) employs the minimum spanning tree (MST) to form an intra-class manifold (set) and then imposes a metric function over such pairs in the formed manifold to promote classwise compactness. Such a scheme is essentially similar to the just described approaches (Sohn, 2016; Song et al., 2016; Sun et al., 2020; Wang et al., 2019b;a; Manmatha et al., 2017), which model targeted pairs via specific strategies and then impose metric function to promote embedding discrimination. Finally, we remark that all these approaches implicitly assume a form of graph convolution with self-connections and thus result in inter-sample, or *sample-to-sample*, optimization.

2.2 Manifold Learning

Manifold learning typically aims to retain intrinsic neighboring structures in the underlying lower-dimensional feature space. The classical manifold learning techniques such as LLE (Roweis & Saul, 2000) and Isomap (Tenenbaum et al., 2000) estimate the local manifolds via justifiable assumptions. However, the effectiveness of these approaches mainly relies on the training data and lacks the capacity to handle unseen data. Chien & Chen (2016); van der Maaten (2009) consider the parametric framework based on a deep neural network to enhance the generalization ability of manifold learning. No matter learning-based or spectral-based optimization, the neighboring relations act as an important reference to realize the objective. Despite the implementations of metric learning and manifold learning are quite different, their common nature likewise focuses on learning an embedding to establish a discriminative mapping accounting for unseen data. Observing that MLP-based deep metric learning does not utilize the neighboring relations during the embedding construction, our proposed graph local embedding lifts this limitation to not only improve the effectiveness of neural network based metric learning but also ameliorate the issue of redundant training pairs.

2.3 MESSAGE PASSING FOR METRIC LEARNING

Graph convolutional networks (GCNs) have been extensively discussed in recent years and show impressive capability in handling non-Euclidean data distribution, *e.g.*, graph data (Kipf & Welling, 2017; Hamilton et al., 2017; Li et al., 2019; Chen et al., 2020; Xu et al., 2020). The main concept of GCN is to learn an embedding, which analyzes the neighborhood structure to yield a discriminative representation. In deep metric learning, *message passing* is popular for refining the feature representation based on the graph neighboring structures. By iteratively exchanging the information among nodes, it can generate an attention score to ensure that the model pays more attention to those similar nodes. Group Loss in Elezi et al. (2020) considers a class-prior matrix and takes the hand-crafted rules to iteratively refine the class information. Instead of the hard assignment, MPN (Seidenschwarz et al., 2021) exploits the Transformer design with multiple layers to learn the attention scores. Unlike learning the neighboring structure of the data points, ProxyGML (Zhu et al., 2020) adopts multiple *intra-class proxies* as neighboring data points. However, these designs are established on the closswise scenario, meaning that the constructed final representation is encouraged only to gather around the center of each class and far away from other inter-class ones. In other words, the constructed graph and message passing are coupled to impose *sample-to-manifold* relationships based on the resulting neighborhood structures.

We instead do not refine the attention scores or use the label information to explicitly specify the local relationships. Our approach follows the learned CNN mapping to define the local relationships and draws on the graph convolutions to better encode the features for metric learning. Whenever the neighboring structures of two inter-class samples are *similar* to each other, it would cause a large penalty by our proposed framework. To the best of our knowledge, we are the first to formulate a robust embedding for pairwise metric learning by exploring local neighboring relations via GCNs.

3 GRAPH LOCAL EMBEDDING

In pairwise metric learning, the redundant pairs, which are less informative and cause unexpected limitation in performance gain, often become the main obstacle for discriminative embedding learning. To tackle this challenging issue, we propose an embedding framework, termed as *graph local embedding* (GLE), to construct a robust embedding with the local *neighborhood* and *class* correlations of each data point for pairwise metric learning. Our approach can be simplified as a pipeline of three key steps, including local manifold construction, GCN embedding, and metric learning.

3.1 GCN VIEW OF LINEAR METRIC LEARNING

To learn a discriminative embedding, deep pairwise metric learning often considers the following procedure to encode input into high-level representations. Given an image batch $B = \{I_i\}_{i=1}^{n}$ in the RGB space and each image I_i is labeled with a class label y_i, where n is the batch size. To begin with, we assume that a convolutional neural network is applied to learn a non-linear transformation to encode B into the deep features $X \in \mathbb{R}^{n\times d}$. Then, a linear MLP layer for metric learning is exploited to learn a mapping into an embedding space $F_{linear} \in \mathbb{R}^{n\times m}$. Finally, CNN and the linear layer will be jointly optimized via a specific metric loss function to realize the discriminative embedding. Formally, the underlying embedding via linear mapping can be expressed by

$$F_{linear} = XW \tag{1}$$

where $W \in \mathbb{R}^{d\times m}$ is a linear mapping, transforming deep features X into F_{linear}.

It is pivotal to point out that the above deep metric learning in (1) can be regarded as a form of graph convolution with self-connections. That is, each data point is associated only with itself and does not link to others—that is, it can be reformulated as deep features X, multiplied by an identity matrix which represents an $n \times n$ graph with only self-connections. Thus we can rewrite (1) into

$$F_{linear} = I_n XW \tag{2}$$

where $I_n \in \mathbb{R}^{n\times n}$ indicates an identity matrix.

With (2), it can be readily observed that the linear learning adopted by most previous studies assumes self-connections I_n and overlooks the neighboring relations, which could further suffer from

an ineffective learning procedure due to information mining from redundant pairs. To resolve the issues, we propose to model the local manifold of each data point and develop a robust GCN embedding for pairwise deep metric learning. Naturally, the drawbacks of redundant pairs are greatly eased owing to paying attention to the neighboring relations. Moreover, effectively integrating the local neighboring relations via non-linear graph convolutions can properly realize the inter local neighborhood optimization, leading to a more general formulation of deep metric learning.

3.2 LOCAL MANIFOLD CONSTRUCTION

To form a local manifold of each data point in the feature space, we consider using an affinity graph to encode neighbor relations. Given a batch B of size n, we construct an affinity graph $\mathcal{G} \in \mathbb{R}^{n\times n}$ where each node represents an encoded data point $\mathbf{x}_i$ from B, and the edge between each pair of nodes is initialized via their cosine similarity. That is, an edge affinity g_{ij} in $\mathcal{G}$ can be formulated as:

$$g_{ij} =< \frac{\mathbf{x}_i}{\|\mathbf{x}_i\|}, \frac{\mathbf{x}_j}{\|\mathbf{x}_j\|} > \tag{3}$$

where i and j respectively indicate the indexes of the underlying samples in B.

Notice that although the class label information is available in the supervised metric learning, the construction of $\mathcal{G}$ does not use it to specify the neighbors of each node. The main reasoning is that not only the intra-class neighbors are useful but also the inter-class ones can provide important information for metric learning. For example, in the optimization process of metric learning, it would be useful to yield a larger penalty when observing local manifolds, now not restricting to samples of the same class, of an arbitrary pair of inter-class samples are significantly *overlapped*.

We show below several popular strategies for local manifold construction and justify their validity in our experimental results. The intuitive idea is to adopt either the k-nearest neighbors (k-nn) or ϵ-ball to define the local neighbors of each sample. The former works by keeping the top-k entries from the given ranking list, while the latter preserves those elements whose similarities are higher than the specified ϵ. The k-nn and ϵ-ball neighborhoods can be respectively expressed as

$$g_{ij} \underset{k\text{-nn}}{=} \begin{cases} g_{ij}, & \text{if } \mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i, k), \\ 0, & \text{otherwise}, \end{cases} \quad \text{and} \quad g_{ij} \underset{\epsilon\text{-ball}}{=} \begin{cases} g_{ij}, & \text{if } g_{ij} > \epsilon, \\ 0, & \text{otherwise}, \end{cases} \tag{4}$$

where $\mathcal{N}(\mathbf{x}_i, k)$ is the set of top-k samples of $\mathbf{x}_i$ according to the ranking list of $\{g_{ij}\}_{j=1}^n$, and ϵ denotes the similarity threshold of the ϵ-ball.

Yet, another strategy for constructing local manifolds from $\mathcal{G}$ is to apply random sampling over a complete graph to form the neighbors of each sample. We here consider two kinds of sampling schemes, *purely random* and *Bernoulli*, to preserve the edges from the given graph $\mathcal{G}$. In the former sampling scheme, we employ uniform sampling to randomly select the neighbors. For the latter, the probability of a neighbor will be selected depending on its similarity to a given anchor i.

3.3 GCN EMBEDDING

After building the local manifold $\mathcal{G}$, we introduce a general GCN (Kipf & Welling, 2017) to construct an embedding for metric learning. Comparing with the typical graph convolution operation, the obtained local manifold $\mathcal{G}$ is sparse and exhibits essential local neighboring relations. It can consequently result in a robust embedding more relevant to the subsequent metric learning. By replacing the self-connections I_n in (2), we establish the proposed GCN local embedding F by

$$F = \sigma(\hat{A}XW) \tag{5}$$

$$\hat{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}} \tag{6}$$

where σ represents an activation function for non-linear mapping. $\tilde{A} = \mathcal{G} + I_n$ is an adjacency matrix by the propagation rule in Kipf & Welling (2017) to prevent exploding/vanishing gradients and $\tilde{D}$ is a normalized degree matrix with its ith diagonal entry $\tilde{D}_{ii} = \sum_{j=1}^{N} \tilde{A}_{ij}$.

3.4 METRIC LEARNING

To learn a discriminative embedding, one can consider using any popular loss function for metric learning. In our formulation, we consider using a simple objective function, which aims to reduce

the $s_n - s_p$, where s_n and s_p denote the similarity of a negative pair (different-class) and of a positive pair (same-class), respectively (Sun et al., 2020). Unlike the linear MLP metric learning, GLE explores the local relationships and draws on the graph convolution to construct a discriminative mapping for deep metric learning. It enhances metric learning by exploring the inter-neighborhood (manifold-to-manifold) relationships. As GLE enriches the representation of each sample with local and informative neighboring relations, either of the similarity s_n for a negative pair or the similarity s_p for a positive pair will not simply entail the pure inter-sample similarity. In particular, the neighboring associations of the anchor, its respective positives and negatives will all turn out to be important references. In the experiments, we find as expected that the burden of redundant pairs will be greatly eased owing to exploring the complex variants of neighboring relations. Importantly, when the model is trained to meet the requirement of the resulting metric, not only each negative sample (with respect to a given anchor) must be far away from the underlying anchor but also its local neighborhood has to respect the same criterion simultaneously. In addition, the anchor will receive the neighboring connections via its positive samples and enforces these neighbors to stay nearby, and simultaneously be far away from the implied negatives.

4 Experimental Results

4.1 Datasets

We assess our method, *i.e.*, GLE, on three fine-grained retrieval benchmark datasets, including CUB (Wah et al., 2011), Cars-196 (Krause et al., 2013), and Stanford Online Products (SOP) (Song et al., 2016). Briefly, the benchmark datasets CUB, Cars-196, SOP respectively contain $11,788$ images of 200 bird categories, $16,185$ images of 196 vehicle categories, and $120,053$ images of $22,634$ online product categories. Following Sun et al. (2020), we use half of the classes for training and the remaining classes for testing by measuring the image retrieval performance with Recall@k (R@k).

4.2 Implementation Details

Our backbone network employs ResNet-50 (He et al., 2016) pre-trained on ImageNet (Deng et al., 2009) to extract deep features. We train one-layer GCN and an embedding layer of 512-dimension representation for neighborhood relationships modeling and metric learning, respectively. Our data augmentation follows the same settings as Kim et al. (2020) for both training and testing phases, and the input images are of size 227×227 as (Seidenschwarz et al., 2021; Sun et al., 2020; Xuan et al., 2018). GLE is trained with AdamW optimizer (Loshchilov & Hutter, 2019) for 60 epochs, and each batch is constructed with a PK batch sampler of P classes and K instances per class ($K = 5$) as Hermans et al. (2017). While testing, we augment the query image with random flip and crop to construct the neighboring nodes of the query's local manifold. In addition, we also provide the result, marked as GLE*, concerning only the self-connections during local manifold construction.

4.3 Ablations

In the ablation analysis, we discuss how the batch size, local neighborhood (training/testing), and pairwise metric learning loss affect the model performance.

4.3.1 Batch Size

Table 1 shows the effect of different batch sizes constructed by the PK batch sampler, where the instances per class are fixed and $K = 5$. The results show that our model is not sensitive to the batch size, and it has the best or the second-best performances by adopting the batch size of 120.

4.3.2 Local Neighborhood in Training

Table 2 shows the different local neighborhood construction methods during training. Notice that all models in Table 2 adopt the same local neighborhood construction in testing, *i.e.*, the self-connections. This experiment employs the proposed GCN embedding step and the MS loss (Wang et al., 2019b) in the metric learning step. To validate the effectiveness of local neighborhood in train-

Table 1: Effect of batch size on Recall@k (R@k).

Batch Size	CUB				Cars-196			
	R@1	R@2	R@4	R@8	R@1	R@2	R@4	R@8
80	69.1	79.7	86.9	91.9	88.6	**93.5**	95.9	**97.7**
120	**70.1**	**80.0**	**87.2**	**92.6**	**88.7**	93.3	**96.1**	97.6
160	69.3	78.3	86.3	92.1	88.3	93.0	95.9	**97.7**
200	69.3	79.2	86.4	91.8	87.1	92.5	95.4	97.5
240	69.4	79.4	86.9	92.0	87.8	92.6	95.4	97.4

Table 2: Effect of local manifold construction in training while using self-connections in testing.

Local Manifold (training)	CUB		Cars-196		SOP	
	R@1	R@2	R@1	R@2	R@1	R@10
Self-connections	68.5	78.5	85.2	91.6	80.7	92.2
Full-connections	66.9	77.9	79.6	87.5	75.2	88.6
Purely random	66.4	77.5	80.7	88.7	77.0	89.7
Bernoulli	66.8	77.8	81.4	88.7	78.2	90.6
k-nn	66.9	77.3	80.1	87.9	79.2	91.0
ϵ-ball	69.4	79.4	87.2	92.4	81.0	92.6
k-nn + ϵ-ball	**70.1**	**80.0**	**88.7**	**93.3**	**81.2**	**93.1**

Table 3: Effect of local manifold construction in test-ing while using k-nn + ϵ-ball in training.

Local Manifold (testing)	CUB				Cars-196			
	R@1	R@2	R@4	R@8	R@1	R@2	R@4	R@8
Self-connections (GLE*)	70.1	80.0	87.2	92.6	88.7	93.3	96.1	97.6
k-random	71.5	81.3	88.1	92.5	91.0	94.7	97.0	98.3
$\bar{k}$-nn	67.5	78.0	86.1	91.7	89.9	94.3	96.7	98.1
k-nn	72.2	82.4	88.9	**93.2**	91.3	95.0	97.2	98.5
ϵ-ball (GLE)	**72.9**	**82.5**	**89.0**	**93.2**	**91.6**	**95.1**	97.2	98.4
ϵ-ball + k-random	72.0	81.6	88.3	92.8	90.3	96.9	**98.1**	**98.8**
ϵ-ball + $\bar{k}$-nn	69.3	79.8	87.4	92.3	89.0	93.7	96.0	97.6
ϵ-ball + k-nn	72.1	82.3	88.9	93.1	91.1	**95.1**	97.3	98.5

ing, our baseline model employs the self-connections, *i.e.*, the traditional learning manner using pure linear mapping (2). We discuss the methods for local manifold construction as follows:

Full-connections. This method denotes the anchor point links to all other samples within a batch to learn the embedding jointly and can be regarded as a *complete graph* in our GLE. Despite the normalization diagonal degree matrix D can emphasize the importance of each entry, the large number of negative pairs leads to an imbalance learning and hence results in the worst performance.

Purely random & Bernoulli. Section 3.2 elaborates two random sampling methods on neighbors within the given graph to construct the local manifold in our GLE. The selected neighbors using *purely random* sampling cannot capture the local manifold structure and hence fail to encode the helpful graph representations for learning the local neighborhood. The *Bernoulli* sampling shows slightly better performance than purely random sampling, yet its effectiveness highly relied on the quality of the deep feature space concerning the local relationships.

k-nn & ϵ-ball. The k-nn method selects the top-k similar nodes for the GLE by ranking the edges among the anchor to other samples. The performance is not good enough since the k-nn method may include irrelevant edges in combination with enlarging the graph's sparsity. The ϵ-ball can exclude irrelevant edges significantly owing to its ϵ threshold, where a larger ϵ involves highly similar connected neighbors, and we set ϵ to be 0.7 in our experiments. We observed that the combination of k-nn and ϵ-ball by simultaneously concerning the sparsity of the graph and high similarity among anchor and edges, we can obtain the best performance, as shown in the last row of Table 2. Hence, our GLE employs such a configuration in the following experiments.

4.3.3 Local Neighborhood in Testing

While testing a GLE-based image retrieval task, we augment the query image with random flip and crop to construct the query's local manifold since the other images besides the query should not be available for use. Table 3 shows the different local manifold construction methods during testing. Precisely, we augment each query image to ten images for constructing the query's local manifold, and we respectively select the most similar k images (k-nn), the most dissimilar k images ($\bar{k}$-nn), and random k images (k-random) for discussing the effect of the local manifold construction. Notice that we use the local neighborhood construction method by 'k-nn + ϵ-ball' and optimized by MS loss for all configurations in Table 3. As the results in Table 2, the baseline using the self-connections in testing also merely shows moderate performance. Though the query's local manifold is constructed

Table 4: Effect of pairwise metric learning losses and GCN embeddings. Table 3 indicates the configurations on local manifold construction of GLE and GLE*.

Pairwise Metric Learning Loss	GCN Embedding	CUB				Cars-196				SOP			
		R@1	R@2	R@4	R@8	R@1	R@2	R@4	R@8	R@1	R@10	R@10^2	R@10^3
Contrastive	Linear	61.4	72.4	81.7	88.5	81.5	88.5	93.1	95.6	75.2	88.6	95.1	98.2
	GLE*	63.7	74.0	83.1	89.9	82.8	89.1	92.9	95.8	77.8	90.0	96.0	98.7
	GLE	**68.1**	**78.2**	**85.9**	**91.1**	**85.2**	**91.3**	**95.1**	**97.2**	**79.2**	**91.2**	**96.4**	**98.9**
Triplet	Linear	55.5	66.6	77.3	84.9	63.1	73.9	82.2	88.6	73.6	88.0	**95.3**	**98.7**
	GLE*	56.1	67.1	77.8	85.2	64.0	74.5	83.2	89.5	75.9	**88.4**	95.2	98.6
	GLE	**57.4**	**69.2**	**78.4**	**86.8**	**65.0**	**75.3**	**83.6**	**90.0**	**77.1**	**88.4**	95.2	98.6
MS	Linear	68.5	78.5	87.0	92.5	85.2	91.6	94.7	96.8	80.7	92.2	96.9	98.9
	GLE*	70.1	80.0	87.2	92.6	88.7	93.3	96.1	97.6	81.2	**93.1**	97.3	99.1
	GLE	**72.9**	**82.5**	**89.0**	**93.2**	**91.6**	**95.1**	**97.2**	**98.4**	**82.3**	**93.1**	**97.5**	**99.2**
Circle	Linear	68.4	78.6	86.5	92.0	83.6	90.4	94.8	97.1	80.4	91.7	96.7	98.9
	GLE*	69.6	80.5	88.1	93.0	85.2	91.2	95.2	97.5	80.9	92.0	97.0	**99.0**
	GLE	**73.5**	**82.6**	**89.2**	**93.7**	**89.3**	**94.1**	**96.4**	**98.1**	**81.6**	**92.7**	**97.4**	**99.0**

from its augmentation, the results in Table 2 show that the ϵ-ball ($\epsilon = 0.7$) is still preferred, yet the construction method 'ϵ-ball + k-nn' does not bring more performance gain as in the training phase.

4.3.4 Pairwise Metric Learning Loss

Our GLE can leverage any popular metric learning loss function to learn a discriminative embedding. This experiment discusses the effect of pairwise metric learning losses by equipping with Contrastive loss (Chopra et al., 2005), Triplet loss (Schroff et al., 2015), MS loss (Wang et al., 2019b), and Circle loss (Sun et al., 2020). Table 4 shows that the proposed graph local embedding constructing the local manifold in both training and testing (GLE) achieves the best performance compared to the traditional learning manner (Linear) without utilizing any local manifold. Even though the GLE*, which only constructs the local manifold in training, also shares the benefits from the GCN embedding within the introduced graph local embedding. While equipping with the MS loss, the proposed model shows the best performance besides the CUB dataset. The performance gain of *GLE*, derived from that GLE can raise the ranking of those positive samples, in Table 4 supports our strategy that exploring the manifold-to-manifold relationships can enhance metric learning.

4.4 Comparison with the State-of-the-art methods

Table 5 compares the proposed GLE against other state-of-the-art metric learning methods on CUB, Cars-196, and SOP benchmark datasets. The results show the superior performance of GLE using the standard metric of Recall@k on all datasets.

The first group in Table 5 denotes the conventional linear metric learning methods. For example, the method DAMLRRM (Xu et al., 2019) employs the minimum spanning tree for modeling metric learning. Then it forms an intra-class manifold and then imposes a metric function over pairs in the formed manifold to promote classwise compactness. Its embedding is formulated by one single MLP layer, similar to self-connections, and can be regarded as a sample-to-sample relationships. Such a group of methods learning sample-to-sample relationships may suffer the burden of redundant pairs. In contrast, our GLE explores the manifold-to-manifold relationships and eases the redundant pairs.

In Table 5, the second group denotes the ensemble metric learning methods. For example, the method XBM (Wang et al., 2020) enriches informative pairs via an additional cross-batch memory module to obtain performance gain compared with the conventional linear metric learning methods. However, such a group of ensemble methods suffers the additional cost of a memory dictionary to store high-level representations for the entire dataset. In contrast, our GLE surpasses these ensemble metric learning methods without the extra cost of any additional storage.

The third group in Table 5 denotes the message passing based metric learning methods. This group of methods such as MPN (Seidenschwarz et al., 2021), ProxyGML (Zhu et al., 2020), and Group Loss (Elezi et al., 2020) concerning sample-to-manifold relationships. While learning the sample-

Table 5: Comparison of GLE against the state-of-the-art metric learning methods. From top to bottom, the first group denotes the conventional linear method, the second group denotes the ensemble methods, and the third group denotes the message passing based method. The backbone column shows the backbone network and embedding size. B: BN-Inception, G: GoogLeNet, R: ResNet-50.

Methods	Backbone	CUB				Cars-196				SOP			
		R@1	R@2	R@4	R@8	R@1	R@2	R@4	R@8	R@1	R@10	R@10^2	R@10^3
DAMLRRM (Xu et al., 2019)	G^{512}	55.1	66.5	76.8	85.3	73.5	82.6	89.1	93.5	69.7	85.2	93.2	–
HTL (Ge et al., 2018)	G^{512}	57.1	68.8	78.7	86.5	81.4	88.0	92.7	95.7	74.8	88.3	94.8	98.4
RLL-H (Wang et al., 2019a)	B^{512}	57.4	69.7	79.2	86.9	74.0	83.6	90.1	94.1	76.1	89.1	95.4	–
HDC (Song et al., 2017)	G^{384}	60.7	72.4	81.9	89.2	83.8	89.8	93.6	96.2	75.9	88.4	94.9	98.1
SoftTriple Qian et al. (2019)	B^{512}	65.4	76.4	84.5	90.4	84.5	90.7	94.5	96.9	78.3	90.3	95.9	–
MS (Wang et al., 2019b)	B^{512}	65.7	77.0	86.6	91.2	84.1	90.4	94.0	96.5	78.2	90.5	96.0	98.7
Circle (Sun et al., 2020)	B^{512}	66.7	77.4	86.2	91.2	83.4	89.8	94.1	96.5	78.3	90.5	96.1	98.6
ABIER (Opitz et al., 2017)	G^{512}	57.5	68.7	78.3	86.2	82.0	89.0	93.2	96.1	74.2	86.9	94.0	97.8
ABE (Kim et al., 2018)	G^{512}	60.6	71.5	79.8	87.4	85.2	90.5	94.0	96.1	76.3	88.4	94.8	98.2
RLL-(L,M,H) (Wang et al., 2019a)	B^{1536}	61.3	72.7	82.7	89.4	82.1	89.3	93.7	96.7	79.8	91.3	96.3	–
DREML (Xuan et al., 2018)	R^{128}	63.9	75.0	83.1	89.7	86.0	91.7	95.0	97.2	–	–	–	–
XBM (Wang et al., 2020)	B^{512}	65.8	75.9	84.0	89.9	82.0	88.7	93.1	96.1	79.5	90.8	96.1	98.7
D&C (Sanakoyeu et al., 2019)	R^{128}	65.9	76.6	84.4	90.6	84.6	90.7	94.1	96.5	75.9	88.4	94.9	98.1
ProxyGML (Zhu et al., 2020)	B^{512}	66.6	77.6	86.4	–	85.5	91.8	95.3	–	78.0	90.6	96.2	–
Group Loss (Elezi et al., 2020)	B^{512}	66.9	77.1	85.4	91.5	88.0	92.5	95.7	97.5	76.3	88.3	94.6	–
MPN (w/o Auxiliary CE Loss)	R^{512}	68.1	–	–	–	87.2	–	–	–	–	–	–	–
MPN (w/ Auxiliary CE Loss)	R^{512}	70.3	80.3	87.6	92.7	88.1	93.3	96.2	98.2	81.4	91.3	95.9	–
MPN (Seidenschwarz et al., 2021)	R^{512}	70.8	–	–	–	88.6	–	–	–	–	–	–	–
GLE*	R^{512}	70.1	80.0	87.2	92.6	88.7	93.3	96.1	97.6	81.2	**93.1**	97.3	99.1
GLE	R^{512}	**72.9**	**82.5**	**89.0**	**93.2**	**91.6**	**95.1**	**97.2**	**98.4**	**82.3**	**93.1**	**97.5**	**99.2**

to-manifold relationships, this group of methods may suffer the class-driven limitation mentioned in the introduction. Namely, they tend to select the sample leaning toward its class center and simultaneously being far away from other class centers. Our GLE instead considers manifold-to-manifold relationships to formulate a robust embedding and surpasses these methods over datasets CUB, Cars-196, and SOP.

5 Conclusion

While training with random sampling, the redundant pairs inevitably happening and are harmful to embedding learning. To address this issue, previous studies make an extensive discussion on how to mining the informative samples. This paper identifies the issue of *missing* neighborhood relationships in conventional CNN-based embedding by pointing out that the linear MLP layer is a form of graph convolution with self-connections, which implies that the pairwise metric learning techniques only discuss the *sample-to-sample* relationships during the targeted embedding and thus suffer redundant pairs. We propose a novel deep metric learning called Graph Local Embedding (GLE) to instead model the local manifold for constructing a robust embedding for deep metric learning. By encoding the local neighborhood relationship into the GCN feature of each sample, our GLE can significantly enrich the context of each encoded data and thus build a better learning pattern while enforcing the metric function. Through the experiments on the public benchmarks, we demonstrate the effectiveness of our GLE. In particular, our GLE can perform great results and surpass state-of-the-art approaches with a clear margin, even if comparing with ensemble approaches.

References

Nicolas Aziere and Sinisa Todorovic. Ensemble deep manifold similarity learning using hard proxies. In *CVPR*, pp. 7299–7307, 2019.

Jintai Chen, Biwen Lei, Qingyu Song, Haochao Ying, Danny Z. Chen, and Jian Wu. A hierarchical graph network for 3d object detection on point clouds. In *CVPR*, pp. 389–398, 2020.

Jen-Tzung Chien and Ching-Huai Chen. Deep discriminative manifold learning. In *ICASSP*, pp. 2672–2676, 2016.

Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, pp. 539–546, 2005.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pp. 248–255, 2009.

Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, pp. 4690–4699, 2019.

Yueqi Duan, Wenzhao Zheng, Xudong Lin, Jiwen Lu, and Jie Zhou. Deep adversarial metric learning. In *CVPR*, pp. 2780–2789, 2018.

Ismail Elezi, Sebastiano Vascon, Alessandro Torcinovich, Marcello Pelillo, and Laura Leal-Taixé. The group loss for deep metric learning. In *ECCV*, volume 12352 of *Lecture Notes in Computer Science*, pp. 277–294, 2020.

Weifeng Ge, Weilin Huang, Dengke Dong, and Matthew R. Scott. Deep metric learning with hierarchical triplet loss. In *ECCV*, volume 11210 of *Lecture Notes in Computer Science*, pp. 272–288, 2018.

William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NeurIPS*, pp. 1024–1034, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person reidentification. *CoRR*, abs/1703.07737, 2017.

Sungyeon Kim, Dongwon Kim, Minsu Cho, and Suha Kwak. Proxy anchor loss for deep metric learning. In *CVPR*, pp. 3235–3244, 2020.

Wonsik Kim, Bhavya Goyal, Kunal Chawla, Jungmin Lee, and Keunjoo Kwon. Attention-based ensemble for deep metric learning. In *ECCV*, volume 11205 of *Lecture Notes in Computer Science*, pp. 760–777, 2018.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCVW*, pp. 554–561, 2013.

Guohao Li, Matthias Müller, Ali K. Thabet, and Bernard Ghanem. Deepgcns: Can gcns go as deep as cnns? In *ICCV*, pp. 9266–9275, 2019.

Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *CVPR*, pp. 6738–6746, 2017.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

R. Manmatha, Chao-Yuan Wu, Alexander J. Smola, and Philipp Krähenbühl. Sampling matters in deep embedding learning. In *ICCV*, pp. 2859–2867, 2017.

Chengzhi Mao, Ziyuan Zhong, Junfeng Yang, Carl Vondrick, and Baishakhi Ray. Metric learning for adversarial robustness. In *NeurIPS*, pp. 478–489, 2019.

Yair Movshovitz-Attias, Alexander Toshev, Thomas K. Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. In *ICCV*, pp. 360–368, 2017.

Michael Opitz, Georg Waltner, Horst Possegger, and Horst Bischof. BIER - boosting independent embeddings robustly. In *ICCV*, pp. 5199–5208, 2017.

Qi Qian, Lei Shang, Baigui Sun, Juhua Hu, Tacoma Tacoma, Hao Li, and Rong Jin. Softtriple loss: Deep metric learning without triplet sampling. In *ICCV*, pp. 6449–6457, 2019.

Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

Artsiom Sanakoyeu, Vadim Tschernezki, Uta Büchler, and Björn Ommer. Divide and conquer the embedding space for metric learning. In *CVPR*, pp. 471–480, 2019.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, pp. 815–823, 2015.

Jenny Denise Seidenschwarz, Ismail Elezi, and Laura Leal-Taixé. Learning intra-batch connections for deep metric learning. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9410–9421, 2021.

Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett (eds.), *NeurIPS*, pp. 1849–1857, 2016.

Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *CVPR*, pp. 4004–4012, 2016.

Hyun Oh Song, Stefanie Jegelka, Vivek Rathod, and Kevin Murphy. Deep metric learning via facility location. In *CVPR*, pp. 2206–2214, 2017.

Yifan Sun, Changmao Cheng, Yuhan Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei. Circle loss: A unified perspective of pair similarity optimization. In *CVPR*, pp. 6397–6406, 2020.

Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

Laurens van der Maaten. Learning a parametric embedding by preserving local structure. In David A. Van Dyk and Max Welling (eds.), *AISTATS*, volume 5 of *JMLR Proceedings*, pp. 384–391, 2009.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu. Additive margin softmax for face verification. *IEEE Signal Process. Lett.*, 25(7):926–930, 2018.

Xinshao Wang, Yang Hua, Elyor Kodirov, Guosheng Hu, Romain Garnier, and Neil Martin Robertson. Ranked list loss for deep metric learning. In *CVPR*, pp. 5207–5216. Computer Vision Foundation / IEEE, 2019a.

Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R. Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *CVPR*, pp. 5022–5030, 2019b.

Xun Wang, Haozhi Zhang, Weilin Huang, and Matthew R. Scott. Cross-batch memory for embedding learning. In *CVPR*, pp. 6387–6396, 2020.

Qiangeng Xu, Xudong Sun, Cho-Ying Wu, Panqu Wang, and Ulrich Neumann. Grid-gcn for fast and scalable point cloud learning. In *CVPR*, pp. 5660–5669, 2020.

Xinyi Xu, Yanhua Yang, Cheng Deng, and Feng Zheng. Deep asymmetric metric learning via rich relationship mining. In *CVPR*, pp. 4076–4085, 2019.

Hong Xuan, Richard Souvenir, and Robert Pless. Deep randomized ensembles for metric learning. In *ECCV*, volume 11220 of *Lecture Notes in Computer Science*, pp. 751–762, 2018.

Wenzhao Zheng, Zhaodong Chen, Jiwen Lu, and Jie Zhou. Hardness-aware deep metric learning. In *CVPR*, pp. 72–81, 2019.

Yuehua Zhu, Muli Yang, Cheng Deng, and Wei Liu. Fewer is more: A deep graph metric learning perspective using fewer proxies. In *NeurIPS*, 2020.