Received November 27, 2021, accepted January 10, 2022, date of publication January 14, 2022, date of current version January 20, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3143129

Probabilistic Principal Geodesic Deep Metric Learning

DAE HA KIM, (Associate Member, IEEE), AND BYUNG CHEOL SONG, (Senior Member, IEEE)
Department of Electrical and Computer Engineering, Inha University, Incheon 22212, South Korea

Corresponding author: Byung Cheol Song (bcsong@inha.ac.kr)

This work was supported in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) Grant by the Korea Government through MSIT (Ministry of Science and ICT) (Artificial Intelligence Convergence Research Center, Inha University) under Grant 2020-0-01389; in part by IITP Grant by the Korea Government through MSIT (AI Innovation Hub) under Grant 2021-0-02068; and in part by MSIT, South Korea, under the Information Technology Research Center (ITRC) Support Program, supervised by IITP, under Grant IITP-2021-0-02052.

ABSTRACT Similarity learning which is useful for the purpose of comparing various characteristics of images in the computer vision field has been often applied for deep metric learning (DML). Also, a lot of combinations of pairwise similarity metrics such as Euclidean distance and cosine similarity have been studied actively. However, such a local similarity-based approach can be rather a bottleneck for a retrieval task in which global characteristics of images must be considered important. Therefore, this paper proposes a new similarity metric structure that considers the local similarity as well as the global characteristic on the representation space, i.e., class variability. Also, based on an insight that better class variability analysis can be accomplished on the Stiefel (or Riemannian) manifold, manifold geometry is employed to generate class variability information. Finally, we show that the proposed method designed through in-depth analysis of generalization bound of DML outperforms conventional DML methods theoretically and experimentally.

INDEX TERMS Deep metric learning, image retrieval, Stiefel manifold, non-linear mapping.

I. INTRODUCTION

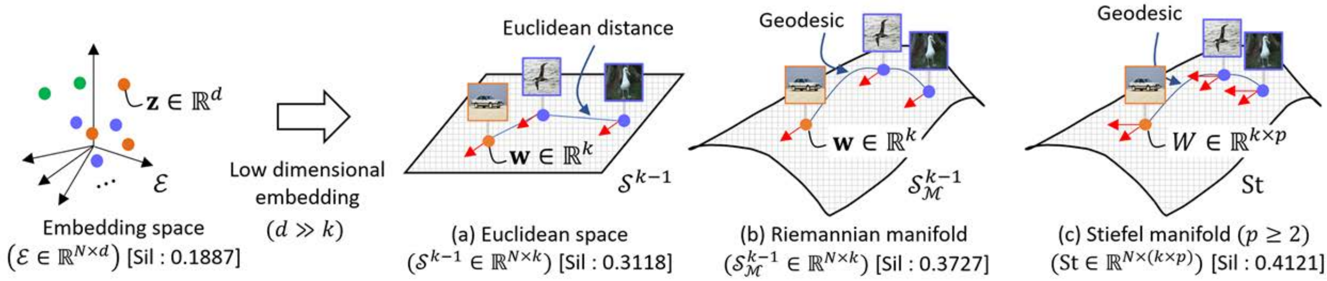
Deep metric learning (DML) is a learning method that can increase intra-class compactness and inter-class variability by quantifying intrinsic or extrinsic relationship between images. Since the existing DML methods [14], [37] are based on a similarity metric that can successfully encode the images of various attributes, they have been widely applied to product searching [32], face verification [37], perturbation analysis [30], etc.

The similarity metric is mainly defined using the pairwise sample-based Euclidean distance or cosine similarity [23], [37]. Among popular triplet-based methods [37], some used the mining [15] for complexity reduction and the others [39] adopted multiple samples with margin. Even episode-based learning scheme with meta data [65] was grafted into [37]. Recently, the similarity metric [23], [43] applying cosine similarity (or dot product) to the softplus function has received a lot of attention owing to high performance. Also, some methods [34], [41] tried to maximize global features on the embedding space by using a histogram or beta function. To further improve the performance, [31], [44], [48] suggested the ensemble of the above-mentioned similarity metrics at the feature-level or network-level.

On the other hand, class variability, one of the core properties of DML, can be analyzed through discriminant analysis (DA) [2]. Basically, the goal of DA is to find a projection component that maximizes the ratio of intra-class compactness and inter-class variability, and was mainly used to analyze dimension reduction and discriminative eigenpair. Eigenpair indicates a tuple of eigenvalue and eigenvector. The performance of DA was guaranteed even on manifolds [29] and the association of DA with pairwise sample-based DML was already verified [1].

Note that most DML studies only considered local similarity based on pairwise connection. Since DML must be able to improve the retrieval performance, global features such as semantic representation need to be reflected in the similarity metric [13], [66]. In addition, since existing DML methods defined the embedding space produced by convolutional neural networks (CNNs) as a vector space, they had a limit in reflecting nonlinear characteristics such as multi-variate covariance (see Sec. II.B for detailed motivation).

The associate editor coordinating the review of this manuscript and approving it for publication was Genoveffa Tortora.


$\mathbf{z} \in \mathbb{R}^d$
$\mathcal{E}$
Low dimensional embedding
$(d \gg k)$
Euclidean distance
$\mathbf{w} \in \mathbb{R}^k$
$\mathcal{S}^{k-1}$
Geodesic
$\mathbf{w} \in \mathbb{R}^k$
$\mathcal{S}_{\mathcal{M}}^{k-1}$
Geodesic
$W \in \mathbb{R}^{k \times p}$
St

Embedding space ($\mathcal{E} \in \mathbb{R}^{N\times d}$) [Sil : 0.1887]

(a) Euclidean space ($\mathcal{S}^{k-1} \in \mathbb{R}^{N\times k}$) [Sil : 0.3118]

(b) Riemannian manifold ($\mathcal{S}_{\mathcal{M}}^{k-1} \in \mathbb{R}^{N\times k}$) [Sil : 0.3727]

(c) Stiefel manifold ($p \geq 2$) (St $\in \mathbb{R}^{N\times(k\times p)}$) [Sil : 0.4121]

FIGURE 1. Some methods of representing data of the embedding space in a low-dimensional representation space. Unlike the Euclidean space assumed by existing DML techniques, the manifold can analyze the intrinsic characteristics of data through geodesic and matrix form. For example, in the manifold, the similarity can be calculated by considering the class attribute of 'bird' rather than features such as angles and colors of the image (see Section IV.A for details). The red arrow indicates the direction of latent vector, i.e., the representation component of data, and it enables compact representation on Stiefel manifold. Here, [Sil] indicates the silhouette score. Triplet-semi with ResNet50 [16] and the CUB200-2011 dataset were used for this experiment.

In order to overcome the drawbacks of the conventional DMLs, we come up with two ideas as follows: 1) Based on a pairwise similarity metric reflecting local characteristics, we reflect even a global characteristic in the representation space, i.e., class variability. 2) We also consider nonlinear characteristic by analyzing the class variability in the low-dimensional space (LDS) derived from the embedding space.

Class variability is observed through the global characteristics of samples corresponding to discrete labels [45]. This global property is space dependent. This fact provides justification for analyzing class variability from the manifold and global characteristics of the space.

As in Fig. 1, we compared the silhouette scores [64] of a few representation spaces to choose an appropriate representation space. The silhouette score is a metric that shows the degree of partitioning of clustered data, and can quantify the learning tendency of retrieval tasks. Silhouette scores were highest in the order of Stiefel, Riemannian, and Euclidean space. This result shows that compact representation in a matrix form enables discriminative class variability analysis. In this experiment, Euclidean space and manifold employed PCA and local linear embedding for low dimensional embedding [3], respectively. On the other hand, [17] reported that matrix representation of embedding space is more useful for learning the intrinsic characteristics of images than vector representation. This report supports analysis of Fig. 1.

Therefore, this paper presents a new metric loss for DML. First, we derive class variability *factor* by using manifold sampling [55] and eigenpair obtained through DA in a nonlinear manifold such as Stiefel. Next, we map this *factor* into a one-dimensional line manifold to associate with a 1D similarity metric like Euclidean distance. Then, we define the projected *factor* as the geodesic factor (GF) (see Sec. III.A). Finally, we present several geodesic metric losses (GMLs) based on locality-perspective metric and GF (see Sec. III.B). Note that GF can be interpreted from the following two points of view.

- GF for analyzing class variability on LDS has the same goal as subspace clustering [54]. In other words, we can regard that GF explicitly grafts subspace clustering paradigm to the retrieval task.
- GF plays a role of *self-supervision* that allows the embedding layer to further reflect class variability in learning. Note that GF is not used in testing, and only the embedding layer that can understand the class variability of data distribution is used in testing.

This paper is organized as follows. Sec. II describes our motivation from a technical point of view, and then previews preliminary knowledge for the proposed method. Sec. III depicts the design procedure of GF. Sec. IV evaluates the performance of the proposed method qualitatively and quantitatively.

II. PRELIMINARIES

A. NOTATION

Let $\mathbf{z}_i = f(\mathbf{x}_i) \in \mathcal{Z}$ be a d-dimensional (embedding) vector of an image $\mathbf{x}_i \in \mathcal{X}$ corresponding class labels $y_i \in \mathcal{Y}$, where f is CNN-based embedding network [16], [18]. Also, let a set of embedding vectors and the number of classes be $Z = \{\mathbf{z}_i\}_{i=1}^N$ and C, where N is the minibatch size. Linear DA (LDA) is used to find the d-dimensional optimal eigenvectors $\{\mathbf{e}_i\}_{i=1}^m$ that maximize the ratio of intra-class compactness and inter-class variability. LDA can be reinterpreted by finding the optimal eigenmatrix $E \in \mathbb{R}^{d\times m}$: $E^* = \max_E \frac{|E^T \Sigma_b E|}{|E^T \Sigma_w E|}$, where $E^* = (\mathbf{e}_1|\mathbf{e}_2|\ldots|\mathbf{e}_m)$, $\Sigma_b = \Sigma_t - \Sigma_w$ (inter-class covariance), Σ_t (total covariance), and Σ_w (intra-class covariance) [2]. The eigenpair of E^*, i.e., $\{(\lambda_i, \mathbf{e}_i)\}_{i=1}^m$ can be found using the eigenvalue equation. Here, $\lambda_1 \geq \ldots \geq \lambda_m$. And the eigenvalue solver plays a technical role to perform forward/backward pass operation (e.g., DA and gradient calculation) during end-to-end learning of LDA (SDA in Fig. 2).

B. MOTIVATION

As a representative similarity metric, Mahalanobis distance between $\mathbf{z}_i$ and $\mathbf{z}_j$ is defined by

$$D_{\Sigma}^2(\mathbf{z}_i, \mathbf{z}_j) = (\mathbf{z}_i - \mathbf{z}_j)^T \Sigma^{-1} (\mathbf{z}_i - \mathbf{z}_j)$$
$$\cong \sum_{k=1}^{d} (\mathbf{z}_i - \mathbf{z}_j)^T \frac{\mathbf{e}_k \mathbf{e}_k^T}{\lambda_k} (\mathbf{z}_i - \mathbf{z}_j)\,(1) \quad (1)$$

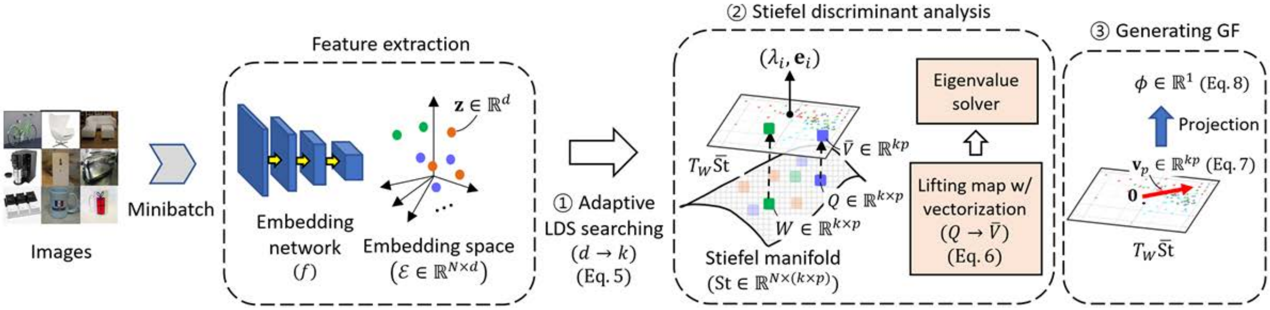

Images
Minibatch
Feature extraction
Embedding network
(f)
$\mathbf{z} \in \mathbb{R}^d$
Embedding space
$(\mathcal{E} \in \mathbb{R}^{N \times d})$
① Adaptive
LDS searching
$(d \to k)$
(Eq. 5)
② Stiefel discriminant analysis
$(\lambda_i, \mathbf{e}_i)$
$T_W\bar{\mathrm{St}}$
$\bar{V} \in \mathbb{R}^{kp}$
$Q \in \mathbb{R}^{k \times p}$
$W \in \mathbb{R}^{k \times p}$
Stiefel manifold
$(\mathrm{St} \in \mathbb{R}^{N \times (k \times p)})$
Eigenvalue solver
Lifting map w/ vectorization
$(Q \to \bar{V})$
(Eq. 6)
③ Generating GF
$\phi \in \mathbb{R}^1$ (Eq. 8)
Projection
$\mathbf{v}_p \in \mathbb{R}^{kp}$ (Eq. 7)
$\mathbf{0}$
$T_W\bar{\mathrm{St}}$

FIGURE 2. **Generation process of geodesic factor (GF) ϕ. The class variability factor $\mathbf{v}_p$ is obtained by uniform sampling on the tangent space $T_W\bar{St}$. W and Q indicate two points on Stiefel manifold St, respectively.**

If covariance matrix Σ is I, Eq. (1) becomes the Euclidean distance $D\left(\mathbf{z}_i, \mathbf{z}_j\right)\left(=D_{ij}\right)$ and Σ^{-1} can be approximated by a scale term $\frac{\mathbf{e}_k\mathbf{e}_k^{\mathbf{T}}}{\lambda_k}$ based on eigenpair $(\mathbf{e}, \lambda)$ [4]. Early metric learning techniques [9], [45] assumed $\mathbf{z}$ as data and they focused on projecting data into a discriminative LDS through a covariance matrix. On the other hand, the latest metric learning techniques [23], [37] could handle the retrieval task only with the local linearity of the embedding vector $\mathbf{z}$ under the assumption of $\Sigma^{-1} = I$, thanks to CNN's feature analysis capability. However, the previous approaches that seldom took into account the global information of feature (or data) and nonlinearity in designing metric showed limitations in improving performance in the fine-grained retrieval task. Reference [36] designed the scale term of Eq. (1) using an orthogonal layer and a scale layer on the Stiefel manifold. Since the two layers were only designed for scale-invariance purpose, eigenpairs could not be dealt with explicitly. This indicates that the global characteristics such as class variability could not be reflected properly. Therefore, we intend to design a GF that can reflect the spectrum of eigenpairs, i.e., global information on a nonlinear space.

C. GEOMETRY OF MATRIX MANIFOLD

1) RIEMANNIAN MANIFOLD

A Riemannian manifold $(\mathcal{M}, g)$ is a smooth manifold $\mathcal{M}$ equipped with a Riemannian metric g.g is defined as the inner product on the tangent space $T_{\mathbf{w}}\mathcal{M}$ for each point $\mathbf{w}\,(\in \mathcal{M})$, $g\,(\cdot, \cdot) : T_{\mathbf{w}}\mathcal{M} \times T_{\mathbf{w}}\mathcal{M} \to \mathbb{R}$.

2) GEODESIC

A geodesic is a locally shortest curve on $\mathcal{M}$. The geodesic $\gamma\,(t) : t \in [0, 1]$ on $\mathcal{M}$, s.t. $\gamma\,(0) = \mathbf{w}$, $\dot{\gamma}\,(0) = \mathbf{v}\,(\in T_{\mathbf{w}}\mathcal{M})$ can be represented by $\mathrm{Exp}_{\mathbf{w}}\,(t\mathbf{v})$ [5].

3) GEOMETRY OF STIEFEL MANIFOLD

Stiefel manifold is composed of a set of $k \times p$ orthonormal matrices as in Eq. (2):

$$\mathrm{St}\,(k, p) \triangleq \left\{W \in \mathbb{R}^{k \times p} | W^T W = I_p\right\}, \tag{2}$$

where I_p denotes a $p \times p$ identity matrix and $p \leq k$. Note that Stiefel manifold of Eq. (2) can be considered as a unitary group or a hypersphere. For example, if $p = 1$, $\mathrm{St}\,(k, p)$ can be regarded as a unit hypersphere, that is, sphere manifold $\mathcal{S}_{\mathcal{M}}^{k-1}$. Also, when $p = k$, $\mathrm{St}\,(k, p)$ is regarded as a unitary group $\mathcal{U}_k$. Then, at a point on Stiefel manifold, i.e., $W \in \mathrm{St}\,(k, p)$, the tangent space is defined by

$$T_W\mathrm{St}\,(k, p) = \left\{V \in \mathbb{R}^{k \times p} | W^T V + V^T W = 0\right\} \tag{3}$$

Also, Lie group $\mathrm{SO}\,(k)$, which operates through group action like matrix multiplication on Stiefel manifold, is defined by

$$\mathrm{SO}\,(k) \triangleq \left\{R \in \mathbb{R}^{k \times k} | R^T R = I_k, \det(R) = 1\right\} \tag{4}$$

Then, the group action between $R \in \mathrm{SO}\,(k)$ and $W \in \mathrm{St}\,(k, p)$ is expressed as $RW \in \mathrm{St}\,(k, p)$. As a result, the concept of this group action is used to design a retraction map or lifting map on Stiefel manifold (② in Fig. 2) [7].

Usually, (length) normalization [46] is applied to embedding vectors to meet Stiefel constraint, and orthogonal initialization is applied to the other trainable parameters.

III. METHOD

A. GEODESIC FACTOR (GF) GENERATION

The process of generating GF consisting of three steps begins just after feature extraction process as in Fig. 2. Assume the Stiefel manifold is LDS.

1) ADAPTIVE LDS SEARCHING

The first step is to use a specific dimension reduction tool [3] that maps embedding space to a low-dimensional representation space (① in Fig. 2). This data-adaptive dimension reduction tools in [3] are simple, but they are powerful projection methods that allow generalization even for out-of-distribution (OOD) samples. And then, pairwise inner product (PIP) criterion [52] is used to find the optimal dimension in the dimensionality reduction process.

$$\left\|ZZ^T - \hat{Z}\hat{Z}^T\right\| = d - k + 2\left\|\hat{Z}^T Z^{\perp}\right\|^2, \tag{5}$$

where $Z \in \mathbb{R}^{N \times d}$ and $\hat{Z} \in \mathbb{R}^{N \times k}$. Note that the criterion of Eq. (5) consists of a bias component $d - k$ and a variance component $2\left\|\hat{Z}^T Z^{\perp}\right\|^2$ [52]. The optimal trade-off between bias and variance terms becomes the sweet spot we find.

Specifically, k that minimizes Eq. (5) becomes the dimension of LDS. Based on k selected by Eq. (5), LDS $\mathrm{St} \in \mathbb{R}^{N\times(k\times p)}$ is generated. Note that when $p = 1$, LDS is regarded as a Riemannian manifold, and from when $p \geq 2$, LDS is considered to be Stiefel manifold. Refer to Sec. IV for an experiment to analyze variation of p-values.

2) STIEFEL DISCRIMINANT ANALYSIS (SDA)

The second step is to obtain the eigenpair required to quantify the class variability on St through SDA. SDA consists of a nonlinear mapping onto tangent space and an eigenvalue solver.

First, a point Q on St is mapped to V on the tangent space $T_W\mathrm{St}$ by nonlinear mapping. Here, W and Q mean the intrinsic mean [11] of St and an arbitrary point, respectively. Since a well-known logarithmic map [11] may not be defined as a closed form, lifting map RP_W^{-1}, which is the inverse process of retraction map RP_W, is used for this step [22] (② in Fig. 2).

$$V = RP_W^{-1}(Q) \equiv QS_p - W, \tag{6}$$

where S_p is the symmetric positive semidefinite $p \times p$ matrix such that $QS_p - W \in T_W\mathrm{St}$. Next, RP_W^{-1} of Eq. (6) is computed as follows [22]: 1) Compute $M = W^TQ$, 2) solve Continuous-time Algebraic Riccati Equation (CARE) [22] $(-M)S_p + S_p\left(-M^T\right) + 2I_p = 0$ for S_p, and 3) compute $QS_p - W$. Before obtaining the eigenpair on $T_W\mathrm{St}$, we transform the matrix space into the vector space according to the existing approaches [51] for transforming the data shape on manifold. Specifically, based on Kronecker product-based vectorization operation $T_W\mathrm{St}(k,p) \otimes T_W\mathrm{St}(k,p) \rightarrow T_W\bar{St}(kp,1)$, St is transformed into product space $\bar{\mathrm{St}}$ [51]. So, an eigenpair that can reflect data consistency is obtained. Finally, based on $\bar{V}$ on $T_W\bar{St}$, eigenpair $(\mathbf{e}, \lambda)$ is produced by LDA.

3) GENERATING GF

The third step is to quantify class variability in the representation space. First, the quantification process is based on an eigenpair and a probability model-based sampling [55]. In detail, as in Eq. (7), class variability factor $\mathbf{v}_p$ is sampled on $T_W\bar{\mathrm{St}}$ by using the average eigenvalue λ_{avg} reflecting separability per class label as the range of the probability model [55] (③ in Fig. 2).

$$\mathbf{v}_p \sim \mathrm{U}_{T_W\bar{\mathrm{St}}}\left(RP_W C_m \pm \epsilon_1 \lambda_{avg}\right) \in \mathbb{R}^{kp}, \tag{7}$$

where the probability model assumes a uniform distribution U, ϵ_1 is a scale factor, and the center value of U is computed by $RP_W C_m$ that is the retraction map of C_m and kp-dimensional zero vector $\mathbf{0}$, i.e., $C_m + 0$ [22]. $C_m = \frac{1}{|C_{all}|}\sum_{(\mathbf{z}_a,\mathbf{z}_p,\mathbf{z}_n)\in C_{all}}\left(D_{ap} + D_{an}\right)$, where $\mathbf{z}$'s subscripts a, p, and n indicate anchor, positive, and negative points, respectively and C_{all} stands for a set of all possible pairs.

Then, GF ϕ is determined by placing $\mathbf{v}_p$ on the line manifold that reflects the separability characteristics as much as possible through the discriminative eigenvector $\mathbf{e}_1 \in \mathbb{R}^{kp}$:

$$\phi = \mathbf{e}_1^T \mathbf{v}_p \tag{8}$$

Since ϕ is the global information that quantifies class variability in the representation space and it lies on the line manifold, it can be fused with the scalar similarity metric reflecting local characteristics. Also, because ϕ is defined based on (inter-)class variability, it gradually becomes larger as latent vectors corresponding to different class supervision are separated, that is, similarity learning progresses. It can alleviate the inherent overfitting problem of similarity learning by playing a momentum in the gradient operation (see Sec. III.C). In the next section, the one-dimensional GF ϕ generated through Eq. (8) is used for the purpose of reflecting the class variability factor in the Euclidean distance or cosine similarity.

B. DEFINING GEODESIC METRIC LOSS (GML)

This section presents three versions of GML based on representative or latest DMLs [23], [31], [37].

GML-Tri is based on matrix multiplication of squared Euclidean distance D^2 and ϕ as in Eq. (9).

$$\begin{aligned}\mathcal{L}_{Tri}\left(\mathbf{z}_a, \mathbf{z}_p, \mathbf{z}_n\right) = {} & \frac{1}{N_p}\sum\nolimits_{i=1}^{N_p}\left[\left(\mathbf{D}_{ap}^2 1_N^T \Phi\right)_i - \beta_1\right]_+ \\ & + \frac{1}{N_n}\sum\nolimits_{i=1}^{N_n}\left[\beta_1 - \left(\mathbf{D}_{an}^2 1_N^T \Phi\right)_i\right]_+ \\ & - \epsilon_2 \sum_{i=1}^{kp} \lambda_i \end{aligned} \tag{9}$$

where $\mathbf{D}^2 = \left[D_i^2\right]^T$, $\Phi = [\phi_i]^T$, $i = 1\ldots, N$. To reflect all elements of ϕ in $\mathbf{D}^2$, an N-dimensional all-one vector 1_N is used. Similar to [14], [37], the first and second terms of Eq. (9) are defined by hinge function $[\cdot]_+$ and class specific margin β_1 [46], and they are trained so that $\mathbf{z}_a$ and $\mathbf{z}_p$ are located close to each other, and $\mathbf{z}_a$ and $\mathbf{z}_n$ are located far from each other. The third term encourages tangent space $T_W\bar{\mathrm{St}}$ to find the discriminative eigenpair $(\mathbf{e}, \lambda)$. On the other hand, in order to reduce the computational burden of Eq. (9), $\left(\mathbf{D}_{ap}^2 1_N^T \Phi\right)_i \in \mathbb{R}^N$ is reconfigured as follows:

$$\left(\mathbf{D}_{ap}^2 1_N^T \Phi\right)_i = \mathbf{D}_{ap_i}^2\left(\phi_1 + \cdots + \phi_N\right) = \mathbf{D}_{ap_i}^2 \phi_s.$$

Similarly, $\mathbf{D}_{an_i}^2\phi_s$ is reconfigured. Among N samples, the numbers of pairs satisfying $D_{ap}^2\phi_s > \beta_1$ and $D_{an}^2\phi_s < \beta_1$ are N_p and N_n, respectively.

GML-PA employs ϕ_s like GML-Tri, and utilizes Proxy-Anchor (PA) [23] as the baseline.

$$\begin{aligned}&\mathcal{L}_{PA}(\mathbf{z}, \mathbf{p}) \\ &= \frac{1}{|P^+|}\sum_{\mathbf{p}\in P^+}\log\left(1 + \phi_s \sum_{\mathbf{z}\in C_p} e^{-\alpha(CS(\mathbf{z},\mathbf{p})-\beta_2)}\right) \\ &\quad + \frac{1}{|P|}\sum_{\mathbf{p}\in P}\log\left(1 + \phi_s \sum_{\mathbf{z}\in C_n} e^{\alpha(CS(\mathbf{z},\mathbf{p})+\beta_2)}\right) - \epsilon_2 \sum_{i=1}^{kp}\lambda_i \end{aligned} \tag{10}$$

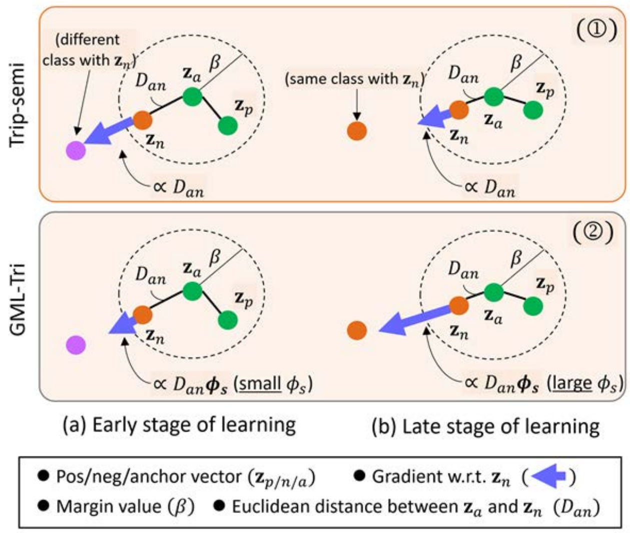

Trip-semi
GML-Tri
(①)
(②)
(different class with $\mathbf{z}_n$)
(same class with $\mathbf{z}_n$)
D_{an}
$\mathbf{z}_a$
β
$\mathbf{z}_p$
$\mathbf{z}_n$
$\propto D_{an}$
$\propto D_{an}\boldsymbol{\phi}_s$ (small ϕ_s)
$\propto D_{an}\boldsymbol{\phi}_s$ (large ϕ_s)
(a) Early stage of learning
(b) Late stage of learning
● Pos/neg/anchor vector ($\mathbf{z}_{p/n/a}$)
● Gradient w.r.t. $\mathbf{z}_n$
● Margin value (β)
● Euclidean distance between $\mathbf{z}_a$ and $\mathbf{z}_n$ (D_{an})

FIGURE 3. Conceptual view of the proposed method. Here, the color of each vector represents class label information.

where CS indicates cosine similarity, and C_p and C_n denote the sets of vectors in positive and negative relationships with a proxy vector $\mathbf{p}$, respectively. P and P^+ denote the set of all and positive proxies, respectively [23].

The exponentially exploding or asymptotically vanishing phenomenon that occurs frequently in [23] is controlled by ϕ_s. Thus, ϕ_s can give stability to similarity learning.

GML-DiVA is designed using DiVA [31] as the baseline, which ensembles several similarity metrics at the network level. The loss function of GML-DiVA is as follows:

$$\mathcal{L}_{DiVA} = \mathcal{L}_{disc} + \mathcal{L}_{sha} + \mathcal{L}_{int} + \mathcal{L}_{reg} \tag{11}$$

where $\mathcal{L}_{disc}$ is the margin loss [46] based on the class label triplet set $\{y_a, y_p, y_n\}$ with $y_a = y_p$ and $y_a \neq y_n$. $\mathcal{L}_{sha}$ and $\mathcal{L}_{int}$ are losses when the configuration conditions of the triplet set are $y_a \neq y_p \neq y_n$ and $y_a = y_p = y_n$, respectively. $\mathcal{L}_{reg}$ is the regularization loss proposed in [31]. Similar to GML-Tri in Eq. (9), $\mathcal{L}_{disc}$ is defined by multiplying the Euclidean distance of margin loss by ϕ_s. $\mathcal{L}_{sha}$ and $\mathcal{L}_{int}$ are also defined like $\mathcal{L}_{disc}$.

The goal of GML-DiVA is to create synergy with loss functions defined through ϕ_s for understanding global semantic similarity. Please refer to **Appendix I** for GML-Cont and GML-MS handled in the experiment section.

1) COMPLEXITY ANALYSIS

$\mathcal{O}\left(Np^3\right)$ and $\mathcal{O}\left(Nkp^2\right)$ are required for CARE equation and non-linear mapping to $T_{\mathrm{W}}\bar{\mathrm{S}}\mathrm{t}$, respectively. Here, $N \gg k, p$. In addition, $\mathcal{O}\left(N^2d\right)$ is required for eigenvalue solver, which is less than $\mathcal{O}\left(N^3C\right)$ of Triplet [10]. On the other hand, compared to $\mathcal{O}\left(NC\right)$ of PA [23], the N-squared complexity of eigenvalue solver is somewhat burdensome. Table 1 shows that when considering the performance improvement by the proposed method, the additional time required for learning is sufficiently tolerated. In the test phase, the proposed method uses only the embedding layer like other techniques, so it consumes the equivalent testing time.

TABLE 1. Elapsed training time per epoch on CUB200. Since all methods use only the embedding layer, testing times are the same as 10.5 seconds (CPU: Intel XEON E5-1650, GPU: GTX 1080TI).

Methods	Training time (second)
PA / Trip-semi / DiVA	12 / 41 / 134
GML – PA / Trip / DiVA	17 (+5) / 47 (+6) / 149 (+15)

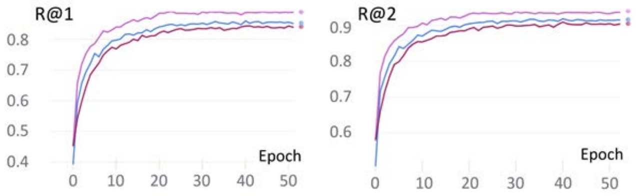


FIGURE 4. Performance analysis of R@1 and R@2 on Cars196. Here, magenta, blue, and brown represent GML-PA, Proxy-Anchor, and MS, respectively.

C. GRADIENT ANALYSIS OF GML

Eq. (9) is differentiated to analyze the effect of ϕ_s on the behavior of the embedding vector.

$$\frac{\partial \mathcal{L}_{tri}}{\partial \mathbf{z}_p} = \frac{\partial}{\partial \mathbf{z}_p}\left(D_{ap}^2\phi_s - \beta_1\right) = \frac{\partial D_{ap}^2}{\partial \mathbf{z}_p}\phi_s + D_{ap}^2\frac{\partial \phi_s}{\partial \mathbf{z}_p} \tag{12}$$

$$\cong 2\phi_s\left(\mathbf{z}_p - \mathbf{z}_a\right) = 2D_{ap}\phi_s\left(\frac{\mathbf{z}_p - \mathbf{z}_a}{D_{ap}}\right) \tag{12}$$

$$\frac{\partial \mathcal{L}_{tri}}{\partial \mathbf{z}_n} = \frac{\partial}{\partial \mathbf{z}_n}\left(\beta_1 - D_{an}^2\phi_s\right) = -\frac{\partial D_{an}^2}{\partial \mathbf{z}_n}\phi_s - D_{an}^2\frac{\partial \phi_s}{\partial \mathbf{z}_n}$$

$$\cong 2\phi_s\left(\mathbf{z}_a - \mathbf{z}_n\right) = 2D_{an}\phi_s\left(\frac{\mathbf{z}_a - \mathbf{z}_n}{D_{an}}\right) \tag{13}$$

Here, only the case where the hinge function is greater than 0 is handled, and the gradient term of ϕ_s can be omitted because it is relatively smaller than the gradient term of D^2. In addition, the gradient of D_{ap}^2 can be decomposed into a size component $2D_{ap}\phi_s$ and a direction component $\left(\frac{\mathbf{z}_p - \mathbf{z}_a}{D_{ap}}\right)$, and the size component can be tuned by ϕ_s unlike the triplet.

Fig. 3 conceptually describes the movements of vectors according to metric losses such as Trip-semi and GML-Tri, through gradient analysis. In case of Trip-semi, since the gradient is only proportional to the size component D_{an}, the negative vector $\mathbf{z}_n$ can be easily belong to different class label region at the beginning of learning. As such, it will be difficult for $\mathbf{z}_n$, which was initially mis-learned, to be included in the region of the original class label. On the contrary, $\mathbf{z}_n$ which belongs to $\mathbf{z}_a$'s class in the latter part of learning is hard to escape from the region of this class (① of Fig. 3). On the other hand, the gradient of GML-Tri, i.e., GML based on triplet loss is affected by D_{an} and ϕ_s that is the sum of all components of GF. As a result, ϕ_s, which quantifies class variability, plays a role of clipping the gradient of D_{an} at the beginning of learning, and acting as a momentum at the end of learning. (② of Fig. 3).

Fig. 4 shows the positive impact of GF in the early stages of learning. GML-PA showed a steep increase in recall rate

TABLE 2. Performance comparison on CUB200-2011, CARS196, and SOP. Superscript 128 and 512 represent the embedding vector size, and superscript 512B represents the results of train/test images of 256 × 256 size with 512 embedding vector size. Here, 'PA', 'TRI', and 'CONT' represent proxy-anchor, Trip-Semi, and contrastive, respectively. * indicates our own implementation results.

Datasets		CUB200-2011				Cars196				SOP		
Recall@Q (%)	Backbone	R1	R2	R4	R8	R1	R2	R4	R8	R1	R10	R100
Contrastive^{128} [14]	R50	55.2	68.2	77.9	85.0	64.2	74.7	82.2	88.4	65.5	82.4	92.5
GML-Cont^{128}		61.2	71.3	80.6	87.6	78.4	86.3	91.2	94.8	71.3	84.7	93.2
Trip-semi^{128} [37]		57.5	68.8	78.3	85.4	65.5	76.9	85.2	90.4	68.7	92.6	91.5
DSML-Tri^{128*}[53]		63.1	74.4	83.4	90.4	80.9	87.5	92.6	95.9	76.4	88.3	95.1
PADS-Tri^{128} [35]		64.0	75.5	84.3	-	79.9	87.5	92.3	-	74.8	88.2	95.0
GML-Tri^{128}		**64.3**	**75.6**	**84.6**	**90.7**	**82.0**	**88.8**	**92.9**	**95.9**	**77.2**	**89.7**	**95.9**
SoftTriplet^{512} [33]	BN	65.4	76.4	84.5	90.4	84.5	90.7	94.5	96.9	78.3	90.3	95.9
MS^{512} [43]		65.7	77.0	86.3	91.2	84.1	90.4	94.0	96.5	78.2	90.5	96.0
GML-MS^{512}		67.2	78.4	86.6	91.5	85.0	91.3	94.9	97.1	79.4	90.6	96.1
JDR^{512} [8]		67.9	78.7	86.2	91.3	84.7	90.7	94.4	97.2	79.2	90.5	96.0
Proxy-Anchor^{512} [23]		68.4	79.2	86.8	91.6	86.1	91.7	95.0	97.3	79.1	90.8	96.2
GML-PA^{512}		**70.7**	**80.2**	**87.5**	**92.4**	**87.7**	**92.8**	**95.8**	**97.7**	**80.2**	**91.4**	**96.4**
HORDE^{512B} [9]	BN	66.3	76.7	84.7	90.6	83.9	90.3	94.1	96.3	80.1	91.3	96.2
GML-MS^{512B}		70.2	80.7	88.1	92.4	86.3	91.3	94.9	97.0	80.0	91.0	96.3
Proxy-Anchor^{512B}		71.1	80.4	87.4	92.5	88.3	93.1	95.7	97.5	80.3	91.4	96.4
GML-PA^{512B}		**72.2**	**81.6**	**88.3**	**93.3**	**89.3**	**94.1**	**96.5**	**98.1**	**81.0**	**91.8**	**96.5**

in the early stage of learning (0-10 epochs) than PA and MS. This momentum is maintained even when the k of recall@k increases, and it affects the peak performance.

See **Appendix I** for *gradient analysis* of anchor $\mathbf{z}_a$, Stiefel manifold, and derivation of GML-PA.

D. GENERALIZATION BOUNDS OF GML

This section analyzes the generalization bound of the similarity metric structure based on the representative covering number [25]. Let $\mathcal{A}$ be a parameterized model that is optimized by C_{all} and the similarity metric ρ of $\mathcal{Z}$. In addition, let the expected triplet set generated by a probability distribution P be $\left(\mathbf{z}, \mathbf{z}', \mathbf{z}''\right) \sim P$. Then, the generalization bound of $\mathcal{A}$ in the metric space $(\mathcal{Z}, \rho)$ is defined as follows.

Theorem 1: Assume that $\rho\left(\mathbf{z}_a, \mathbf{z}\right) \leq \gamma$, $\rho\left(\mathbf{z}_p, \mathbf{z}'\right) \leq \gamma$, $\rho\left(\mathbf{z}_n, \mathbf{z}''\right) \leq \gamma$ for $\forall \mathbf{z}_a, \mathbf{z}_p, \mathbf{z}_n, \mathbf{z}, \mathbf{z}', \mathbf{z}''$ and $\gamma > 0$. If $\mathcal{A}$ satisfies

$$\left|\mathcal{L}\left(\mathcal{A}_{C_{all}}; \mathbf{z}_a, \mathbf{z}_p, \mathbf{z}_n\right) - \mathcal{L}\left(\mathcal{A}_{C_{all}}; \mathbf{z}, \mathbf{z}', \mathbf{z}''\right)\right| \leq \eta\left(C_{all}\right) \tag{14}$$

and $\mathcal{N}(\gamma, \mathcal{Z}, \rho) < \infty$, then $\mathcal{A}$ is $(|\mathcal{Y}| \mathcal{N}(\gamma, \mathcal{Z}, \rho),\ \eta\left(C_{all}\right))$-robust. Here, $\mathcal{L}\left(\mathcal{A}_{C_{all}}; \cdot\right)$ is the loss function of $\mathcal{A}$ optimized with C_{all} as input, $\mathcal{N}(\cdot)$ is $\mathcal{Z}$'s covering number [25], and $\eta(\cdot) : (\mathcal{Z} \times \mathcal{Z})^N \rightarrow \mathbb{R}$ indicates a real-valued function.

Proof: For the proof of **Theorem 1** and the details of $\eta\left(C_{all}\right)$ and the covering number, see **Appendix II**.

On the other hand, based on **Theorem 1**, the bounds of different similarity metric structures can be compared as follows.

Theorem 2: Let the generalization bound of the similarity metric structure [23] based on cosine similarity be η_L. η_L assumes a softplus function type. Also, as in [37], let the hinge function with margin that is based on Euclidean distance be η_E. Then, the following relationship is established between η_L and η_E.

$$\eta_L \leq \eta_E \leq \eta\left(C_{all}\right) \tag{15}$$

Proof: Refer to **Appendix II** for the proof of **Theorem 2**.

η_E and η_L correspond to the bounds of GML-Tri and GML-PA, respectively. Eq. (15) provides theoretical indicators regarding optimization stability and convergence between different similarity metric structures. In addition, the structure with tight generalization bounds improves retrieval performance (see Table 2 of Sec. IV.A). From this point of view, **Theorem 2** can be an explicit tool for analyzing the relationship between the retrieval performance and the generalization bound.

IV. EXPERIMENTS

A. CONFIGURATION DETAILS

We used PyTorch library for network design and parameter optimization. All experiments were performed five times on 4 NVIDIA GeForce GTX 1080 TI GPUs. We adopted the training protocol of the base techniques [23], [31], [37] to which GML is applied. Two backbone networks of ResNet50 (R50) [16] and BN-Inception (BN) [18] were used. For preprocessing, a center crop of 224 × 224 (or 256 × 256) and horizontal flip were applied to input images. As in the other techniques, only the embedding layer trained by GML was used for evaluation, and the GF generation process was excluded. Please refer to **Appendix III** for specific details.

B. HYPER-PARAMETER SETTING

For PA as the base technique, AdamW [28] with learning rate of 10^{-4} and $N = 180$ were used, and for the others,

TABLE 3. Performance comparison with ensemble methods on CUB200-2011, CARS196, and SOP. 'G' indicates GoogleNet V2. The experiment setup is the same as in table 2.

Datasets		CUB200-2011			Cars196			SOP		
Recall@Q / NMI (%)	Backbone	R1	R2	NMI	R1	R2	NMI	R1	R10	NMI
DREML^{9216} [48]	R18, BN	63.9	75.0	67.8	86.0	91.7	76.4	-	-	-
RLL^{1536} [44]	G	61.3	72.7	66.	82.1	89.3	71.8	79.8	91.3	90.4
DiVA^{512} [31]	R50	69.2	79.3	71.4	87.6	92.9	72.2	79.6	91.2	90.6
GML-DiVA^{512}	R50	**70.0**	**79.9**	**72.1**	**88.0**	**93.2**	**74.3**	**81.3**	**92.1**	**92.1**

TABLE 4. Performance comparison according to the different manifolds on CUB200-2011. Other experiment settings are the same as table 2.

Methods	R1	R2	NMI
GML-PA (Riem. $\mathcal{S}_{M}^{k-1}$, $p = 1$) [63]	69.8	79.4	71.8
GML-PA (Unitary $\mathcal{U}_k$, $p = k$)	69.4	79.3	71.6
GML-PA (Stiefel w/ Log. map [11])	70.4	**80.4**	72.1
GML-PA (Stiefel)	**70.7**	80.2	**72.3**

Adam [21] with learning rate of 10^{-5} and $N = 112$ were used. ϵ_1 of Eq. (7) and ϵ_2 of Eq. (9) were set to 1e-1 and 1e-6, respectively. Also, α and β_2 of Eq. (10) were set to 48 and 1e-1. Isomap [3] was used for LDS searching, and the default set of LDS dimension k was $\{3, 4, 5\}$, and p was set to 2.

C. DATASETS

Four natural and artificial image datasets were adopted for training and testing. Cars196 [26] contains 16,185 images of 196 car classes. The first 98 classes (8,054 images) for training and the remaining 98 classes (8,131 images) for testing. CUB200-2011 [42] consists of 11,788 images of 200 bird classes. The first 100 classes (5,864 images) for training and the remaining 100 classes (5,924 images) for testing. Stanford online products (SOP) [32] consists of 120,053 images of 22,634 product classes. 11,318 classes (59,551 images) for training and 11,316 classes (60,502 images) for testing. In-shop clothes [27] consists of 72,712 images of 7,986 classes: 3,997 classes for training and 3,985 classes for testing. The test set is divided into query set (14,218 images) and gallery set (12,612 images). See Fig. 5 for more details.

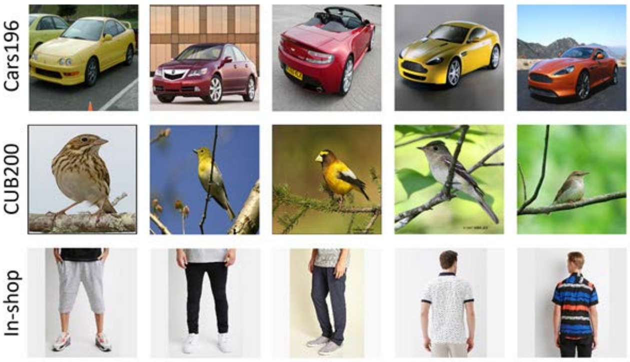


FIGURE 5. An example of datasets for DML.

D. EXPERIMENTAL RESULTS

1) QUANTITATIVE RESULTS

Quantitative evaluation was performed on the CUB200-2011, Cars196, and SOP datasets. Recall@Q (R@Q) [20] was employed for retrieval performance evaluation, and normalized mutual information (NMI) [38] was adopted for clustering performance evaluation. Table 2 listed Euclidean distance-based and cosine similarity-based techniques [23], [33], [39], [43] in the order of embedding vector size, and then showed their performance.

The greatest performance improvement was observed in GML-Cont and GML-Tri. For example, in CUB200-2011, GML improved R@1 by 6% and 6.8%, respectively, in comparison to the base technique (see the 2nd and 6th rows in Table 2). Note that among the triplet-based DML techniques, GML-Tri was the best. GML-Tri showed R@1 gap of only 0.3% from PADS-Tri [35] for CUB200-2011, but it provided significant R@1 differences of 2.1% and 2.4% for Cars196 and SOP datasets, which can guarantee generalization performance. Also, it showed a steady performance gain of more than 1% over DSML [53]. This proves that a loss function inspired by subspace-clustering is more effective in retrieval tasks than SNR-based class variability analysis [53] and reinforcement learning-based sampling [35].

GML-PA achieved the highest performance improvement among cosine similarity-based techniques. Seeing the 12th row, R@1 was 70.7% in CUB200-2011 and 87.5% in Cars196. In case that the input image size is set to 256 × 256, GML boosted the performance of PA [23] and HORDE [19] by about 1% (see the last row). This is SOTA performance in this embedding vector size.

Next, Table 3 compares GML-DiVA with conventional ensemble methods [31], [44], [48]. GML-DiVA showed better performance in terms of both recall and NMI than [44] and [48] with embedding vector size of 512 or more.

Although GML-DiVA showed a meaningful R@1 of 1.7% for the SOP dataset, its overall performance improvement was not significant, compared to GML-Tri and GML-PA of Table 2. Actually, DiVA is somewhat different from the purpose of GML designed for improving a single metric because DiVA boosts performance through fusion of metrics. Based on a study [50] showing that the ensemble performance depends on the performance of single metrics, we can interpret that the metric used in GML-DiVA has a positive effect on the classifier fusion process.

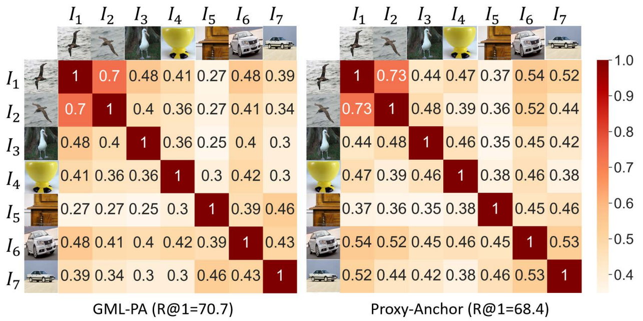


FIGURE 6. An examples of confusion matrix based on the distance correlation (DC). A larger DC value means a similar image. Both GML-PA and PA were trained with CUB200-2011 and then evaluated with test split of CUB200-2011, Cars196, and SOP.

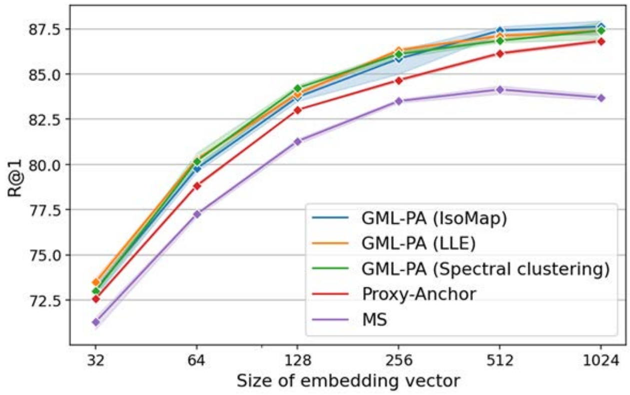


FIGURE 7. R@1 accuracy versus embedding vector size on Cars196.

E. QUALITATIVE RESULTS

In this section, a qualitative evaluation in terms of similarity was performed using a confusion matrix based on the distance correlation (DC) index. Note that well-known Euclidean distance is difficult to effectively reflect the similarity of latent vectors as well as *disentanglement characteristics*. So, we analyzed the relationship between vectors based on DC. As in Fig. 6, for the same class label images I_1 and I_2, both GML-PA and PA showed high DC values of about 0.7. However, the DC value between a 'bird' image I_1 and a 'cup' image I_4 was 0.41 for GML-PA and 0.47 for PA. This supports the outstanding discrimination performance of GML. In the case of PA, although the class label is different, the DC (0.44) between I_3 and I_1, which have a common attribute of 'bird', is smaller than the DC (0.45) between I_3 and I_6, which have completely different attributes. On the other hand, GML-PA shows an ordinary DC distribution. This experiment result shows that GML-PA can learn similarity metric by reflecting not only class label but also image attribute. See **Appendix IV** for more examples of Fig. 6 and latent space visualization.

F. ABLATION STUDY

In this section, we performed comparative analysis on the hyper-parameters of GF. Fig. 7 evaluates the techniques used for LDS searching of Sec. III.A according to the embedding vector size. The performance was highest in the order of GML-PA, Proxy-Anchor, and MS. As the embedding vector size increased from 32 to 256, R@1 also improved in proportion to the capacity of the embedding layer. However, from 512 and above, the performance of all techniques was saturated around 87%. This indicates that the embedding layer no longer provides sensible information in calculating the similarity metric. Also, the performance variation according to LDS search methods amounted to about 1%. On the other hand, it is noteworthy that the technique showing the best performance differs depending on the embedding vector size. For example, LLE showed the best performance of about 73.8% at 32, and IsoMap of about 87.8% was the best at 1024.

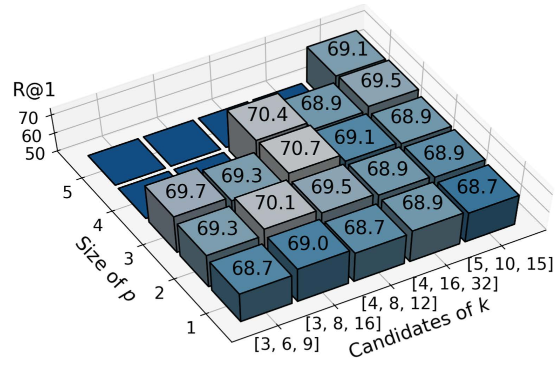


FIGURE 8. Performance variation according to k and p of the St(k, p) on CUB200-2011. Since we assumed a compact Stiefel manifold, we experimented only when $p \leq k$.

Figure 8 shows the performance of GML-PA according to k and p of St (k, p). GML-PA showed a performance variation of about 2.0% at various combinations of k and p. Although GML-PA is somewhat sensitive to parameters, overall performance was clearly improved compared to authentic PA. Note that GML-PA with $p = 1$ also showed higher performance than PA (68.4%). This proves that class variability has a positive effect on similarity learning. In detail, the performance according to k rarely shows a certain trend, whereas p is somewhat proportional to the R@1. This is because p indicates the capacity for data representation on Stiefel manifold.

Finally, Table 4 evaluates the performance for different manifolds and non-linear mappings. We performed this experiment with manifolds created while changing the orthogonal parameter p of the Stiefel manifold. From Table 4, we can observe that the Stiefel-based method has a performance advantage of about 1% R@1 over Riemannian and Unitary. Next, the logarithmic map [11] was used to generate the tangent space, but only the variation within 0.3% of the recall rate was observed. As a result, applying the same interpretation as in Fig. 7, finding an appropriate p on Stiefel manifold is equivalent to creating a GF that reflects the class variability of the embedding space well. Refer to **Appendix V** for additional materials such as experimental results on the in-shop clothes dataset and performance changes according to minibatch size.

V. RELATED WORKS

A. METRIC LEARNING SUMMARY

The goal of DML is to learn the projection matrix from the input space into the representation space [9], [45]. Various regularization terms have been proposed to mitigate overfitting in the DML process [6], [47]. In addition, CNN-based DML techniques such as pair losses [24] and triplet losses [37] have appeared to deal with high image dimensions. Also, there were studies on the concept of neighborhood component analysis (NCA) that locate points on the representation space in the discriminative decision space based on class supervision information [12]. On the other hand, unlike the previous studies, there were cases where global information on embedding space was quantified in various ways and used as a tool for similarity learning [34], [41]. Finally, a novel efficient retrieval approach for the image retrieval task from bulk databases has emerged [67]. [67] has in common with the proposed method in that color and texture features are extracted considering both local and global views of the image. And it is noteworthy that features useful for improving retrieval performance were extracted from the frequency domain.

B. LATEST DML APPROACHES

Recently, several metric learning techniques based on the analysis of class variability have been proposed. For instance, DSML [53] defined the signal-to-ratio (SNR) from the (Euclidean) distance and variance between samples in a mini-batch, and then used it as a loss function. However, due to the inherent characteristics of SNR metric, DSML can only be applied to metric learning based on triplet or contrastive distance. That is, it is difficult to apply to different techniques in terms of formula such as PA [23]. PADS [35] proposed an adaptive negative sampling mechanism through a class variability factor and a feedback loop based on reinforcement learning. PADS, which suggested a new sampling concept, is easy to attach and detach. However, since PADS does not structurally change the loss function, it does not affect similarity learning. Also, since PADS analyzes the statistical property on Euclidean space, it cannot capture the non-linear properties of the data. Finally, [63] analyzed class variability on a manifold. Reference [63] quantified class variability on the Riemannian manifold and applied this to basic metric learning techniques such as triplet. However, [63] is numerically unstable and sensitive to outlier samples because other constraint terms such as orthogonality are not employed when mapping to the manifold.

C. DISCRIMINANT ANALYSIS (DA)

The objective of DA is to obtain a factor that maximizes inter-class variability in a given space, and then perform dimension reduction and eigenvalue analysis using the factor. For example, dimension reduction could be performed after defining the relationship between samples through Laplacian graphs [49] or Laplacian graphs based on the Gaussian kernel could be used for constructing an intra/inter-class covariance matrix [40]. One of the major features of DA is that it can perform pattern analysis in association with manifold geometry. For example, Louis *et al.* proposed a probabilistic DA based on probabilistic model to deal with manifold characteristics [29]. On the other hand, since DA has a common goal with metric learning, the relationship between DA and metric learning has been theoretically analyzed [1]. However, as far as we know, there have been no recent studies that have performed the retrieval task using both DA and metric learning at the same time.

VI. CONCLUSION AND FUTURE WORK

This paper proposes a method for successfully obtaining nonlinear characteristics in embedding space and presents a novel metric structure based on this characteristic. The proposed method will inspire recent DML studies which consider only the Euclidean distance or cosine similarity in vector space. Future study will be to apply nonlinear characteristics to diverse DML techniques and to propose an optimization method for efficiently computing matrix computation.

APPENDIX I (GRADIENT ANALYSIS OF GML)

A. FORM OF GML-CONT AND GML-MS

The GML-Cont and GML-MS used in the experiments of the main body are defined by

$$\begin{aligned}&\mathcal{L}_{Cont}\left(\mathbf{z}_a,\mathbf{z}_p,\mathbf{z}_n\right)\\&\quad=\frac{1}{N_p}\sum\nolimits_{i=1}^{N_p}\left(\mathbf{D}^2_{ap_i}\phi_s\right)\\&\qquad+\frac{1}{N_n}\sum\nolimits_{i=1}^{N_n}\left[\beta_1-\mathbf{D}^2_{an_i}\phi_s\right]_+-\epsilon_2\sum_{i=1}^{kp}\lambda_i\end{aligned}\tag{I.1}$$

$$\begin{aligned}&\mathcal{L}_{MS}\left(\mathbf{z},\mathbf{z}_p,\mathbf{z}_n\right)\\&\quad=\frac{1}{N}\sum_{\mathbf{z}\in Z}\frac{1}{\alpha_{ms}}\log\left[1+\phi_s\sum_{\mathbf{z}_p\in C_p}e^{-\alpha_{ms}(<\mathbf{z},\mathbf{z}_p\succ\lambda_{ms})}\right]\\&\qquad+\frac{1}{\beta_{ms}}\log\left[1+\phi_s\sum_{\mathbf{z}_n\in C_n}e^{\beta_{ms}(<\mathbf{z},\mathbf{z}_n\succ\lambda_{ms})}\right]-\epsilon_2\sum_{i=1}^{kp}\lambda_i\end{aligned}\tag{I.2}$$

where $< \cdot,\cdot >$ denotes dot product and α_{ms}, β_{ms}, and λ_{ms} are set to 2, 50, and 1, respectively. Here, C_p and C_n denote the sets of vectors in positive and negative relationship with $\mathbf{z}$, respectively.

B. GRADIENT OF GML-TRI

The final form of Eq. (9) of the main body is given by

$$\begin{aligned}\mathcal{L}_{Tri}\left(\mathbf{z}_a,\mathbf{z}_p,\mathbf{z}_n\right)&=\frac{1}{N_p}\sum\nolimits_{i=1}^{N_p}\left[\mathbf{D}^2_{ap_i}\phi_s-\beta_1\right]_+\\&\quad+\frac{1}{N_n}\sum\nolimits_{i=1}^{N_n}\left[\beta_1-\mathbf{D}^2_{an_i}\phi_s\right]_+-\epsilon_2\sum_{i=1}^{kp}\lambda_i\end{aligned}\tag{I.3}$$

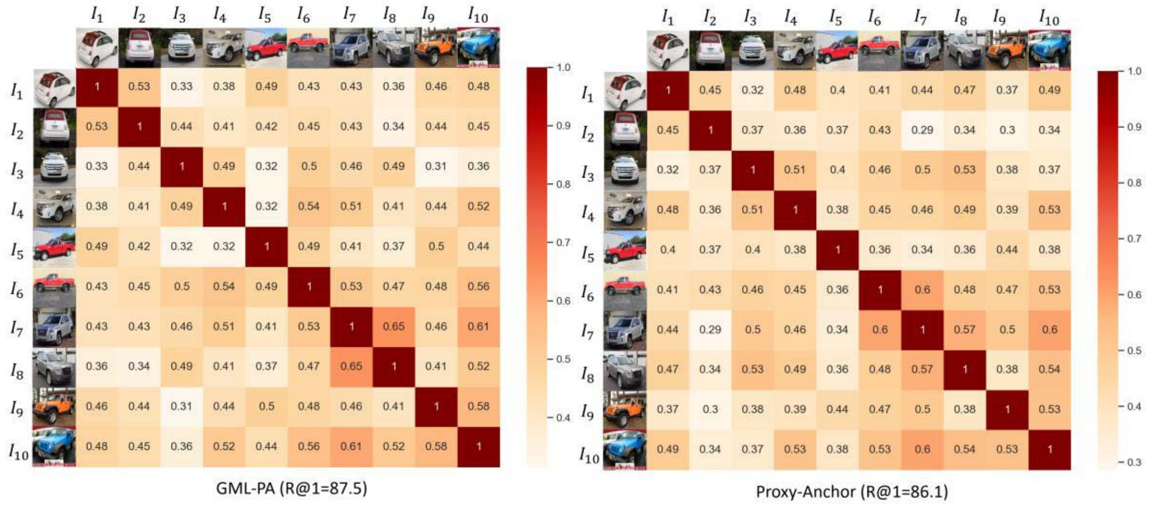


FIGURE 9. DC confusion matrices on Cars196 test split. Image sets belonging to the intra-class are as follows: {I_k, I_{k+1}}, $k = 1, 3, 5, 7, 9$.

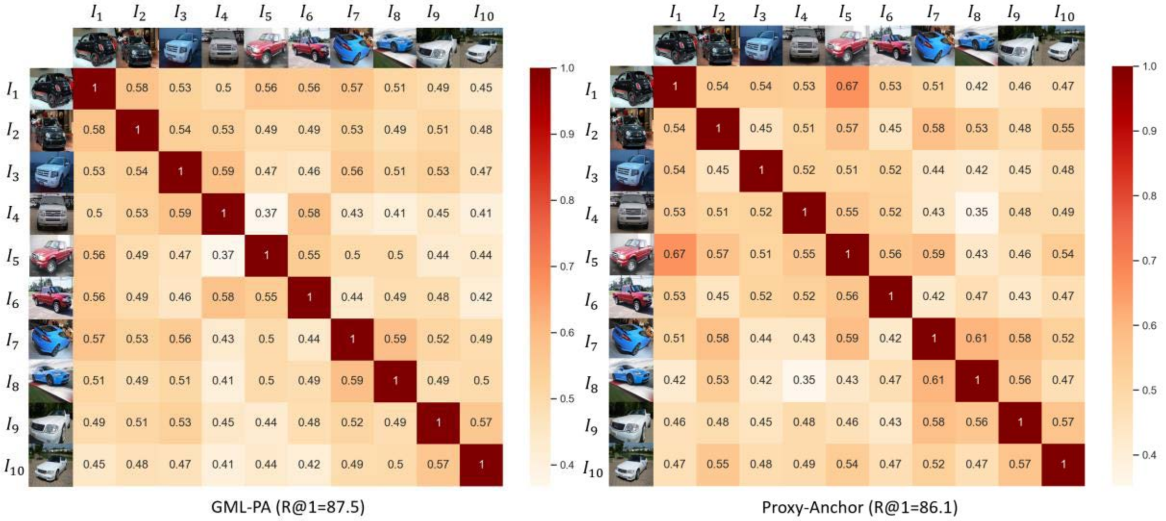


FIGURE 10. DC confusion matrices on Cars196 test split. The intra-class configuration is the same as Fig. 9.

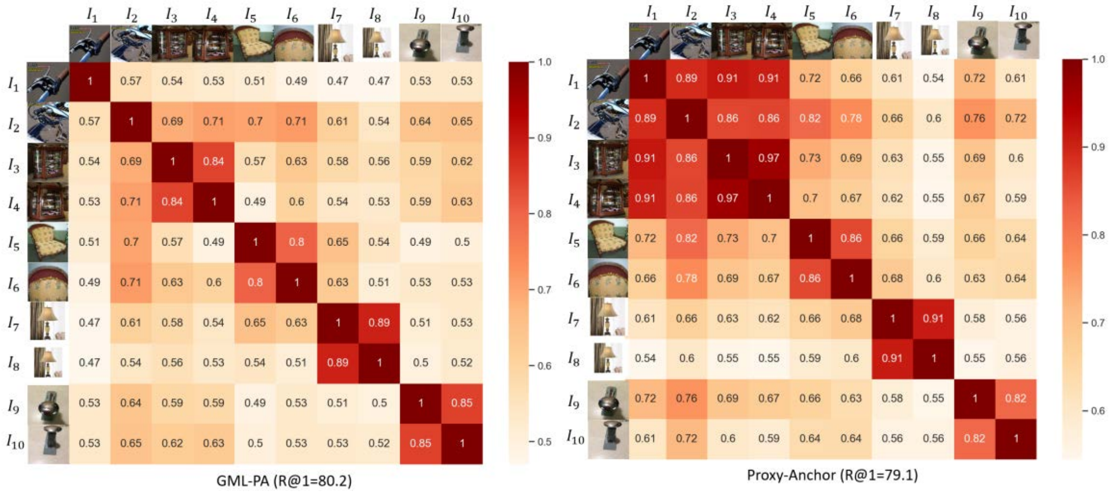


FIGURE 11. DC confusion matrices on SOP test split. The intra-class configuration is the same as Fig. 9.

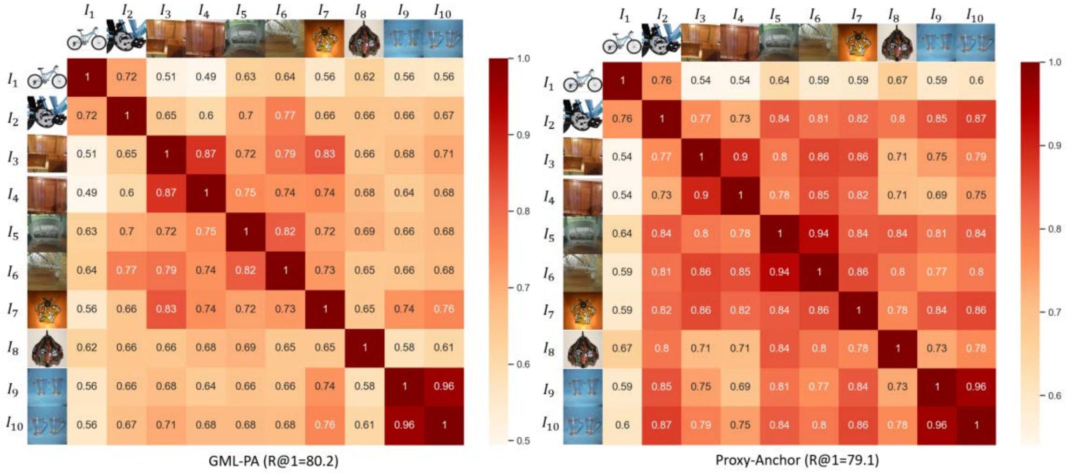


FIGURE 12. DC confusion matrices on SOP test split. The intra-class configuration is the same as Fig. 9.

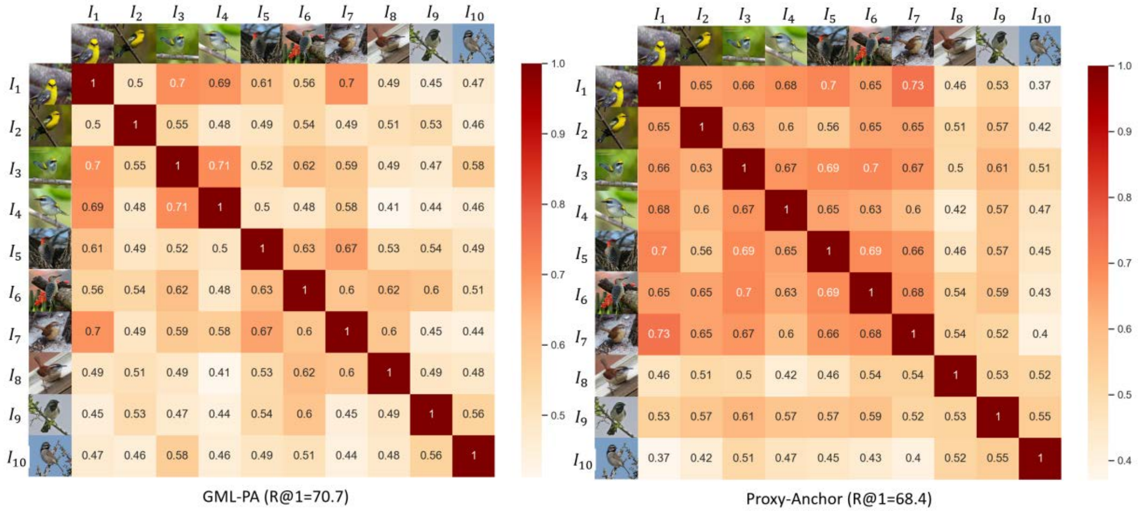


FIGURE 13. DC confusion matrices on CUB-200-2011 test split. The intra-class configuration is the same as Fig. 9.

The partial derivatives of Eq. (I.3) with respect to a triplet set $\mathbf{z}_a$, $\mathbf{z}_p$, $\mathbf{z}_n$ and β_1 are as follows:

$$\frac{\partial \mathcal{L}_{tri}}{\partial \mathbf{z}_p} = \frac{\partial}{\partial \mathbf{z}_p}\left[D_{ap}^2\phi_s - \beta_1\right]_+ = \frac{\partial D_{ap}^2}{\partial \mathbf{z}_p}\phi_s + D_{ap}^2\frac{\partial \phi_s}{\partial \mathbf{z}_p} \cong \begin{cases} 2\phi_s\left(\mathbf{z}_p - \mathbf{z}_a\right) = 2D_{ap}\phi_s\left(\dfrac{\mathbf{z}_p - \mathbf{z}_a}{D_{ap}}\right), & \text{if } D_{ap}^2\phi_s > \beta_1 \\ 0, & \text{otherwise} \end{cases} \tag{I.4}$$

$$\frac{\partial \mathcal{L}_{tri}}{\partial \mathbf{z}_n} = \frac{\partial}{\partial \mathbf{z}_n}\left[\beta_1 - D_{an}^2\phi_s\right]_+ = -\frac{\partial D_{an}^2}{\partial \mathbf{z}_n}\phi_s - D_{an}^2\frac{\partial \phi_s}{\partial \mathbf{z}_n} \cong \begin{cases} 2\phi_s\left(\mathbf{z}_a - \mathbf{z}_n\right) = 2D_{ap}\phi_s\left(\dfrac{\mathbf{z}_a - \mathbf{z}_n}{D_{an}}\right), & \text{if } D_{an}^2\phi_s < \beta_1 \\ 0, & \text{otherwise} \end{cases} \tag{I.5}$$

$$\frac{\partial \mathcal{L}_{tri}}{\partial \mathbf{z}_a} = \begin{cases} D_{ap}\left(\dfrac{2\left(\mathbf{z}_a - \mathbf{z}_p\right)}{D_{ap}}\phi_s\right) + D_{an}\left(\dfrac{2\left(\mathbf{z}_n - \mathbf{z}_a\right)}{D_{an}}\phi_s\right), \\ \quad \text{if } D_{an}^2\phi_s < \beta_1 \text{ and } D_{ap}^2\phi_s > \beta_1 \\ D_{ap}\left(\dfrac{2\left(\mathbf{z}_a - \mathbf{z}_p\right)}{D_{ap}}\phi_s\right), \\ \quad \text{if } D_{an}^2\phi_s > \beta_1 \text{ and } D_{ap}^2\phi_s > \beta_1 \\ D_{an}\left(\dfrac{2\left(\mathbf{z}_n - \mathbf{z}_a\right)}{D_{an}}\phi_s\right), \\ \quad \text{if } D_{an}^2\phi_s < \beta_1 \text{ and } D_{ap}^2\phi_s < \beta_1 \\ 0, \\ \quad \text{if } D_{an}^2\phi_s > \beta_1 \text{ and } D_{ap}^2\phi_s < \beta_1 \end{cases} \tag{I.6}$$

$$\frac{\partial \mathcal{L}_{tri}}{\partial \beta_1} = -1\left\{\beta_1 < D_{ap}^2\phi_s\right\} + 1\left\{\beta_1 > D_{an}^2\phi_s\right\} \tag{I.7}$$

The magnitues of embedding vector gradients are tuned by ϕ_s. In Eq. (I.7), $1\{\cdot\}$ outputs 1 when the given condition is satisfied, and 0 otherwise.

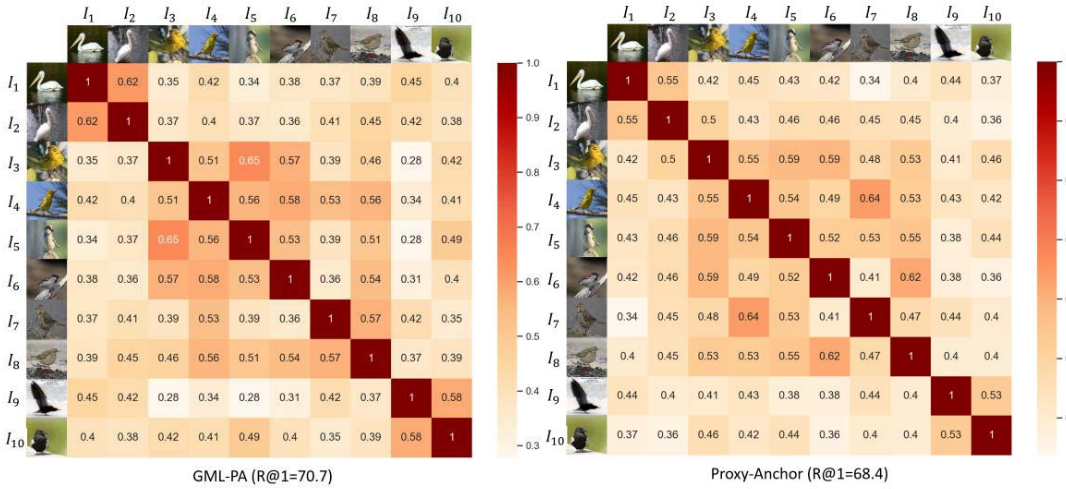


FIGURE 14. DC confusion matrices on CUB-200-2011 test split. The intra-class configuration is the same as Fig. 9.

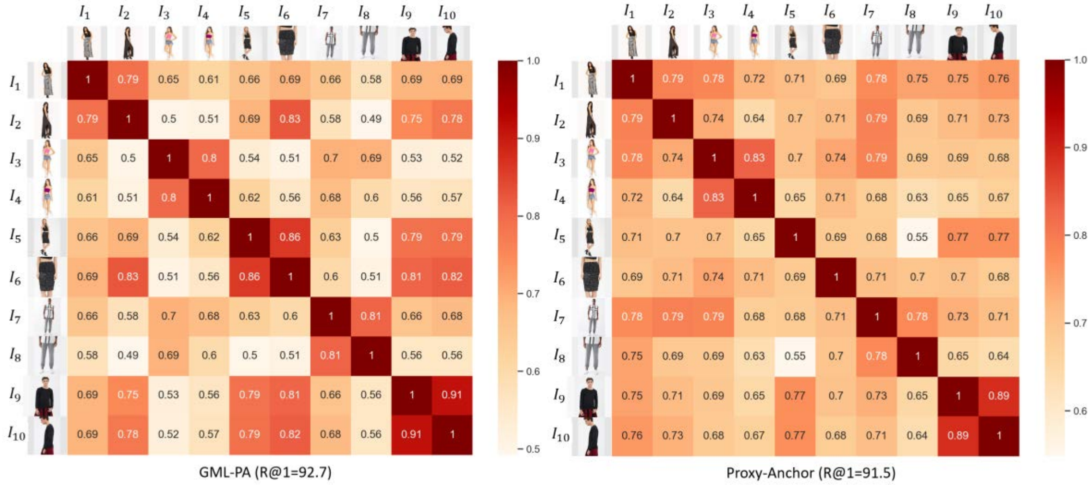


FIGURE 15. DC confusion matrices on In-shop query split. The intra-class configuration is the same as Fig. 9.

C. GRADIENT OF GML-PA

The loss function of GML-PA is defined by Eq. (I.8), and the derivative analysis on cosine similarity CS is given by Eq. (I.9).

$$\mathcal{L}_{PA}(\mathbf{z},\mathbf{p}) = \frac{1}{|P^{+}|}\sum_{\mathbf{p}\in P^{+}}\log\left(1+\phi_s\sum_{\mathbf{z}\in C_p} e^{-\alpha(CS(\mathbf{z},\mathbf{p})-\beta_2)}\right) + \frac{1}{|P|}\sum_{\mathbf{p}\in P}\log\left(1+\phi_s\sum_{\mathbf{z}\in C_n} e^{\alpha(CS(\mathbf{z},\mathbf{p})+\beta_2)}\right) - \epsilon_2\sum_{i=1}^{kp}\lambda_i \tag{I.8}$$

$$\frac{\partial\mathcal{L}_{PA}}{\partial CS} = \begin{cases} \dfrac{1}{|P^{+}|}\dfrac{-\alpha\phi_s h_p^{+}(\mathbf{z}) + \dfrac{\partial\phi_s}{\partial CS}h_p^{+}(\mathbf{z})}{1+\phi_s\sum_{\mathbf{z}'\in C_p}h_p^{+}(\mathbf{z}')}, & \forall\mathbf{z}\in C_p \\ \dfrac{1}{|P|}\dfrac{\alpha\phi_s h_p^{-}(\mathbf{z}) + \dfrac{\partial\phi_s}{\partial CS}h_p^{-}(\mathbf{z})}{1+\phi_s\sum_{\mathbf{z}'\in C_n}h_p^{-}(\mathbf{z}')}, & \forall\mathbf{z}\in C_n \end{cases} \tag{I.9}$$

where $h_p^{+}(\mathbf{z}) = e^{-\alpha(s(\mathbf{z},\mathbf{p})-\beta_2)}$ and $h_p^{-}(\mathbf{z}) = e^{\alpha(s(\mathbf{z},\mathbf{p})+\beta_2)}$ are positive and negative similarity metrics for embedding vector $\mathbf{z}$ given proxy $\mathbf{p}$, respectively. In Eq. (I.8), the scale factor ϕ_s is multiplied by the value of the cosine similarity value of the cosine function.

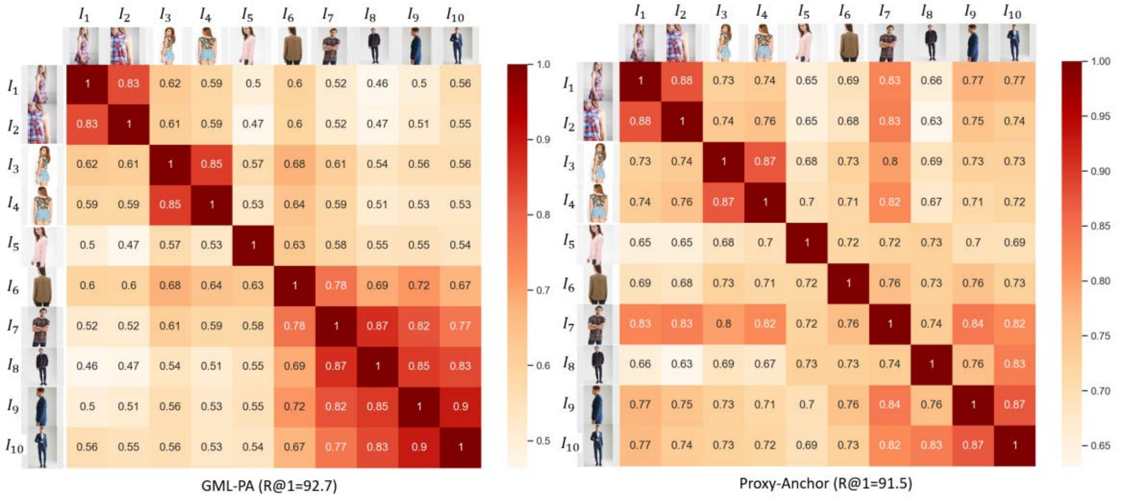
Note that the gradients of GML-Cont and GML-MS can be analyzed in the same way as GML-Tri and GML-PA, respectively.

D. GRADIENT OF ϕ_s

In general, ϕ_s created through the sampling process cannot perform the backpropagation process. To solve this problem, we build an end-to-end algorithm by referring to the re-parameterization trick of [56].

E. GRADIENT ON THE STIEFEL MANIFOLD

Let a point W be a trainable parameter. The process of updating W on the Stiefel manifold St consists of 4 steps as follows: First, find the derivative of the objective function $\nabla\mathcal{L}_{*}(W_t)$ of W at iteration t. Second, map $\nabla\mathcal{L}_{*}(\mathrm{W_t})$ to $\nabla_{\mathrm{St}}\mathcal{L}_{*}(W_t)$ on $\mathrm{T_{W_t}St}$ using the lifting map. Third, through



GML-PA (R@1=92.7)

	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}
I_1	1	0.83	0.62	0.59	0.5	0.6	0.52	0.46	0.5	0.56
I_2	0.83	1	0.61	0.59	0.47	0.6	0.52	0.47	0.51	0.55
I_3	0.62	0.61	1	0.85	0.57	0.68	0.61	0.54	0.56	0.56
I_4	0.59	0.59	0.85	1	0.53	0.64	0.59	0.51	0.53	0.53
I_5	0.5	0.47	0.57	0.53	1	0.63	0.58	0.55	0.55	0.54
I_6	0.6	0.6	0.68	0.64	0.63	1	0.78	0.69	0.72	0.67
I_7	0.52	0.52	0.61	0.59	0.58	0.78	1	0.87	0.82	0.77
I_8	0.46	0.47	0.54	0.51	0.55	0.69	0.87	1	0.85	0.83
I_9	0.5	0.51	0.56	0.53	0.55	0.72	0.82	0.85	1	0.9
I_{10}	0.56	0.55	0.56	0.53	0.54	0.67	0.77	0.83	0.9	1

Proxy-Anchor (R@1=91.5)

	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}
I_1	1	0.88	0.73	0.74	0.65	0.69	0.83	0.66	0.77	0.77
I_2	0.88	1	0.74	0.76	0.65	0.68	0.83	0.63	0.75	0.74
I_3	0.73	0.74	1	0.87	0.68	0.73	0.8	0.69	0.73	0.73
I_4	0.74	0.76	0.87	1	0.7	0.71	0.82	0.67	0.71	0.72
I_5	0.65	0.65	0.68	0.7	1	0.72	0.72	0.73	0.7	0.69
I_6	0.69	0.68	0.73	0.71	0.72	1	0.76	0.73	0.76	0.73
I_7	0.83	0.83	0.8	0.82	0.72	0.76	1	0.74	0.84	0.82
I_8	0.66	0.63	0.69	0.67	0.73	0.73	0.74	1	0.76	0.83
I_9	0.77	0.75	0.73	0.71	0.7	0.76	0.84	0.76	1	0.87
I_{10}	0.77	0.74	0.73	0.72	0.69	0.73	0.82	0.83	0.87	1

FIGURE 16. DC confusion matrices on In-shop query split. The intra-class configuration is the same as Fig. 9.

the specifically determined momentum M_t and $\nabla_{\mathrm{St}}\mathcal{L}_*(W_t)$ at iteration t, calculate the momentum M_{t+1} for the next step. Finally, update W_t to W_{t+1} on the Stiefel manifold along the direction of M_{t+1}. For more details, please refer [57] and [58].

APPENDIX II (GENERALIZATION BOUNDS OF GML)

In this section, background knowledge for generalization bound analysis and proof of Theorem II.2 are examined.

F. BACKGROUND

Definition II.1 (Covering Number [25]): For a metric space $(\mathcal{Z}, \rho)$ and $\mathcal{V} \subset \mathcal{Z}$, we say that $\hat{\mathcal{V}} \subset \mathcal{V}$ is a γ-cover of $\mathcal{V}$ if $\forall \mathbf{t} \in \mathcal{V}, \exists \hat{t} \in \hat{\mathcal{V}}$ such that $\rho(\mathbf{t}, \hat{t}) \leq \gamma$. The γ-covering number of $\mathcal{V}$ is defined by

$$\mathcal{N}(\gamma, \mathcal{V}, \rho) = \min\left\{\left\lceil \hat{\mathcal{V}} \right\rceil : \hat{\mathcal{V}} \mathit{isa}\gamma - \mathit{cover\ of}\ \mathcal{V}\right\} \quad \text{(II.1)}$$

In particular, when $\mathcal{V}$ is compact, $\mathcal{N}(\gamma, \mathcal{V}, \rho)$ is finite, leading to a finite cover. Then, $\mathcal{Z}$ can be *partitioned* into $|\mathcal{Y}|\mathcal{N}(\gamma, \mathcal{V}, \rho)$ subsets such that two vectors $\mathbf{z}$ and $\mathbf{z}^*$ belong to the same subset, i.e., $y = y^*$ and $\rho(\mathbf{z}, \mathbf{z}^*) \leq \gamma$.

The robustness of the pairwise similarity metric through subset samples placed on a compact space is defined as follows.

Definition II.2 (Robustness of Metric Learning [59]): An algorithm $\mathcal{A}$ is $(\mathcal{N}(\cdot), \eta(\cdot))$-robust for $\mathcal{N}(\cdot) \in \mathbb{N}$ and $\eta(\cdot) : (\mathcal{Z} \times \mathcal{Z})^N \rightarrow \mathbb{R}$ if $\mathcal{Z}$ can be partitioned into $\mathcal{N}(\cdot)$ disjoints sets, denoted by $\{Q_i\}_{i=1}^{\mathcal{N}(\cdot)}$, such that the following holds for all $C_{all} \in \mathcal{Z}^N$: $\forall \left(\mathbf{z}_a, \mathbf{z}_p, \mathbf{z}_n\right) \in C_{all}, \forall \mathbf{z}, \mathbf{z}', \mathbf{z}'' \in \mathcal{Z}, \forall i, j, k \in [\mathcal{N}(\cdot)]$.

If $\mathbf{z}_a, \mathbf{z} \in Q_i, \mathbf{z}_p, \mathbf{z}' \in Q_j, \mathbf{z}_n, \mathbf{z}'' \in Q_k$, then

$$\left|\mathcal{L}\left(\mathcal{A}_{C_{all}}, \mathbf{z}_a, \mathbf{z}_p, \mathbf{z}_n\right) - \mathcal{L}\left(\mathcal{A}_{C_{all}}, \mathbf{z}, \mathbf{z}', \mathbf{z}''\right)\right| \leq \eta\left(C_{all}\right) \quad \text{(II.2)}$$

where $\mathcal{A}_{C_{all}}$ is the hypothesis learned by $\mathcal{A}$ on C_{all}.

$\mathcal{N}(\cdot)$ and $\eta(\cdot)$ *quantify* the robustness of the algorithm and depend on the training sample. A shown in [59], **Definition II.2** guarantees the following generalization bound.

Theorem II.1 (Generalization Bound of Metric Learning [59]): If a learning algorithm $\mathcal{A}$ is $(\mathcal{N}(\cdot), \eta(\cdot))$-robust and the training sample consists of the triplets C_{all}, then for any $\delta > 0$, with probability at least $1 - \delta$ we have:

$$\left|\mathcal{R}\left(\mathcal{A}_{C_{all}}\right) - \mathcal{R}_{emp}\left(\mathcal{A}_{C_{all}}\right)\right| \leq \eta\left(C_{all}\right) + 3U\sqrt{\frac{\mathcal{N}\ln 2 + \ln\frac{1}{\delta}}{0.5N}} \quad \text{(II.3)}$$

where $\mathcal{R}$ and $\mathcal{R}_{emp}$ are expected and empirical losses based on $\left(\mathbf{z}, \mathbf{z}', \mathbf{z}''\right)$ and $\left(\mathbf{z}_a, \mathbf{z}_p, \mathbf{z}_n\right)$, respectively. U is a pre-defined constant.

It is noteworthy that the generalization bound of $\mathcal{A}$ can be analyzed more clearly with **Theorem II.2** than **Theorem II.1**. **Theorem II.2** shows that the loss of triplet tuples located close to each other in the partitions split by the covering number has a certain range.

Theorem II.2: Fix $\gamma > 0$ and a metric ρ of $\mathcal{Z}$, suppose that $\forall \mathbf{z}_a, \mathbf{z}_p, \mathbf{z}_n, \mathbf{z}, \mathbf{z}', \mathbf{z}''$: $\left(\mathbf{z}_a, \mathbf{z}_p, \mathbf{z}_n\right) \in C_{all}$, $\rho\left(\mathbf{z}_a, \mathbf{z}\right) \leq \gamma$, $\rho\left(\mathbf{z}_p, \mathbf{z}'\right) \leq \gamma$, $\rho\left(\mathbf{z}_n, \mathbf{z}''\right) \leq \gamma$. If $\mathcal{A}$ satisfies

$$\left|\mathcal{L}\left(\mathcal{A}_{C_{all}}, \mathbf{z}_a, \mathbf{z}_p, \mathbf{z}_n\right) - \mathcal{L}\left(\mathcal{A}_{C_{all}}, \mathbf{z}, \mathbf{z}', \mathbf{z}''\right)\right| \leq \eta\left(C_{all}\right) \quad \text{(II.4)}$$

and $\mathcal{N}(\gamma, \mathcal{Z}, \rho) < \infty$. Then the algorithm $\mathcal{A}$ is $(|\mathcal{Y}|\mathcal{N}(\gamma, \mathcal{Z}, \rho), \eta(C_{all}))$-robust.

G. GENERALIZATION BOUND FOR GML-TRI

We focus on GML-Tri in Eq. (I.3) and derive a generalization bound by showing that GML-Tri satisfies **Theorem II.2.** First, assume $\|\mathbf{z}\|_2 \leq R$. Using **Definition II.1**, we can partition $\mathcal{Z}$ into $|\mathcal{Y}|\mathcal{N}(\gamma, \mathcal{V}, \rho)$ subsets such that if two examples

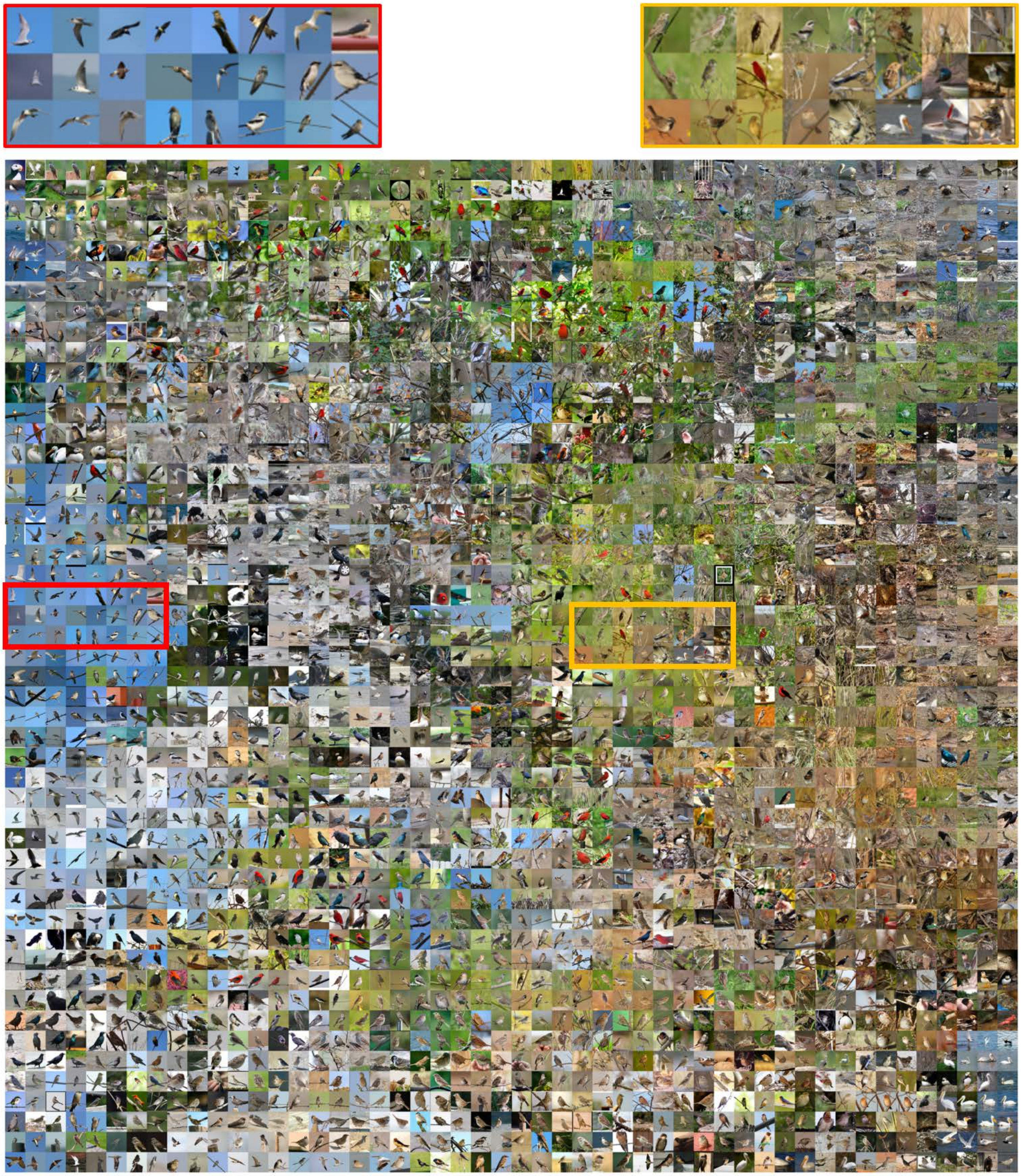


FIGURE 17. Grid-wise T-SNE visualization of the embedding space on the test split of CUB-200-2011 dataset.

$\mathbf{z}$ and $\mathbf{z}'$ belong to the same subset, then $y = y'$ and $\rho\left(\mathbf{z}, \mathbf{z}'\right) \leq \gamma$. For $\forall \mathbf{z}_a, \mathbf{z}_p, \mathbf{z}_n, \mathbf{z}, \mathbf{z}', \mathbf{z}'' \in Z$ and $y_a, y_p, y_n, y, y', y'' \in Y$, if $y_a = y$, $\|\mathbf{z}_a - \mathbf{z}\|_2 \leq \gamma$, $y_p = y'$, $\left\|\mathbf{z}_p - \mathbf{z}'\right\|_2 \leq \gamma$, $y_n = y''$, $\left\|\mathbf{z}_n - \mathbf{z}''\right\|_2 \leq \gamma$, then $\left(\mathbf{z}_a, \mathbf{z}_p, \mathbf{z}_n\right)$ and $\left(\mathbf{z}, \mathbf{z}', \mathbf{z}''\right)$ are either both admissible or both non-admissible triplets. We do not consider the non-admissible case as the respective loss

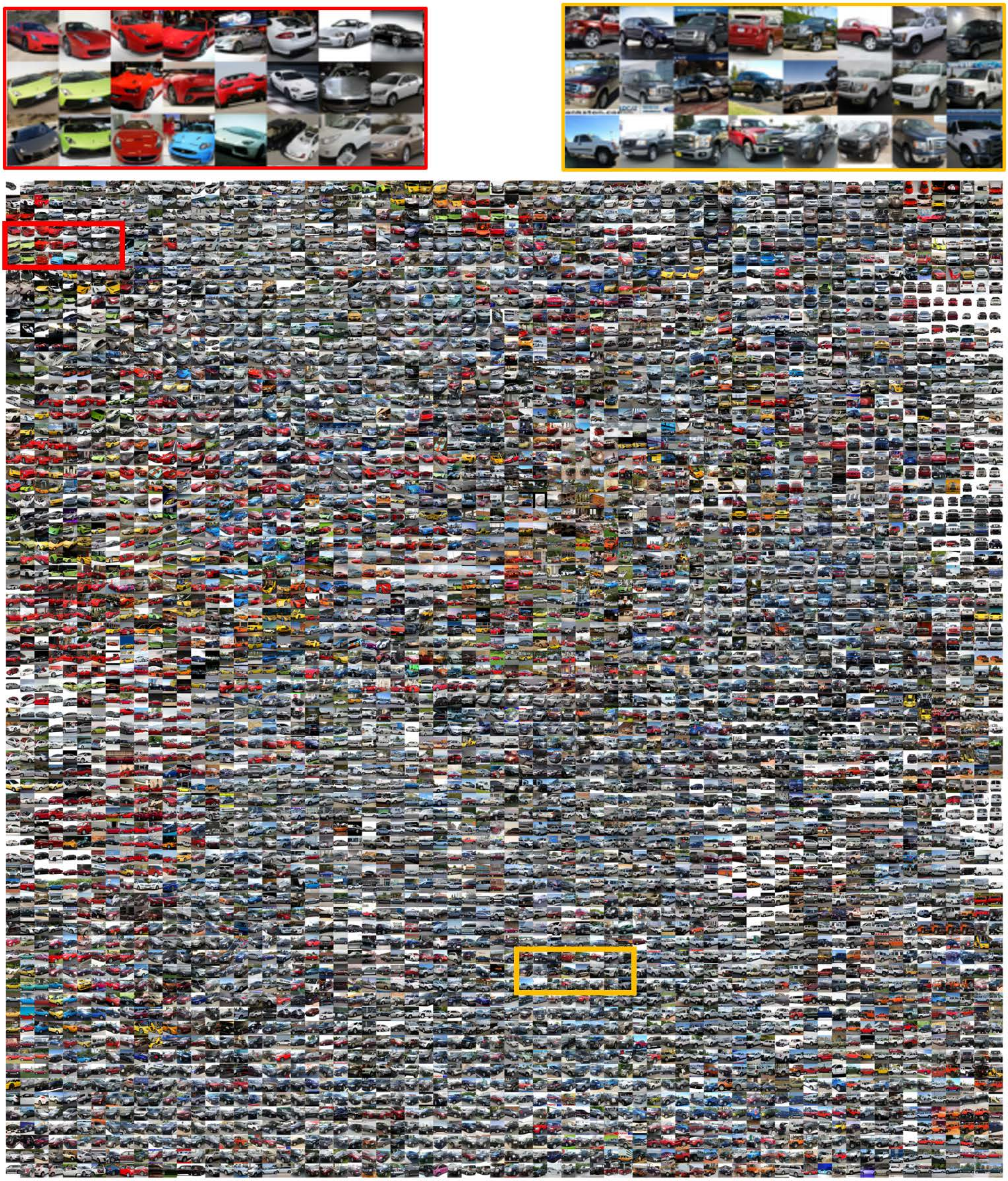


FIGURE 18. Grid-wise T-SNE visualization of the embedding space on the test split of Cars196 dataset.

has 0. Thus, we only consider the case of admissible triplets as follow:

$$
\begin{aligned}
&\mathcal{L}\left(\mathbf{z}_a, \mathbf{z}_p, \mathbf{z}_n\right) - \mathcal{L}\left(\mathbf{z}, \mathbf{z}', \mathbf{z}''\right) \\
&= \left[D^2\left(\mathbf{z}_a, \mathbf{z}_p\right)\phi_s - \beta_1\right]_+ + \left[\beta_1 - D^2\left(\mathbf{z}_a, \mathbf{z}_n\right)\phi_s\right]_+ \\
&\quad - \left[D^2\left(\mathbf{z}, \mathbf{z}'\right)\phi_s' - \beta_1\right]_+ - \left[\beta_1 - D^2\left(\mathbf{z}, \mathbf{z}''\right)\phi_s'\right]_+ \qquad \text{(II.5)}
\end{aligned}
$$

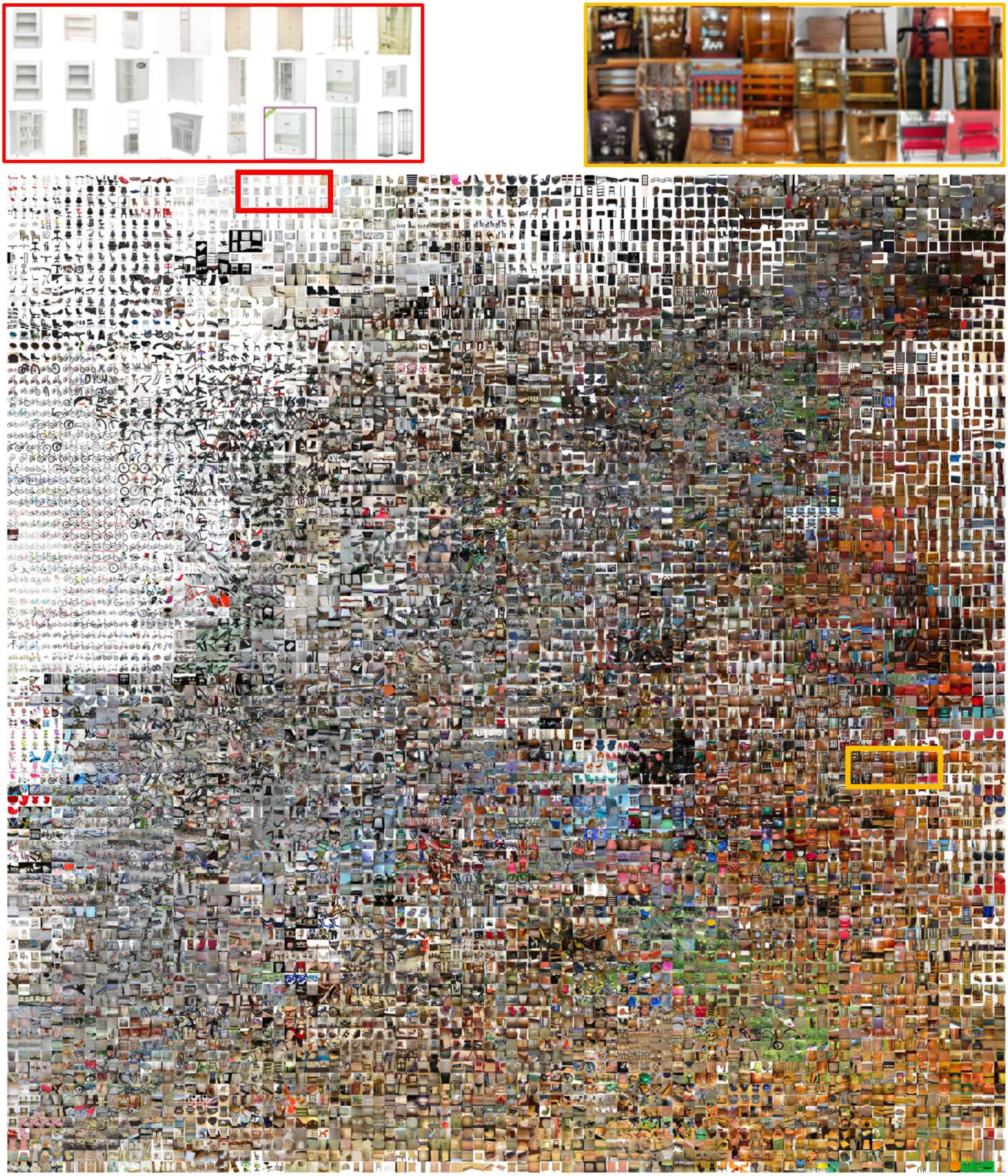


FIGURE 19. Grid-wise T-SNE visualization of the embedding space on the test split of SOP dataset.

Since hinge loss is a function with a 1-Lipschitz bound, we can rewrite Eq. (II.5) as follows:

$$
\begin{aligned}
&\mathcal{L}\left(\mathbf{z}_a, \mathbf{z}_p, \mathbf{z}_n\right) - \mathcal{L}\left(\mathbf{z}, \mathbf{z}', \mathbf{z}''\right)\\
&\quad \leq |\sqrt{\phi}_s\left(\mathbf{z}_a - \mathbf{z}_p\right)^T\left(\mathbf{z}_a - \mathbf{z}_p\right)\sqrt{\phi}_s\\
&\quad -\sqrt{\phi}_s\left(\mathbf{z}_a - \mathbf{z}_n\right)^T\left(\mathbf{z}_a - \mathbf{z}_n\right)\sqrt{\phi}_s\\
&\quad +\sqrt{\phi}'_s\left(\mathbf{z} - \mathbf{z}''\right)^T\left(\mathbf{z} - \mathbf{z}''\right)\sqrt{\phi}'_s\\
&\quad -\sqrt{\phi}'_s\left(\mathbf{z} - \mathbf{z}'\right)^T\left(\mathbf{z} - \mathbf{z}'\right)\sqrt{\phi}'_s| \qquad \text{(II.6)}
\end{aligned}
$$

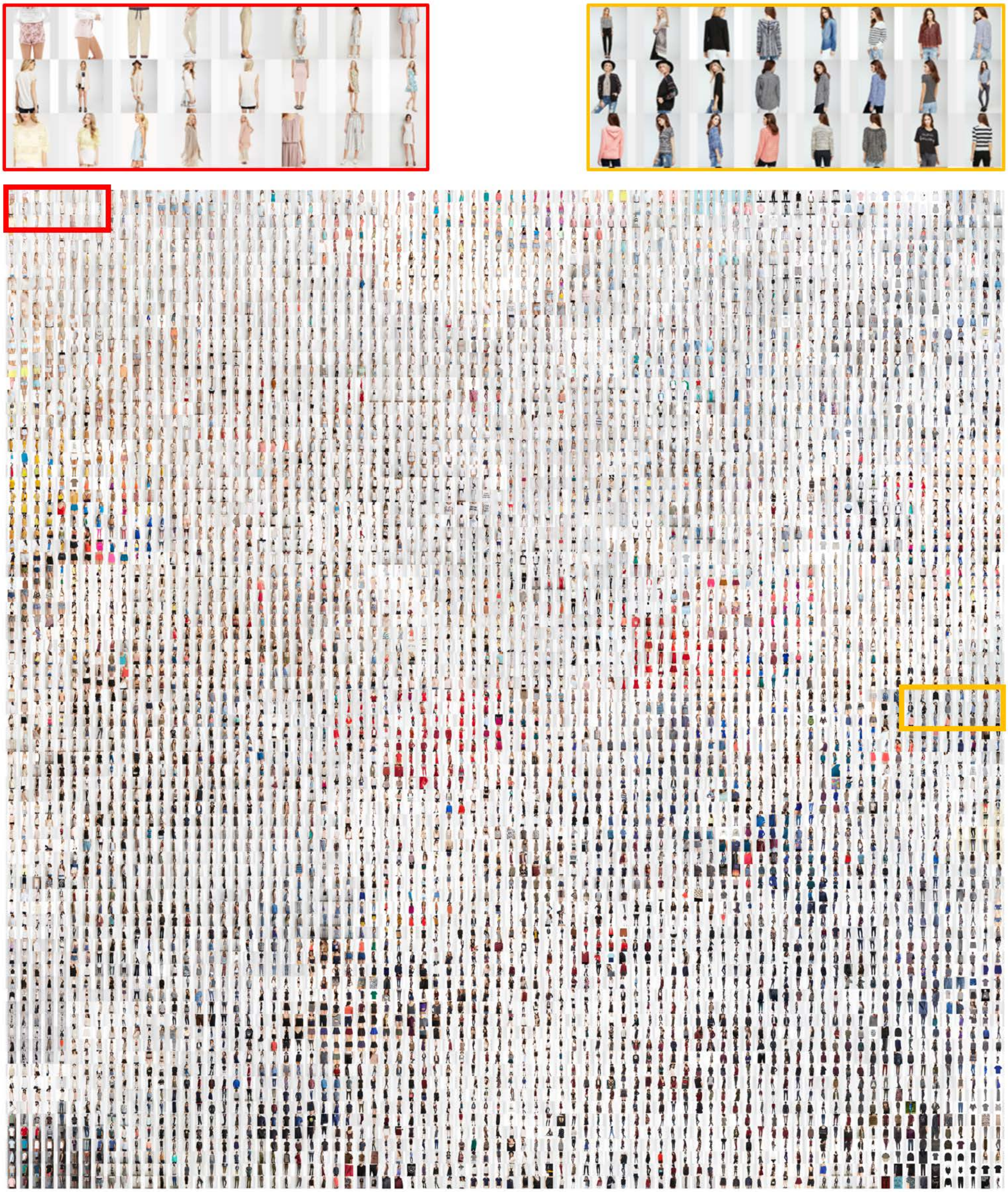


FIGURE 20. Grid-wise T-SNE visualization of the embedding space on the query split of In-shop dataset.

In Eq. (II.6), additional terms may be introduced to bind terms having similar properties to each other.

$$
\begin{aligned}
(\text{II.6}) \leq\ & |\sqrt{\phi_s}\left(\mathbf{z}_a-\mathbf{z}_p\right)^T\left(\mathbf{z}_a-\mathbf{z}_p\right)\sqrt{\phi_s} \\
& -\sqrt{\phi_s}\left(\mathbf{z}_a-\mathbf{z}_p\right)^T\left(\mathbf{z}-\mathbf{z}'\right)\sqrt{\phi_s'} \\
& +\sqrt{\phi_s}\left(\mathbf{z}_a-\mathbf{z}_p\right)^T\left(\mathbf{z}-\mathbf{z}'\right)\sqrt{\phi_s'} \\
& -\sqrt{\phi_s'}\left(\mathbf{z}-\mathbf{z}'\right)^T\left(\mathbf{z}-\mathbf{z}'\right)\sqrt{\phi_s'}
\end{aligned}
$$

$$\begin{aligned}
&+\sqrt{\phi_s'}\left(\mathbf{z}-\mathbf{z}''\right)^T\left(\mathbf{z}-\mathbf{z}''\right)\sqrt{\phi_s'}\\
&-\sqrt{\phi_s'}\left(\mathbf{z}-\mathbf{z}''\right)^T\left(\mathbf{z}_a-\mathbf{z}_n\right)\sqrt{\phi_s}\\
&+\sqrt{\phi_s'}\left(\mathbf{z}-\mathbf{z}''\right)^T\left(\mathbf{z}_a-\mathbf{z}_n\right)\sqrt{\phi_s}\\
&-\sqrt{\phi_s}\left(\mathbf{z}_a-\mathbf{z}_n\right)^T\left(\mathbf{z}_a-\mathbf{z}_n\right)\sqrt{\phi_s}|\\
=&\ |\sqrt{\phi_s}\left(\mathbf{z}_a-\mathbf{z}_p\right)^T\left(\left(\mathbf{z}_a-\mathbf{z}_p\right)\sqrt{\phi_s}-\left(\mathbf{z}-\mathbf{z}'\right)\sqrt{\phi_s'}\right)\\
&+\left(\sqrt{\phi_s}\left(\mathbf{z}_a-\mathbf{z}_p\right)-\sqrt{\phi_s'}\left(\mathbf{z}-\mathbf{z}'\right)\right)^T\left(\mathbf{z}-\mathbf{z}'\right)\sqrt{\phi_s'}\\
&+\sqrt{\phi_s'}\left(\mathbf{z}-\mathbf{z}''\right)^T\left(\left(\mathbf{z}-\mathbf{z}''\right)\sqrt{\phi_s'}-\left(\mathbf{z}_a-\mathbf{z}_n\right)\sqrt{\phi_s}\right)\\
&+\left(\sqrt{\phi_s'}\left(\mathbf{z}-\mathbf{z}''\right)-\sqrt{\phi_s}\left(\mathbf{z}_a-\mathbf{z}_n\right)\right)^T\left(\mathbf{z}_a-\mathbf{z}_n\right)\sqrt{\phi_s}|
\end{aligned}\tag{II.7}$$

$$\begin{aligned}
(\text{II}.7)\leq&\left|\sqrt{\phi_s}\left(\mathbf{z}_a-\mathbf{z}_p\right)^T\left(\sqrt{\phi_s}\mathbf{z}_a-\sqrt{\phi_s'}\mathbf{z}\right)\right|\\
&+\left|\sqrt{\phi_s}\left(\mathbf{z}_a-\mathbf{z}_p\right)^T\left(\sqrt{\phi_s'}\mathbf{z}'-\sqrt{\phi_s}\mathbf{z}_p\right)\right|\\
&+\left|\left(\sqrt{\phi_s}\mathbf{z}_a-\sqrt{\phi_s'}\mathbf{z}\right)^T\left(\mathbf{z}-\mathbf{z}'\right)\sqrt{\phi_s'}\right|\\
&+\left|\left(\sqrt{\phi_s'}\mathbf{z}'-\sqrt{\phi_s}\mathbf{z}_p\right)^T\left(\mathbf{z}-\mathbf{z}'\right)\sqrt{\phi_s'}\right|\\
&+\left|\sqrt{\phi_s'}\left(\mathbf{z}-\mathbf{z}''\right)^T\left(\sqrt{\phi_s'}\mathbf{z}-\sqrt{\phi_s}\mathbf{z}_a\right)\right|\\
&+\left|\sqrt{\phi_s'}\left(\mathbf{z}-\mathbf{z}''\right)^T\left(\sqrt{\phi_s}\mathbf{z}_n-\sqrt{\phi_s'}\mathbf{z}''\right)\right|\\
&+\left|\left(\sqrt{\phi_s'}\mathbf{z}-\sqrt{\phi_s}\mathbf{z}_a\right)^T\left(\mathbf{z}_a-\mathbf{z}_n\right)\sqrt{\phi_s}\right|\\
&+\left|\left(\sqrt{\phi_s}\mathbf{z}_n-\sqrt{\phi_s'}\mathbf{z}''\right)^T\left(\mathbf{z}_a-\mathbf{z}_n\right)\sqrt{\phi_s}\right|
\end{aligned}\tag{II.8}$$

Eq. (II.8) is expressed by Holder's inequality as follows:

$$\begin{aligned}
(\text{II}.8)\leq&\left\|\sqrt{\phi_s}\left(\mathbf{z}_a-\mathbf{z}_p\right)\right\|_2\left\|\sqrt{\phi_s}\mathbf{z}_a-\sqrt{\phi_s'}\mathbf{z}\right\|_2\\
&+\left\|\sqrt{\phi_s}\left(\mathbf{z}_a-\mathbf{z}_p\right)\right\|_2\left\|\sqrt{\phi_s'}\mathbf{z}'-\sqrt{\phi_s}\mathbf{z}_p\right\|_2\\
&+\left\|\sqrt{\phi_s}\mathbf{z}_a-\sqrt{\phi_s'}\mathbf{z}\right\|_2\left\|\left(\mathbf{z}-\mathbf{z}'\right)\sqrt{\phi_s'}\right\|_2\\
&+\left\|\sqrt{\phi_s'}\mathbf{z}'-\sqrt{\phi_s}\mathbf{z}_p\right\|_2\left\|\left(\mathbf{z}-\mathbf{z}'\right)\sqrt{\phi_s'}\right\|_2\\
&+\left\|\sqrt{\phi_s'}\left(\mathbf{z}-\mathbf{z}''\right)\right\|_2\left\|\sqrt{\phi_s'}\mathbf{z}-\sqrt{\phi_s}\mathbf{z}_a\right\|_2\\
&+\left\|\sqrt{\phi_s'}\left(\mathbf{z}-\mathbf{z}''\right)\right\|_2\left\|\sqrt{\phi_s}\mathbf{z}_n-\sqrt{\phi_s'}\mathbf{z}''\right\|_2\\
&+\left\|\sqrt{\phi_s'}\mathbf{z}-\sqrt{\phi_s}\mathbf{z}_a\right\|_2\left\|\left(\mathbf{z}_a-\mathbf{z}_n\right)\sqrt{\phi_s}|\right\|_2\\
&+\left\|\sqrt{\phi_s}\mathbf{z}_n-\sqrt{\phi_s'}\mathbf{z}''\right\|_2\left\|\left(\mathbf{z}_a-\mathbf{z}_n\right)\sqrt{\phi_s}\right\|_2\\
\leq&\left\|\tilde{z}_a-\tilde{z}_p\right\|_2\|\tilde{z}_a-\tilde{z}\|_2+\left\|\tilde{z}_a-\tilde{z}_p\right\|_2\left\|\tilde{z}'-\tilde{z}_p\right\|_2\\
&+\|\tilde{z}_a-\tilde{z}\|_2\left\|\tilde{z}-\tilde{z}'\right\|_2+\left\|\tilde{z}'-\tilde{z}_p\right\|_2\left\|\tilde{z}-\tilde{z}'\right\|_2\\
&+\left\|\tilde{z}-\tilde{z}''\right\|_2\|\tilde{z}-\tilde{z}_a\|_2+\left\|\tilde{z}-\tilde{z}''\right\|_2\left\|\tilde{z}_n-\tilde{z}''\right\|_2\\
&+\|\tilde{z}-\tilde{z}_a\|_2\|\tilde{z}_a-\tilde{z}_n\|_2+\left\|\tilde{z}_n-\tilde{z}''\right\|_2\|\tilde{z}_a-\tilde{z}_n\|_2
\end{aligned}\tag{II.9}$$

where the product of each element of triplets $\left(\mathbf{z}_a,\mathbf{z}_p,\mathbf{z}_n\right)$ and $\sqrt{\phi_s}$ is redefined as $\tilde{z}_a,\tilde{z}_p,\tilde{z}_n$, and the product of each element of triplets $\left(\mathbf{z},\mathbf{z}',\mathbf{z}''\right)$ and $\sqrt{\phi_s'}$ is redefined as $\tilde{z},\tilde{z}',\tilde{z}''$, respectively.

Since ϕ_s and ϕ_s' obtained through the bounded eigen spectrum have specific bounds, $\tilde{z}$ also have a certain bound, i.e., $\|\tilde{z}_a-\tilde{z}\|_2\leq\tilde{\gamma}$, $\left\|\tilde{z}_p-\tilde{z}'\right\|_2\leq\tilde{\gamma}$, $\left\|\tilde{z}_n-\tilde{z}''\right\|_2\leq\tilde{\gamma}$, and $\|\tilde{z}\|_2\leq\tilde{R}$. The covering number $\tilde{\mathcal{N}}(\cdot)$ is also changed accordingly. Finally, we can conclude the generalization bound of GML-Tri is given by

$$(\text{II}.9)\leq 2\tilde{\gamma}\tilde{R}\times 8=16\tilde{\gamma}\tilde{R}\tag{II.10}$$

We can say that GML-Tri is $\left(|Y|\tilde{\mathcal{N}}(\tilde{\gamma},Z,\|\cdot\|_2),16\tilde{\gamma}\tilde{R}\right)$-robust, which means $\eta\left(C_{all}\right)$ in Eq. (II.4) is equal to $16\tilde{\gamma}\tilde{R}$. □

H. GENERALIZATION BOUND FOR GML-PA

In this section, we guarantee the generalization bound of GML-PA by using **Theorem II.2**. Let ρ and ρ' denote the cosine similarity corresponding to the part of empirical and expected loss, respectively. Then, we can show the generalization bound of GML-PA is given by

$$\left|\log\left(1+\rho\right)-\log\left(1+\rho'\right)\right|\leq\eta\left(C_{all}\right)\tag{II.11}$$

Here, since the positive and negative terms are derived identically, it can be assumed that Eq. (II.11) has the same meaning as **Theorem II.2**.

By the way, the cosine similarity metric ρ having a finite range in the compact space satisfies the Taylor expansion as follows:

$$\log\left(1+\rho\right)=\sum_{i=1}^{\infty}(-1)^{i+1}\frac{\rho^i}{i}\leq(-1)^{1+1}\frac{\rho^1}{1}=\rho\tag{II.12}$$

According to Eq. (II.12), the relationship between GML-PA with softplus function structure and GML-Tri with hinge function structure can be expressed by

$$\left|\log\left(1+\rho\right)-\log\left(1+\rho'\right)\right|\leq\left|\rho-\rho'\right|\leq\eta\left(C_{all}\right)\tag{II.13}$$

If the generalization bounds of GML-PA and GML-Tri are defined as η_L and η_E, respectively, this can be re-written by $\eta_L\leq\eta_E\leq\eta\left(C_{all}\right)=16\tilde{\gamma}\tilde{R}$.

Therefore, it can be inferred that GML-PA also has a finite generalization bound, which is a more compact bound than η_E. □

I. GENERALIZATION BOUNDS FOR GML-CONT AND GML-MS

GML-Cont/MS has the same structure as GML-Tri/PA, but only the composition of the constant term has changed. Therefore, the generalization bounds of GML-Cont and GML-MS can be equally derived from Eqs. (II.4) and (II.13).

APPENDIX III (HYPER-PARAMETER SETTING)

J. GML-PA/MS

Embedding network is trained for 80 epochs with the initial learning rate of 10^{-4} on the CUB200-2011 and Cars196, and for 120 epochs with the initial learning rate of 6×10^{-4} on the SOP and In-shop. Decay factor of optimizer is set to 10^{-4}.

Step-wise learning rate reducer (StepLR) with the step size of 10 and the ratio of 0.5 is used for convergence stability on the CUB200-2011 and Cars196, and the step size of 20 and the ratio of 0.25 on the SOP and In-shop. In case of GML-PA, the learning rate for proxies is scaled up 200 times for faster convergence. Random sampling is used to configure minibatch during training phase. Default minibatch size is 180.

K. GML-TRI/CONT

Embedding network is trained for 100 epochs with the initial learning rate of 10^{-5} on the CUB200-2011 and Cars196, and for 100 epochs with the initial learning rate of 4×10^{-5} on the SOP and In-shop. Decay factor of optimizer is set to 4×10^{-4}. StepLR with the step size of 25 and the ratio of 0.3 is used for convergence stability on the four DML public datasets. Distance-based sampling [46] is used to configure minibatch during training phase. Default minibatch size is 112.

L. GML-DiVA

Embedding network is trained for 150 epochs with the initial learning rate of 10^{-5} on the four DML public datasets. The configuration of StepLR is the same as GML-Tri. Distance-based sampling [46] is used to configure minibatch during training phase. Default minibatch size is 112. Other detailed coefficients are set by referring to [31].

APPENDIX IV (ADDITIONAL QUALITATIVE RESULTS)

In this section, qualitative results in terms of confusion matrix based on the distance correlation (DC) index is additionally examined.

We qualitatively compare the results in Figs. 9-16 obtained through the CUB200-2011 [42], Cars196 [26], and Stanford Online Products (SOP) [32] datasets. The main comparison method is Proxy-Anchor (PA) [23]. The overall DC value between inter-class images is lower in GML-PA than in PA. This is because the inter-class variability property has been further considered in GML-PA. For example, in Fig. 13, the DC value between I_1 and I_5 images of PA is 0.7, which is higher than that between all intra-class images.

Figures 17, 18, 19, and 20 show the embedding spaces using grid-wise T-SNE visualization tool [60]. We can see the images which contain similar attributes locate close to each other, vice versa. Compared to the visualization results of [23], our 2D embedding plot shows discriminative location of images.

APPENDIX V (ABLATION STUDY)

TABLE 5. **Experimental results on in-shop clothes dataset. The experiment setup is the same as in table 2 in main body.**

Recall@Q (%)		1	10	20	40
Contrastive^{128}	R50	75.5	89.7	92.5	93.1
GML-Cont^{128}	R50	82.5	93.4	94.6	95.8
Trip-semi^{128}	R50	78.4	92.0	94.8	95.2
GML-Tri^{128}	R50	**87.2**	**97.2**	**98.3**	**98.8**
HTL^{128} [61]	BN	80.9	94.3	95.8	97.4
MS^{128}	BN	88.0	97.2	98.1	98.7
GML-MS^{128}	BN	88.7	97.7	98.3	98.8
Proxy-Anchor^{128}	BN	90.8	97.9	98.5	99.0
GML-PA^{128}	BN	**91.7**	**97.9**	**98.5**	**98.9**
A-BIER^{512} [62]	G	83.1	95.1	96.9	997.8
MS^{512}	BN	89.7	97.9	98.5	99.1
GML-MS^{512}	BN	91.1	98.1	98.7	99.2
Proxy-Anchor^{512}	BN	91.5	98.1	98.8	99.1
GML-PA^{512}	BN	**92.7**	**98.4**	**98.8**	**99.3**
HORDE^{512B}	BN	90.4	97.8	98.4	98.9
Proxy-Anchor^{512B}	BN	92.6	98.3	**98.9**	99.3
GML-PA^{512B}	BN	**93.1**	**98.5**	**98.9**	**99.3**

TABLE 6. **Performance according to minibatch size.**

Minibatch size	Recall@1 (%) (Proxy-Anchor / GML-PA)	
	CUB200-2011	Cars196
30	65.9 / **66.8** (+0.9)	84.6 / **84.8** (+0.2)
60	67.0 / **68.7** (+1.7)	86.2 / **86.3** (+0.1)
90	68.4 / **69.3** (+0.9)	86.2 / **87.4** (+1.2)
120	68.5 / **69.4** (+0.9)	86.3 / **87.1** (+0.8)
150	68.6 / **70.4** (+1.8)	86.4 / **87.5** (+1.1)
180	69.0 / **70.7** (+1.7)	86.2 / **87.4** (+1.2)
-	SOP	In-shop
30	76.0 / **76.5** (+0.5)	91.3 / **92.0** (+0.7)
60	78.0 / **79.2** (+1.2)	91.3 / **92.6** (+1.3)
90	78.5 / **79.5** (+1.0)	91.5 / **92.6** (+1.1)
120	78.9 / **80.0** (+1.1)	91.7 / **92.6** (+0.9)
150	79.1 / **80.2** (+1.1)	91.9 / **92.8** (+0.9)
300	79.3 / **80.4** (+1.1)	92.0 / **92.7** (+0.7)

Table 5 shows the performance for the in-shop clothes dataset. As in the SOP dataset, GML-PA shows an excellent R@1 performance improvement of 1.2% for the embedding vector size of 512. Since the performance has already reached the upper limit, it is difficult to expect further improvement. On the other hand, GML-Cont and GML-Tri, which have a weaker saturation in performance, show noticeable R@1 improvements of 7% and 8.8%, respectively.

Next, 6 shows the performance according to the minibatch size. Since the similarity metric is calculated using piecewise components in a minibatch, smaller minibatch size tends to show lower performance. However, GML-PA consistently shows higher performance than PA even in a relatively small minibatch size. This demonstrates that GML-PA is less sensitive to the minibatch size, which is important for extracting local information.

REFERENCES

[1] B. Alipanahi, M. Biggs, and A. Ghodsi, "Distance metric learning vs. Fisher discriminant analysis," in *Proc. 23rd Nat. Conf. Artif. Intell.*, Chicago, IL, USA, Jul. 2008, pp. 598–603.

[2] S. Balakrishnama and A. Ganapathiraju, "Linear discriminant analysis—A brief tutorial," *Inst. Signal Inf. Process.*, vol. 18, pp. 1–8, Mar. 1998.

[3] Y. Bengio, J.-F. Paiement, P. Vincent, O. Delalleau, N. Le Roux, and M. Ouimet, "Out-of-sample extensions for LLE, Isomap, MDS, Eigenmaps, and spectral clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 16, 2003, pp. 177–184.

[4] C. M. Bishop, *Pattern Recognition and Machine Learning*. Berlin, Germany: Springer-Verlag, 2006.

[5] M. P. D. Carmo, *Riemannian Geometry*. Berlin, Gemany: Springer, 2013.

[6] M. A. Carreira-Perpinán and R. Raziperchikolaei, "An ensemble diversity approach to supervised binary hashing," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 757–765.

[7] R. Chakraborty and B. C. Vemuri, "Statistics on the Stiefel manifold: Theory and applications," *Ann. Statist.*, vol. 47, no. 1, pp. 415–438, Feb. 2019.

[8] X. Chu, Y. Lin, Y. Wang, X. Wang, H. Yu, X. Gao, and Q. Tong, "Distance metric learning with joint representation diversification," in *Proc. Int. Conf. Mach. Learn.*, Nov. 2020, pp. 1962–1973.

[9] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning," in *Proc. 24th Int. Conf. Mach. Learn. (ICML)*, Corvallis, OR, USA, Jun. 2007, pp. 209–216.

[10] T.-T. Do, T. Tran, I. Reid, V. Kumar, T. Hoang, and G. Carneiro, "A theoretically sound upper bound on the triplet loss for improving the efficiency of deep distance metric learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, San Francisco, CA, USA, Jun. 2019, pp. 10404–10413.

[11] P. T. Fletcher, C. Lu, S. M. Pizer, and S. Joshi, "Principal geodesic analysis for the study of nonlinear statistics of shape," *IEEE Trans. Med. Imag.*, vol. 23, no. 8, pp. 995–1005, Aug. 2004.

[12] G. Jacob, R. Sam, H. Geoff, and S. Ruslan, "Neighbourhood components analysis," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 17, 2004, pp. 513–520.

[13] A. Gordo and D. Larlus, "Beyond instance-level image retrieval: Leveraging captions to learn a global visual representation for semantic retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 6589–6598.

[14] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New York, NY, USA, Jun. 2006, pp. 1735–1742.

[15] B. Harwood, V. Kumar B. G., G. Carneiro, I. Reid, and T. Drummond, "Smart mining for deep metric learning," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 2821–2829.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.

[17] Z. Huang, R. Wang, S. Shan, and X. Chen, "Projection metric learning on Grassmann manifold with application to video based face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 140–149.

[18] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Mach. Learn.*, Lille, France, Jun. 2015, vol. 37, pp. 448–456.

[19] P. Jacob, D. Picard, A. Histace, and E. Klein, "Metric learning with HORDE: High-order regularizer for deep embeddings," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul, South Korea, Oct. 2019, pp. 6539–6548.

[20] H. Jegou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 117–128, Jan. 2011.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, San Diego, CA, USA, May 2015, pp. 1–15.

[22] T. Kaneko, S. Fiori, and T. Tanaka, "Empirical arithmetic averaging over the compact Stiefel manifold," *IEEE Trans. Signal Process.*, vol. 61, no. 4, pp. 883–894, Feb. 2013.

[23] S. Kim, D. Kim, M. Cho, and S. Kwak, "Proxy anchor loss for deep metric learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 3238–3247.

[24] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *Proc. ICML Deep Learn. Workshop*, vol. 2, Lille, France, Jul. 2015, pp. 1–30.

[25] A. N. Kolmogorov and V. M. Tikhomirov, "ε-entropy and ε-capacity of sets in function spaces," *Uspekhi Matematicheskikh Nauk*, vol. 14, no. 2, pp. 3–86, 1959.

[26] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3D object representations for fine-grained categorization," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Sydney, NSW, Australia, Dec. 2013, pp. 554–561.

[27] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "DeepFashion: Powering robust clothes recognition and retrieval with rich annotations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 1096–1104.

[28] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Representations*, New Orleans, LA, USA, May 2019, pp. 1–19. [Online]. Available: https://openreview.net/forum?id=Bkg6RiCqY7

[29] M. Louis, B. Charlier, and S. Durrleman, "Geodesic discriminant analysis for manifold-valued data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Salt Lake City, UT, USA, Jun. 2018, pp. 332–340.

[30] C. Mao, Z. Zhong, J. Yang, C. Vondrick, and B. Ray, "Metric learning for adversarial robustness," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 478–489.

[31] T. Milbich, K. Roth, H. Bharadhwaj, S. Sinha, Y. Bengio, B. Ommer, and J. P. Cohen, "DiVA: Diverse visual feature aggregation for deep metric learning," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Aug. 2020, pp. 590–607.

[32] H. O. Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 4004–4012.

[33] Q. Qian, L. Shang, B. Sun, J. Hu, T. Tacoma, H. Li, and R. Jin, "SoftTriple loss: Deep metric learning without triplet sampling," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul, South Korea, Oct. 2019, pp. 6450–6458.

[34] K. Ridgeway and M. C. Mozer, "Learning deep disentangled embeddings with the f-statistic loss," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 185–194.

[35] K. Roth, T. Milbich, and B. Ommer, "PADS: Policy-adapted sampling for visual similarity learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6568–6577.

[36] S. Roy, M. Harandi, R. Nock, and R. Hartley, "Siamese networks: The tale of two manifolds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul, South Korea, Oct. 2019, pp. 3046–3055.

[37] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 815–823.

[38] H. Schutze, C. D. Manning, and P. Raghavan, *Introduction to Information Retrieval*, vol. 39. Cambridge, U.K.: Cambridge Univ. Press, 2008, pp. 234–265.

[39] K. Sohn, "Improved deep metric learning with multi-class N-pair loss objective," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 1857–1865.

[40] M. Sugiyama, "Dimensionality reduction of multimodal labeled data by local Fisher discriminant analysis," *J. Mach. Learn. Res.*, vol. 8, no. 1, pp. 1027–1061, Jan. 2007.

[41] E. Ustinova and V. Lempitsky, "Learning deep embeddings with histogram loss," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 4170–4178.

[42] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD birds-200-2011 dataset," Comput. Neural Syst., California Inst. Technol., Pasadena, CA, USA, Tech. Rep. CNS-TR-2011-001, 2011.

[43] X. Wang, X. Han, W. Huang, D. Dong, and M. R. Scott, "Multi-similarity loss with general pair weighting for deep metric learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 5022–5030.

[44] X. Wang, Y. Hua, E. Kodirov, G. Hu, R. Garnier, and N. M. Robertson, "Ranked list loss for deep metric learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 5207–5216.

[45] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *J. Mach. Learn. Res.*, vol. 10, no. 2, pp. 1–38, 2009.

[46] C.-Y. Wu, R. Manmatha, A. J. Smola, and P. Krahenbuhl, "Sampling matters in deep embedding learning," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 2840–2848.

[47] P. Xie, W. Wu, Y. Zhu, and E. Xing, "Orthogonality-promoting distance metric learning: Convex relaxation and theoretical analysis," in *Proc. Int. Conf. Mach. Learn.*, Stockholm, Sweden, Jul. 2018, pp. 5403–5412.

[48] H. Xuan, R. Souvenir, and R. Pless, "Deep randomized ensembles for metric learning," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Munich, Germany, Sep. 2018, pp. 723–734.

[49] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: A general framework for dimensionality reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 40–51, Jan. 2007.

[50] H. J. Ye, D. C. Zhan, X. M. Si, Y. Jiang, and Z. H. Zhou, ''What makes objects similar: A unified multi-metric learning approach,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 1235–1243.

[51] W. Yin and Z. Ma, ''High order discriminant analysis based on Riemannian optimization,'' *Knowl.-Based Syst.*, vol. 195, May 2020, Art. no. 105630.

[52] Z. Yin and Y. Shen, ''On the dimensionality of word embedding,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 887–898.

[53] T. Yuan, W. Deng, J. Tang, Y. Tang, and B. Chen, ''Signal-to-noise ratio: A robust distance metric for deep metric learning,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 4815–4824.

[54] J. Zhang, C.-G. Li, C. You, X. Qi, H. Zhang, J. Guo, and Z. Lin, ''Self-supervised convolutional subspace clustering network,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 5473–5482.

[55] M. Zhang and T. Fletcher, ''Probabilistic principal geodesic analysis,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 26, 2013, pp. 1178–1186.

[56] D. P. Kingma and M. Welling, ''Auto-encoding variational Bayes,'' in *Proc. 2nd Int. Conf. Learn. Representations*, Banff, AB, Canada, Apr. 2014, pp. 1–14.

[57] J. Li, F. Li, and S. Todorovic, ''Efficient Riemannian optimization on the Stiefel manifold via the Cayley transform,'' in *Proc. Int. Conf. Learn. Representations*, New Orleans, LA, USA, May 2019, pp. 1–16. [Online]. Available: https://openreview.net/forum?id=HJxV-ANKDH

[58] Y. Nishimori and S. Akaho, ''Learning algorithms utilizing quasi-geodesic flows on the Stiefel manifold,'' *Neurocomputing*, vol. 67, pp. 106–135, Aug. 2005.

[59] A. Bellet and A. Habrard, ''Robustness and generalization for metric learning,'' *Neurocomputing*, vol. 151, pp. 259–267, Mar. 2015.

[60] L. Van der Maaten and G. Hinton, ''Visualizing data using t-SNE,'' *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 1–27, 2008.

[61] W. Ge, ''Deep metric learning with hierarchical triplet loss,'' in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Munich, Germany, Sep. 2018, pp. 269–285.

[62] M. Opitz, G. Waltner, H. Possegger, and H. Bischof, ''Deep metric learning with BIER: Boosting independent embeddings robustly,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 276–290, Feb. 2020.

[63] D. H. Kim and B. C. Song, ''Deep metric learning with manifold class variability analysis,'' *IEEE Trans. Multimedia*, early access, Aug. 4, 2021, doi: 10.1109/TMM.2021.3101944.

[64] P. J. Rousseeuw, ''Silhouettes: A graphical aid to the interpretation and validation of cluster analysis,'' *J. Comput. Appl. Math.*, vol. 20, no. 1, pp. 53–65, 1987.

[65] W. Zheng, J. Lu, and J. Zhou, ''Deep metric learning via adaptive learnable assessment,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 2960–2969.

[66] Y. Li, T. Yao, Y. Pan, H. Chao, and T. Mei, ''Deep metric learning with density adaptivity,'' *IEEE Trans. Multimedia*, vol. 22, no. 5, pp. 1285–1297, May 2020.

[67] N. T. Bani and S. Fekri-Ershad, ''Content-based image retrieval based on combination of texture and colour information extracted in spatial and frequency domains,'' *Electron. Library*, vol. 37, no. 4, pp. 650–666, 2019.

DAE HA KIM (Associate Member, IEEE) received the B.S. degree in electronic engineering from Inha University, Incheon, South Korea, in 2017, where he is currently pursuing the Ph.D. degree in electronic engineering. His research interests include deep metric learning and video emotion recognition.

BYUNG CHEOL SONG (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 1994, 1996, and 2001, respectively. From 2001 to 2008, he was a Senior Engineer at the Digital Media Research and Development Center, Samsung Electronics Company Ltd., Suwon, South Korea. In March 2008, he joined the Department of Electronic Engineering, Inha University, Incheon, South Korea, and is currently a Professor. His research interests include image processing and computer vision.