Deep Disentangled Metric Learning

Jinhee Park[1], **Jisoo Park**[2], **Dagyeong Na**[2], **Junseok Kwon**[1,2]

[1]School of Computer Science and Engineering, Chung-Ang University, Seoul, Korea
[2]Department of Artificial Intelligence, Chung-Ang University, Seoul, Korea
{iv4084em, susiehome, dakunglove, jskwon}@cau.ac.kr

Abstract

Proxy-based metric learning has enhanced semantic similarity with class representatives and exhibited noteworthy performance in deep metric learning tasks. While these methods alleviate computational demands by learning instance-to-class relationships rather than instance-to-instance relationships, they often limit features to be class-specific, thereby degrading generalization performance for unseen class. In this paper, we introduce a novel perspective called Disentangled Deep Metric Learning (DDML), grounded in the framework of information bottleneck, which applies class-agnostic regularization to existing DML methods. Unlike conventional NormSoftmax methods, which primarily emphasize distinct class-specific features, our DDML enables a diverse feature representation by seamlessly transitioning between class-specific features with the aid of class-agnostic features. It smooths decision boundaries, allowing unseen classes to have stable semantic representations in the embedding space. To achieve this, we learn disentangled representations of both class-specific and class-agnostic features in the context of DML. Our method easily integrates into existing proxy-based algorithms, consistently delivering improved performance.

Introduction

Owing to the remarkable representation capabilities of deep neural networks, recent deep metric learning (DML) methods have achieved significant performance advancements. These advancements have facilitated various practical applications in computer vision tasks, including image retrieval (Yi et al. 2014; Schroff, Kalenichenko, and Philbin 2015; Movshovitz-Attias et al. 2017; Oh Song et al. 2016), clustering (Xing et al. 2002; Hershey et al. 2016), and classification (Snell, Swersky, and Zemel 2017; Sung et al. 2018).

The goal of DML is to learn a low-dimensional embedding space using DNNs, in which semantically similar images are positioned close to each other, while dissimilar images are placed farther apart. In other words, the distance in the embedding space plays an important role in representing semantic relationships. To achieve this goal, proxy-based methods have emerged, involving the comparison of instances to a proxy sample that represents specific classes (Zhai and Wu 2018; Kim et al. 2020). These methods reduce computational complexity by considering instance-to-class relationships rather than instance-to-instance relationships. However, learning instance-to-class relations limits the diversity of image features, making them overly class-specific. Consequently, several studies (Milbich et al. 2020; Zheng et al. 2021a,b; Gu, Ko, and Kim 2021) have advanced to address this issue through feature diversification, with the goal of improving generalization for unseen classes.

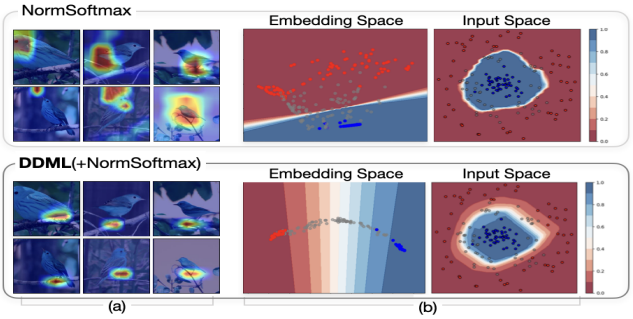

Figure 1: **Motivation of the proposed DDML.** (a) shows Grad-cam (Selvaraju et al. 2017) results for the same unseen class. While the conventional NormSoftmax (Zhai and Wu 2018) method induces inconsistent activations for unseen classes, our method demonstrates consistent activations. (b) illustrates the positioning of training samples (*i.e.*, red and blue points) and test samples (*i.e.*, gray points) in both the embedding space and the input space for 2D synthesized data. The NormSoftmax approach places testing samples in either the red or blue regions due to the narrow uncertain region (*i.e.*, white regions) caused by emphasizing only class-specific features. In contrast, our method semantically distributes the testing samples gradually within the smoothed uncertain region by leveraging class-agnostic features.

Our insight stems from observing the NormSoftmax (Zhai and Wu 2018) method, which is frequently used as a baseline in proxy-based DML. In Fig.1(a), the NormSoftmax approach induces inconsistent gradient activations for samples belonging to the same unseen class in the CUB dataset (Wah et al. 2011). This observation is evident from the experiment with 2D synthesized data. In Fig.1(b), red and blue points denote the training samples from two distinct classes in

the embedding space, whereas gray points represent testing samples belonging to the same unseen class. In conventional NormSoftmax approaches, these testing samples tend to be located in either the red or blue regions (*i.e.*, high probability regions of the two distinct classes, respectively) because the uncertain region around the decision boundary (*i.e.*, white regions) is very narrow. This observation implies that NormSoftmax primarily emphasizes distinct class-specific features. For this reason, samples from unseen classes, which lack predefined class-specific features, may be randomly scattered when every region in the embedding space is deterministically associated with a particular class. Thus, the conventional approach destabilizes the meaning of distances in regions containing samples from unseen classes.

In pursuit of a better embedding space from this perspective, we propose a novel approach called Deep Disentangled Metric Learning (DDML) within the framework of information bottleneck. DDML learns disentangled representations of both class-specific and class-agnostic features to infuse the meaning of distance into uncertain regions. Our class-agnostic features serve to bridge the gap between class-specific features, thus widening the uncertain regions in the embedding space. As shown in Fig.1(a), our approach facilitates consistent gradient activations for samples belonging to the same unseen class by introducing class-agnostic features. Similarly, in Fig.1(b), the uncertain region around the decision boundary becomes broader and smoother, leading to a seamless transition between class-specific feature regions. The gray points, which represent testing samples from the same unseen class, are gradually distributed in the expanded uncertain region around the decision boundary. This can be interpreted as testing samples similar to the red (blue) points being located close to the red (blue) region. These observations suggest that features capturing the relationships between classes enable unseen samples to be placed relatively based on their semantic representations in the embedding space, achieving the goal of DML even in uncertain region. In conclusion, the proposed DDML enhances feature diversity, improving generalization for unseen classes.

The proposed method trains both class-agnostic and class-specific features in the context of DML. The trade-off between learning class-specific and class-agnostic features can be resolved by expanding the variational information bottleneck (VIB) (Alemi et al. 2016) objectives. Class-specific features are learned to extract the essential information shared within identical classes, while class-agnostic features are learned to enhance the generalization of DML. For this, we design a class-agnostic regularization loss function composed of three loss terms: specific, agnostic, and split losses. The specific loss term aims to increase the similarity of class-specific features to their corresponding class proxy, while the agnostic loss term decreases this similarity by leveraging class-agnostic features. The split loss term is responsible for mitigate the inherent interdependence between class-agnostic and class-specific features. Fig.2 illustrates the proposed network structure and the contribution of each term in the class agnostic regularization loss function.

The contributions of the proposed method are as follows.

- We propose a novel Deep Disentangled Metric Learning method (DDML) that leverages disentangled representations of both class-specific and class-agnostic features.
- We introduce a loss function for class-agnostic regularization and propose an extended framework along with an optimization strategy that enables effective learning.
- Our loss function can be easily attached to existing DML losses and enhances generalization performance.

Related Work

Proxy-based DML Proxy-based metric learning encodes image similarity as distances in the embedding space by capturing the relationship between sample points and class representatives. A critical concern in proxy-based DML is acquiring intra-class variation. ProxyGML (Zhu et al. 2020) addressed issue by using fewer, trainable proxies per class, while RankMI (Kemertas et al. 2020) estimated a tight lower bound on joint probability divergence. Recent advancements in proxy-based DML have increasingly focused on strategies that address both intra-class and inter-class variations. DRML (Zheng et al. 2021b) captured both inter-class and intra-class distributions by extracting diverse features, thereby mitigating the issue of discarding intra-class variations. DiVA (Milbich et al. 2020) learned domain-invariant representations by disentangling latent subspaces for domain, class, and residual variations, thus improving generalization to unseen domains. DCML-MDM (Zheng et al. 2021a) employed losses across different sub-embedding compositions to enhance the diversity of the encoded features. HIST (Lim et al. 2022) used hypergraph modeling with semantic tuples to capture multilateral relations, while S2SD (Roth et al. 2021) addressed dimension bottleneck through knowledge distillation, improving generalization.

In contrast, our method emphasizes class-agnostic feature acquisition to promote feature diversification.

Regularization on DML If the embedding space is continuously arranged, it results in a smooth decision boundary, which is crucial for robust model performance. Recognizing class relationships enhances generalization by embedding test samples more accurately in the embedding space (Gu, Ko, and Kim 2021; Venkataramanan et al. 2021). Techniques such as mixup (Zhang et al. 2017; Verma et al. 2019) interpolate between samples to generate synthetic data points, thereby promoting a smooth and semantically consistent embedding space. The integrated mixup methodology (Venkataramanan et al. 2021) further extends this by performing mixup at the input, feature, and embedding levels, significantly enhancing metric learning. The embedding expansion method (Ko and Gu 2020) has been proposed based on query expansion, which synthesizes new features by combining existing features. These new features are used during training to enhance network performance, demonstrating that incorporating samples with additional meaning can help construct a more robust embedding space. Additionally, an augmentation method was implemented to address class imbalance by controlling confidence levels across classes with varying sample sizes (Liu et al. 2020). Classes with a large number of samples tend to dominate the embedding space, leading to inflated confi-

dence values, while tail classes may exhibit erroneously low confidence. To address this issue, artificial samples were introduced for the tail classes, allowing for large confidence values and long-tail data distributions.

In contrast, our method does not rely on sample augmentation. Instead, it uses a probabilistic framework to smooth uncertain regions by leveraging class-agnostic features, offering a more systematic approach than heuristic methods.

Variational Information Bottleneck The broader literature on the information bottleneck (IB) (Tishby, Pereira, and Bialek 2000) introduces a strategy of maximizing the mutual information $I(Z;Y)$ while simultaneously minimizing the mutual information $I(Z;X)$. This is intended to preserve information relevant to the label Y within the latent vector Z, while compacting the information associated with the input X. In other words, IB extracts relevant elements by maintaining only important information while discarding unnecessary information. However, due to the computational challenges of calculating mutual information, a variational information bottleneck (VIB) (Alemi et al. 2016) has been proposed. It utilized variational inference to obtain the lower bound of the IB objective. The variational approach allows the IB model to be parameterized using a neural network, known as Deep VIB. It has been successfully applied to high dimensional data across various fields, such as domain generalization (Du et al. 2020), detecting out-of-distributions (Alemi, Fischer, and Dillon 2018), unsupervised clustering (Uğur, Arvanitakis, and Zaidi 2020), dimensionality reduction (Abdelaleem, Nemenman, and Martini 2023), and multiview representation learning (Bao 2021).

We apply Deep VIB to DML and show that the tradeoff relationship between learning class-specific and class-agnostic features can be resolved by expanding the VIB objectives. Class-specific features capture essential information, while class-agnostic features enhance generalization.

Deep Disentangled Metric Learning

Assumption 1. *X is a highly complex, high-dimensional vector represented by an image, containing features of both known and unknown classes.*

Definition 1 (Domain)**.** *Given an input space $\mathcal{X}$ and a label space $\mathcal{Y}$, a domain is defined by the joint distribution $P_{X,Y}$, where the random variables $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$. The known classes form a subset of $\mathcal{Y}$, with their label space denoted by $\mathcal{Y}_k$. The unknown classes are represented by $\mathcal{Y}_u = \mathcal{Y}/\mathcal{Y}_k$.*

We propose an extended Variational Information Bottleneck (VIB) method (Alemi et al. 2016) designed to capture features of unknown classes within the bottleneck latent space Z, while learning class-specific features through an additional bottleneck latent Z^s. The Markov chain structure can be expanded from $Y(\text{Label}) \leftrightarrow X(\text{Input}) \rightarrow Z(\text{Latent vector})$ to $Y \leftrightarrow X \rightarrow Z \rightarrow Z^s$. In this framework, Z serves as a bottleneck with a VIB objective for the overall Y, encompassing both known and unknown classes, and is referred to as the *embedding bottleneck*. Moreover, Z^s, derived from Z through a probabilistic encoder, is designed with a VIB objective specifically targeting the known class Y_k, which we refer to as the *specific bottleneck*. By extending the traditional VIB objective to the open-set scenario, we propose a more robust variational information bottleneck objective as follows:

$$\max \left\{ I(Z;Y_k) + \alpha I(Z;Y_u) + \beta I(Z^s;Y_k) - \gamma I(Z^s;Z) \right\}, \tag{1}$$

where α, β and γ are hyper-parameters that controls the relative influence of the terms. In (1), for the embedding bottleneck Z, mutual information with both the known class Y_k and the unknown class Y_u is maximized (*i.e.*, $I(Z;Y_k) + \alpha I(Z;Y_u)$). For the specific bottleneck Z^s, mutual information with Y_k is maximized (*i.e.*, $\beta I(Z^s;Y_k)$), similar to the original VIB (Alemi et al. 2016). At the same time, mutual information between Z^s and the previous bottleneck Z is minimized to disentangle class-specific features from class-agnostic features (*i.e.*, $-\gamma I(Z^s;Z)$). This strategy ensures that Z^s retains specific and relevant information about Y_k, while minimizing irrelevant information.

To facilitate learning with the proposed neural network with the above objective, we derive the lower bound of (1) and solve the problem by maximizing this lower bound, thereby indirectly but efficiently optimizing the objective. Using the proposed Markov chain and the chain rule of mutual information, the lower bound of (1) can be obtained as

$$\begin{aligned}
\mathcal{L} &= I(Z;Y_k) + \alpha I(Z;Y_u) + \beta I(Z^s;Y_k) - \gamma I(Z;Z^s) \\
&\geq \int dx dy_k dz p(x) p(y_k|x) p(z|x) \log q(y_k|z) \\
&+ \alpha \int dx dy_u dz p(x) p(y_u|x) p(z|x) \log q(y_u|z) \\
&+ \beta \int dx dy_k dz dz^s p(x) p(y_k|x) p(z|x) p(z^s|z) \log q(y_k|z^s) \\
&- \gamma \int dx dy_k dz dz^s p(x) p(y_k|x) p(z|x) p(z^s|z) \log \frac{p(z^s|z)}{r(z^s)},
\end{aligned} \tag{2}$$

where x, y_k, y_u, and z are instances of random variables X, Y_k, Y_u, and Z, respectively. $q(y_k|z), q(y_u|z)$ and $q(y_k|z^s)$ are the variational approximations for $p(y_k|z), p(y_u|z)$ and $p(y_k|z^s)$ respectively, while $r(z^s)$ is the variational approximation for the marginal distribution $p(z^s)$. *Please note that the detailed derivation is found in supplementary materials*.

Optimization of (1)

We approximate (2) using training samples via probabilistic encoders. For this, $p(x, y_k)$ is approximated by the empirical distribution over the training samples, as follows.

$$p(x, y_k) = p(x)p(y_k|x) \approx \frac{1}{N} \sum_{n=1}^{N} \delta(x - x^n) \delta(y_k - y_k^n), \tag{3}$$

where x^n and y_k^n denote the n-th training data point and its corresponding label, respectively. N is the total number of training samples, and $\delta(\cdot)$ denotes the Dirac delta function.

In (2), the first, third, and fourth terms are probabilistic formulations concerning y^k and can be approximated using the empirical distribution of $p(x, y_k)$ in (3). However, the formulation of the second term involving y^u requires an additional approximation. We define $p(y_u|x)$ as

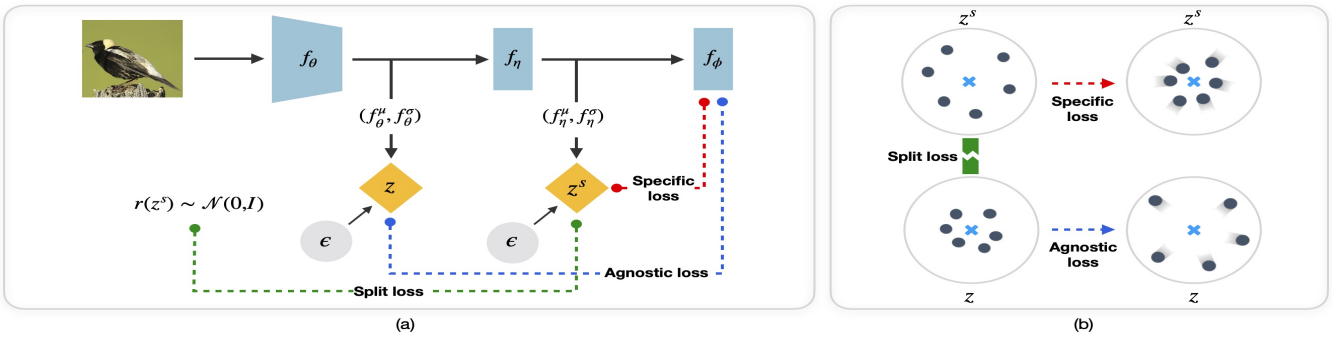

Figure 2: **Illustration of the proposed network structure and the contribution of each loss term**. (a) depicts the overall network structure. The network includes two encoders (f_θ, f_η) and a decoder (f_ϕ), producing class-agnostic (z) and class-specific (z^s) features whose likelihoods are evaluated by the decoder. The specific loss increases the likelihood of z^s, the agnostic loss reduces that of z, and the split loss mitigates their interdependence. (b) illustrates these effects in embedding space, where the black circle represents a sample and the blue cross represents the class proxy.

$p\left(E_k[p(y_k|x)] = \frac{1}{P}\right)$, which represents the likelihood that the model assigns equal probabilities $\frac{1}{P}$ to all classes y_k where P denotes the number of classes in label Y_k. This approach is both practical and reasonable, as it ensures that the network remains unbiased towards any specific class when encountering an unknown class, thereby evenly distributing the uncertainty across all possible known classes. Therefore, the joint distribution for the unknown class y_u can be approximated as follows:

$$p(x, y_u) = p(x)p(y_u|x) \approx \frac{1}{N}\sum_{n=1}^{N} \delta(x-x^n)p(y_u^n|x^n). \quad (4)$$

Then, maximizing the likelihood of $p(y_u|x)$ is equivalent to ensuring that $E_k[p(y_k|x)] = \frac{1}{P}$, which can be reframed as minimizing the negative log-likelihood as follows.

$$\mathbb{E}_{p(z|x^n)}[-\log q(y_u^n|z)]$$
$$\approx -\frac{1}{N}\sum_{n=1}^{N}\left[\int dz\, p(z|x^n)\log\left(\sum_{j=1}^{P} q(y_k^n = j|z)\,\frac{1}{P}\right)\right], \quad (5)$$

where we can exclude y_u by expressing it in terms of y_k. Thus, y^n indicates the training data point for y_k, hereafter.

Using (3) and (5), the proposed loss, $\mathcal{L}_{DDML}$, is derived by converting the maximization of the lower bound in (2) into a minimization objective by adding a negative sign:

$$\mathcal{L}_{DDML} \approx \frac{1}{N}\sum_{n=1}^{N}\Big[\underbrace{\mathbb{E}_{p(z|x^n)}[-\log q(y^n|z)]}_{\textbf{DML Loss}}$$
$$+\,\alpha\underbrace{\mathbb{E}_{p(z|x^n)}\left[-\log\sum_{j=1}^{P} q(y^n = j|z)\cdot\frac{1}{P}\right]}_{\textbf{Agnostic Loss}}$$
$$+\,\beta\underbrace{\mathbb{E}_{p(z^s|x^n)}\left[-\log q(y^n|z^s)\right]}_{\textbf{Specific Loss}} + \gamma\underbrace{KL[p(Z^s|x^n), r(Z^s)]}_{\textbf{Split Loss}}\Big], \quad (6)$$

where KL refers to Kullback-Leibler divergence.

Implementation Details

To optimize (6), we implement two stochastic encoders $p(z|x)$ and $p(z^s|z)$ and one decoder $q(y|z^+), z^+ \in \{z, z^s\}$.

- We design the encoder $p(z|x)$ as a Gaussian distribution $\mathcal{N}(z|f_\theta^\mu(x), f_\theta^\sigma(x))$, where f_θ represents a DNN with trainable parameters θ. This network outputs the mean $f_\theta^\mu(x)$ and variance $f_\theta^\sigma(x)$. The network has an output dimension of $2D$, with the first D dimensions dimensions corresponding to the mean and the remaining D dimensions corresponding to the variance.
- We design the encoder $p(z^s|z) = \mathcal{N}(z^s|f_\eta^\mu(z), f_\eta^\sigma(z))$, where f_η consists of a single fully connected layer with the shape of $D \times 2D$ using the trainable parameter η.
- We design the decoder $q(y|z^+) = Softmax(y|f_\phi(z^+))$ by adopting a straightforward logistic regression model with the trainable parameter ϕ. Here, $z^+ \in \{z, z^s\}$, $f_\phi(z^+) = Wz^+$, and W maps the D-dimensional embedding space to the logit space of y.
- To compute the KL divergence, we design $r(z^s) = \mathcal{N}(z^s|0, I)$ as a D-dimensional Gaussian distribution.

Specific loss: The third term of (6) facilitates the learning of distinct features z_s (Fig.2(b)). The input image is processed sequentially through DNNs f_θ and f_η, resulting in class-specific features f_η^μ and f_η^σ via the reparameterization trick (Fig.2(a)). The decoder f_ϕ computes the likelihood of the input image belonging to the target class by projecting the feature into the logit space.

Agnostic loss: The second term in (6) learns class-agnostic features $f_\theta^\mu, f_\theta^\sigma$ using the reparameterization trick by feeding the input image into f_θ. The agnostic features are then fed into the decoder f_ϕ to enforce a uniform probability of $\frac{1}{P}$ for all classes. Using the same decoder f_ϕ is crucial, as it has been trained with class-specific features. Following the learning of representative features through the specific loss

	CUB			CAR			SOP		
	R@1	R@2	R@4	R@1	R@2	R@4	R@1	R@10	R@100
NS(Zhai and Wu 2018)	64.65 ± 0.09	75.68 ± 0.14	84.20 ± 0.13	83.54 ± 0.09	89.67 ± 0.05	94.06 ± 0.05	78.55 ± 0.04	90.43 ± 0.05	96.02 ± 0.02
DDML(+NS)	$\mathbf{65.96 \pm 0.06}$	$\mathbf{76.58 \pm 0.03}$	$\mathbf{84.56 \pm 0.26}$	$\mathbf{84.85 \pm 0.07}$	$\mathbf{90.06 \pm 0.07}$	$\mathbf{94.16 \pm 0.04}$	$\mathbf{79.38 \pm 0.03}$	$\mathbf{90.56 \pm 0.06}$	$\mathbf{96.15 \pm 0.02}$
PA(Kim et al. 2020)	68.36 ± 0.04	78.92 ± 0.11	86.45 ± 0.65	85.99 ± 0.07	91.65 ± 0.17	95.07 ± 0.20	78.84 ± 0.18	90.70 ± 0.05	96.09 ± 0.08
DDML(+PA)	$\mathbf{69.91 \pm 0.05}$	$\mathbf{79.55 \pm 0.06}$	$\mathbf{87.24 \pm 0.15}$	$\mathbf{87.86 \pm 0.06}$	$\mathbf{92.57 \pm 0.16}$	$\mathbf{95.71 \pm 0.05}$	$\mathbf{79.87 \pm 0.04}$	$\mathbf{91.09 \pm 0.03}$	$\mathbf{98.80 \pm 0.02}$
HypViT(Ermolov et al. 2022)	85.53 ± 0.07	91.29 ± 0.08	94.75 ± 0.06	89.08 ± 0.06	93.97 ± 0.12	96.52 ± 0.18	85.86 ± 0.04	94.86 ± 0.04	98.06 ± 0.05
DDML(+HypViT)	$\mathbf{85.98 \pm 0.05}$	$\mathbf{91.53 \pm 0.03}$	$\mathbf{95.10 \pm 0.06}$	$\mathbf{89.40 \pm 0.02}$	$\mathbf{94.14 \pm 0.05}$	$\mathbf{96.73 \pm 0.03}$	$\mathbf{86.08 \pm 0.05}$	$\mathbf{95.01 \pm 0.05}$	$\mathbf{98.19 \pm 0.04}$

Table 1: **Performance evaluation of our method combined with different metric learning algorithms** in terms of Recall@k. We used NormSoftmax(NS) (Zhai and Wu 2018), ProxyAnchor(PA) (Kim et al. 2020), and HypViT (Ermolov et al. 2022), as the baseline algorithms. The proposed disentangled module based on class-agnostic regularization in (6) was plugged into the baseline networks, resulting in an increase in accuracy for each baseline network. The notation $\mu \pm \sigma$ represents the mean and standard deviation of three runs, and bold numbers indicate performance improvements by adding DDML to the baseline.

(Fig.2(b)), the agnostic loss ensures that the similarity between the agnostic feature and the representative features of all classes follows a uniform distribution.

Split loss: The fourth term in (6) minimizes the KL divergence between $p(z^s|z)$ and the prior z^s (Fig.2(a)). This enforces z to have a diminished effect on z^s, as shown in Fig.2(b). When f_η is trained using this loss, the relevance between class-agnostic and class-specific features is reduced.

Please note that our method supports regularization, and to ensure seamless integration with the existing DML loss, the first term in (6) can be formulated to apply the standard DML loss such as NormSoftmax (Zhai and Wu 2018) and Proxy Anchor (Kim et al. 2020).

Experiments

We analyzed the performance of our method by integrating our disentangled module into existing DML methods to highlight the benefits of disentanglement. We compared the proposed method with other state-of-the-art DML approaches and evaluated our method against regularization techniques. Finally, we conducted an ablation study on hyperparameters to interpret the impact of each loss term.

Experimental Settings

For experiments, we followed the protocol outlined in (Oh Song et al. 2016). To evaluate the DML methods, we utilized several benchmark datasets for metric learning: Caltech-UCSD Birds (CUB) (Wah et al. 2011), CARS196 (CAR) (Krause et al. 2013), and Stanford Online Products (SOP) (Oh Song et al. 2016). To demonstrate the applicability of the proposed method to conventional DML methods, we selected three baseline methods to which we attached $\mathcal{L}_{DDML}$: NormSoftmax (Zhai and Wu 2018), which is a fundamental method, and Proxy Anchor (Kim et al. 2020), known for its strong performance in CNN-based approaches, and the recent HypViT (Ermolov et al. 2022) based on Vision transformer (Dosovitskiy et al. 2020).

Since the proposed method is a regularization technique added to the baseline, it is essential to follow the baseline's experimental settings closely to ensure a fair performance comparison. Therefore, hyperparameters, such as the optimizer type, learning rate, weight decay parameter, embedding space dimension, and batch size, were kept consistent with those used in the baseline methods. In all three baseline methods, the structure up to the embedding space Z remained consistent with each respective method, and a linear layer was added from the embedding bottleneck Z to the specific bottleneck Z^s. For NormSoftmax and ProxyAnchor, which are proxy-based metric learning methods, the existing decoders from the embedding space to the label space were utilized. However, for HypViT, a pair-based metric learning method, the same decoder used in NormSoftmax was added. *Note that details are found in the supplementary materials.*

Performance Improvement By Our DDML

Table 1 shows that the proposed method consistently improved performance for all baseline algorithms across all datasets. We used NormSoftmax(NS) (Zhai and Wu 2018), ProxyAnchor(PA) (Kim et al. 2020), and HypViT (Ermolov et al. 2022) as the baseline algorithms. The DDML framework utilized the baseline network as the DML loss in (6), resulting in enhanced performance for each of the baseline network, with an improvement in recall of up to 1.4%. The combination of HypViT (Ermolov et al. 2022) and our method exhibited the best performance. Although the baseline methods already demonstrated high performance, the incorporation of our disentangled module in (6) yielded a further substantial enhancement in performance.

Comparison With Other Methods

Table 2 compares our method with other state-of-the-art methods, with results organized by the backbone network. When BN-Inception was used as the backbone, the proposed method, which applies DDML to ProxyAnchor, achieved the best performance across all recall values on the SOP dataset. Although it might not have the highest performance across all recall values on the CUB and CAR datasets, it did achieve the best R@1, which is the most critical metric. Similarly, when ResNet was used as the backbone, the proposed ProxyAnchor with DDML method also delivered the best R@1 performance across all datasets, while also demonstrating strong performance in other recall metrics. When ViT was used as the backbone, the proposed HypViT (Ermolov et al. 2022) with the DDML method achieved the best performance across all datasets, with the exception of the R@1 score on the CUB dataset. It is noteworthy that VPTSP-G, which achieved the best performance on the CUB dataset in

		CUB				CAR				SOP			
	R@k	1	2	4	8	1	2	4	8	1	10	100	1000
BN-Inception	NormSoftmax(Zhai and Wu 2018)†	55.3	67.0	77.6	85.4	75.2	84.7	90.4	94.2	69.0	84.5	93.1	-
	MS(Wang et al. 2019)†	65.7	77.0	86.3	91.2	84.1	90.4	94.0	96.5	78.2	90.5	96.0	98.7
	SoftTriple(Qian et al. 2019)†	65.4	76.4	84.5	90.4	84.5	90.7	94.5	96.9	78.3	90.3	95.9	-
	ProxyAnchor(Kim et al. 2020)†	68.4	79.2	86.8	91.6	86.1	91.7	95.0	97.3	80.3	91.4	96.4	98.7
	ProxyGML(Zhu et al. 2020)†	66.6	77.6	86.4	-	85.5	91.8	95.6	-	78.0	90.6	96.2	-
	DRML(Zheng et al. 2021b)†	68.7	78.6	86.3	91.6	86.9	92.1	95.2	97.4	**79.9**	90.7	96.1	-
	CircleLoss(Sun et al. 2020)†	66.7	77.4	86.2	91.2	83.4	89.8	94.1	96.5	78.3	90.5	96.1	98.6
	DAM(Xu et al. 2021)†	69.1	79.8	87.2	91.8	86.9	92.1	95.3	**97.9**	-	-	-	-
	PADs(Roth, Milbich, and Ommer 2020)†	66.6	77.2	85.6	-	81.7	88.3	93.0	-	-	-	-	-
	HIST(Lim et al. 2022)†	69.7	**80.0**	**87.3**	-	87.4	92.5	95.4	-	79.6	91.0	96.2	-
	DDML(+ProxyAnchor(Kim et al. 2020))†	**70.0**	79.6	87.2	**92.0**	**87.8**	**92.6**	**95.7**	97.4	**79.9**	**91.1**	**96.4**	**98.8**
ResNet50	NormSoftmax(Zhai and Wu 2018)†	61.3	73.9	83.5	90.0	84.2	90.4	94.4	96.9	78.2	90.6	96.2	-
	Div&Conq(Sanakoyeu et al. 2019)*	65.9	76.6	84.4	90.6	84.6	90.7	94.1	96.5	75.9	88.4	94.9	98.1
	MIC(Roth, Brattoli, and Ommer 2019)*	66.1	76.8	85.6	-	82.6	89.1	93.2	-	77.2	89.4	95.6	-
	PADs(Roth, Milbich, and Ommer 2020)*	67.3	78.0	85.9	-	83.5	89.7	93.8	-	76.5	89.0	95.4	-
	RankMI(Kemertas et al. 2020)*	66.7	77.2	85.1	91.0	83.3	89.8	93.8	96.5	74.3	87.9	94.9	98.3
	EPSHN(Xuan, Stylianou, and Pless 2020)†	64.9	75.3	83.5	-	82.7	89.3	93.0	-	78.3	90.7	96.3	-
	DiVA(Milbich et al. 2020)†	69.2	79.3	-	-	87.6	92.9	-	-	79.6	91.2	-	-
	ProxyAnchor(Kim et al. 2020)†	69.7	80.0	87.0	92.4	87.7	92.9	95.8	97.9	-	-	-	-
	DCML-MDW(Zheng et al. 2021b)†	68.4	77.9	86.1	91.7	85.2	91.8	96.0	98.0	79.8	90.8	95.8	-
	IBC(Seidenschwarz, Elezi, and Leal-Taixé 2021)†	70.3	80.3	87.6	**92.7**	88.1	93.3	96.2	**98.2**	81.4	91.3	95.9	-
	HIST(Lim et al. 2022)†	71.4	81.1	**88.1**	-	**89.6**	**93.9**	**96.4**	-	81.4	92.0	96.7	-
	HIER(Kim, Jeong, and Kwak 2023)†	70.1	79.4	86.9	-	88.2	93.0	95.6	-	80.2	91.5	96.6	-
	DDML(+cosFace(Wang et al. 2018))†	66.6	76.7	85.0	90.6	84.6	90.2	94.4	96.6	79.1	90.6	95.9	98.5
	DDML(+SphereFace(Liu et al. 2017))†	65.7	76.8	84.9	90.5	84.8	90.5	94.2	97.2	78.9	90.7	95.9	98.5
	DDML(+ArcFace(Deng et al. 2019))†	66.7	77.4	85.0	90.7	85.0	90.8	94.8	96.8	79.1	90.6	95.7	98.6
	DDML(+ProxyAnchor(Kim et al. 2020))†	**71.5**	**81.2**	88.0	92.6	**89.6**	**93.9**	95.9	97.6	**81.5**	91.7	96.6	**99.0**
ViT	HypViT(Ermolov et al. 2022)§	85.6	91.4	94.8	96.7	89.2	94.1	96.7	98.1	85.9	94.9	98.1	**99.5**
	HIER(Kim, Jeong, and Kwak 2023)§	85.7	91.3	94.4	-	88.3	93.2	96.1	-	86.1	95.0	98.0	-
	VPTSP-G(Ren et al. 2024)§	**86.6**	**91.7**	94.8	-	87.7	93.3	96.1	-	84.4	93.6	97.3	-
	DFML-PA(Wang et al. 2023)§	79.1	86.8	-	-	**89.5**	93.9	-	-	84.2	93.8	-	-
	DDML(+HypViT(Ermolov et al. 2022))§	86.0	**91.7**	**95.2**	**96.8**	**89.5**	**94.2**	**96.8**	**98.2**	**86.1**	**95.1**	**98.2**	**99.5**

Table 2: **Quantitative comparison** on the CUB, CAR, and SOP datasets in terms of Recall@k. The symbols $*$, § and † represent embedding space sizes of 128, 384 and 512, respectively. The best performance for each backbone was boldfaced.

BB	BN-Inception			ResNet50			ViT	
Base	ProxyAnchor			ProxyAnchor			HypViT	
Reg.	-	*PS*	***DDML***	-	*Metrix*	***DDML***	-	***DDML***
CUB	68.4	69.2	**70.0**	69.7	70.4	**71.5**	85.6	**86.0**
CAR	86.1	86.9	**87.8**	87.7	88.5	**89.6**	89.2	**89.4**
SOP	79.1	79.8	**79.9**	-	81.3	**81.5**	85.9	**86.1**

Table 3: Quantitative comparison with other **regularization techniques** in terms of Recall@1. BB refers to the backbone, Base indicates the type of baseline algorithm, and *Reg.* represents the regularization method.

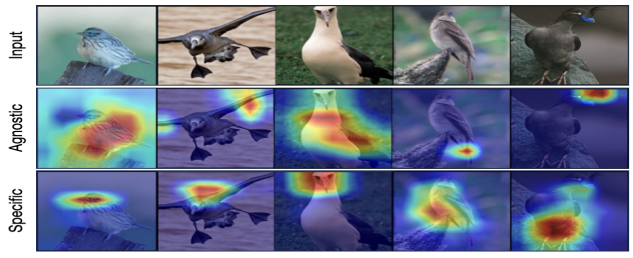

Figure 3: **Visualization on Z and Z^s.** First row: input images, Second row: GradCam visualization for the agnostic loss, Third row: GradCam visualization for the specific loss. We can observe a complementary relationship between agnostic and specific losses.

terms of R@1, leveraged Visual Prompts (VPT) as an additional component during training. Although the baseline HypViT showed superior performance attributed to a larger pre-training set, the performance was further enhanced by incorporating our DDML method. *More experiment results (e.g., MLRC evaluation, proxy-to-proxy affinity matrix, logit visualization and experiments with class-imbalanced data) are included in supplementary materials.*

Table 3 provides a quantitative comparison with recent DML regularization methods. PS (Gu, Ko, and Kim 2021) employed a mixup strategy in the embedding space to augment synthetic samples, thereby enhancing generalization performance. Metrix (Venkataramanan et al. 2021) extended this approach with an integrated mixup at the input, embedding, and feature levels. Both studies presented experimental

Agnostic loss	Specific loss	Split loss	CUB	CAR	SOP
-	-	-	64.7	83.5	78.6
✓	-	-	64.9	84.2	78.8
-	✓	-	64.8	84.0	78.6
-	-	✓	64.5	83.1	78.6
✓	✓	-	65.1	84.5	79.1
✓	-	✓	64.7	83.4	78.0
-	✓	✓	65.0	84.2	78.7
✓	✓	✓	**66.2**	**84.9**	**79.4**

Table 4: **Ablation study on the proposed losses**. Numbers indicate the Recall@1 values.

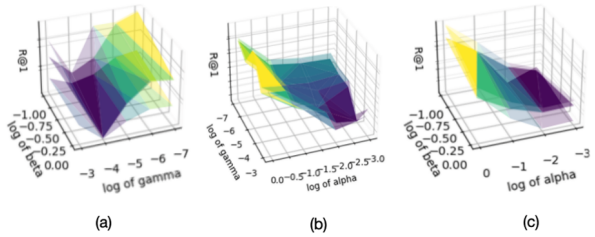

Figure 4: **Relationship between the three hyperparameters and performance.** R@1 performance with ProxyAnchor: (a) varying β and γ while fixing α, (b) varying α and γ while fixing β, and (c) varying α and β while fixing γ. These results indicate a consistent relationship between each hyperparameter and performance. *Expanded figures are in the supplementary materials.*

results using ProxyAnchor as the baseline. However, PS was evaluated solely with a BN-Inception backbone, and Metrix was evaluated only with a ResNet50 backbone. In contrast, our proposed method, DDML(+PA), provides a comprehensive comparison by demonstrating performance across both BN-Inception and ResNet50 backbones. In the CUB and CAR datasets, DDML(+PA) showed the highest performance improvement, achieving more than a 1.5 % gain over the baseline method, at least a 0.8 % gain over PS, and over a 1.0 % gain compared to Metrix. For the SOP dataset, while DDML(+PA) significantly improved upon the baseline, its performance was comparable to that of other regularization methods. Although the performance of the probabilistic model may vary with the number of classes and images per class, the consistent improvement over the baseline and the comparable performance with other regularization methods highlight the effectiveness of the proposed DDML as a regularization approach.

Analysis on Proposed Losses

Fig.3 shows the activation map obtained by GradCam (Selvaraju et al. 2017) for both the agnostic loss and the specific loss in $\mathcal{L}_{DDML}$ of (6) for CUB training data. For all input images, the GradCam was activated for different regions for each loss. As shown in the last row of Fig.3, the GradCam visualization for the specific loss exhibited activation either on the facial region or the entirety of the bird, both of which could be crucial attributes associated with birds. In contrast, the GradCam visualization for the agnostic loss exhibits different activation patterns, as shown in the middle row of Fig.3. Especially concerning the rightmost bird, the specific loss tends to activate the beak and legs, whereas the agnostic loss focuses on the head feathers. This observation indicates a complementary relationship between the agnostic loss and the specific loss.

Table 4 compares our disentangled module in (6) across various combinations of agnostic, specific, and split losses on the CUB, CAR, and SOP datasets. Using the proxy-based algorithm NormSoftmax to compute the DML loss ($\mathcal{L}DML$), we consistently applied $\mathcal{L}DML$ while controlling the additional losses by setting hyperparameters α, β, γ to specific values ($\alpha = 1$, $\beta = 1$, $\gamma = 1e-7$) or to zero to disable them. The baseline performance (Row 1) represents NormSoftmax without additional losses. Rows 2 to 4 indicate that applying any single loss term improves performance, with agnostic and specific losses increasing performance by approximately 0.5% on the CAR dataset. Rows 5 to 7 demonstrate that combining agnostic and specific losses yields a 0.4% to 1.0% improvement across all datasets, outperforming each loss individually. The final row shows that incorporating all three losses achieves the highest performance, with improvements of 1.5%, 1.4%, and 0.8% on the CUB, CAR, and SOP datasets, respectively. Adding the split loss further enhances performance by 1.1%, 0.4%, and 0.3%, demonstrating its effectiveness in stabilizing learning by reducing dependency between class-agnostic and class-specific features. In conclusion, all three losses are essential for improving DML performance.

We investigated the effects of three hyperparameters—agnostic loss (α: $1e-7, 1e-6, 1e-5, 1e-4, 1e-3$), specific loss ($\beta$: 0.1, 1), and split loss (γ: 0.001, 0.01, 0.1, 1)—on R@1 performance using the CUB dataset with ProxyAnchor as the baseline and BN-Inception as the backbone. Fig.4 illustrates that R@1 performance improves as γ decreases for fixed α values (a) and as α increases for fixed β values (b). Additionally, higher α values consistently enhance performance regardless of γ (c). These results demonstrate a stable and consistent relationship between each hyperparameter and performance, highlighting the robustness of the proposed method.

Conclusion

We propose a DML method for enhancing generalization performance by leveraging the disentangled representations of both class-specific and class-agnostic features. Our motivation stems from the limitations of NormSoftmax methods, which often exhibit inconsistent gradient activations for unseen classes. We introduce class-agnostic regularization that can smooth uncertain regions and induce stable semantic representations for unseen classes in the embedding space. Furthermore, we have optimized our method by integrating the VIB objectives into DML. Experimental results validate the effectiveness of the proposed optimization method for class-agnostic regularization, emphasizing its seamless integration with existing DML algorithms and consistent performance improvements.

Acknowledgments

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (2021-0-01341, Artificial Intelligence Graduate School Program (Chung-Ang university)).

References

Abdelaleem, E.; Nemenman, I.; and Martini, K. M. 2023. Deep Variational Multivariate Information Bottleneck– A Framework for Variational Losses. *arXiv preprint arXiv:2310.03311*.

Alemi, A. A.; Fischer, I.; and Dillon, J. V. 2018. Uncertainty in the variational information bottleneck. *arXiv preprint arXiv:1807.00906*.

Alemi, A. A.; Fischer, I.; Dillon, J. V.; and Murphy, K. 2016. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*.

Bao, F. 2021. Disentangled variational information bottleneck for multiview representation learning. In *CICAI*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. ImageNet: A large-scale hierarchical image database. In *CVPR*.

Deng, J.; Guo, J.; Xue, N.; and Zafeiriou, S. 2019. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Du, Y.; Xu, J.; Xiong, H.; Qiu, Q.; Zhen, X.; Snoek, C. G.; and Shao, L. 2020. Learning to learn with variational information bottleneck for domain generalization. In *ECCV*.

Ermolov, A.; Mirvakhabova, L.; Khrulkov, V.; Sebe, N.; and Oseledets, I. 2022. Hyperbolic vision transformers: Combining improvements in metric learning. In *CVPR*.

Gu, G.; Ko, B.; and Kim, H.-G. 2021. Proxy Synthesis: Learning with Synthetic Classes for Deep Metric Learning. In *AAAI*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.

Hershey, J. R.; Chen, Z.; Le Roux, J.; and Watanabe, S. 2016. Deep clustering: Discriminative embeddings for segmentation and separation. In *ICASSP*.

Ioffe, S.; and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*.

Kemertas, M.; Pishdad, L.; Derpanis, K. G.; and Fazly, A. 2020. Rankmi: A mutual information maximizing ranking loss. In *CVPR*.

Kim, S.; Jeong, B.; and Kwak, S. 2023. HIER: Metric Learning Beyond Class Labels via Hierarchical Regularization. In *CVPR*.

Kim, S.; Kim, D.; Cho, M.; and Kwak, S. 2020. Proxy anchor loss for deep metric learning. In *CVPR*.

Ko, B.; and Gu, G. 2020. Embedding expansion: Augmentation in embedding space for deep metric learning. In *CVPR*.

Krause, J.; Stark, M.; Deng, J.; and Fei-Fei, L. 2013. 3d object representations for fine-grained categorization. In *ICCV Workshops*.

Lim, J.; Yun, S.; Park, S.; and Choi, J. Y. 2022. Hypergraph-induced semantic tuplet loss for deep metric learning. In *CVPR*.

Liu, J.; Sun, Y.; Han, C.; Dou, Z.; and Li, W. 2020. Deep representation learning on long-tailed data: A learnable embedding augmentation perspective. In *CVPR*.

Liu, W.; Wen, Y.; Yu, Z.; Li, M.; Raj, B.; and Song, L. 2017. SphereFace: Deep Hypersphere Embedding for Face Recognition. In *CVPR*.

Milbich, T.; Roth, K.; Bharadhwaj, H.; Sinha, S.; Bengio, Y.; Ommer, B.; and Cohen, J. P. 2020. Diva: Diverse visual feature aggregation for deep metric learning. In *ECCV*.

Movshovitz-Attias, Y.; Toshev, A.; Leung, T. K.; Ioffe, S.; and Singh, S. 2017. No fuss distance metric learning using proxies. In *ICCV*.

Oh Song, H.; Xiang, Y.; Jegelka, S.; and Savarese, S. 2016. Deep metric learning via lifted structured feature embedding. In *CVPR*.

Qian, Q.; Shang, L.; Sun, B.; Hu, J.; Li, H.; and Jin, R. 2019. Softtriple loss: Deep metric learning without triplet sampling. In *ICCV*.

Rangwani, H.; Aithal, S. K.; Mishra, M.; et al. 2022. Escaping saddle points for effective generalization on class-imbalanced data. In *NeurIPS*.

Ren, L.; Chen, C.; Wang, L.; and Hua, K. 2024. Learning Semantic Proxies from Visual Prompts for Parameter-Efficient Fine-Tuning in Deep Metric Learning. *arXiv preprint arXiv:2402.02340*.

Roth, K.; Brattoli, B.; and Ommer, B. 2019. Mic: Mining interclass characteristics for improved metric learning. In *ICCV*.

Roth, K.; Milbich, T.; and Ommer, B. 2020. Pads: Policy-adapted sampling for visual similarity learning. In *CVPR*.

Roth, K.; Milbich, T.; Ommer, B.; Cohen, J. P.; and Ghassemi, M. 2021. Simultaneous similarity-based self-distillation for deep metric learning. In *ICML*.

Sanakoyeu, A.; Tschernezki, V.; Buchler, U.; and Ommer, B. 2019. Divide and conquer the embedding space for metric learning. In *CVPR*.

Schroff, F.; Kalenichenko, D.; and Philbin, J. 2015. Facenet: A unified embedding for face recognition and clustering. In *CVPR*.

Seidenschwarz, J. D.; Elezi, I.; and Leal-Taixé, L. 2021. Learning intra-batch connections for deep metric learning. In *ICML*.

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*.

Snell, J.; Swersky, K.; and Zemel, R. S. 2017. Prototypical networks for few-shot learning. *arXiv preprin arXiv:1703.05175t*.

Sun, Y.; Cheng, C.; Zhang, Y.; Zhang, C.; Zheng, L.; Wang, Z.; and Wei, Y. 2020. Circle loss: A unified perspective of pair similarity optimization. In *CVPR*.

Sung, F.; Yang, Y.; Zhang, L.; Xiang, T.; Torr, P. H.; and Hospedales, T. M. 2018. Learning to compare: Relation network for few-shot learning. In *CVPR*.

Tishby, N.; Pereira, F. C.; and Bialek, W. 2000. The information bottleneck method. *arXiv preprint arXiv:physics/0004057*.

Uğur, Y.; Arvanitakis, G.; and Zaidi, A. 2020. Variational information bottleneck for unsupervised clustering: Deep gaussian mixture embedding. *Entropy*, 22(2): 213.

Venkataramanan, S.; Psomas, B.; Avrithis, Y.; Kijak, E.; Amsaleg, L.; and Karantzalos, K. 2021. It Takes Two to Tango: Mixup for Deep Metric Learning. *arXiv preprint arXiv:2106.04990*.

Verma, V.; Lamb, A.; Beckham, C.; Najafi, A.; Mitliagkas, I.; Courville, A.; Lopez-Paz, D.; and Bengio, Y. 2019. Manifold Mixup: Better Representations by Interpolating Hidden States. *arXiv preprint arXiv:1806.05236*.

Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie, S. 2011. The caltech-ucsd birds-200-2011 dataset.

Wang, C.; Zheng, W.; Li, J.; Zhou, J.; and Lu, J. 2023. Deep factorized metric learning. In *CVPR*.

Wang, H.; Wang, Y.; Zhou, Z.; Ji, X.; Gong, D.; Zhou, J.; Li, Z.; and Liu, W. 2018. Cosface: Large margin cosine loss for deep face recognition. In *CVPR*.

Wang, X.; Han, X.; Huang, W.; Dong, D.; and Scott, M. R. 2019. Multi-similarity loss with general pair weighting for deep metric learning. In *CVPR*.

Xing, E.; Jordan, M.; Russell, S. J.; and Ng, A. 2002. Distance metric learning with application to clustering with side-information. In *NIPS*.

Xu, F.; Wang, M.; Zhang, W.; Cheng, Y.; and Chu, W. 2021. Discrimination-aware mechanism for fine-grained representation learning. In *CVPR*.

Xuan, H.; Stylianou, A.; and Pless, R. 2020. Improved embeddings with easy positive triplet mining. In *WACV*.

Yi, D.; Lei, Z.; Liao, S.; and Li, S. Z. 2014. Deep metric learning for person re-identification. In *ICPR*.

Zhai, A.; and Wu, H.-Y. 2018. Classification is a strong baseline for deep metric learning. *arXiv preprint arXiv:1811.12649*.

Zhang, H.; Cissé, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2017. mixup: Beyond Empirical Risk Minimization. *arXiv preprint arXiv:1710.09412*.

Zheng, W.; Wang, C.; Lu, J.; and Zhou, J. 2021a. Deep Compositional Metric Learning. In *CVPR*.

Zheng, W.; Zhang, B.; Lu, J.; and Zhou, J. 2021b. Deep relational metric learning. In *ICCV*.

Zhu, Y.; Yang, M.; Deng, C.; and Liu, W. 2020. Fewer is more: A deep graph metric learning perspective using fewer proxies. In *NeurIPS*.