Introspective Deep Metric Learning

Chengkun Wang, Wenzhao Zheng, *Student Member, IEEE*, Zheng Zhu, *Member, IEEE*, Jie Zhou, *Senior Member, IEEE*, and Jiwen Lu, *Fellow, IEEE*

***Abstract*—This paper proposes an introspective deep metric learning (IDML) framework for uncertainty-aware comparisons of images. Conventional deep metric learning methods focus on learning a discriminative embedding to describe the semantic features of images, which ignore the existence of uncertainty in each image resulting from noise or semantic ambiguity. Training without awareness of these uncertainties causes the model to overfit the annotated labels during training and produce overconfident judgments during inference. Motivated by this, we argue that a good similarity model should consider the semantic discrepancies with awareness of the uncertainty to better deal with ambiguous images for more robust training. To achieve this, we propose to represent an image using not only a semantic embedding but also an accompanying uncertainty embedding, which describes the semantic characteristics and ambiguity of an image, respectively. We further propose an introspective similarity metric to make similarity judgments between images considering both their semantic differences and ambiguities. The gradient analysis of the proposed metric shows that it enables the model to learn at an adaptive and slower pace to deal with the uncertainty during training. Our framework attains state-of-the-art performance on the widely used CUB-200-2011, Cars196, and Stanford Online Products datasets for image retrieval. We further evaluate our framework for image classification on the ImageNet-1 K, CIFAR-10, and CIFAR-100 datasets, which shows that equipping existing data mixing methods with the proposed introspective metric consistently achieves better results (e.g., +0.44% for CutMix on ImageNet-1 K).**

***Index Terms*—Deep metric learning, representation learning, uncertainty-aware similarity judgments.**

I. Introduction

LEARNING an effective metric to measure the similarity between data is a long-standing problem in computer vision, which serves as a fundamental step in various downstream tasks, such as face recognition [21], [60], [89], image retrieval [1], [65], [102] and image classification [10], [62]. The general objective of metric learning is to reduce the distances between positive pairs and enlarge the distances between negative pairs, which has recently powered the rapid developments for both supervised learning [83], [104] and unsupervised learning [7], [24].

Generally, deep metric learning (DML) employs deep neural networks [25], [63], [69] to map an image to a discriminative embedding space [98]. Most methods represent images using a deterministic embedding which only describes the characteristic features [60], [85]. Nevertheless, when asked to classify a certain image, humans are able to additionally provide the corresponding confidence as well as the semantic features of the image since an image might be ambiguous. Motivated by this, researchers have proposed a variety of probabilistic embedding methods using distributions to model images in the embedding space [9], [50], [66], [96]. They typically use KL-divergence [26] or Monte-Carlo-sampling-based [50] distances to measure the similarity between images. They regard the variance of the distributions as the uncertainty measure of images, yet they still provide a confident judgment of similarity regardless of the uncertainty. Specifically, though the variance affects the distribution discrepancy, a larger variance of an image does not necessarily blur its differences from other images. Subsequently, existing probabilistic embedding methods only output an uncertainty score but the training process is still unaware of the uncertainty. However, we argue that the model should focus more on certain samples and put less weight on the uncertain ones. This resembles humans who tend to ignore uncertain and uncredible information and occlude it from their learning process (e.g., people only believe what they want to believe), which enables humans to selectively and stably learn useful knowledge. Therefore, given a highly ambiguous image (e.g., an extremely blurred image), we think it is more reasonable to weaken the semantic discrepancies and consider it similar to other images since it literally could be anything.

In this paper, we propose an introspective similarity metric to achieve this and further present an introspective deep metric learning (IDML) framework for both image retrieval and classification. Different from existing methods, we represent an image using a semantic embedding to capture the semantic characteristics and further accompany it with an uncertainty embedding to model the uncertainty. An introspective similarity metric (ISM) then takes as input both embeddings and outputs an uncertainty-aware similarity score, which softens the semantic discrepancies by the degree of uncertainty. Different from the conventional metric, the proposed introspective metric deems a pair of images similar if they are either semantically similar or ambiguous to judge. It provides more flexibility to the training process to avoid the harmful influence of inaccurately labeled data, as illustrated in Fig. 1. The proposed introspective similarity metric

Manuscript received 4 September 2022; revised 28 June 2023; accepted 31 August 2023. Date of publication 5 September 2023; date of current version 6 March 2024. This work was supported in part by the National Natural Science Foundation of China under Grants 62336004, 62321005, and 62125603. Recommended for acceptance by J. Verbeek. *(Chengkun Wang and Wenzhao Zheng contribute equally to this work.) (Corresponding author: Jiwen Lu.)*

Chengkun Wang, Wenzhao Zheng, Jie Zhou, and Jiwen Lu are with the Beijing National Research Center for Information Science and Technology (BNRist), Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: wck20@mails.tsinghua.edu.cn; zhengwz18@mails.tsinghua.edu.cn; jzhou@tsinghua.edu.cn; lujiwen@tsinghua.edu.cn).

Zheng Zhu is with PhiGent Robotics, Beijing 100084, China (e-mail: zhengzhu@ieee.org).

Digital Object Identifier 10.1109/TPAMI.2023.3312311

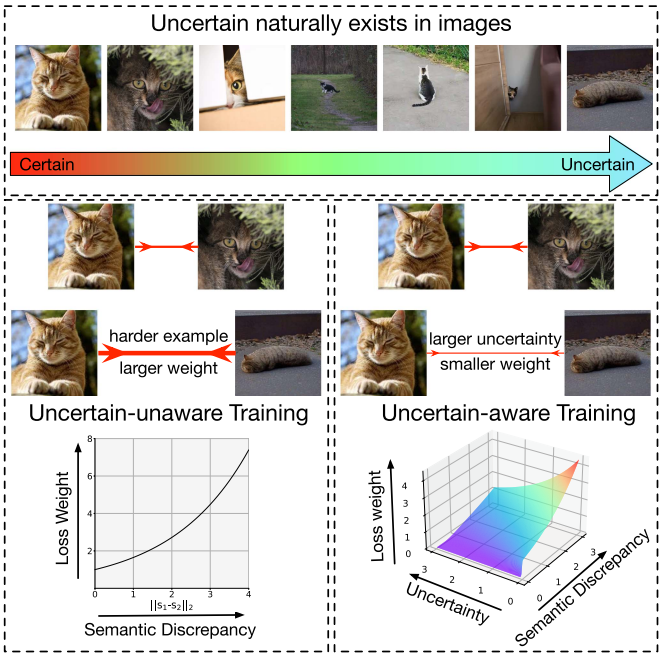


Fig. 1. Motivation of the proposed IDML framework. Uncertainty naturally exists in images due to occlusion, scale, pose, low resolution, or semantic ambiguity. Most existing deep metric learning methods only consider the semantic distance of an image pair and determine its loss weight solely based on the semantic discrepancy. However, we argue that the model should focus more on certain samples and put less weight on the more uncertain ones. Our IDML achieves this by using both a semantic embedding and an uncertainty embedding to describe an image and employing an introspective similarity metric to compute an uncertainty-aware distance.

can alternatively enlarge the uncertainty level instead of rigidly enforcing the semantic constraint for an uncertain image. We perform a gradient analysis of the optimization process equipped with the proposed introspective similarity metric, which shows that IDML enables the model to learn at an uncertainty-aware pace. The overall framework of the proposed IDML can be trained efficiently in an end-to-end manner and generally applied to existing deep metric learning and classification methods. Furthermore, we argue that the data uncertainty issue widely exists even for the general deep representation learning due to the commonly used data augmentation techniques. We conduct experiments with different data augmentations including resizing, blurring, occlusion, and data mixing [8], [74], [97] and observe consistent improvements for our framework. We perform extensive experiments on the CUB-200-2011, Cars196, and Stanford Online Products datasets for image retrieval, which shows that our framework generally improves the performance of existing deep metric learning methods by a large margin and attains state-of-the-art results. We also evaluate our framework for the image classification task on the widely used ImageNet-1 K, CIFAR-10, and CIFAR-100 datasets. Experimental results show that our introspective similarity metric further boosts existing data mixing methods including Mixup and CutMix [95], [97] (e.g., +0.44% for CutMix on ImageNet-1 K). We additionally provide an in-depth analysis of our framework including an ablation study of different components, effects of different hyperparameters, and qualitative visualizations.

We summarize the key contributions as follows:

1) We propose an uncertainty-aware representation of images consisting of a semantic embedding and an uncertainty embedding. We further propose an introspective similarity metric to measure the similarity considering both semantic discrepancies and uncertainty levels.
2) We propose an end-to-end introspective deep metric learning framework which can be generally applied to various deep metric learning and classification methods. Our framework enables the model to learn at an uncertainty-adaptive pace.
3) We conduct extensive experiments on both image retrieval and image classification tasks and observe the state-of-the-art performance of the proposed framework. We also provide an in-depth analysis of the working mechanism of our framework quantitively and qualitatively.

II. Related Work

A. Deep Metric Learning

Deep metric learning aims to construct an effective embedding space to measure the similarity between images, where the general objective is to decrease intraclass distances and increase interclass distances. Most existing methods [2], [12], [80], [91] employ a discriminative loss to learn the image embeddings. Instance-based methods directly restrict the distances between sample embeddings [29], [65], [79], [82]. For example, the commonly used contrastive loss [29] pulls embeddings from the same class as close as possible while maintaining a fixed margin between embeddings from different classes. Wang et al. [82] further formulated three types of similarities between embeddings and proposed a multi-similarity loss to restrict them. Proxy-based methods represent each class using proxy embeddings and instead constrain the similarities between sample embeddings and proxy embeddings [10], [36], [45], [78]. The ProxyNCA loss [45] generates a proxy for each class and simultaneously updates the proxies and sample embeddings. Kim et al. [36] further proposed to constrain both data-to-data relations and data-to-proxy relations. Classification-based losses (e.g., CosFace [78] and ArcFace [10]) can also be regarded as proxy-based loss, where the class proxies are implicitly learned in the softmax classifier.

In addition to the design of loss functions, various methods have explored effective sampling strategies for better training [15], [23], [60], [85], [94], [95], [97]. For example, hard negative mining [23], [60], [94] selects challenging negative samples for more efficient learning of the metric. To further alleviate the lack of informative training samples, recent works [15], [101] proposed to generate synthetic samples for training. Also, a variety of data augmentation methods improve the performance by mixing original images for better generalization [95], [97]. The aforementioned methods employ synthetic images for training, which can be semantically ambiguous. We design an introspective similarity metric to consider the uncertainty and further incorporate them to make similarity judgments.

B. Contrastive Learning

Similar to metric learning, contrastive learning also adopts a discriminative objective to increase the similarities between positive pairs and decrease the similarities between negative pairs. Despite the high resemblance of their working mechanisms, the community usually uses contrastive learning to denote methods that target self-supervised learning [3], [5], [6], [7], [20], [24]. They usually regard different augmentations of the same images as positive pairs and instances of different samples as negative pairs. SimCLR [6] adopt three simple augmentations to create positive pairs and use a normalized temperature-scaled cross-entropy loss [64] as the learning objective. MoCo [24] exploits a momentum-updated memory queue to store past embeddings as negative samples, which facilitates learning with a small batch size. SwAV [3] and NNCLR [16] restrict the similarities between samples and clusters or neighboring samples similar to proxy-based deep metric learning methods. An important reason why contrastive learning attracts increasing attention in recent years is its ability to learn meaningful representations without ground truth labels for various tasks [28], [41], [42], [77], [86], [87]. DetCo [86] additionally considers the relations between global images and local patches to exploit more fine-grained information for object detection. PointContrast [87] contrasts the match/mismatch points between different views of the same point clouds to learn useful 3D representations.

In addition to self-supervised learning, recent methods also reveal the potential of contrastive learning or metric learning for the pretraining of deep networks [17], [34], [59], [99]. Khosla et al. [34] proposed a new supervised contrastive loss which generalizes to an arbitrary number of positive samples. Feng et al. [17] only constrained the relations between a sample and its K-nearest neighbor to allow interclass distributions for better generalization. Pretraining using a contrastive objective has been shown to demonstrate better generalization performance than pretraining with the conventional classification-based objective [59]. Orthogonally, the proposed IDML framework can be generally applied to training with both contrastive or classification objectives. Our framework further considers the uncertainty of samples during training and implicitly achieves uncertainty-aware training with negligible additional training cost.

C. Uncertainty Modeling

Uncertainty modeling is widely adopted in natural language processing to model the inherent hierarchies of words [47], [48], [75]. Vilnis et al. [75] proposed the Gaussian formation in word embedding and Nguyen et al. [48] presented a mixture model to learn multi-sense word embeddings. Computer vision has also benefited from uncertainty modeling due to the natural uncertainty in images caused by factors such as occlusion and blur [33], [61]. Various methods have attempted to model the uncertainty for better robustness and generalization in face recognition [4], [62], point cloud segmentation [96], and age estimation [40].

A prevailing method is to model each image as a statistical distribution and regard the variance as the uncertainty measure. For example, Oh et al. [50] employed Gaussian distributions to represent images and used Monte-Carlo sampling to sample several point embeddings from the distributions. They then imposed a soft contrastive loss on the sampled embeddings to optimize the metric. Similar strategy has also been used in pose estimation [66], cross-model retrieval [9] and unsupervised embedding learning [90]. However, they still make confident similarity judgments regardless of the uncertainty and a larger variance would not necessarily weaken the semantic discrepancies between samples. Differently, we propose an introspective metric to measure the similarity between images, which tends to omit the semantic differences of two images given a large uncertainty level. Our method bypasses the optimization of distributions and can further increase the robustness of the model.

III. Proposed Approach

In this section, we first revisit existing deterministic and probabilistic deep metric learning methods and introduce the motivation of our framework. We then present the introspective similarity metric and elaborate on the introspective deep metric learning framework. Lastly, we conduct a gradient analysis to reveal the advantage of our framework and demonstrate how to apply our framework to existing deep metric learning methods.

A. Motivation of an Uncertainty-Aware Metric

Let $\mathbf{X}$ be an image set with N training samples $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ and $\mathbf{L}$ be the ground truth label set $\{l_1, \ldots, l_N\}$. Deep metric learning methods aim at learning a mapping to transform each image $\mathbf{x}_i$ to an embedding space, where conventional methods [29], [45] use a deterministic vector embedding $\mathbf{y}_i$ to represent an image. They usually adopt the Euclidean distance as the distance measure

$$D(\mathbf{x}_1, \mathbf{x}_2) = D_E(\mathbf{x}_1, \mathbf{x}_2) = ||\mathbf{y}_1 - \mathbf{y}_2||_2, \tag{1}$$

where $||\cdot||_2$ denotes the L2-norm. They then impose various constraints on the pairwise distances, which generally enlarge interclass distances and reduce intraclass distances.

Conventional DML methods only encode semantic information in the embedding space, ignoring the possible uncertainty in images. However, the semantic uncertainty ubiquitously exists due to low resolution, blur, occlusion, or semantic ambiguity, which motivates probabilistic embedding learning methods [9], [50] to model images as statistical distributions $\mathbf{Y}$ in the embedding space. They further use the distribution variance $\boldsymbol{\sigma}$ to measure the uncertainty of the image in the embedding space. One popular way is to employ a Gaussian distribution to describe an image [50], i.e., $\mathbf{Y} \sim N(\boldsymbol{\mu}, \boldsymbol{\sigma})$, where they use a network to learn two vectors $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ as the mean and variance of the distribution, respectively, assuming each dimension is independent. They adopt distribution divergences [26], [92] or Monte-Carlo-sampling-based [50] distances as the similarity metrics. For instance, Hershey et al. [26] used KL-divergence to measure the discrepancy of two Gaussian distributions $\mathbf{Y}_1$

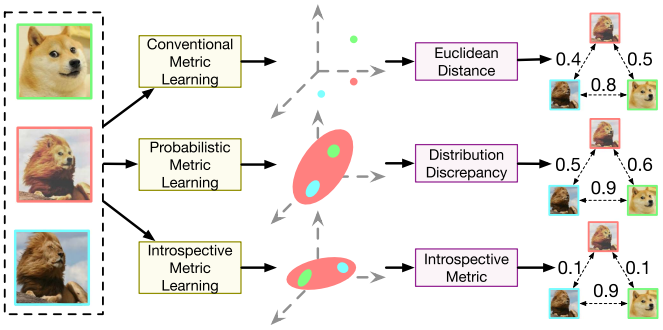


Fig. 2. Comparisons between different similarity metrics. Conventional metric learning and probabilistic metric learning both produce a discriminative distance for a pair of images regardless of the uncertainty level. Our introspective similarity metric weakens the semantic discrepancies for uncertain pairs.

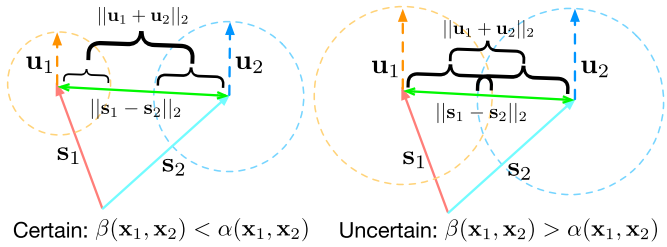


Fig. 3. Illustration of the proposed uncertain-aware comparison of images. We consider both the semantic discrepancy $\alpha(\mathbf{x}_1,\mathbf{x}_2) = ||\mathbf{s}_1 - \mathbf{s}_2||_2$ and the uncertainty level $\beta(\mathbf{x}_1,\mathbf{x}_2) = ||\mathbf{u}_1 + \mathbf{u}_2||_2$ to compute the similarity. We deem it uncertain to distinguish a pair when $\beta(\mathbf{x}_1,\mathbf{x}_2) > \alpha(\mathbf{x}_1,\mathbf{x}_2)$. We only demonstrate the case when $\mathbf{u}_1$ and $\mathbf{u}_2$ align with each other for simplicity. Practically, we use $||\mathbf{u}_1 + \mathbf{u}_2||_2$ instead of $||\mathbf{u}_1||_2 + ||\mathbf{u}_2||_2$ to facilitate more capacity.

and $\mathbf{Y}_2$

$$D_{KL} = -\frac{1}{2}\sum_{k=1}^{d}\left[\log\frac{\sigma_{1,k}^2}{\sigma_{2,k}^2} - \frac{\sigma_{1,k}^2}{\sigma_{2,k}^2} - \frac{(\mu_{1,k}-\mu_{2,k})^2}{\sigma_{2,k}^2} + 1\right], \tag{2}$$

where d denotes the dimension of the Gaussian distributions, $\mathbf{Y}_1 \sim N(\boldsymbol{\mu}_1, \boldsymbol{\sigma}_1)$, and $\mathbf{Y}_2 \sim N(\boldsymbol{\mu}_2, \boldsymbol{\sigma}_2)$, and $\sigma_{1,k}$ denotes the kth component of $\boldsymbol{\sigma}_1$.

One can find that for two distributions with the same mean, their discrepancy solely depends on the ratio of the variance. The discrepancy still varies greatly when the variance of one image is large, i.e., of large uncertainty. In other words, it still provides confident judgments about the similarity even when uncertain. However, we argue that a good similarity metric should weaken the semantic discrepancies for uncertain image pairs to allow confusion during training, which has been proven to be useful in knowledge distillation [27]. This avoids the false pulling of ambiguous pairs to improve the generalization of the learned model. Fig. 2 presents the comparison between different metrics.

B. Introspective Similarity Metric

To facilitate an uncertainty-aware similarity metric, we first need to model the uncertainty in images. Different from existing probabilistic embedding learning methods to model images as distributions, we represent an image using a semantic embedding $\mathbf{s}$ and an uncertainty embedding $\mathbf{u}$, i.e., $\mathbf{y}_{IN} = \{\mathbf{s}, \mathbf{u}\}$. The semantic embedding $\mathbf{s}$ describes the semantic characteristics of an image while the uncertainty embedding $\mathbf{u}$ models the ambiguity.

For comparing two images $\mathbf{x}_1$ and $\mathbf{x}_2$, we define the semantic distance as $\alpha(\mathbf{x}_1,\mathbf{x}_2) = ||\mathbf{s}_1 - \mathbf{s}_2||_2$ similar to conventional DML methods but further compute a similarity uncertainty as $\beta(\mathbf{x}_1,\mathbf{x}_2) = ||\mathbf{u}_1 + \mathbf{u}_2||_2$. Note that we add the vectors of the uncertainty embeddings before computing the norm instead of directly adding their norms. The reason is that the uncertainty should depend on both concerning images. For example, it might be difficult to differentiate a wolf from a dog, but it can be affirmatively distinguished from a person.

When determining the semantic similarity between two images, an introspective metric needs to consider both the semantic distance and the similarity uncertainty. And when not certain enough, the metric refuses to distinguish semantic discrepancies, as illustrated in Fig. 3. Formally, we consider it uncertain to identify the semantic discrepancy between $\mathbf{x}_1$ and $\mathbf{x}_2$ when

$$\beta(\mathbf{x}_1,\mathbf{x}_2) + \gamma \geq \alpha(\mathbf{x}_1,\mathbf{x}_2), \tag{3}$$

where $\gamma \geq 0$ is the introspective bias indicating the introspective degree of the metric. The positive value of γ represents that the metric is still suspicious even if the image representation model provides no uncertainty. We then define a strict introspective similarity metric as

$$\tilde{D}_{IN}(\mathbf{x}_1,\mathbf{x}_2) = \alpha(\mathbf{x}_1,\mathbf{x}_2) \cdot I(\alpha(\mathbf{x}_1,\mathbf{x}_2) - \beta(\mathbf{x}_1,\mathbf{x}_2) - \gamma), \tag{4}$$

where $I(x)$ is an indicator function which outputs 1 if $x > 0$ and 0 otherwise.

However, the use of an indicator function is too strict and hard to optimize during training. We instead compare the semantic distance and the similarity uncertainty to define the relative uncertainty of two images

$$\widetilde{\beta}(\mathbf{x}_1,\mathbf{x}_2) = \frac{\beta(\mathbf{x}_1,\mathbf{x}_2) + \gamma}{\alpha(\mathbf{x}_1,\mathbf{x}_2)}. \tag{5}$$

Note that the relative uncertainty is constantly non-negative. We then use it to soften the semantic discrepancy to obtain our introspective similarity metric (ISM)

$$D_{IN}(\mathbf{x}_1,\mathbf{x}_2) = \alpha(\mathbf{x}_1,\mathbf{x}_2) \cdot e^{(-\frac{1}{\tau}\,\widetilde{\beta}(\mathbf{x}_1,\mathbf{x}_2))}, \tag{6}$$

$\tau > 0$ is a hyperparameter to control the weakening degree. Note that the proposed ISM does not satisfy the triangular equation and thus is not a mathematically strict metric. We follow existing works [49], [93] to still refer to it as a metric.

Intuitively, the proposed introspectively metric considers both the semantic distance and the similarity uncertainty between two images to conduct the final semantic discrepancies. It generally produces a smaller distance for two images due to the awareness of the uncertainty. Given two pairs of images with the same semantic distance, the introspective metric distinguishes better for the pair with a smaller similarity uncertainty. Also,

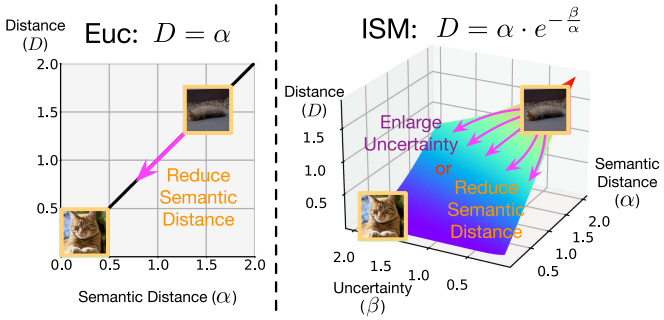


Fig. 4. Illustration of the proposed ISM compared with the conventional Euclidean (Euc) metric. For a semantically ambiguous image, conventional DML explicitly reduces its distance with other intraclass images unaware of the uncertainty. Differently, the proposed introspective similarity metric provides an alternative way to enlarge the uncertainty level to allow confusion in the network.

when the uncertainty of two images outweighs the semantic distance to a great extent, the introspective metric simply outputs a near-zero semantic distance avoiding unnecessary influence on the network. Fig. 4 illustrates the effect of the proposed introspective similarity metric compared with the conventional Euclidean metric. Our ISM provides an alternate way to enlarge the uncertainty for an ambiguous pair instead of rigidly altering their semantic distance.

C. Introspective Deep Metric Learning

In this subsection, we demonstrate how to apply the proposed introspective similarity metric to existing methods and present the overall framework of IDML, as illustrated in Fig. 5.

Although data uncertainty ubiquitously exists in the original images, it is hard to accurately quantify and compare the uncertainty for each image. On the other hand, data augmentation [15], [95], [97] is a widely-used technique for training modern deep learning models. They expand the training data to improve the generalization ability of the learned model, yet many of the augmented images (e.g., mixed, blurred, and partially occluded images) show larger semantic uncertainties compared to the original ones. For example, one of the most effective data augmentation methods is Mixup [97], which randomly interpolates the pixel values of two images to generate a synthetic image with multiple concepts. The generated images thus show characteristics of both the original images and are semantically uncertain to a great extent. Therefore, we mainly accompany our framework with Mixup [97] to demonstrate the advantage of our framework to deal with data with large uncertainty, though we find simply using ISM also improves the performance (Section IV-E).

Formally, Mixup [97] mixes the original images $\mathbf{x}_1$ and $\mathbf{x}_2$ to obtain $\mathbf{x}_m = \lambda \cdot \mathbf{x}_1 + (1-\lambda) \cdot \mathbf{x}_2$. Different from the original method which combines the labels of two images by $l_m = \lambda \cdot l_1 + (1-\lambda) \cdot l_2$, we treat l_m as a set which simultaneously includes l_1 and l_2, noted as $l_m = \{l_1, l_2\}$. We define $l_i = l_j$ if $l_i \cap l_j \neq \emptyset$ and $l_i \neq l_j$ otherwise. In other words, for a mixed image of dog-lion, we think it belongs to both classes of dog and lion.

We adopt a deep neural network to extract the feature embeddings of both original images and mixed images $\mathbf{y}_{IN} = f(\mathbf{x}) = \{\mathbf{s}, \mathbf{u}\}$, where $\mathbf{s}$ and $\mathbf{u}$ denote the semantic embedding and the uncertainty embedding of the image $\mathbf{x}$, respectively. The distance (or similarity) computation of a pair of samples then follows the proposed introspective similarity metric (6). Our IDML can be generally applied to various methods with different loss functions and sampling strategies. For a training objective $J(\mathbf{y}, \mathbf{L}; D)$, where D can be the Euclidean distance, we simply substitute the embedding $\mathbf{y}$ and the metric D with the proposed introspective embedding $\mathbf{y}_I$ and the corresponding metric D_{IN} as the objective $J(\mathbf{y}_{IN}, \mathbf{L}; D_{IN})$ of the proposed IDML framework. We detail the specific objective with different methods in Section III-D.

For inference, we directly use the Euclidean distance between semantic embeddings as the similarity metric and can optionally use the uncertainty embedding to indicate the uncertainty level. Therefore, the proposed IDML framework introduces no additional computation load compared to the original method.

D. Applications of IDML to Various Methods

The proposed ISM is generally compatible with a variety of loss formulations and sampling strategies. Our framework can be readily applied to existing methods and can be very easily implemented with only a few additional lines of code. We provide the PyTorch-like pseudocode of IDML in Algorithm 1.

Most pair-based losses compute distances between samples, so we can directly employ the proposed ISM (6). However, many proxy-based losses such as the softmax loss [10], [67] usually compute the cosine similarity $C(\mathbf{x}_i, \mathbf{p}_j)$ instead of the distance between an image $\mathbf{x}_i$ and a class-level representative $\mathbf{p}_j$ (i.e., proxy). To accommodate this, we propose a similarity-based version of the proposed ISM

$$C_{IN}(\mathbf{x}_i, \mathbf{p}_j) = 1 - (1 - C(\mathbf{x}_i, \mathbf{p}_j)) \cdot e^{(-\frac{1}{\tau}\ \widetilde{\beta}(\mathbf{x}_i, \mathbf{p}_j))}, \quad (7)$$

where $\widetilde{\beta}(\mathbf{x}_i, \mathbf{p}_j)$ denotes the relative uncertainty between the image and the representative. We see that it similarly blurs the similarity discrepancy for a larger uncertainty.

Contrastive Loss: The contrastive loss [29] directly pulls closer positive samples and pushes away negative samples

$$J_{con}(\mathbf{y}_{IN}, \mathbf{L}; D_{IN}) = \sum_{l_i = l_j} D_{IN}(\mathbf{x}_i, \mathbf{x}_j) + \sum_{l_i \neq l_j} [\delta - D_{IN}(\mathbf{x}_i, \mathbf{x}_j)]_+, \quad (8)$$

where δ is the margin.

Margin Loss & Distance Weighted Sampling: The margin loss [85] is similar to the contrastive and additionally stipulates a margin to positive pairs. It is often equipped with the distance-weighted sampling strategy [85] for more uniform sampling

$$J_m(\mathbf{y}_{IN}, \mathbf{L}; D_{IN}) = \sum_{l_i = l_j} [D_{IN}(\mathbf{x}_i, \mathbf{x}_j) - \xi]_+ - \sum_{l_i \neq l_j} I(p(D_{IN}(\mathbf{x}_i, \mathbf{x}_j)))[\omega - D_{IN}(\mathbf{x}_i, \mathbf{x}_j)]_+, \quad (9)$$

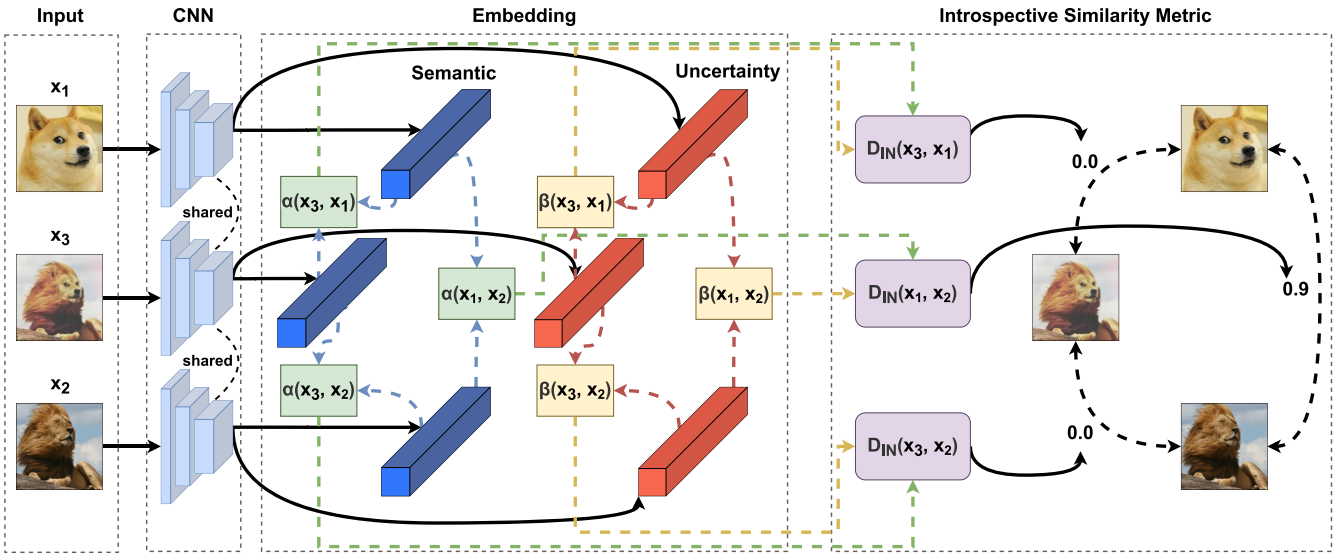


Fig. 5. Illustration of the proposed IDML framework. We employ a convolutional neural network to represent each image by a semantic embedding and an uncertainty embedding. We then use the distance between semantic embeddings as the semantic discrepancy and add the uncertainty embeddings for uncertainty measure. The introspective similarity metric then uses the uncertainty level to weaken the semantic discrepancy to make a discreet similarity judgment.

where $D_{IN}(\mathbf{x}_i, \mathbf{x}_j)$ follows (6), the random variable $I(p)$ has a probability of p to be 1 and 0 otherwise, $p(d) = \min(\phi, d^{2-n}[1-\frac{1}{4}d^2]^{\frac{3-n}{2}})$, $[\cdot]_+ = \max(\cdot, 0)$, ξ and ω are two pre-defined margins, and ϕ is a positive constant.

Triplet Loss & Semi-Hard Negative Sampling: The triplet loss [60] requires a distance ranking within triplets and maintains a margin between positive pairs and negative pairs. The IDML with the triplet loss can be formulated as follows:

$$J_t(\mathbf{y}_{IN}, \mathbf{L}; D_{IN}) = \sum_{a,p,n} [D_{IN}(\mathbf{x}_a, \mathbf{x}_p) - D_{IN}(\mathbf{x}_a, \mathbf{x}_n) + \delta]_+, \tag{10}$$

where $D_{IN}(\cdot)$ denotes our introspective similarity metric, $\{a, p, n\}$ denote the indices of all the possible triplets, $[\cdot]_+ = \max(\cdot, 0)$, and δ is a pre-defined margin. Furthermore, we employ the semi-hard negative sampling strategy to select challenging samples while avoiding noisy ones to boost training. Given an anchor $\mathbf{x}_a$ and a positive sample $\mathbf{x}_p$, we select the negative sample $\mathbf{x}_n$ which satisfies the following constraint

$$\mathbf{n}^*_{\mathbf{a},\mathbf{p}} = \underset{n: D_{IN}(\mathbf{x_a}, \mathbf{x_p}) < D_{IN}(\mathbf{x_a}, \mathbf{x_n})}{\arg\min} D_{IN}(\mathbf{x_a}, \mathbf{x_n}). \tag{11}$$

Multi-Similarity Loss: The multi-similarity (MS) loss [82] uses the cosine similarity to measure the relations between samples. We first define a modified introspective similarity as follows:

$$C^*_{IN}(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} C_{IN}(\mathbf{x}_i, \mathbf{x}_j), & C_{IN}(\mathbf{x}_i, \mathbf{x}_j) > \min_{l_k = l_i} C_{IN}(\mathbf{x}_i, \mathbf{x}_k) - \epsilon, \\ C_{IN}(\mathbf{x}_i, \mathbf{x}_j), & C_{IN}(\mathbf{x}_i, \mathbf{x}_j) < \max_{l_k \neq l_i} C_{IN}(\mathbf{x}_i, \mathbf{x}_k) - \epsilon, \\ 0, & \text{otherwise}, \end{cases} \tag{12}$$

Algorithm 1: Pseudocode of IDML in a PyTorch-Like Style.

```
# X_tr, X_te: the training set, the testing set
# L_tr, L_te: the ground truth label sets
# f: the original network
# f_s, f_u: the semantic layer and uncertainty layer
# norm: l2 norm
# gamma, tau: pre-defined hyper-parameters
# J: the loss function
# V: the evaluation function

#training:
model.train()
for (x1, l1), (x2, l2) in (X_tr, L_tr):
    z1 = f(x1), z2 = f(x2) # feature extraction
    s1 = f_s(z1), s2 = f_s(z2) # semantic embeddings
    u1 = f_u(z1), u2 = f_u(z2) # uncertainty embeddings
    alpha(x1, x2) = norm(s1 - s2)
    beta(x1, x2) = norm(u1 + u2)
    beta_r(x1, x2) = (beta(x1, x2) + gamma)/alpha(x1, x2)
    D_in(x1, x2) = alpha(x1, x2)*exp(-beta_r(x1, x2)/tau)
    loss = J(D_in(x1, x2), l1, l2)
    loss.backward()

# evaluation:
S = [] # embeddings
model.eval()
for (x, l) in (X_te, L_te):
    z = f(x) # feature extraction
    s = f_s(z) # the semantic embedding
    S.append(s) # same as conventional methods

evaluation_results = V(S) # introduce no additional load
```

where ϵ is a hyperparameter. We then instantiate the IDML framework with the MS loss as follows:

$$J_{MS}(\mathbf{y}_{IN}, \mathbf{L}; C^*_{IN}) = \frac{1}{N}\sum_{i=1}^{N}\left\{\frac{1}{\alpha}\log\left[1 + \sum_{l_i = l_j} e^{-\alpha(C^*_{IN}(\mathbf{x}_i, \mathbf{x}_j) - \lambda)}\right] + \frac{1}{\beta}\log\left[1 + \sum_{l_i \neq l_j} e^{\beta(C^*_{IN}(\mathbf{x}_i, \mathbf{x}_j) - \lambda)}\right]\right\}, \tag{13}$$

where α, β, and λ are hyperparameters.

Softmax Loss: The softmax loss is a very commonly used loss function mainly for the classification task. It can be seen as a proxy-based metric learning loss, which maximizes the similarity of a sample with the positive proxy and minimizes the added similarities with other proxies

$$J_s(\mathbf{y}_{IN}, \mathbf{L}; C_{IN}) = \frac{1}{N}\sum_{i=1}^{N}\left(-log\frac{\sum_{l_{\mathbf{p}_j}=l_i} e^{C_{IN}(\mathbf{x}_i,\mathbf{p}_j)}}{\sum_{l_{\mathbf{p}_j}\neq l_i} e^{C_{IN}(\mathbf{x}_i,\mathbf{p}_j)}}\right), \tag{14}$$

where $l_{\mathbf{p}_j}$ is the category of the class representative $\mathbf{p}_j$. $\mathbf{p}_j$ can be implemented by the jth row vector of a linear classifier.

ProxyNCA Loss: Very similar to the softmax loss, the ProxyNCA loss [45] optimizes the distances between a sample and all the proxies instead of the similarities

$$J_p(\mathbf{y}_{IN}, \mathbf{L}; D_{IN}) = \sum_{i}\left(-log\frac{\sum_{l_{\mathbf{p}}=l_i} e^{-D_{IN}(\mathbf{x}_i,\mathbf{p})}}{\sum_{l_{\mathbf{p}}\neq l_i} e^{-D_{IN}(\mathbf{x}_i,\mathbf{p})}}\right), \tag{15}$$

where l_i denotes the corresponding label of $\mathbf{x}_i$.

ProxyAnchor Loss: The ProxyAnchor loss [36] takes advantage of both sample-sample and sample-proxy relations with the cosine similarity to improve the discriminativeness of the embedding space. Our IDML framework can be implemented for the ProxyAnchor loss as follows:

$$\begin{aligned} &J_{pa}(\mathbf{y}_{IN}, \mathbf{L}; C_{IN}) \\ &= \frac{1}{|\mathbf{P}^+|}\sum_{\mathbf{p}\in\mathbf{P}^+} log\left(1+\sum_{l_i=l_{\mathbf{p}}} e^{-\alpha(C_{IN}(\mathbf{x}_i,\mathbf{p})-\delta)}\right) \\ &\quad + \frac{1}{|\mathbf{P}|}\sum_{\mathbf{p}\in\mathbf{P}} log\left(1+\sum_{l_i\neq l_{\mathbf{p}}} e^{\alpha(C_{IN}(\mathbf{x}_i,\mathbf{p})+\delta)}\right), \end{aligned} \tag{16}$$

where $\mathbf{P}$ denotes the set of all proxies, $\mathbf{P}^+$ denotes the set of positive proxies, $|\cdot|$ represents the size of the set, $\alpha > 0$ is a scaling factor, and $\delta > 0$ is a pre-defined margin.

E. Gradient Analysis

We provide a gradient analysis to demonstrate the effect of our introspective similarity metric on the learning of semantic embeddings. Generally, our proposed similarity metric results in reduced semantic discrepancies to lower the influences of uncertain samples. These influences are measured by the magnitude of gradients on the model parameters. For a conventional metric learning method $J^E = J(\mathbf{y}, \mathbf{L}; D_E)$ equipped with the Euclidean distance $D_E(\mathbf{x}_1, \mathbf{x}_2) = \alpha(\mathbf{x}_1, \mathbf{x}_2) = ||\mathbf{s}_1 - \mathbf{s}_2||_2$, we can decompose the loss gradient on the model parameters according to the chain rule as follows:

$$\frac{\partial J^E}{\partial W} = \frac{\partial J^E}{\partial \mathbf{s}}\cdot\frac{\partial \mathbf{s}}{\partial W} = \frac{\partial J^E}{\partial \alpha}\cdot\frac{\partial \alpha}{\partial \mathbf{s}}\cdot\frac{\partial \mathbf{s}}{\partial W}, \tag{17}$$

where W represents the model parameters that affect the semantic embedding $\mathbf{s}$. Similarly, for the IDML objective $J^{IN} = J(\mathbf{y}_{IN}, \mathbf{L}; D_{IN})$, we compute the loss gradient for our IDML framework as follows:

$$\frac{\partial J^{IN}}{\partial W} = \frac{\partial J^{IN}}{\partial D_{IN}}\cdot\frac{\partial D_{IN}}{\partial \mathbf{s}}\cdot\frac{\partial \mathbf{s}}{\partial W} = \frac{\partial J^{IN}}{\partial D_{IN}}\cdot\frac{\partial D_{IN}}{\partial \alpha}\cdot\frac{\partial \alpha}{\partial \mathbf{s}}\cdot\frac{\partial \mathbf{s}}{\partial W}. \tag{18}$$

In both (17) and (18), the partial term $\frac{\partial \mathbf{S}}{\partial W^t}$ is only relevant to the architecture of the backbone network. As our IDML only substitutes the Euclidean distance D_E in the original training objective with the proposed ISM D_{IN}, we have $\frac{\partial J^E}{\partial \mathbf{s}} = \frac{\partial J^{IN}}{\partial D_{IN}}$, which is determined by the form of the loss function. Therefore, our IDML multiplies the conventional loss gradient with an additional term related to the uncertainty level $\beta(\mathbf{x}_1, \mathbf{x}_2)$

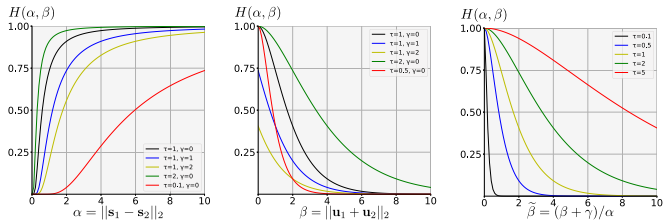
$$\frac{\partial J^{IN}}{\partial W} = \frac{\partial J^E}{\partial W}\cdot\frac{\partial D_{IN}}{\partial \alpha} = \frac{\partial J^E}{\partial W}\cdot H(\alpha, \beta), \tag{19}$$

where

$$H(\alpha,\beta) := \frac{\partial D_{IN}}{\partial \alpha} = \frac{\partial\left(\alpha\cdot e^{-\frac{1}{\tau}\frac{\beta+\gamma}{\alpha}}\right)}{\partial \alpha} = e^{-\frac{\widetilde{\beta}}{\tau}}\cdot\left(1+\frac{\widetilde{\beta}}{\tau}\right), \tag{20}$$

where $\widetilde{\beta} = \frac{\beta+\gamma}{\alpha}$ is the relative uncertainty defined in (5).

We plot the effect of the semantic distance, the uncertainty level, and the relative uncertainty on $H(\alpha, \beta)$ in Fig. 6(a), (b), and (c), respectively. With the function $g(x) = e^{-x}\cdot(1+x)$ being monotonically decreasing when $x \geq 0$ and the relative uncertainty $\widetilde{\beta}$ being constantly non-negative, we see that $H(\alpha, \beta)$ has a maximum value of 1 when $\widetilde{\beta} = 0$ and decreases as the uncertainty level increases. This indicates that samples with larger uncertainties will result in smaller gradients compared to the original method, and thus impose fewer influences on the network. Our method reduces the effects of semantically ambiguous images to avoid false training signals. In particular, when $\widetilde{\beta} = 0$ (i.e., absolutely certain), we have $H(\alpha, \beta) = 1$ and $\frac{\partial J^{IN}}{\partial W} = \frac{\partial J^{IN}}{\partial W}$. The proposed IDML framework then degenerates to the conventional method with the Euclidean distance. Our framework is a generalization of conventional metric learning if we set the introspective bias $\gamma = 0$ and assign a zero uncertainty level to all the sample pairs.



(a) Effect of α. (b) Effect of β. (c) Effect of $\widetilde{\beta}$.

Fig. 6. Influence of the (a) the semantic distance, (b) the uncertainty level, and (c) the relative uncertainty on the gradient weights. With a weight generally less than 1, our ISM reduces the loss effects (i.e., gradients) on the network to avoid false training signals. Furthermore, a larger similarity uncertainty and a smaller semantic distance indicate that the model is less confident to differentiate two images, and the proposed ISM thus implicitly induces a smaller gradient.

IV. Experiments

In this section, we conducted various experiments to evaluate the performance of our IDML framework on both image retrieval and classification. We show that employing the proposed introspective similarity metric consistently improves the performance of existing deep metric learning and data mixing methods. We also provide in-depth analyses of the effectiveness of our framework.

A. Settings

Image Retrieval: We first evaluated our framework under the conventional deep metric learning setting [65], [82] and conducted experiments on three widely-used datasets: CUB-200-2011 [76], Cars196 [38], and Stanford Online Products [65]. We adopted the ImageNet [57] pretrained ResNet-50 [25] as the backbone and two randomly initialized fully connected layers to obtain the semantic embedding and uncertainty embedding, respectively. We set the embedding size to 512 for the main experiments. The training images were first resized to 256×256 and then augmented with random cropping to 224×224 as well as random horizontal flipping with the probability of 50%. We employed Mixup [97] for our framework to generate images with large uncertainty for training unless otherwise stated. We fixed the batch size to 120 and used the AdamW optimizer with the learning rate of 10^{-5}. We set $\gamma = 3$ for the Cars196 dataset and $\gamma = 0$ for the other datasets and fixed $\epsilon = 5$ for all the datasets during training. We adopted the original similarity metric without our uncertainty embedding for testing and thus introducing no additional computational workload. The reported evaluation metrics include Recall@Ks, normalized mutual information (NMI), R-Precision (RP), and Mean Average Precision at R (M@R). See Musgrave et al. [46] for more details.

Image Classification: We evaluated our framework for the image classification task on three datasets: ImageNet-1K [57], CIFAR-10 [39], and CIFAR-100 [39]. We directly implemented the proposed introspective similarity metric on Mixup [97] and CutMix [95] using the official code[1] and used the same hyperparameter without tuning. Specifically, for ImageNet-1 K, we adopted the ResNet-50 [25] as the base model and fixed the probability of the Mixup and CutMix to 1.0. We set the batch size to 256 and the total training epochs to 300. The learning rate was initialized to 0.1 and decayed by 0.1 at epochs 75, 150, and 225. For CIFAR-10 and CIFAR-100, we adopted the PyramidNet-200 [22] as the backbone and fixed the probability of the Mixup and Cutmix to 0.5 during training. We set the batch size to 64 and the total training epochs to 300. The learning rate was 0.25 with a 0.1 decay rate at epochs 150 and 225. We used $\gamma = 0$ and $\tau = 1$ for our introspective similarity metric on all datasets. We report the classification accuracy for evaluation.

B. Dataset

For the image retrieval task, we followed existing DML methods [36], [65], [82], [101] to conduct experiments on the CUB-200-2011 [76], Cars196 [38], and Stanford Online Products [65]. CUB-200-2011 contains 11,788 images of 200 bird species. We used the first 100 species with 5,864 images for training and the rest 100 species with 5,924 images are for testing. Cars196 includes 16,183 images of 196 car models. We used the first 96 classes with 8,054 images for training and the rest 98 classes with 8,131 images for testing. Stanford Online Products is relatively large and contains 120,053 images of 22,634 products. We used the first 11,318 products with 59,551 images in the training set and the rest of 11,318 products with 60,502 images for testing.

For the image classification task, we conducted experiments on the widely used ImageNet-1K [57], CIFAR-10 [39], and CIFAR-100 [39]. ImageNet-1 K contains 1,200,000 training images and 50,000 validation images from to 1,000 categories. CIFAR-10 and CIFAR-100 comprise the same 50,000 images for training and 10,000 images for validation. CIFAR-10 contains 10 classes with 6,000 images per class while CIFAR-100 further provides more fine-grained labels of 100 classes with 600 images per class.

C. Main Results: Image Retrieval

We evaluated the proposed IDML framework under the conventional deep metric learning setting and compared it with state-of-the-art methods. To demonstrate the versatility of our framework, we applied the introspective similarity metric to various loss functions, including the triplet loss with the semi-hard sampling (Triplet-SH) [60], the ProxyNCA loss [45], the FastAP loss [2], the contrastive loss [29], the margin loss with the distance-weighed sampling (Margin-DW) [85], the multi-similarity loss (Multi-Sim) [82], and the ProxyAnchor loss [36].

Table I shows the experimental results on the CUB-200-2011 [76], Cars196 [38], and Stanford Online Products [65] datasets. The n-BN/R denotes the model setting where n is the embedding size and BN, R represents BN-Incertion [31] and ResNet-50 [25], respectively. The bold numbers highlight the improvement of our framework compared with the original method. We indicate the best results using red colors and the second best results using blue colors. We observe that our framework achieves a constant performance boost to all the associated methods. Furthermore, we attain state-of-the-art performance on all three datasets by applying our framework to the ProxyAnchor loss, which surpasses the original performance by 3.3% at Recall@1 and 2.0% at M@R on the Cars196 dataset, respectively. This is because the proposed similarity metric is aware of the data uncertainty in images so that the uncertain samples only provide limited training signals.

D. Main Results: Image Classification

For the image classification task, we applied our framework to the widely used Mixup [97] and CutMix [95] method to evaluate the effectiveness. Tables II, III, and IV shows the experimental results on the ImageNet-1K [57], CIFAR-100 [39], and CIFAR-10 [39] datasets, respectively. We first applied the proposed ISM to the baseline vanilla method without using any data mixing methods. We see consistent performance improvements on all datasets, which demonstrate the significance of exploiting the natural ambiguity in the original images. We also observe that

[1][Online]. Available: https://github.com/clovaai/CutMix-PyTorch

TABLE I
EXPERIMENTAL RESULTS (%) ON THE CUB-200-2011, CARS196, AND STANFORD ONLINE PRODUCTS DATASETS COMPARED WITH STATE-OF-THE-ART METHODS

		CUB-200-2011					Cars196					Stanford Online Products				
Method	Setting	R@1	R@2	NMI	RP	M@R	R@1	R@2	NMI	RP	M@R	R@1	R@10	NMI	RP	M@R
N-Pair [64]	512G	50.1	63.3	60.4	-	-	71.1	79.7	64.0	-	-	67.7	83.8	88.1	-	-
Angular [79]	512G	53.6	65.0	61.0	-	-	71.3	80.7	62.4	-	-	67.9	83.2	87.8	-	-
HDML [101]	512G	53.7	65.7	62.6	-	-	79.1	87.1	69.7	-	-	68.7	83.2	89.3	-	-
HTL [18]	512BN	57.1	68.8	-	-	-	81.4	88.0	-	-	-	74.8	88.3	-	-	-
RLL-H [81]	512BN	57.4	69.7	63.6	-	-	74.0	83.6	65.4	-	-	76.1	89.1	89.7	-	-
A-BIER [51]	512G	57.5	68.7	-	-	-	82.0	89.0	-	-	-	74.2	86.9	-	-	-
ABE-8 [37]	512G	60.6	71.5	-	-	-	85.2	90.5	-	-	-	76.3	88.4	-	-	-
Ranked [81]	1536BN	61.3	72.7	66.1	-	-	82.1	89.3	71.8	-	-	79.8	91.3	90.4	-	-
DREML [88]	9216R	63.9	75.0	67.8	-	-	86.0	91.7	76.4	-	-	-	-	-	-	-
SoftTriple [53]	512BN	65.4	76.4	69.3	-	-	84.5	90.7	70.1	-	-	78.3	90.3	92.0	-	-
D & C [58]	128R	65.9	76.6	69.6	-	-	84.6	90.7	70.3	-	-	75.9	88.4	90.2	-	-
MIC [54]	128R	66.1	76.8	69.7	-	-	82.6	89.1	68.4	-	-	77.2	89.4	90.0	-	-
RankMI [32]	128R	66.7	77.2	71.3	-	-	83.3	89.8	69.4	-	-	74.3	87.9	90.5	-	-
CircleLoss [67]	512R	66.7	77.4	-	-	-	83.4	89.8	-	-	-	78.3	90.5	-	-	-
PADS [55]	128BN	67.3	78.0	69.9	-	-	83.5	89.7	68.8	-	-	76.5	89.0	89.9	-	-
DIML [100]	512R	68.2	-	-	37.9	26.9	87.0	-	-	39.0	29.4	79.3	-	-	46.4	43.2
DCML [102]	512R	68.4	77.9	71.8	-	-	85.2	91.8	73.9	-	-	79.8	90.8	90.8	-	-
DRML [103]	512BN	68.7	78.6	69.3	-	-	86.9	92.1	72.1	-	-	79.9	90.7	90.1	-	-
ProxyNCA++ [70]	512R	69.0	79.8	73.9	-	-	86.5	92.5	73.8	-	-	80.7	92.0	-	-	-
DiVA [44]	512R	69.2	79.3	71.4	-	-	87.6	92.9	72.2	-	-	79.6	91.2	90.6	-	-
NIR [56]	512R	70.5	80.6	72.5	-	-	89.1	93.4	75.0	-	-	80.7	91.5	90.9	-	-
Triplet-SH* [60]	512R	63.6	75.5	67.9	35.1	24.0	70.8	81.7	64.8	31.7	21.1	76.5	89.1	89.7	51.3	48.4
IDML-TSH	512R	**65.3**	**76.5**	**69.5**	**36.2**	**25.0**	**73.7**	**84.0**	**67.3**	**33.8**	**24.1**	**77.4**	**89.4**	**90.1**	**51.9**	**49.0**
ProxyNCA* [45]	512R	64.6	75.6	69.1	35.5	24.7	82.6	89.0	66.4	33.5	23.5	77.0	89.1	89.5	51.9	49.0
IDML-PN	512R	**66.0**	**76.4**	**70.1**	**36.5**	**25.4**	**85.5**	**91.3**	**69.0**	**36.1**	**26.4**	**78.3**	**90.1**	**89.9**	**53.0**	**49.9**
FastAP* [2]	512R	65.1	75.4	68.5	35.9	24.1	81.6	88.5	68.8	35.1	25.2	75.9	89.2	89.7	50.1	46.8
IDML-FAP	512R	**66.4**	**76.4**	**69.7**	**36.7**	**25.5**	**83.9**	**89.9**	**71.9**	**36.5**	**26.7**	**76.8**	**89.7**	**90.9**	**50.9**	**47.9**
Contrastive* [29]	512R	65.6	76.5	68.9	36.5	24.7	82.7	89.6	69.5	35.8	25.7	76.4	88.5	88.9	50.9	47.9
IDML-Con	512R	**67.2**	**77.6**	**71.3**	**37.5**	**25.7**	**85.5**	**91.5**	**72.5**	**38.8**	**29.0**	**77.3**	**89.7**	**90.0**	**51.7**	**48.5**
Margin-DW* [85]	512R	65.9	77.0	69.5	36.0	24.9	82.6	88.7	69.3	36.4	26.5	78.5	89.9	90.1	53.4	50.2
IDML-MDW	512R	**67.9**	**78.3**	**72.1**	**37.2**	**26.1**	**86.1**	**91.7**	**73.0**	**39.2**	**29.7**	**79.4**	**90.6**	**91.0**	**53.7**	**50.4**
Multi-Sim* [82]	512R	67.3	78.2	72.7	36.6	25.5	83.3	90.9	72.2	37.4	27.4	78.1	90.0	89.9	52.9	49.9
IDML-MS	512R	**69.0**	**79.5**	**73.5**	**38.5**	**27.2**	**86.3**	**92.2**	**74.1**	**40.0**	**30.8**	**79.7**	**91.4**	**91.2**	**53.7**	**50.9**
ProxyAnchor* [36]	512R	69.0	79.4	72.3	38.5	27.5	87.3	92.7	75.7	40.9	31.8	79.5	91.1	91.0	53.7	50.5
IDML-PA	512R	**70.7**	**80.2**	**73.5**	**39.3**	**28.4**	**90.6**	**94.5**	**76.9**	**42.6**	**33.8**	**81.5**	**92.7**	**92.3**	**54.8**	**51.3**

* denotes our reproduced results under the same settings.

our framework further improves the data mixing methods at top-1 and top-5 accuracy. In particular, the IDML framework with Mixup outperforms the original method by 0.53%, 0.37%, and 0.22% at top-1 accuracy on ImageNet-1 K, CIFAR-100, and CIFAR-10, respectively. In addition, we improve CutMix by 0.44% on ImageNet-1 K and attain a top-1 accuracy of 79.04% using exactly the same recipe. Despite no hyperparameter tuning, this is already one of the strongest performances for a vanilla ResNet-50 using a regular training recipe without extra data or distillation.

E. Analysis

1) Ablation Study of Different Components: We conducted experiments with the margin loss and ProxyAnchor loss to analyze the effect of different components of our framework. Table V shows the experimental results on the CUB-200-2011, Cars196, and Stanford Online Products datasets.

We first applied Mixup to the baseline method (Mixup-MDW) without using our introspective similarity metric and then only employed the proposed metric without mixup (ISM-MDW). We see that the Mixup method and the proposed ISM can independently boost the performances of the baseline method. Our IDML framework further improves the performance by combining Mixup and our ISM. Furthermore, we reproduced the probabilistic embedding learning (PEL) framework [50] on each loss (PEL-MDW) and also equipped it with Mixup (PEL-Mixup-MDW) for fair comparisons with our framework. We observe that it achieves lower performance than the baseline method, and further using Mixup improves the performance. The performance drop might result from the compromise of discriminativeness when representing images as distributions. Differently, our framework uses an uncertainty embedding to model the uncertainty which does not affect the discriminativeness of the semantic embedding.

TABLE II
EXPERIMENTAL RESULTS (%) OF THE PROPOSED IDML WITH RESNET-50 ON THE IMAGENET-1K DATASET

Method	Top-1 Acc.	Top-5 Acc.
Baseline	76.32	92.95
+ Cutout [11]	77.07	93.34
+ StochDepth [30]	77.54	93.73
+ Manifold Mixup [74]	77.50	93.79
+ DropBlock [19]	78.13	94.02
+ Feature CutMix [95]	78.20	93.94
+ SaliencyMix [72]	78.74	94.24
+ PuzzleMix [35]	78.76	94.29
+ ISM (ours)	**76.94**	**93.24**
+ Mixup [97]	77.42	93.60
+ Mixup + ISM (ours)	**77.95**	**93.93**
+ CutMix [95]	78.60	94.08
+ CutMix + ISM (ours)	**79.04**	**94.47**

TABLE III
EXPERIMENTAL RESULTS (%) OF THE PROPOSED IDML WITH PYRAMIDNET-200 ON THE CIFAR-100 DATASET

Method	Top-1 Acc.	Top-5 Acc.
Baseline	83.55	96.31
+ Cutout [11]	83.47	96.35
+ StochDepth [30]	84.14	96.67
+ Label Smoothing [68]	83.27	96.63
+ Cutout + Label Smoothing	84.39	96.12
+ Manifold Mixup [74]	83.86	95.93
+ Cutout + Manifold Mixup	84.91	96.65
+ DropBlock [19]	84.27	96.74
+ DropBlock + Label Smoothing	84.84	96.14
+ ISM (ours)	**84.08**	**96.46**
+ Mixup [97]	84.22	95.96
+ Mixup + Cutout	84.54	96.58
+ Mixup + ISM (ours)	**84.59**	**96.79**
+ CutMix [95]	85.53	97.03
+ CutMix + ISM (ours)	**85.65**	**97.21**

2) Uncertainty of Other Forms: In addition to Mixup, we further conducted experiments when training with lowered-resolution, blurred, and occluded images, as shown in Table VI. Though certain augmentations (e.g., low-resolution) reduce the performance of the baseline, further applying our ISM consistently attains better results than training without these augmentations. This verifies the effectiveness of our method to deal with various forms of uncertain data.

3) Comparisons of Different Training Recipes: We provide comparisons of different training recipes for ResNet-50 in Table VII. See [84] for more comprehensive descriptions. We instantiate our framework using a vanilla setting since the main objective is to evaluate the effectiveness of using the proposed metric for training. One can find that our IDML achieves very competitive performance despite using a very constrained setting (e.g., a small batch size). In particular, we outperform CutMix [95] using the same hyperparameters. Several recipes achieve better performances including FAMS [13], Timm-A1 [84], and Timm-A2 [84]. However, they all use a larger batch size (at least $4\times$ than our setting) and a more advanced data augmentation strategy. FAMS [13] and Timm-A1 [84] further train more epochs to achieve better performance. In contrast, our framework achieves competitive performance without bells and whistles.

TABLE IV
EXPERIMENTAL RESULTS (%) OF THE PROPOSED IDML WITH PYRAMIDNET-200 ON THE CIFAR-10 DATASET

Method	Top-1 Acc.
Baseline	96.15
+ Cutout [11]	96.90
+ Manifold Mixup [74]	96.85
+ ISM (ours)	**96.43**
+ Mixup [97]	96.91
+ Mixup + ISM (ours)	**97.13**
+ CutMix [95]	97.12
+ CutMix + ISM (ours)	**97.32**

4) Effect of the Hyper-Parameters γ and τ: γ determines the introspective bias and τ controls the weakening degree in our introspective similarity metric. They jointly affect the final performance of our framework. We experimentally evaluated their impacts on three datasets, as demonstrated in Fig. 7. We first fixed γ to 0 and set τ to 1, 3, 5, 7, 9. We see that our framework achieves the best recall@1 when $\tau = 5$ for the CUB-200-2011 and Cars196 datasets, indicating the favor of a modest weakening degree. In addition, we fixed $\tau = 5$ and set γ to 0, 1, 2, 3, 4 for training. The experimental results vary on the three datasets. Specifically, our framework achieves the best performance when $\gamma = 0$ on the CUB-200-2011 and Stanford Online Products datasets while $\gamma = 3$ on the Cars196 dataset. This indicates that the metric is more discreet when comparing images on the Cars196 dataset.

5) Effect of the Batch Size: We conducted experiments on the CUB-200-2011 [76] and Cars196 [38] datasets to investigate the influence of the batch size during training. Specifically, we set the batch size from 40 to 180, as shown in Table VIII. We observe a relatively consistent performance improvement as the batch size increases on both datasets. This is because larger batch sizes enable richer relation mining among data. Still, we see that the performance plateaus and even decreases when the batch sizes exceed 120. Therefore, we set the batch size to 120 for the main experiments for a better balance of performance and computation.

6) Effect of Embedding Sizes: The dimension of the embedding is a crucial factor for the final performance, as verified by several works [36], [103]. During training, the proposed IDML framework simultaneously obtains a semantic embedding and an uncertainty embedding for each image. During testing, we only use the semantic embeddings for inference, introducing no additional computation cost. Still, the uncertainty embedding influences the training of the semantic embedding and thus affects the inference performance.

TABLE V
ANALYSIS OF DIFFERENT COMPONENTS OF IDML ON THE CUB-200-2011, CARS196, AND STANFORD ONLINE PRODUCTS DATASETS

	CUB-200-2011				Cars196				Stanford Online Products			
Method	R@1	NMI	RP	M@R	R@1	NMI	RP	M@R	R@1	NMI	RP	M@R
Margin-DW [85]	65.9	69.5	36.0	24.9	82.6	69.3	36.4	26.5	78.5	90.1	53.4	50.2
Mixup-MDW	67.1	71.6	36.7	25.5	84.7	72.4	38.0	28.0	79.1	90.5	53.6	**50.4**
ISM-MDW	67.0	71.4	36.9	25.7	84.4	71.9	37.9	28.1	78.9	90.4	53.6	50.3
PEL-MDW [50]	63.3	67.1	34.6	24.2	80.4	67.1	34.8	25.5	76.4	88.7	51.2	48.7
PEL-Mixup-MDW	64.5	68.6	35.3	24.7	82.3	68.9	35.9	26.1	77.2	89.4	52.1	49.3
IDML-MDW	**67.9**	**72.1**	**37.2**	**26.1**	**86.1**	**73.0**	**39.2**	**29.7**	**79.4**	**91.0**	**53.7**	**50.4**
ProxyAnchor [36]	69.0	72.3	38.5	27.5	87.3	75.7	40.9	31.8	79.5	91.0	53.7	50.5
Mixup-PA	69.8	73.0	39.1	28.1	88.5	75.8	41.0	32.1	80.6	91.8	54.4	50.7
ISM-PA	69.5	73.1	38.9	28.0	88.8	75.8	41.2	32.2	80.3	91.8	54.3	50.9
PEL-PA [50]	64.9	67.1	34.5	23.7	83.4	66.4	34.4	24.9	76.8	89.7	51.8	48.7
PEL-Mixup-PA	65.7	68.0	35.6	24.7	84.5	66.6	34.6	25.1	77.9	90.5	52.6	49.9
IDML-PA	**70.7**	**73.5**	**39.3**	**28.4**	**90.6**	**76.9**	**42.6**	**33.8**	**81.5**	**92.3**	**54.8**	**51.3**

TABLE VI
EXPERIMENTAL RESULTS WITH VARIOUS FORMS OF UNCERTAIN DATA

Method	CUB-200-2011				Cars196				Stanford Online Products			
	R@1	NMI	RP	M@R	R@1	NMI	RP	M@R	R@1	NMI	RP	M@R
Baseline	69.0	72.3	38.5	27.5	87.3	75.7	40.9	31.8	79.5	91.0	53.7	50.5
Baseline + ISM	69.5	73.1	38.9	28.0	88.8	75.8	41.2	32.2	80.3	91.8	54.3	50.9
Low-res	67.4	71.3	37.7	26.2	87.0	75.3	40.8	31.2	78.7	90.0	52.6	49.1
Low-res + ISM	68.9	71.9	38.2	27.4	89.0	76.2	41.3	32.5	79.3	90.9	53.4	50.4
Blur	69.0	72.4	38.5	27.7	88.2	75.8	41.1	32.0	79.7	91.0	53.9	50.5
Blur + ISM	69.2	72.5	38.6	27.7	89.6	76.4	41.8	32.8	79.7	91.1	53.9	50.3
Occlusion	69.3	72.4	38.6	27.9	87.9	75.7	41.1	31.8	80.2	91.2	54.2	50.7
Occlusion + ISM	69.6	72.8	38.8	28.0	89.2	76.4	41.5	32.6	80.6	91.7	54.6	50.9
Mixup	69.8	73.0	39.1	28.1	88.5	75.8	41.0	32.1	80.6	91.8	54.4	50.7
Mixup + ISM	**70.7**	**73.5**	**39.3**	**28.4**	**90.6**	**76.9**	**42.6**	**33.8**	**81.5**	**92.3**	**54.8**	**51.3**

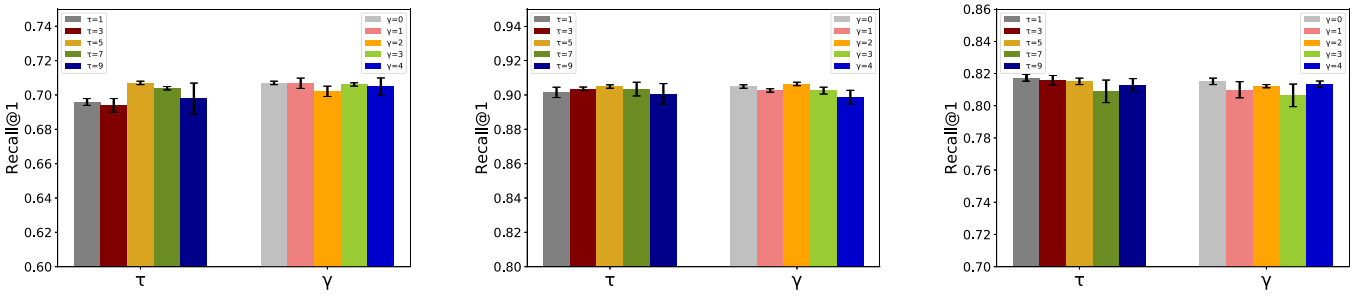
TABLE VII
COMPARISONS OF DIFFERENT TRAINING RECIPES FOR RESNET-50

Procedure	Epochs	Batch Size	Optimizer	LR decay	Rand Augment	Label Smoothing	Stoch.Depth	Mixup	BCE Loss	Top-1 Acc.
PyTorch [52]	90	256	SGD-M	step	✗	✗	✗	✗	✗	76.1
DeiT [71]	300	1024	AdamW	cosine	9/0.5	0.1	0.1	0.8	✗	78.4
FAMS (×4) [13]	400	1024	SGD-M	step	✗	0.1	✗	0.2	✗	79.5
Timm-A1 [84]	600	2048	LAMB	cosine	7/0.5	0.1	0.05	0.2	✓	80.5
Timm-A2 [84]	300	2048	LAMB	cosine	7/0.5	✗	0.05	0.1	✓	79.8
Timm-A3 [84]	100	2048	LAMB	cosine	6/0.5	✗	✗	0.1	✓	78.1
ConvNet [43]	300	4096	AdamW	cosine	9/0.5	0.1	0.1	0.8	✗	78.8
CutMix [95]	300	256	SGD-M	step	✗	✗	✗	✗	✗	78.6
IDML (ours)	300	256	SGD-M	step	✗	✗	✗	✗	✗	**79.0**

We first fixed the dimension of the uncertainty embeddings to 512 and used the dimension of 32, 64, 128, 256, 512, and 1,024 for the semantic embeddings as shown in Table IX. We observe that the performance improves as the size of the semantic embedding increases and reaches the top at 512 and 1,024. We also see that using a dimension of 1,024 does not prominently enhance the results, which might result from the information redundancy.

Furthermore, we fixed the semantic embedding size to 512 and tested the performance using the dimension of 32, 64, 128, 256, 512, and 1,024 for the uncertainty embeddings, as shown in Table X. We see that the performance gradually improves as the uncertainty embedding size increases, and the model achieves the best result when the dimension is 512. In summary, we find that using the dimension of 512 for both the semantic embeddings and the uncertainty embeddings achieves the best accuracy/computation trade-off, and we thus adopted them in the main experiments.

7) Effect of the Metric Formulation During Training: When uncertain, the proposed metric tends to treat the pair similarly



(a) CUB-200-2011 (b) Cars196 (c) Stanford Online Products

Fig. 7. Impact of metric parameters on the CUB-200-2011, Cars196 and Stanford Online Products datasets.

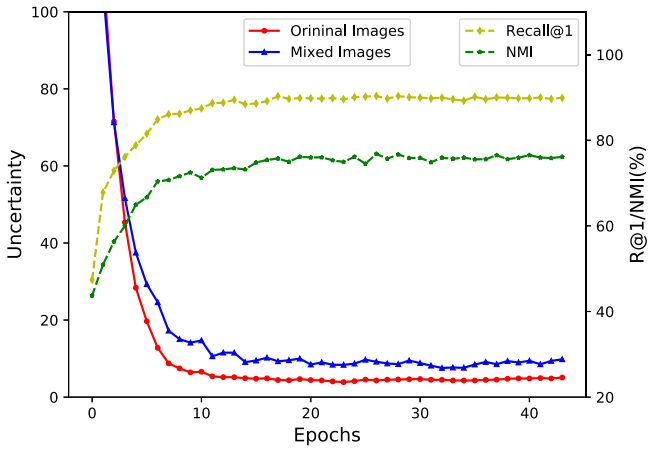


Fig. 8. Uncertainty trend during training on the Cars196 dataset.

TABLE VIII
ANALYSIS OF THE BATCH SIZE (BS) ON THE CUB-200-2011, CARS196, AND STANFORD ONLINE PRODUCTS DATASETS

	CUB-200-2011		Cars196		SOP	
BS	R@1	NMI	R@1	NMI	R@1	NMI
40	66.3	70.6	88.2	74.1	78.1	89.5
60	67.1	71.3	88.9	75.3	79.9	91.3
80	68.5	72.0	89.6	75.6	80.5	91.6
100	69.6	72.5	90.0	76.3	81.2	92.1
120	**70.7**	73.5	90.6	76.9	81.5	**92.3**
140	70.5	73.2	90.5	77.2	**81.6**	92.2
160	**70.7**	73.3	**90.7**	76.8	81.3	**92.3**
180	**70.7**	**73.6**	90.5	**77.5**	81.3	92.1

TABLE IX
ANALYSIS OF THE SEMANTIC EMBEDDING (SE) SIZE ON THE CUB-200-2011, CARS196, AND STANFORD ONLINE PRODUCTS DATASETS

	CUB-200-2011		Cars196		SOP	
SE Size	R@1	NMI	R@1	NMI	R@1	NMI
32	56.9	64.0	76.8	68.0	73.0	86.2
64	63.1	67.2	83.1	70.6	77.3	89.1
128	66.4	70.2	86.0	73.7	79.6	91.1
256	68.5	71.7	89.2	76.1	80.8	92.0
512	70.7	73.5	**90.6**	**76.9**	81.5	**92.3**
1024	**71.0**	**73.6**	90.3	76.4	**81.7**	**92.3**

TABLE X
ANALYSIS OF THE UNCERTAINTY EMBEDDING (UE) SIZE ON THE CUB-200-2011, CARS196, AND STANDORD ONLINE PRODUCTS DATASETS

	CUB-200-2011		Cars196		SOP	
UE Size	R@1	NMI	R@1	NMI	R@1	NMI
32	69.4	72.5	89.2	75.9	80.3	91.3
64	69.2	72.9	89.8	76.5	80.6	91.5
128	69.5	73.3	89.8	76.7	81.1	92.0
256	70.0	73.2	90.1	76.8	81.4	92.1
512	**70.7**	**73.5**	**90.6**	**76.9**	**81.5**	92.3
1024	70.4	73.4	90.4	76.7	81.4	**92.4**

TABLE XI
EFFECT OF DIFFERENT METRIC FORMULATIONS DURING TRAINING

Dataset	Training Metric	R@1	NMI	RP	M@R
CUB-200-2011	Euclidean	69.0	72.3	38.5	27.5
	ISM-Dis (22)	69.2	72.0	38.7	28.1
	ISM-Sim (21)	**70.7**	**73.5**	**39.3**	**28.4**
Cars196	Euclidean	87.3	75.7	40.9	31.8
	ISM-Dis (22)	89.4	75.4	41.6	32.5
	ISM-Sim (21)	**90.6**	**76.9**	**42.6**	**33.8**
SOP	Euclidean	79.5	91.0	53.7	50.5
	ISM-Dis (22)	80.2	91.3	54.0	50.6
	ISM-Sim (21)	**81.5**	**92.3**	**54.8**	**51.3**

since we think an uncertain metric should not be able to differentiate all pairs. The proposed introspective similarity metric (ISM-Sim) based on the cosine similarity is defined as follows:

$$C_{IN}(\mathbf{x}_i, \mathbf{p}_j) = 1 - (1 - C(\mathbf{x}_i, \mathbf{p}_j)) \cdot e^{(-\frac{1}{\tau}\ \text{r\_conf}(\mathbf{x}_i, \mathbf{p}_j))}. \tag{21}$$

Alternatively, we may also weaken the similarity judgment by encouraging the metric to output large distances to all uncertainty pairs. As a comparison, we additionally modified the metric to treat each ambiguous pair dissimilar (ISM-Dis) as follows:

$$C_{IN}(\mathbf{x}_i, \mathbf{p}_j) = C(\mathbf{x}_i, \mathbf{p}_j) \cdot e^{(-\frac{1}{\tau}\ \text{r\_conf}(\mathbf{x}_i, \mathbf{p}_j))}. \tag{22}$$

We conducted experiments on the CUB-200–2011 and Cars196 datasets to test the performances of using different metric formulations for training in Table XI. We observe that treating each ambiguous pair dissimilar performs worse than the original

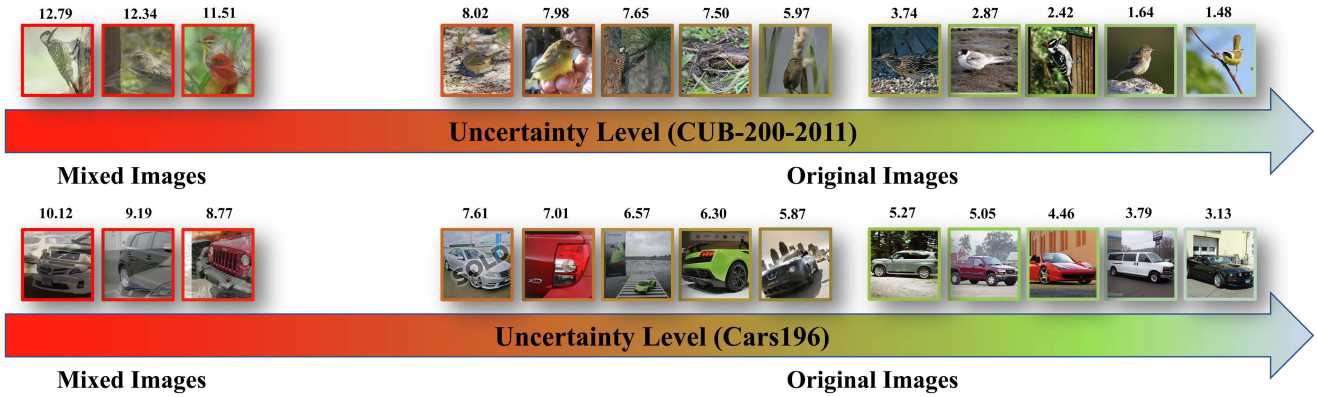
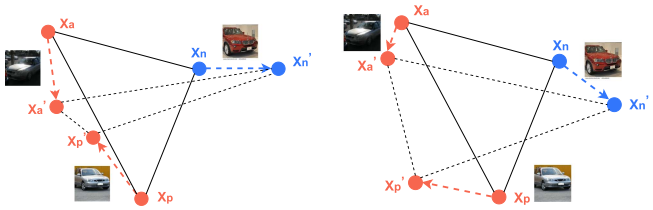
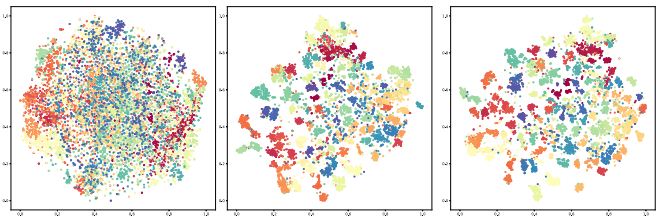


Fig. 9. Uncertainty levels produced by the proposed IDML framework on the test split of CUB-200-2011 and Cars196.



(a) Effect of the triplet loss. (b) Effect of IDML-Triplet.

Fig. 10. t-SNE [73] analysis of one-step updating of embeddings.



(a) Initialization (b) Proxy Anchor (c) IDML

Fig. 11. t-SNE visualization of the embeddings.

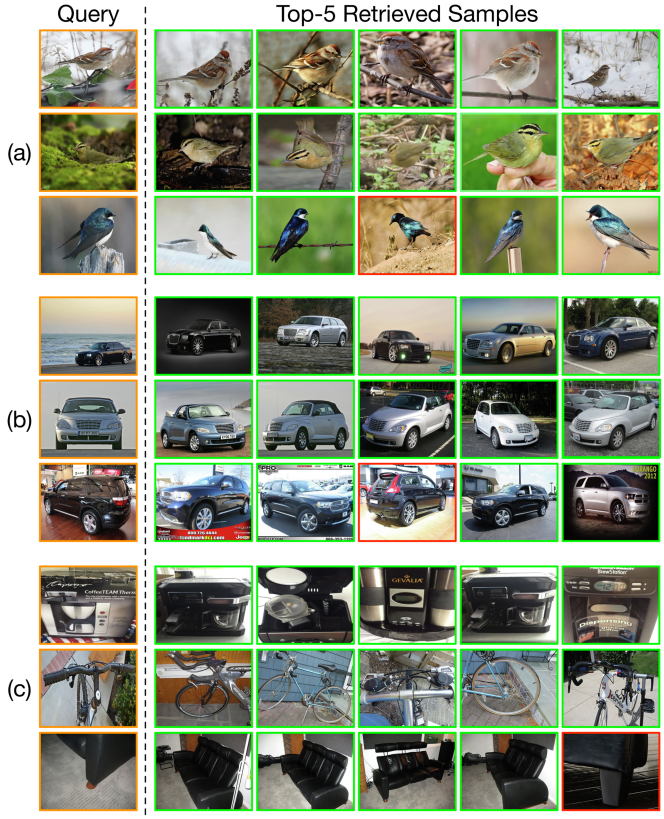


Fig. 12. Visualization of the top-5 retrieved samples of our IDML framework on (a) CUB-200-2011, (b) Cars196, and (c) Stanford Online Products datasets. We use the orange, green, and red color to denote the query, positive, and negative sample, respectively.

metric. This verifies our motivation for using uncertainty to weaken the semantic discrepancy.

8) Effect of the Metric Formulation During Testing: During testing, we adopt the original similarity metric without our uncertainty-aware term. As an alternative, we conducted an experiment using the introspective similarity metric (ISM) during testing on the CUB-200-2011 and Cars196 datasets, as shown in Table XII. We observe a decrease in performance when using ISM during testing, indicating a harmful effect of using uncertainty to weaken the similarity discrepancy during inference. This is reasonable since providing a clear and confident similarity judgment is more beneficial to discriminative tasks such as image classification.

9) Application on Different Backbone Architectures: We applied our IDML framework to different backbone architectures to show its generalization. Given the recent success of Vision Transformers (ViTs) [14], we adopt DeiT [71] as the backbone model. ViTs first formulate each image into a sequence of patches and flatten them to construct patch tokens. They additionally use a class token to extract the image-level feature. To apply the proposed IDML framework to ViTs, we further employ an uncertainty token for uncertainty measure. We treat the output class token and uncertainty token as the semantic embedding and uncertainty embedding, respectively. We then employ the our ISM to replace the cosine similarity in the classifier and use the same softmax loss during training. We discard the uncertainty token during inference and use the same architecture as the original transformer without extra computation cost. The experimental results are shown in Table XIII. We observe

TABLE XII
EFFECT OF DIFFERENT METRIC FORMULATIONS DURING TESTING

Dataset	Testing Metric	R@1	NMI	RP	M@R
CUB-200-2011	Euclidean (baseline)	69.0	72.3	38.5	27.5
	ISM (IDML)	69.8	73.1	39.0	27.8
	Euclidean (IDML)	**70.7**	**73.5**	**39.3**	**28.4**
Cars196	Euclidean (baseline)	87.3	75.7	40.9	31.8
	ISM (IDML)	89.9	76.2	42.3	33.3
	Euclidean (IDML)	**90.6**	**76.9**	**42.6**	**33.8**
SOP	Euclidean (baseline)	79.5	91.0	53.7	50.5
	ISM (IDML)	80.7	92.1	54.2	50.8
	Euclidean (IDML)	**81.5**	**92.3**	**54.8**	**51.3**

TABLE XIII
APPLICATION OF IDML ON VISION TRANSFORMERS

Network	#Param.	FLOPs	Image Size	Top-1 Acc.
DeiT-Ti [71]	5M	1.3G	224×224	72.2
DeiT-Ti-IDML	5M	1.3G	224×224	**72.7 (+0.5)**
DeiT-S [71]	22M	4.6G	224×224	79.8
DeiT-S-IDML	22M	4.6G	224×224	**80.5 (+0.7)**

consistent performance boosts with relatively large margins when applying IDML to both DeiT-Tiny and DeiT-Small, showing the universality of IDML.

10) Uncertainty Trend During Training: To demonstrate that our IDML framework properly handles mixed images with high uncertainty, we visualize the trend of uncertainty level for both original images and mixed images during training, as shown in Fig. 8. We define the uncertainty level of an image to be the L2-norm of its uncertainty embedding. We see that the uncertainty decreases for both original and mixed images as the training proceeds and becomes stable as the model converges. We also observe that the uncertainty level for mixed images is larger than that of original images. This verifies that our framework can indeed learn the uncertainty in images.

11) Uncertainty Levels on the Test Split: We visualize the uncertainty levels on the test split of the CUB-200-2011 and Cars196 datasets, as shown in Fig. 9. We obtain the uncertainty levels of the mixed images together with original images in the test set after the model converges. We observe that the uncertainty of mixed images is much larger than that of the original test images since the mixed images contain the information of two images. Also, we see that several original images result in relatively higher uncertainty than others because of the natural noise such as occlusion and improper directions. This further verifies that the proposed framework can successfully learn the uncertainty in images.

12) Qualitative Analysis of the Learning Process: We provide a t-SNE [73] visualized analysis of how embeddings are learned using a toy example on the Cars196 dataset in Fig. 10. We visualized the embeddings before and after updating the model with one gradient step. For a dark (and ambiguous) image $\mathbf{x}_\alpha$, the original method pulls it quite closer to the positive sample $\mathbf{x}_p$, while IDML is more cautious and only slightly pulls it together due to the uncertainty to prevent the influence of possible noise.

13) Visualization of the Embeddings: We employ t-SNE to visualize the embeddings in Fig. 11 and use colors to differentiate different classes. We observe that both methods can cluster similar samples, but we can hardly identify the better one due to the large number of samples.

14) Qualitative Results: We provide a visualization of several retrieved examples of the proposed IDML framework from the CUB-200-2011, Cars196, and Stanford Online Products datasets in Fig. 12. The main challenge is the various backgrounds, poses, and viewpoints for the same class on CUB-200-2011, Cars196, and Stanford Online Products, respectively. Still, our framework can successfully identify the positive samples despite the large intraclass variations. Additionally, the failure cases only possess subtle differences with the query, which are difficult even for humans to capture.

V. CONCLUSION

In this paper, we have presented an introspective deep metric learning framework to more effectively process the semantic uncertainty in training data for better performance. We represent an image with a semantic embedding and an uncertainty embedding to model the semantic characteristics and the uncertainty, respectively. We have further proposed an introspective similarity metric to compute an uncertainty-aware similarity score, which weakens semantic discrepancies for uncertain images. We have performed various experiments on six benchmark datasets on both image retrieval and image classification to analyze the effectiveness of our framework. Experimental results have demonstrated a constant performance boost to various methods in different settings.

REFERENCES

[1] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, "Neural codes for image retrieval," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 584–599.
[2] F. Cakir, K. He, X. Xia, B. Kulis, and S. Sclaroff, "Deep metric learning to rank," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1861–1870.
[3] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 9912–9924.
[4] J. Chang, Z. Lan, C. Cheng, and Y. Wei, "Data uncertainty learning in face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 5710–5719.
[5] K. Chen, L. Hong, H. Xu, Z. Li, and D.-Y. Yeung, "MultiSiam: Self-supervised multi-instance Siamese representation learning for autonomous driving," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 7546–7554.
[6] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. 37th Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.
[7] X. Chen and K. He, "Exploring simple Siamese representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 15750–15758.
[8] H.-P. Chou, S.-C. Chang, J.-Y. Pan, W. Wei, and D.-C. Juan, "Remix: Rebalanced mixup," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 95–110.
[9] S. Chun, S. J. Oh, R. S. de Rezende, Y. Kalantidis, and D. Larlus, "Probabilistic embeddings for cross-modal retrieval," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8415–8424.
[10] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4690–4699.
[11] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," 2017, arXiv: *1708.04552*.

[12] 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.*, 2019, pp. 10404–10413.

[13] P. Dollár, M. Singh, and R. Girshick, "Fast and accurate model scaling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 924–932.

[14] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Representations*, 2020, pp. 1–22.

[15] Y. Duan, W. Zheng, X. Lin, J. Lu, and J. Zhou, "Deep adversarial metric learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2780–2789.

[16] D. Dwibedi, Y. Aytar, J. Tompson, P. Sermanet, and A. Zisserman, "With a little help from my friends: Nearest-neighbor contrastive learning of visual representations," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 9588–9597.

[17] Y. Feng, J. Jiang, M. Tang, R. Jin, and Y. Gao, "Rethinking supervised pre-training for better downstream transferring," in *Proc. Int. Conf. Learn. Representations*, 2021, pp. 1–22.

[18] W. Ge, W. Huang, D. Dong, and M. R. Scott, "Deep metric learning with hierarchical triplet loss," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 269–285.

[19] G. Ghiasi, T.-Y. Lin, and Q. V. Le, "DropBlock: A regularization method for convolutional networks," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 10750–10760.

[20] J.-B. Grill et al., "Bootstrap your own latent-a new approach to self-supervised learning," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 21271–21284.

[21] S. Guo, J. Xu, D. Chen, C. Zhang, X. Wang, and R. Zhao, "Density-aware feature embedding for face clustering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 6698–6706.

[22] D. Han, J. Kim, and J. Kim, "Deep pyramidal residual networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5927–5935.

[23] B. Harwood, V. Kumar, B. G. G. Carneiro, I. Reid, and T. Drummond, "Smart mining for deep metric learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2017, pp. 2840–2848.

[24] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9729–9738.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[26] J. R. Hershey and P. A. Olsen, "Approximating the Kullback Leibler divergence between Gaussian mixture models," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2007, pp. IV-317–IV-320.

[27] G. Hinton et al., "Distilling the knowledge in a neural network," in *Proc. Adv. Neural Inf. Process. Syst. Workshop*, 2015, pp. 1–9.

[28] J. Hou, S. Xie, B. Graham, A. Dai, and M. Nießner, "Pri3D: Can 3D priors help 2D representation learning?," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 5693–5702.

[29] J. Hu, J. Lu, and Y.-P. Tan, "Discriminative deep metric learning for face verification in the wild," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1875–1882.

[30] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, "Deep networks with stochastic depth," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 646–661.

[31] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 448–456.

[32] M. Kemertas, L. Pishdad, K. G. Derpanis, and A. Fazly, "RankMI: A mutual information maximizing ranking loss," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 14362–14371.

[33] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5574–5584.

[34] P. Khosla et al., "Supervised contrastive learning," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 18661–18673.

[35] J.-H. Kim, W. Choo, and H. O. Song, "Puzzle mix: Exploiting saliency and local statistics for optimal mixup," in *Proc. 37th Int. Conf. Mach. Learn.*, 2020, pp. 5275–5285.

[36] 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.*, 2020, pp. 3238–3247.

[37] W. Kim, B. Goyal, K. Chawla, J. Lee, and K. Kwon, "Attention-based ensemble for deep metric learning," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 760–777.

[38] 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*, 2013, pp. 554–561.

[39] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Tech. Rep., Univ. Toronto, 2009, pp. 1–60.

[40] W. Li, X. Huang, J. Lu, J. Feng, and J. Zhou, "Learning probabilistic ordinal embeddings for uncertainty-aware regression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 13896–13905.

[41] H. Liang et al., "Exploring geometry-aware contrast and clustering harmonization for self-supervised 3D object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 3293–3302.

[42] S. Liu, Z. Li, and J. Sun, "Self-EMD: Self-supervised object detection without ImageNet," 2020, arXiv: *2011.13677*.

[43] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," 2022, arXiv: *2201.03545*.

[44] T. Milbich et al., "DiVA: Diverse visual feature aggregation fordeep metric learning," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 590–607.

[45] Y. Movshovitz-Attias, A. Toshev, T. K. Leung, S. Ioffe, and S. Singh, "No fuss distance metric learning using proxies," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2017, pp. 360–368.

[46] K. Musgrave, S. Belongie, and S.-N. Lim, "A metric learning reality check," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 681–699.

[47] A. Neelakantan, J. Shankar, A. Passos, and A. McCallum, "Efficient non-parametric estimation of multiple embeddings per word in vector space," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2014, pp. 1059–1069.

[48] D. Q. Nguyen, D. Q. Nguyen, A. Modi, S. Thater, and M. Pinkal, "A mixture model for learning multi-sense word embeddings," 2017, arXiv: *1706.05111*.

[49] H. V. Nguyen and L. Bai, "Cosine similarity metric learning for face verification," in *Proc. Asian Conf. Comput. Vis.*, 2010, pp. 709–720.

[50] S. J. Oh, K. P. Murphy, J. Pan, J. Roth, F. Schroff, and A. C. Gallagher, "Modeling uncertainty with hedged instance embeddings," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–17.

[51] 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.

[52] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 8026–8037.

[53] Q. Qian, L. Shang, B. Sun, and J. Hu, "SoftTriple loss: Deep metric learning without triplet sampling," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 6450–6458.

[54] K. Roth, B. Brattoli, and B. Ommer, "MIC: Mining interclass characteristics for improved metric learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 8000–8009.

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

[56] K. Roth, O. Vinyals, and Z. Akata, "Non-isotropy regularization for proxy-based deep metric learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 7420–7430.

[57] O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.

[58] A. Sanakoyeu, V. Tschernezki, U. Buchler, and B. Ommer, "Divide and conquer the embedding space for metric learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 471–480.

[59] M. B. Sariyildiz, Y. Kalantidis, K. Alahari, and D. Larlus, "Improving the generalization of supervised models," 2022, arXiv: *2206.15369*.

[60] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 815–823.

[61] G. Shaw and D. Manolakis, "Signal processing for hyperspectral image exploitation," *IEEE Signal Process. Mag.*, vol. 19, no. 1, pp. 12–16, Jan. 2002.

[62] Y. Shi and A. K. Jain, "Probabilistic face embeddings," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 6902–6911.

[63] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, arXiv: *1409.1556*.

[64] K. Sohn, "Improved deep metric learning with multi-class N-pair loss objective," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 1857–1865.

[65] H. O. Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4004–4012.
[66] J. J. Sun, J. Zhao, L.-C. Chen, F. Schroff, H. Adam, and T. Liu, "View-invariant probabilistic embedding for human pose," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 53–70.
[67] Y. Sun et al., "Circle loss: A unified perspective of pair similarity optimization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 6398–6407.
[68] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2818–2826.
[69] C. Szegedy et al., "Going deeper with convolutions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.
[70] E. W. Teh, T. DeVries, and G. W. Taylor, "ProxyNCA: Revisiting and revitalizing proxy neighborhood component analysis," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 448–464.
[71] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *Proc. 38th Int. Conf. Mach. Learn.*, 2021, pp. 10347–10357.
[72] A. S. Uddin, M. S. Monira, W. Shin, T. Chung, and S.-H. Bae, "SaliencyMix: A saliency guided data augmentation strategy for better regularization," in *Proc. Int. Conf. Learn. Representations*, 2020, pp. 1–12.
[73] L. Van Der Maaten, "Accelerating t-SNE using tree-based algorithms," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 3221–3245, 2014.
[74] V. Verma et al., "Manifold mixup: Better representations by interpolating hidden states," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 6438–6447.
[75] L. Vilnis and A. McCallum, "Word representations via Gaussian embedding," in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 1–12.
[76] C. Wah, S. Branson, P. Welinder, P. Perona, and S. J. Belongie, "The Caltech-UCSD Birds-200–2011 dataset," California Inst. Technol., Pasadena, CA, Tech. Rep. CNS-TR-2011-001, 2011.
[77] F. Wang, H. Wang, C. Wei, A. Yuille, and W. Shen, "CP2: Copy-paste contrastive pretraining for semantic segmentation," 2022, arXiv: *2203.11709*.
[78] H. Wang et al., "CosFace: Large margin cosine loss for deep face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5265–5274.
[79] J. Wang, F. Zhou, S. Wen, X. Liu, and Y. Lin, "Deep metric learning with angular loss," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2017, pp. 2593–2601.
[80] J. Wang et al., "Learning fine-grained image similarity with deep ranking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1386–1393.
[81] 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.*, 2019, pp. 5207–5216.
[82] 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.*, 2019, pp. 5022–5030.
[83] X. Wang, H. Zhang, W. Huang, and M. R. Scott, "Cross-batch memory for embedding learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 6388–6397.
[84] R. Wightman, H. Touvron, and H. Jégou, "ResNet strikes back: An improved training procedure in TIMM," 2021, arXiv: *2110.00476*.
[85] C.-Y. Wu, R. Manmatha, A. J. Smola, and P. Krähenbühl, "Sampling matters in deep embedding learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2017, pp. 2859–2867.
[86] E. Xie et al., "DetCo: Unsupervised contrastive learning for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 8392–8401.
[87] S. Xie, J. Gu, D. Guo, C. R. Qi, L. Guibas, and O. Litany, "PointContrast: Unsupervised pre-training for 3D point cloud understanding," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 574–591.
[88] H. Xuan, R. Souvenir, and R. Pless, "Deep randomized ensembles for metric learning," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 723–734.
[89] L. Yang, X. Zhan, D. Chen, J. Yan, C. C. Loy, and D. Lin, "Learning to cluster faces on an affinity graph," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2298–2306.
[90] M. Ye and J. Shen, "Probabilistic structural latent representation for unsupervised embedding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 5457–5466.
[91] B. Yu and D. Tao, "Deep metric learning with tuplet margin loss," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 6490–6499.
[92] D. Yu, K. Yao, H. Su, G. Li, and F. Seide, "KL-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2013, pp. 7893–7897.
[93] 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.*, 2019, pp. 4815–4824.
[94] Y. Yuan, K. Yang, and C. Zhang, "Hard-aware deeply cascaded embedding," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2017, pp. 814–823.
[95] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "CutMix: Regularization strategy to train strong classifiers with localizable features," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 6023–6032.
[96] B. Zhang and P. Wonka, "Point cloud instance segmentation using probabilistic embeddings," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8883–8892.
[97] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–13.
[98] L. Zhang, T. Xiang, and S. Gong, "Learning a deep embedding model for zero-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2021–2030.
[99] S. Zhang, R. Xu, C. Xiong, and C. Ramaiah, "Use all the labels: A hierarchical multi-label contrastive learning framework," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 16639–16648.
[100] W. Zhao, Y. Rao, Z. Wang, J. Lu, and J. Zhou, "Towards interpretable deep metric learning with structural matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 9887–9896.
[101] W. Zheng, Z. Chen, J. Lu, and J. Zhou, "Hardness-aware deep metric learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 72–81.
[102] W. Zheng, C. Wang, J. Lu, and J. Zhou, "Deep compositional metric learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 9320–9329.
[103] W. Zheng, B. Zhang, J. Lu, and J. Zhou, "Deep relational metric learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 12065–12074.
[104] H. Zhu, Y. Yuan, G. Hu, X. Wu, and N. Robertson, "Imbalance robust Softmax for deep embedding learning," in *Proc. Asian Conf. Comput. Vis.*, 2020, pp. 274–291.

Chengkun Wang received the BS degree from the Department of Electrical Engineering, Tsinghua University, China, in 2020. He is currently working toward the PhD degree with the Department of Automation, Tsinghua University. His current research interests include computer vision, metric learning, and representation learning. He has authored conference papers in CVPR and ICCV. He serves as a regular reviewer member for a number of journals and conferences, e.g., CVPR, *IEEE Transactions on Image Processing*, ICME, ICIP, and *IEEE Transactions on Neural Networks and Learning Systems*.

Wenzhao Zheng (Student Member, IEEE) received the BS degree from the Department of Physics, Tsinghua University, China, in 2018. He is currently working toward the PhD degree with the Department of Automation, Tsinghua University. His current research interests include computer vision, deep learning, and metric learning. He has authored more than 10 papers in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *IEEE Transactions on Image Processing*, CVPR, ICCV, and ECCV. He serves as a regular reviewer member for a number of journals and conferences, e.g., *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *IEEE Transactions on Image Processing*, *IEEE Transactions on Biometrics, Behavior, and Identity Science*, *ACM Transactions on Intelligent Systems and Technology*, CVPR, ICCV, ECCV, IJCAI, ICME, and ICIP.

Zheng Zhu (Member, IEEE) received the PhD degree from the Institute of Automation, Chinese Academy of Sciences, in 2019. He is currently a postdoctoral fellow with Tsinghua University. He served reviewers in various journals and conferences including *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *IEEE Transactions on Image Processing*, *IEEE Transactions on Multimedia*, *IEEE Transactions on Circuits and Systems for Video Technology*, CVPR, ICCV, ECCV, ICLR. He has co-authored more than 40 journal and conference papers mainly on computer vision and robotics problems, such as face recognition, visual tracking, human pose estimation, and servo control. He has more than 3,000 Google Scholar citations to his work. He organized the Masked Face Recognition Challenge and Workshop in ICCV 2021. He ranked the 1st on NIST-FRVT Masked Face Recognition, won the COCO Keypoint Detection Challenge in ECCV 2020 and Visual Object Tracking (VOT) Real-Time Challenge in ECCV 2018.

Jie Zhou (Senior Member, IEEE) received the BS and MS degrees both from the Department of Mathematics, Nankai University, China, in 1990 and 1992, respectively, and the PhD degree from the Institute of Pattern Recognition and Artificial Intelligence, Huazhong University of Science and Technology (HUST), China, in 1995. From then to 1997, he served as a postdoctoral fellow with the Department of Automation, Tsinghua University, China. Since 2003, he has been a full professor with the Department of Automation, Tsinghua University. His research interests include computer vision and pattern recognition. In recent years, he has authored more than 100 papers have been published in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *IEEE Transactions on Image Processing* and CVPR. He is an associate editor for *IEEE Transactions on Pattern Analysis and Machine Intelligence* and two other journals. He received the National Outstanding Youth Foundation of China Award. He is a fellow of the IAPR.

Jiwen Lu (Fellow, IEEE) received the BEng degree in mechanical engineering and the MEng degree in electrical engineering from the Xi'an University of Technology, China, in 2003 and 2006, respectively, and the PhD degree in electrical engineering from Nanyang Technological University, Singapore, in 2012. He is currently an associate professor with the Department of Automation, Tsinghua University, China. His current research interests include computer vision and pattern recognition. He was/is a member of the Multimedia Signal Processing Technical Committee and the Information Forensics and Security Technical Committee of the IEEE Signal Processing Society, and a member of the Multimedia Systems and Applications Technical Committee and the Visual Signal Processing and Communications Technical Committee of the IEEE Circuits and Systems Society. He serves as the co-editor-of-chief for *Physical Review Letters*, an associate editor for the *IEEE Transactions on Image Processing*, *IEEE Transactions on Circuits and Systems for Video Technology*, *IEEE Transactions on Biometrics, Behavior, and Identity Science*, and *Pattern Recognition*. He also serves as the program co-chair of IEEE FG'2023, VCIP'2022, AVSS'2021 and ICME'2020. He received the National Outstanding Youth Foundation of China Award. He is a fellow of the IAPR.