Deep Causal Metric Learning

Xiang Deng [1] **Zhongfei Zhang** [1]

Abstract

Deep metric learning aims to learn distance metrics that measure similarities and dissimilarities between samples. The existing approaches typically focus on designing different hard sample mining or distance margin strategies and then minimize a pair/triplet-based or proxy-based loss over the training data. However, this can lead the model to recklessly learn all the correlated distances found in training data including the spurious distance (e.g., background differences) that is not the distance of interest and can harm the generalization of the learned metric. To address this issue, we study metric learning from a causality perspective and accordingly propose deep causal metric learning (DCML) that pursues the true causality of the distance between samples. DCML is achieved through explicitly learning environment-invariant attention and task-invariant embedding based on causal inference. Extensive experiments on several benchmark datasets demonstrate the superiority of DCML over the existing methods [1].

1. Introduction

Deep metric learning (DML) is an effective technique to automatically learn a task-specific distance metric with a deep neural network (DNN). It has received much attention in recent years due to its wide applications in different domains, such as image retrieval (Li & Tang, 2015; Wu et al., 2017), self-supervised learning (He et al., 2020; Chen et al., 2020), few-shot learning (Sung et al., 2018; Snell et al., 2017), and face recognition (Wang et al., 2017a; Liu et al., 2017; Wang et al., 2018a). The existing approaches (Suh et al., 2019; Sun et al., 2020; Wang et al., 2018b) mainly focus on designing different sampling strategies (e.g., hard or informative sample mining) or adopting different kinds of distance margins (e.g., an angular or similarity margin). Then a pair/triplet-based or proxy-based loss combining with these strategies is minimized over the training data (Hadsell et al., 2006; Weinberger et al., 2006; Sun et al., 2020). However, simply minimizing the loss over the training data makes the learned metric model recklessly absorb all the correlated distances between samples, even the spurious distances caused by context biases, which hurts the generalization on new (test) classes or samples. In this paper, different from the existing literature designing different sampling or margin strategies, we study DML from a different perspective by exploring the true causality of the distance between samples.

In DML, the distance between a pair of samples varies with the tasks (i.e., learning goals). For example, with two images of tigers in forests, one task is to learn the distance between the tigers in the two images while another task can be learning the distance between the forests in the two images, which results in different distances. The backgrounds and foregrounds (i.e., object) can be switched based on the tasks. However, backgrounds and objects are typically highly correlated in reality, e.g., tigers usually appear in forests instead of water while fish does the opposite. The high correlation between an object and a background makes DML more likely suffer from background (context) biases in the training data, since the classes in the training dataset can be totally different from those in the test dataset in the DML, which is different from the case in classification where the training dataset and the test dataset share the same classes (He et al., 2016; Deng & Zhang, 2021b). An intuitive example is provided in Figure 1 where the task is to learn the distance metric between animals. In the training dataset, there are two classes of animals, i.e., tigers and Alaskan dogs. Since most tigers live in forests and most Alaskan dogs live in snow, when the model learns the distances between tigers and Alaskan dogs, it also absorbs the spurious distances between forests and snow due to the high correlations between tigers and forests, and between Alaskan dogs and snow. In the test dataset with two new classes, i.e., wolves and lions, the learned biased metric mistakenly infers that the wolf in forests (i.e., the query image) is more similar to the lion in forests instead of the two wolves in snow due to the biases induced by the context prior in the training data. Without dealing with the kind of context biases in the training data, it hurts the generalization

[1]Department of Computer Science, State University of New York at Binghamton, NY, US. Correspondence to: Xiang Deng <xdeng7@binghamton.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

[1]Code: https://github.com/Xiang-Deng-DL/DCML.

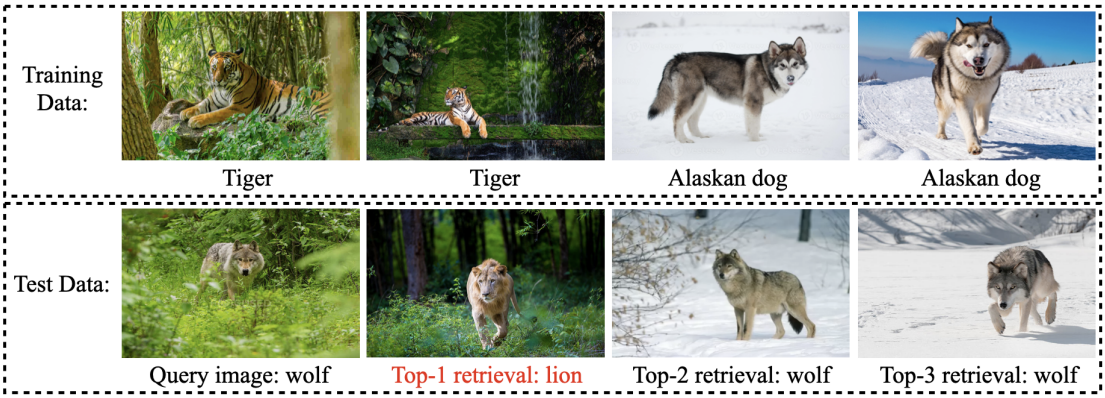


Figure 1. Biased distance metric induced by context prior.

of the learned metric on new (test) classes or samples where the context distribution is different from that in the training data. In light of this, we propose deep causal metric learning (DCML) to learn the true distance between two samples by removing the bad effects of context biases.

DCML is achieved by explicitly learning environment-invariant attention and task-invariant embedding based on causal inference. To construct different context environments, we re-weight the training samples to change the context distribution; since different samples contain different context information, re-weighting the samples is equivalent to generating different context prior distributions. Instead of randomly generating sample weights, we explicitly learn the environments that the attention and embedding are not optimal or consistent across. We iteratively train the metric model and generate new environments, which leads to a causal distance metric that is not affected by context priors.

Our main contributions are summarized as follows:

- Different from the existing DML approaches that focus on designing different sampling or distance margin strategies for pair/triplet-based or proxy-based losses, we study DML from a different perspective by proposing deep causal metric learning (DCML) to pursue the true causality of the distances between samples.
- We design a novel metric learning framework, i.e., DCML, that learns the causal distance between samples through explicitly learning context-environment-invariant attention and task-invariant embedding based on causal inference.
- Extensive experiments on several benchmark datasets demonstrate that DCML outperforms the state-of-the-art approaches substantially.

2. Related Work

Pair and Triplet-Based Metric Learning. Pair and triplet-based metric learning approaches design the loss based on sample pairs or triplets. The contrastive loss (Hadsell et al., 2006) trains the metric model by making the distance between a positive pair less than a threshold, and the distance between a negative pair greater than a threshold. Instead of directly penalizing the distance between a pair, the triplet loss (Weinberger et al., 2006) uses an anchor, an positive sample, and a negative sample and enforces the anchor-positive distances to be smaller than the anchor-negative distances by a threshold. Ever since then, many efforts have been made on improving these two losses. The angular loss (Wang et al., 2017b) modifies the triple loss by taking into account the angle relationship among triplets. The margin loss (Wu et al., 2017) improves the contrastive loss by using a distance weighted sampling strategy and a learnable variable to tune the distance threshold. Kim et al. (2018) apply ensemble techniques on DML, which improves the performance but also introduces substantial overheads. RML-DGATs (Wang et al., 2020b) introduces attention mechanism to relational metric learning but fails to deal with data biases. Other pair/triplet-based approaches mainly focus on designing a new (margin) threshold strategy (Yu & Tao, 2019), using a different sampling strategy (Suh et al., 2019; Sun et al., 2020; Liu et al., 2021; Wang et al., 2020c; Aziere & Todorovic, 2019; Zheng et al., 2019), considering more pairs in the loss (Sohn, 2016; Oh Song et al., 2016; Wang et al., 2019), normalizing the distance between pairs by using softmax or other similar functions (Oh Song et al., 2016; Cakir et al., 2019; Zhang et al., 2020a), or using additional unlabelled data or extra embedding spaces (Duan et al., 2021; Roth et al., 2021; Zheng et al., 2021).

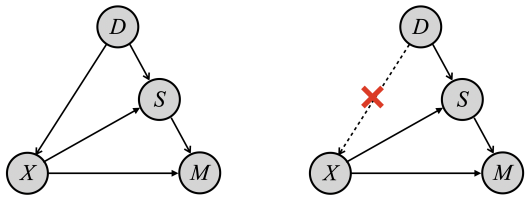
Proxy-Based Metric Learning. Pair and triplet-based approaches have high sampling complexities and thus lead to low convergence, i.e., the sampling complexities of pair and triplet losses are $\mathcal{O}(n^2)$ and $\mathcal{O}(n^3)$ where n is the number of samples, respectively. Proxy-based approaches address this issue by assuming that there is a class proxy vector for each class. They train a metric model by making the samples close to their ground-truth class proxies while far from the other class proxies with a cross-entropy loss, which

leads to a sampling complexity $\mathcal{O}(n)$. These class proxy vectors are discarded after the training process and only the learned embedding is used to calculate the distances among samples. NormFace (Wang et al., 2017a) uses the cosine similarity between an embedding vector and the class proxy vectors as the training logits. SphereFace (Liu et al., 2017) takes a further step by considering sphere embedding. Zhai & Wu (2018) further introduce the temperature to the cross-entropy loss for DML. ProxyNCA (Movshovitz-Attias et al., 2017) employs the cross-entropy loss on the Euclidean distances between an embedding vector and the class proxy vectors. AM-Softmax (Wang et al., 2018a) and COS (Wang et al., 2018b) introduce a cosine similarity margin to the cross-entropy loss. Arcface (Deng et al., 2019) further introduces an angular margin. Softtriple (Qian et al., 2019) uses multiple class proxy vectors for each class instead of one proxy vector. Recently, many proxy-based approaches (Teh et al., 2020; Seidenschwarz et al., 2021; Yang et al., 2022; Deng et al., 2022) have been proposed to use different techniques, e.g., message-passing and hierarchical proxies, to improve the embedding for metric learning.

Causal Inference. Causal inference (Pearl et al., 2016; Pearl & Mackenzie, 2018; Rubin, 2019; Wang & Blei, 2019) is an effective technique to identify the cause-effect relationships between different variables. In recent years, it has been introduced to machine learning (Bengio et al., 2019) and has been used to address the challenges in different tasks, including but not limited to domain adaptation or generalization (Gong et al., 2016; Magliacane et al., 2018; Wang et al., 2021b; Teney et al., 2021; Zhang et al., 2021), imitation learning (de Haan et al., 2019), scene graph generation (Tang et al., 2020b), image captioning (Yang et al., 2020), imbalance classification (Tang et al., 2020a), visual dialog (Qi et al., 2020), few-shot learning (Yue et al., 2020), semantic segmentation (Yu & Koltun, 2016; Zhang et al., 2020b), visual question answering (Niu et al., 2021; Teney et al., 2021), unsupervised learning (Wang et al., 2020a), knowledge distillation (Deng & Zhang, 2021a), and self-supervised learning (Wang et al., 2021a). Different from all these approaches, we propose to learn causal metrics by explicitly learning environment-invariant attention and task-specific embedding based on causal inference.

3. Framework

The goal of DCML is not to design a novel pair/triplet-based or proxy-based loss, or a novel sampling or margin strategy. Instead, it aims to learn the causal distances between samples regarding a task through causal inference and attention mechanism. DCML is based on a classical proxy-based loss (Wang et al., 2018b) as proxy losses have the advantages of small sampling complexity, high convergence, and capturing the global information. We thus first introduce the proxy-based loss and then present DCML.



(a) Causal graph (b) Intervention

Figure 2. Structural causal model.

3.1. Proxy-Based Loss

We denote the training dataset by $\mathbf{D} = \{(x_i, y_i)\}_{i=1}^{n}$ where x_i and y_i are a sample and the corresponding label, respectively. Proxy-based losses (Wang et al., 2018b) assume that there are m proxy vectors $\{c_1, c_2, ..., c_m\}$ for the m classes in $\mathbf{D}$, where c_i denotes the proxy vector of class i. The goal is to pull the embedding vector of each sample x_i close to its ground-truth class proxy c_{y_i} while pushing it far from the other class proxies, which can be expressed as:

$$\mathcal{K}(h_i, y_i, c) = -\log \frac{e^{a*(\mathcal{C}(h_i, c_{y_i})-m)}}{e^{a*(\mathcal{C}(h_i, c_{y_i})-m)} + \sum_{j \neq y_i} e^{a*\mathcal{C}(h_i, c_j)}} \tag{1}$$

where h_i is the embedding vector of samples x_i; c is the class proxy vectors, i.e., $c = \{c_1, c_2, ..., c_m\}$; $\mathcal{C}(h_i, c_i)$ denotes the cosine similarity between vector h_i and c_i; a is a scaler; $*$ denotes multiplication; and m is a threshold.

The existing approaches typically directly minimize the empirical error over the training dataset:

$$\mathcal{L}_{er}(\mathbf{D}, f, c) = \sum_{(x_i, y_i) \in \mathbf{D}} \mathcal{K}(f(x_i), y_i, c) \tag{2}$$

where $f()$ is a DNN encoder. We show below that simply minimizing the empirical error over the training dataset does not lead to the casual distance metric between samples due to the context biases in the dataset, and the resulting biased metric may not generalize well on new (test) classes.

3.2. Structural Causal Model

We present the causalities among context prior D, sample X, and metric M in Figure 2(a) where $A \rightarrow B$ denotes that A is the cause of B. We describe the causal relationships among these variables below.

$D \rightarrow X$: D is the dataset-specific context prior. It determines what the object co-appears with or where the object appears in an image X, e.g., in Figure 1, the tiger appears in green forests instead of snow. Different datasets have different context prior distribution, e.g., one dataset collects most of the tigers in forests while another dataset mainly collects tigers in zoos.

$D \rightarrow S \leftarrow X$**:** S is the spurious (context) representation of X under prior D. This relationship exists due to the fact that even for the same image, its context representation learned under the datasets with different context priors can differ substantially.

$X \rightarrow M \leftarrow S$**:** M is the distance metric. Besides the images X, the distance between two images is also influenced by the context prior D through mediator S. For example, as shown in Figure 1, when most of the tigers in a training dataset are in forests and Alaskan dogs are in snow, this kind of context prior makes the model take into account the representations of grasses and snow when calculating the distance between two animals. This misleads the model to make unreasonable predictions on the test data, i.e., the distance between the wolf in forests and the lion in forests is less than the distance between this wolf and the other two wolves in snow.

As illustrated above, the context prior D is the confounder of image X and the distance metric M. The existing approaches directly learn $P(M|X)$, which introduces the context biases to the learned metric and thus results in a biased metric. To address this issue, we propose DCML to learn $P(M|do(X))$ that learns the true causality of the distance between samples w.r.t. the target task.

3.3. Deep Causal Metric Learning

To remove the effects of the context prior D and learn the causal distance metric, one intuitive way is to collect data samples evenly in different context environments, which is impossible due to the huge cost (Zhang et al., 2020b). Thanks to backdoor adjustment, we can learn $P(M|do(X))$ by cutting off $D \rightarrow X$ so that D is not a confounder anymore as shown in Figure 2(b), which can be achieved by stratifying confounder D into pieces $D = \{d_1, d_2, ..., d_k\}$ where each d_i represents one kind of context prior environment. The de-confounder model is expressed as:

$$P(M|do(X)) = \sum_{d_j \in D} [P(d_j)P(M|X, S = \mathcal{I}(X, d_j))] \tag{3}$$

where $\mathcal{I}(X, d_j)$ is a function to generate the context representation of X under prior d_j.

By adopting the proxy-based loss, maximizing the probability of $P(M|do(X))$ is equivalent to minimizing the proxy-based loss:

$$\mathcal{L}_{bd} = \sum_{d_j \in D} \sum_{(x_i, y_i) \in d_j} \mathcal{K}(\mathcal{G}(x_i, s_i = \mathcal{I}(x_i, d_j)), y_i, c) \tag{4}$$

Two challenges arise in (4): (I) how to obtain environments d_j with different context priors; (II) how to learn spurious representation s_i for each sample x_i, i.e., function $\mathcal{I}$, and the final embedding function $\mathcal{G}$.

To address challenge (I), we construct environments with different context distributions by re-weighting the training samples (Teney et al., 2021; Wang et al., 2021a; Zhang et al., 2021; Wang et al., 2021b). Since different samples have different context information, re-weighting the samples is equivalent to changing the context distribution of a dataset. By adopting sample weights, the loss in context environment d_j in (4) is written as:

$$\mathcal{L}_{env}(d_j, (\mathcal{G}, \mathcal{I}), c) = \sum_{(x_i, y_i, w_{ij}) \in d_j} w_{ij} * \mathcal{K}(\mathcal{G}(x_i, s_i), y_i, c) \tag{5}$$

where w_{ij} is the weight for sample x_i in environment d_j. The sum of the weights for a training sample in all the k environments is set to 1, i.e., $w_{i1} + w_{i2} + ... + w_{ik} = 1$.

The causal distance that is not affected by the context should be invariant across different context environments, which can be achieved by learning environment-invariant embedding. Inspired by invariant risk minimization (IRM) (Arjovsky et al., 2019), we learn the invariant embedding by optimizing (4) under the constraint that there exists a class-proxy that is invariant and optimal across environments:

$$\begin{aligned} \mathcal{L}_{bd} &= \sum_{d_j \in D} \mathcal{L}_{env}(d_j, (\mathcal{G}, \mathcal{I}), c) \\ \text{subject to} \quad & c \in \arg\min_{\hat{c}} \mathcal{L}_{env}(d_j, (\mathcal{G}, \mathcal{I}), \hat{c}), \forall d_j \in D \end{aligned} \tag{6}$$

(6) can be further translated to a practical version:

$$\begin{aligned} \mathcal{L}_{inv} = \sum_{d_j \in D} \Big[& \mathcal{L}_{env}(d_j, (\mathcal{G}, \mathcal{I}), c) \\ & + \alpha * \|\nabla_{c|c=\mathbf{1}} \mathcal{L}_{env}(d_j, (\mathcal{G}, \mathcal{I}), c)\|^2 \Big] \end{aligned} \tag{7}$$

where α is a balancing coefficient; $\|.\|^2$ is the square of l^2-norm; $c = \mathbf{1}$ is a "dummy" class proxy and this term is for measuring the optimality of the dummy proxy at each environment. Please refer to (Arjovsky et al., 2019) for the proof and more details.

Next we need to solve challenge (II), i.e., designing $\mathcal{G}$ and $\mathcal{I}$ for learning the final embedding and spurious feature s, respectively. We suppose that there exits a mask m such that $m \circ f(x)$ is spurious feature s, where each element m_i in m is in the range of $[0, 1]$ and $\circ$ denotes element-wise multiplication. We design $\mathcal{G}$ and $\mathcal{I}$ as the following computation:

$$\begin{aligned} \mathcal{G}(x, s = \mathcal{I}(x, d)) &= f(x) - \mathcal{I}(x, d) \\ &= f(x) - m \circ f(x) = a \circ f(x) \end{aligned} \tag{8}$$

where instead of directly learning mask m in each environment, we learn task-specific attention map a that is the

complement of m to capture the causal embedding. An attention net $\mathcal{T}_\theta$ is used to learn a where θ is the parameters. $\mathcal{T}_\theta$ is implemented as a multilayer perceptron (Woo et al., 2018) (described in Appendix C). The input to $\mathcal{T}_\theta$ is the feature representation h of x (where $h = f(x)$). To effectively learn the attention map, we design two criteria: (a) the attention should be environment-invariant as its goal is to learn the causal embedding that is not influenced by the context environment changes, which can be achieved by penalizing the differences between the attention parameters learned under different environments; (b) the attention map should focus on task-invariant embedding and thus the task-specific embedding can be emphasized. By inserting the attention module and the two criteria to (7), we obtain:

$$\mathcal{L}_{it} = \sum_{d_j \in D} \Big[\mathcal{L}_{env}(d_j, \mathcal{T}_{\theta_j}(h) \circ h, c) + \alpha * \|\nabla_{c|c=\mathbf{1}} \mathcal{L}_{env}(d_j, \mathcal{T}_{\theta_j}(h) \circ h, c)\|^2 + \beta * \|\theta_j - \hat{\theta}\|^2 + \gamma * \mathbf{1}_{[y_i == y_k]} * \|\mathcal{T}_{\theta_j}(h_i) \circ h_i - \mathcal{T}_{\theta_j}(h_k) \circ h_k\|^2 \Big] \tag{9}$$

where β and γ are the balancing coefficients; $\hat{\theta}$ is the mean of the attention parameters learned under different environments; note that at inference time, there is only an attention net whose parameter values are set to $\hat{\theta}$; $\mathbf{1}_{[y_i==y_k]}$ is an indicator function that equals to 1 when $y_i = y_k$, otherwise 0. The third and the forth terms in the right hand side of (9) correspond to attention criteria (a) and (b), respectively. The intuition behind the forth term is that as the distance between two samples in the task is determined by their class labels, the attention net is expected to focus on the invariant "class" (task) embedding of the sample.

As the original dataset $\mathbf{D}$ (i.e., equivalent to sample weight 1) is also an important environment to the task, we also minimize the empirical error on $\mathbf{D}$, which results in the final objective of DCML:

$$\mathcal{L} = \mathcal{L}_{it} + \mathcal{L}_{er}(\mathbf{D}, T_\theta(h) \circ h, c) \tag{10}$$

The metric model is trained by minimizing (10). The remaining issue is how to generate the context environments, i.e., sample weights w_{ij}. Instead of using fixed environments, DCML automatically learns the context environments that the current embedding and the attention are not optimal or consistent across, which is achieved by:

$$\arg\max_{w} \sum_{d_j \in D} [\|\nabla_{c|c=\mathbf{1}} \mathcal{L}_{env}(d_j, \mathcal{T}_{\theta_j}(h) \circ h, c)\|^2 + \|\nabla_{\theta_j|\theta_j=\mathbf{1}} \mathcal{L}_{env}(d_j, \mathcal{T}_{\theta_j}(h) \circ h, c)\|^2] \tag{11}$$

where $\theta_j = \mathbf{1}$ is a "dummy" attention which the intuition behind is similar to that behind "dummy" class proxy $c = \mathbf{1}$. The context environments are updated once every e epochs. The implementation-level framework of DCML is sumarried in Algorithm 1.

Algorithm 1 DCML

Input: Training data $\mathbf{D}$, Encoder f, Attention MLP $\mathcal{T}_\theta$.
for $i = 1$ **to** N epochs **do**
  if $i\%e == 0$ **then**
    for $i = 1$ **to** M **do**
      Update sample weights (environments) with (11)
    end for
  end if
  Update the the model parameters by minimizing (10)
end for

4. Experiments

In this section, we first introduce the experimental settings, then report the comparison results between DCML and the state-of-the-art (SOTA) approaches, and finally present the ablation studies and qualitative results.

4.1. Experimental Settings

Musgrave et al. (2020) find through extensive experiments that when the existing approaches use their optimal hyper-parameters and are compared in a fair manner, the SOTA approaches only marginally outperform or perform on a par with the classical approaches such as the contrastive loss and the triplet loss. For a fair comparison, we adopt the same training strategy and performance metrics as those in (Musgrave et al., 2020).

Training Settings. We exactly follow the training settings in (Musgrave et al., 2020). The BN-Inception (Ioffe & Szegedy, 2015) pretrained on ImageNet (Russakovsky et al., 2015) is adopted as the backbone. 4-fold cross validation on the first half of the classes in each dataset is used for training the model. Specifically, the first half of classes are divided into 4 partitions deterministically. 3 of the 4 partitions are used as the training dataset and the remaining 1 as the validation dataset for tuning the hyper-parameters. The training cycles through all leave-one-out possibilities. The second half of classes are used for test. More details can be found in (Musgrave et al., 2020).

Evaluation Metrics. Musgrave et al. (2020) shows that the existing evaluation metrics fail to provide a complete picture of the embedding space and they further propose more stringent evaluation metrics by combining different metrics. To fairly and comprehensively evaluate different approaches, we adopt the metrics proposed in (Musgrave et al., 2020), i.e., P@1 (also known as Recall@1), RP, and MRP@R. More details can be found in Appendix A.

Datasets. Following the existing literature, we adopt the three widely used metric learning benchmark datasets, i.e., CUB-200 (Wah et al., 2011), Cars-196 (Krause et al., 2013), and Stanford Online Products (SOP) (Oh Song et al., 2016).

Table 1. Comparison results (%) on CUB200.

	Concatenated (512-dim)			Separated (128-dim)		
	P@1	RP	MAP@R	P@1	RP	MAP@R
Pretrained	51.05	24.85	14.21	50.54	25.12	14.53
Contrastive	68.13 ± 0.31	37.24 ± 0.28	26.53 ± 0.29	59.73 ± 0.40	31.98 ± 0.29	21.18 ± 0.28
Triplet	64.24 ± 0.26	34.55 ± 0.24	23.69 ± 0.23	55.76 ± 0.27	29.55 ± 0.16	18.75 ± 0.15
NT-Xent	66.61 ± 0.29	35.96 ± 0.21	25.09 ± 0.22	58.12 ± 0.23	30.81 ± 0.17	19.87 ± 0.16
ProxyNCA	65.69 ± 0.43	35.14 ± 0.26	24.21 ± 0.27	57.88 ± 0.30	30.16 ± 0.22	19.32 ± 0.21
Margin	63.60 ± 0.48	33.94 ± 0.27	23.09 ± 0.27	54.78 ± 0.30	28.86 ± 0.18	18.11 ± 0.17
Margin/class	64.37 ± 0.18	34.59 ± 0.16	23.71 ± 0.16	55.56 ± 0.16	29.32 ± 0.15	18.51 ± 0.13
N. Softmax	65.65 ± 0.30	35.99 ± 0.15	25.25 ± 0.13	58.75 ± 0.19	31.75 ± 0.12	20.96 ± 0.11
COS	67.32 ± 0.32	37.49 ± 0.21	26.70 ± 0.23	59.63 ± 0.36	31.99 ± 0.22	21.21 ± 0.22
ArcFace	67.50 ± 0.25	37.31 ± 0.21	26.45 ± 0.20	60.17 ± 0.32	32.37 ± 0.17	21.49 ± 0.16
FastAP	63.17 ± 0.34	34.20 ± 0.20	23.53 ± 0.20	55.58 ± 0.31	29.72 ± 0.16	19.09 ± 0.16
SNR	66.44 ± 0.56	36.56 ± 0.34	25.75 ± 0.36	58.06 ± 0.39	31.21 ± 0.28	20.43 ± 0.28
MS	65.04 ± 0.28	35.40 ± 0.12	24.70 ± 0.13	57.60 ± 0.24	30.84 ± 0.13	20.15 ± 0.14
MS+Miner	67.73 ± 0.18	37.37 ± 0.19	26.52 ± 0.18	59.41 ± 0.30	31.93 ± 0.15	21.01 ± 0.14
SoftTriple	67.27 ± 0.39	37.34 ± 0.19	26.51 ± 0.20	59.94 ± 0.33	32.12 ± 0.14	21.31 ± 0.14
ProxyNCA++	64.69 ± 0.40	34.37 ± 0.13	23.53 ± 0.12	57.13 ± 0.36	29.52 ± 0.16	18.76 ± 0.15
ContXBM	68.43 ± 1.18	37.66 ± 0.56	26.85 ± 0.63	60.95 ± 0.76	32.69 ± 0.33	21.78 ± 0.35
Proxy-Anchor	67.64 ± 0.42	37.29 ± 0.19	26.47 ± 0.21	60.59 ± 0.24	32.45 ± 0.15	21.57 ± 0.15
DCML (Ours)	**70.09 ± 0.22**	**39.05 ± 0.13**	**28.36 ± 0.13**	**62.28 ± 0.30**	**33.39 ± 0.18**	**22.61 ± 0.15**

Baselines. We compare DCML with both pair/triplet-based and proxy-based SOTA approaches including Contrastive (Hadsell et al., 2006), Triplet (Weinberger et al., 2006), NT-Xent (Sohn, 2016), ProxyNCA (Movshovitz-Attias et al., 2017), Margin (Wu et al., 2017), N. Softmax (Wang et al., 2017a; Zhai & Wu, 2018), COS (Wang et al., 2018a;b), ArcFace (Deng et al., 2019), FastAP (Cakir et al., 2019), SNR (Yuan et al., 2019), MS (Wang et al., 2019), SoftTriple (Qian et al., 2019), ContXBM (Wang et al., 2020c), ProxyNCA++ (Teh et al., 2020), and Proxy-Anchor (Kim et al., 2020).

4.2. Comparison with SOTA approaches

Comparison on CUB-200. We train the network for 20 epochs on CUB-200. The hyper-parameter tuning strategy is given in Appendix B. The comparison results are reported in Table 1. It is observed that DCML outperforms all these baselines by a large margin in terms of all the three metrics in both dim-512 and dim-128 embedding spaces, which demonstrates the effectiveness of DCML. Specifically, DCML improves P@1, RP, and MAP@R over the original proxy loss COS by 2.77%, 1.56%, and 1.66% respectively in the dim-512 embedding space, and by 2.65%, 1.40%, and 1.40% in the dim-128 embedding space. The significant improvements indicate the importance and superiority of learning causal distances.

Comparison on Cars-196. To examine the applicability of DCML on different kinds of datasets, we further conduct experiments on Cars-196. We train the model for 50 epochs with the hyper-parameters given in Appendix B. Table 2 shows that DCML beats both pair/triplet-based and proxy-based approaches substantially on both dim-512 and dim-128 embedding spaces, which demonstrates the usefulness and applicability of DCML on different kinds of datasets. Specifically, the absolute improvements over most of these baselines are consistently over 2% in terms of the three metrics in both embedding spaces. The superior performances of DCML are due to its ability to learn environment-invariant attention and task-invariant embedding.

Comparison on SOP. We further investigate the performance of DCML on large scale dataset SOP, where the model is trained for 70 epochs with hyper-parameters in Appendix B. As shown in Table 3, DCML obtains the best performance among these approaches, which validates the effectiveness and superiority of DCML on large-scale datasets. It also indicates that the causal distance metric is also beneficial to large datasets.

4.3. Ablation Studies

Ablation studies are conducted to examine the effects of the strategies in DCML on CUB-200 in the dim-512 space.

4.3.1. Components in the Objective of DCML

We first examine the effects of the components in (9). We denote DCML without the environment-invariant embedding

Table 2. Comparison results (%) on Car-196.

	Concatenated (512-dim)			Separated (128-dim)		
	P@1	RP	MAP@R	P@1	RP	MAP@R
Pretrained	46.89	13.77	5.91	43.27	13.37	5.64
Contrastive	81.78 ± 0.43	35.11 ± 0.45	24.89 ± 0.50	69.80 ± 0.38	27.78 ± 0.34	17.24 ± 0.35
Triplet	79.13 ± 0.42	33.71 ± 0.45	23.02 ± 0.51	65.68 ± 0.58	26.67 ± 0.36	15.82 ± 0.36
NT-Xent	80.99 ± 0.54	34.96 ± 0.38	24.40 ± 0.41	68.16 ± 0.36	27.66 ± 0.23	16.78 ± 0.24
ProxyNCA	83.56 ± 0.27	35.62 ± 0.28	25.38 ± 0.31	73.46 ± 0.23	28.90 ± 0.22	18.29 ± 0.22
Margin	81.16 ± 0.50	34.82 ± 0.31	24.21 ± 0.34	68.24 ± 0.35	27.25 ± 0.19	16.40 ± 0.20
Margin/class	80.04 ± 0.61	33.78 ± 0.51	23.11 ± 0.55	67.54 ± 0.60	26.68 ± 0.40	15.88 ± 0.39
N. Softmax	83.16 ± 0.25	36.20 ± 0.26	26.00 ± 0.30	72.55 ± 0.18	29.35 ± 0.20	18.73 ± 0.20
COS	85.52 ± 0.24	37.32 ± 0.28	27.57 ± 0.30	74.67 ± 0.20	29.01 ± 0.11	18.80 ± 0.12
ArcFace	85.44 ± 0.28	37.02 ± 0.29	27.22 ± 0.30	72.10 ± 0.37	27.29 ± 0.17	17.11 ± 0.18
FastAP	78.45 ± 0.52	33.61 ± 0.54	23.14 ± 0.56	65.08 ± 0.36	26.59 ± 0.36	15.94 ± 0.34
SNR	82.02 ± 0.48	35.22 ± 0.43	25.03 ± 0.48	69.69 ± 0.46	27.55 ± 0.25	17.13 ± 0.26
MS	85.14 ± 0.29	38.09 ± 0.19	28.07 ± 0.22	73.77 ± 0.19	29.92 ± 0.16	19.32 ± 0.18
MS+Miner	83.67 ± 0.34	37.08 ± 0.31	27.01 ± 0.35	71.80 ± 0.22	29.44 ± 0.21	18.86 ± 0.20
SoftTriple	84.49 ± 0.26	37.03 ± 0.21	27.08 ± 0.21	73.69 ± 0.21	29.29 ± 0.16	18.89 ± 0.16
ProxyNCA++	82.09 ± 0.41	36.31 ± 0.24	26.02 ± 0.26	70.60 ± 0.18	29.35 ± 0.08	18.63 ± 0.09
ContXBM	83.67 ± 0.35	36.10 ± 0.19	26.04 ± 0.24	72.58 ± 0.21	28.55 ± 0.10	18.07 ± 0.11
Proxy-Anchor	86.38 ± 0.15	37.53 ± 0.17	27.77 ± 0.20	76.85 ± 0.13	30.12 ± 0.10	19.82 ± 0.10
DCML (Ours)	**87.43 ± 0.21**	**39.60 ± 0.16**	**30.29 ± 0.12**	**78.58 ± 0.27**	**31.58 ± 0.15**	**21.55 ± 0.14**

Table 3. Comparison results (%) on SOP.

	Concatenated (512-dim)			Separated (128-dim)		
	P@1	RP	MAP@R	P@1	RP	MAP@R
Pretrained	50.71	25.97	23.44	47.25	23.84	21.36
Contrastive	73.12 ± 0.20	47.29 ± 0.24	44.39 ± 0.24	69.34 ± 0.26	43.41 ± 0.28	40.37 ± 0.28
Triplet	72.65 ± 0.28	46.46 ± 0.38	43.37 ± 0.37	67.33 ± 0.34	40.94 ± 0.39	37.70 ± 0.38
NT-Xent	74.22 ± 0.22	48.35 ± 0.26	45.31 ± 0.25	69.88 ± 0.19	43.51 ± 0.21	40.31 ± 0.20
ProxyNCA	75.89 ± 0.17	50.10 ± 0.22	47.22 ± 0.21	71.30 ± 0.20	44.71 ± 0.21	41.74 ± 0.21
Margin	70.99 ± 0.36	44.94 ± 0.43	41.82 ± 0.43	65.78 ± 0.34	39.71 ± 0.40	36.47 ± 0.39
N. Softmax	75.36 ± 0.17	50.01 ± 0.22	47.13 ± 0.22	71.65 ± 0.14	45.32 ± 0.17	42.35 ± 0.16
COS	75.79 ± 0.14	49.77 ± 0.19	46.92 ± 0.19	70.71 ± 0.19	43.56 ± 0.21	40.69 ± 0.21
ArcFace	76.20 ± 0.27	50.27 ± 0.38	47.41 ± 0.40	70.88 ± 1.51	44.00 ± 1.26	41.11 ± 0.22
FastAP	72.59 ± 0.26	46.60 ± 0.29	43.57 ± 0.28	68.13 ± 0.25	42.06 ± 0.25	38.88 ± 0.25
SNR	73.40 ± 0.09	47.43 ± 0.13	44.54 ± 0.13	69.45 ± 0.10	43.34 ± 0.12	40.31 ± 0.12
MS	74.50 ± 0.24	48.77 ± 0.32	45.79 ± 0.32	70.43 ± 0.33	44.25 ± 0.38	41.15 ± 0.38
MS+Miner	75.09 ± 0.17	49.51 ± 0.20	46.55 ± 0.20	71.25 ± 0.15	45.19 ± 0.16	42.10 ± 0.16
SoftTriple	76.12 ± 0.17	50.21 ± 0.18	47.35 ± 0.19	70.88 ± 0.20	43.83 ± 0.20	40.92 ± 0.20
ProxyNCA++	75.10 ± 0.15	49.50 ± 0.19	46.56 ± 0.19	70.43 ± 0.17	43.82 ± 0.20	41.51 ± 0.18
Proxy-Anchor	76.12 ± 0.19	50.82 ± 0.27	47.88 ± 0.26	72.79 ± 0.22	47.00 ± 0.24	43.97 ± 0.25
DCML (Ours)	**77.88 ± 0.19**	**52.81 ± 0.22**	**50.00 ± 0.22**	**73.83 ± 0.21**	**47.38 ± 0.23**	**44.52 ± 0.22**

term (i.e., $\alpha = 0$), without the environment-invariant attention term (i.e., $\beta = 0$), without the task-invariant term (i.e., $\gamma = 0$) by "DCML w/o inv_emb", "DCML w/o inv_at", and "DCML w/o inv_task", respectively. As shown in Table 4, the performance of DCML drops significantly without either of the three terms, which validates the effectiveness and necessity of the three terms.

4.3.2. Environment Strategies and Attention

DCML updates the environments by learning sample weights with (11). To show the effectiveness of this strat-

Table 4. Effects of the terms in the objective function.

	P@1	RP	MAP@R
DCML	70.09	39.05	28.36
DCML w/o inv_emb	68.87	37.98	27.13
DCML w/o inv_at	68.96	38.64	27.78
DCML w/o inv_task	68.86	38.92	28.03

Table 5. Effects of the environments and attention.

	P@1	RP	MAP@R
DCML	70.09	39.05	28.36
DCML w random	69.12	38.30	27.72
DCML w/o envs	67.69	37.34	26.56
DCML w/o at	69.03	38.22	27.88

Table 6. Effects of environment update frequency.

	$e=1$	$e=3$	$e=5$	$e=10$
P@1	69.63	69.38	70.09	69.95
RP	38.86	38.72	39.05	38.80
MAP@R	28.06	27.94	28.36	28.03

Table 7. Effects of the number of environments.

	$k=2$	$k=5$	$k=10$	$k=15$
P@1	70.09	69.89	69.47	69.10
RP	39.05	38.66	38.31	38.39
MAP@R	28.36	27.91	27.52	27.58

Figure 3. Randomly selected 3 query images and the Top-10 retrieval results from Cars-196. The blue frame indicates that the retrieved image and the query image belong to the same class, and the red frame indicates that they are from different classes.

egy, we compare it with DCML with randomly generated sample weights as environments (denoted by "DCML w random"). Table 5 reports the comparison results. It is unsurprising that DCML outperforms the random weight strategy significantly, since DCML explicitly learns the context environments that the attention and embedding are not optimal or consistent across while the randomly generated weights are less informative.

To examine whether the environment-based training and the attention module are necessary, we also report the performances of DCML without environments (denoted by "DCML w/o envs") and without attention module (denoted by "DCML w/o at") in Table 5. It is observed that the performances drop substantially without either of them, which indicates their importance and necessity.

4.3.3. Number of Environments Optimized Simultaneously and Update Frequency

As shown in (9), the losses on k (where $|D| = k$) environments are optimized simultaneously and then these environments are updated once every e epochs with (11) to cover more environments. We investigate how the performances vary with number k and update interval e. It is observed in Table 6 that the performance drops when the update interval is too large or too small, since too large update intervals make DCML not cover enough environments while too small update intervals make DCML not learn well on each environment. Table 7 shows that increasing k leads to a slight performance drop. The reason may be that it is difficult to optimize the model on too many environments simultaneously.

4.4. Qualitative Results

We further present qualitative results in Figure 3. It is observed that the images that are more visually similar to and are from the same class as the query image are typically within Top-3. We also notice that the images that are not very visually similar to the query image but are from the same class are also retrieved in Top-10 by DCML, which indicates the powerful ability of DCML for learning the causal metric regarding a target task (i.e., class here).

5. Conclusion

In this paper, we study deep metric learning from a novel perspective and accordingly propose deep causal metric learning (DCML). DCML learns the causal distance metric regarding a task by removing the effects of the spurious distances. This is achieved by learning environment-invariant attention and task-invariant embedding. Extensive experiments on several metric learning benchmark datasets demonstrate the effectiveness and superiority of DCML.

References

Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Aziere, N. and Todorovic, S. Ensemble deep manifold similarity learning using hard proxies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7299–7307, 2019.

Bengio, Y., Deleu, T., Rahaman, N., Ke, R., Lachapelle, S., Bilaniuk, O., Goyal, A., and Pal, C. A meta-transfer objective for learning to disentangle causal mechanisms. *arXiv preprint arXiv:1901.10912*, 2019.

Cakir, F., He, K., Xia, X., Kulis, B., and Sclaroff, S. Deep metric learning to rank. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1861–1870, 2019.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

de Haan, P., Jayaraman, D., and Levine, S. Causal confusion in imitation learning. In *Advances in Neural Information Processing Systems*, 2019.

Deng, J., Guo, J., Xue, N., and Zafeiriou, S. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4690–4699, 2019.

Deng, X. and Zhang, Z. Comprehensive knowledge distillation with causal intervention. *Advances in Neural Information Processing Systems*, 34, 2021a.

Deng, X. and Zhang, Z. Learning with retrospection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021b.

Deng, X., Xiao, Y., Long, B., and Zhang, Z. Reducing flipping errors in deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

Duan, J., Lin, Y.-L., Tran, S., Davis, L. S., and Kuo, C.-C. J. Slade: A self-training framework for distance metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9644–9653, 2021.

Gong, M., Zhang, K., Liu, T., Tao, D., Glymour, C., and Schölkopf, B. Domain adaptation with conditional transferable components. In *International conference on machine learning*, pp. 2839–2848. PMLR, 2016.

Hadsell, R., Chopra, S., and LeCun, Y. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pp. 1735–1742. IEEE, 2006.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. PMLR, 2015.

Kim, S., Kim, D., Cho, M., and Kwak, S. Proxy anchor loss for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3238–3247, 2020.

Kim, W., Goyal, B., Chawla, K., Lee, J., and Kwon, K. Attention-based ensemble for deep metric learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 736–751, 2018.

Krause, J., Stark, M., Deng, J., and Fei-Fei, L. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pp. 554–561, 2013.

Li, Z. and Tang, J. Weakly supervised deep metric learning for community-contributed image retrieval. *IEEE Transactions on Multimedia*, 17(11):1989–1999, 2015.

Liu, C., Yu, H., Li, B., Shen, Z., Gao, Z., Ren, P., Xie, X., Cui, L., and Miao, C. Noise-resistant deep metric learning with ranking-based instance selection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6811–6820, 2021.

Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., and Song, L. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 212–220, 2017.

Magliacane, S., van Ommen, T., Claassen, T., Bongers, S., Versteeg, P., and Mooij, J. M. Domain adaptation by using causal inference to predict invariant conditional distributions. In *Advances in Neural Information Processing Systems*, 2018.

Movshovitz-Attias, Y., Toshev, A., Leung, T. K., Ioffe, S., and Singh, S. No fuss distance metric learning using proxies. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 360–368, 2017.

Musgrave, K., Belongie, S., and Lim, S.-N. A metric learning reality check. In *European Conference on Computer Vision*, pp. 681–699. Springer, 2020.

Niu, Y., Tang, K., Zhang, H., Lu, Z., Hua, X.-S., and Wen, J.-R. Counterfactual vqa: A cause-effect look at language bias. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

Oh Song, H., Xiang, Y., Jegelka, S., and Savarese, S. Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4004–4012, 2016.

Pearl, J. and Mackenzie, D. *The book of why: the new science of cause and effect*. Basic books, 2018.

Pearl, J., Glymour, M., and Jewell, N. P. *Causal inference in statistics: A primer*. John Wiley & Sons, 2016.

Qi, J., Niu, Y., Huang, J., and Zhang, H. Two causal principles for improving visual dialog. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10860–10869, 2020.

Qian, Q., Shang, L., Sun, B., Hu, J., Li, H., and Jin, R. Softtriple loss: Deep metric learning without triplet sampling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6450–6458, 2019.

Roth, K., Milbich, T., Ommer, B., Cohen, J. P., and Ghassemi, M. Simultaneous similarity-based self-distillation for deep metric learning. In *International Conference on Machine Learning*, pp. 9095–9106. PMLR, 2021.

Rubin, D. B. Essential concepts of causal inference: a remarkable history and an intriguing future. *Biostatistics & Epidemiology*, 3(1):140–155, 2019.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3): 211–252, 2015.

Seidenschwarz, J., Elezi, I., and Leal-Taixé, L. Learning intra-batch connections for deep metric learning. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9410–9421. PMLR, 2021.

Snell, J., Swersky, K., and Zemel, R. Prototypical networks for few-shot learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 4080–4090, 2017.

Sohn, K. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in neural information processing systems*, pp. 1857–1865, 2016.

Suh, Y., Han, B., Kim, W., and Lee, K. M. Stochastic class-based hard example mining for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7251–7259, 2019.

Sun, Y., Cheng, C., Zhang, Y., Zhang, C., Zheng, L., Wang, Z., and Wei, Y. Circle loss: A unified perspective of pair similarity optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6398–6407, 2020.

Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H., and Hospedales, T. M. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1199–1208, 2018.

Tang, K., Huang, J., and Zhang, H. Long-tailed classification by keeping the good and removing the bad momentum causal effect. In *Advances in Neural Information Processing Systems*, 2020a.

Tang, K., Niu, Y., Huang, J., Shi, J., and Zhang, H. Unbiased scene graph generation from biased training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3716–3725, 2020b.

Teh, E. W., DeVries, T., and Taylor, G. W. Proxynca++: Revisiting and revitalizing proxy neighborhood component analysis. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16*, pp. 448–464. Springer, 2020.

Teney, D., Abbasnejad, E., and van den Hengel, A. Unshuffling data for improved generalization in visual question answering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1417–1427, 2021.

Van der Maaten, L. and Hinton, G. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. The caltech-ucsd birds-200-2011 dataset. *California Institute of Technology*, 2011.

Wang, F., Xiang, X., Cheng, J., and Yuille, A. L. Normface: L2 hypersphere embedding for face verification. In *Proceedings of the 25th ACM international conference on Multimedia*, pp. 1041–1049, 2017a.

Wang, F., Cheng, J., Liu, W., and Liu, H. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 25(7):926–930, 2018a.

Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z., and Liu, W. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5265–5274, 2018b.

Wang, J., Zhou, F., Wen, S., Liu, X., and Lin, Y. Deep metric learning with angular loss. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2593–2601, 2017b.

Wang, T., Huang, J., Zhang, H., and Sun, Q. Visual commonsense r-cnn. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10760–10770, 2020a.

Wang, T., Yue, Z., Huang, J., Sun, Q., and Zhang, H. Self-supervised learning disentangled group representation as feature. *Advances in Neural Information Processing Systems*, 34, 2021a.

Wang, T., Zhou, C., Sun, Q., and Zhang, H. Causal attention for unbiased visual recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3091–3100, 2021b.

Wang, X., Han, X., Huang, W., Dong, D., and Scott, M. R. Multi-similarity loss with general pair weighting for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5022–5030, 2019.

Wang, X., Liu, Z., Wang, N., and Fan, W. Relational metric learning with dual graph attention networks for social recommendation. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 104–117. Springer, 2020b.

Wang, X., Zhang, H., Huang, W., and Scott, M. R. Cross-batch memory for embedding learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6388–6397, 2020c.

Wang, Y. and Blei, D. M. The blessings of multiple causes. *Journal of the American Statistical Association*, 114(528): 1574–1596, 2019.

Weinberger, K. Q., Blitzer, J., and Saul, L. K. Distance metric learning for large margin nearest neighbor classification. In *Advances in neural information processing systems*, pp. 1473–1480, 2006.

Woo, S., Park, J., Lee, J.-Y., and Kweon, I. S. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19, 2018.

Wu, C.-Y., Manmatha, R., Smola, A. J., and Krahenbuhl, P. Sampling matters in deep embedding learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2840–2848, 2017.

Yang, X., Zhang, H., and Cai, J. Deconfounded image captioning: A causal retrospect. *arXiv preprint arXiv:2003.03923*, 2020.

Yang, Z., Bastan, M., Zhu, X., Gray, D., and Samaras, D. Hierarchical proxy-based loss for deep metric learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1859–1868, 2022.

Yu, B. and Tao, D. Deep metric learning with tuplet margin loss. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6490–6499, 2019.

Yu, F. and Koltun, V. Multi-scale context aggregation by dilated convolutions. In *International Conference on Learning Representations (ICLR)*, 2016.

Yuan, T., Deng, W., Tang, J., Tang, Y., and Chen, B. Signal-to-noise ratio: A robust distance metric for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Yue, Z., Zhang, H., Sun, Q., and Hua, X.-S. Interventional few-shot learning. In *Advances in Neural Information Processing Systems*, 2020.

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

Zhang, D., Li, Y., and Zhang, Z. Deep metric learning with spherical embedding. *Advances in Neural Information Processing Systems*, 33, 2020a.

Zhang, D., Zhang, H., Tang, J., Hua, X., and Sun, Q. Causal intervention for weakly-supervised semantic segmentation. In *Advances in Neural Information Processing Systems*, 2020b.

Zhang, X., Cui, P., Xu, R., Zhou, L., He, Y., and Shen, Z. Deep stable learning for out-of-distribution generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5372–5382, 2021.

Zheng, W., Chen, Z., Lu, J., and Zhou, J. Hardness-aware deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 72–81, 2019.

Zheng, W., Zhang, B., Lu, J., and Zhou, J. Deep relational metric learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12065–12074, 2021.

A. Evaluation Metrics

For a fair and comprehensive comparison, we adopt the three evaluation metrics that are introduced in (Musgrave et al., 2020), i.e., P@1, RP, and MAP@R. P@1 is also well-known as Recall@1 that has been widely used in different areas. We next introduce the definition of R-precision (RP). For a query, suppose that R is the total number of references that are from the same class as the query. First we find the R nearest references to the query. Suppose r is the number of those nearest references that are from the same class as the query. The RP for the query is calculated with $\frac{r}{R}$ (Musgrave et al., 2020). MAP@R is Mean Average Precision at R that combines the Mean Average Precision and R-precision. It is defined as $\frac{1}{R}\sum_{i=1}^{R} P(i)$ where $P(i)$ equals to the precision at i if the ith retrieval is from the same class as the query; otherwise 0.

B. Training and Hyper-parameters

DCML has 3 hyper-parameters, i.e., α, β, and γ. Instead of using grid search that is time-consuming, we do a very simple search. We first fix α and γ to 0, and tune β in [0, 1]. After the optimal β is obtained, we fix β and α and search the optimal γ in [0, 1]. Finally, we search α while fixing β and γ. Their final values on each dataset are given in the Github repository: https://github.com/Xiang-Deng-DL/DCML. For the hyper-parameters in the proxy loss, we set them to the values searched by (Musgrave et al., 2020). The model is trained with optimizer RMSprop. The learning rates for the backbone and the attention net are set to 1e-6 and 2e-6, respectively. The learning rate for the class proxy vectors are set to the values searched by (Musgrave et al., 2020), i.e., 2.53e-3, 7.41e-3, and 2.16e-3 on CUB-200, Cars-196, and SOP, respectively.

C. Implementation Details of Attention Net

The attention net is implemented as a MLP (Woo et al., 2018). Suppose that the feature maps h in the last convolution layer of an image are in $\mathbb{R}^{c\times h\times w}$ where c is the number of channels; h and w are the height and the width of the feature map in one channel. The attention net in the model consists of a two-layer MLP for learning channel attention a_c of size c and a two-layer MLP for learning the spatial attention map a_s of size $h \times w$. a_s is shared by all the feature maps in different channels. Specifically, we first average all the feature maps in different channels to obtain $\hat{h}_s \in \mathbb{R}^{h\times w}$. We flatten it to a vector and then simply apply the MLP and the sigmoid function mapping it to spatial attention map a_s. We then element-wisely multiply each feature map in h by attention map a_s and obtain h'. We then apply adaptive average pooling to h' and obtain a vector of size c. By applying the other MLP and the sigmoid function to the vector, we obtain the channel attention. We finally apply the channel attention to h'.

D. Embedding Visualization

As DCML maps the samples to the embedding space, we investigate "how close or far the samples are from each other in the embedding space" by visualizing them. The t-SNE (Van der Maaten & Hinton, 2008) visualization of the embedding features of 10 classes from CUB-200 and Cars-196 is presented in Figure 4 and Figure 5, respectively. It is observed that in

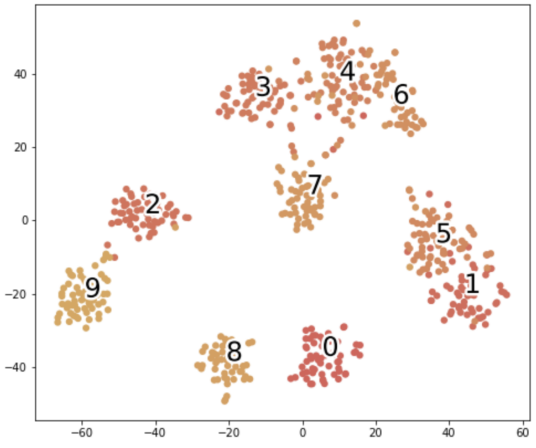


Figure 4. Feature visualization of 10 classes from CUB.

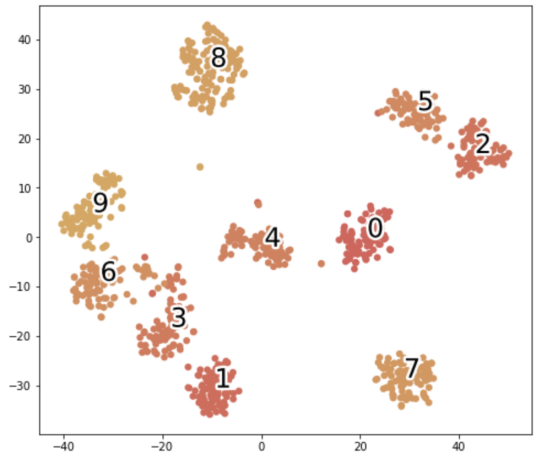


Figure 5. Feature visualization of 10 classes from Cars.

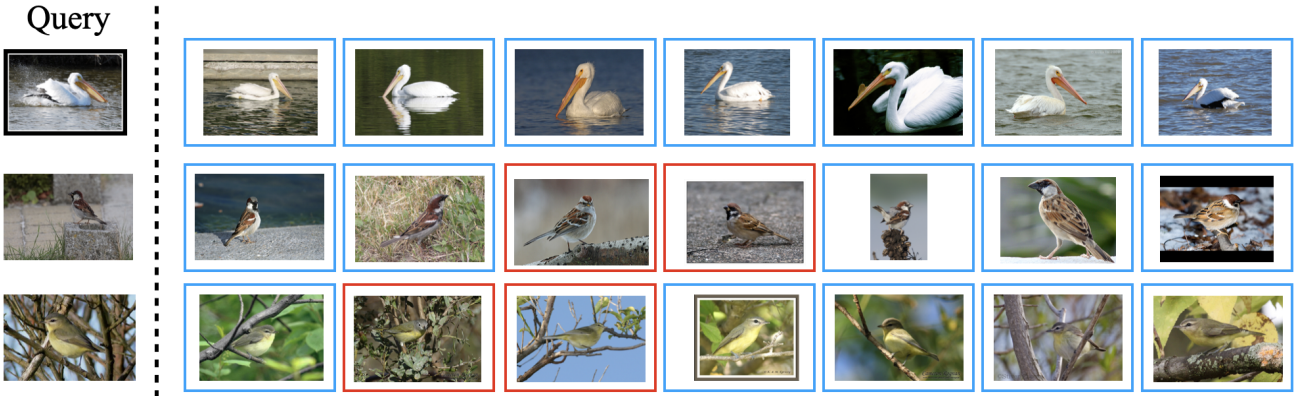


Figure 6. Randomly selected 3 query images and the retrieval results from CUB-200. The blue frame indicates that the retrieved image and the query image belong to the same class, and the red frame indicates that they are from different classes.

the embedding space, the samples belonging the same class are much closer than the samples from different classes, which is consistent with the metric learning goal. This also indicates the effectiveness of DCML.

E. More Qualitative Results

We present more qualitative results on CUB-200. As shown in Figure 6, given a query image, both visually similar and visually-not similar images can be retrieved if they are from the same class as the query image. This validates the powerful ability of the learned causal metric.