

Visual Similarity Attention

Meng Zheng¹, Srikrishna Karanam¹, Terrence Chen¹, Richard J. Radke² and Ziyan Wu¹

¹United Imaging Intelligence, Cambridge MA, USA

²Rensselaer Polytechnic Institute, Troy NY, USA

{meng.zheng, terrence.chen, ziyan.wu}@uui-ai.com, srikrishna@ieee.org, rjradke@ecse.rpi.edu

Abstract

While there has been substantial progress in learning suitable distance metrics, these techniques in general lack transparency and decision reasoning, *i.e.*, explaining why the input set of images is similar or dissimilar. In this work, we solve this key problem by proposing the first method to generate generic visual similarity explanations with gradient-based attention. We demonstrate that our technique is agnostic to the specific similarity model type, *e.g.*, we show applicability to Siamese, triplet, and quadruplet models. Furthermore, we make our proposed similarity attention a principled part of the learning process, resulting in a new paradigm for learning similarity functions. We demonstrate that our learning mechanism results in more generalizable, as well as explainable, similarity models. Finally, we demonstrate the generality of our framework by means of experiments on a variety of tasks, including image retrieval, person re-identification, and low-shot semantic segmentation.

1 Introduction

We consider the problem of learning similarity predictors for metric learning and related applications. Given a query image of an object, our task is to retrieve, from a set of reference images, the object image that is most similar to the query image. This problem finds applications in a variety of tasks, including image retrieval [Chen and Deng, 2019], person re-identification (re-id) [Zheng *et al.*, 2019a], and even low-shot learning [Shaban *et al.*, 2017]. There has been substantial recent progress in learning distance functions for these similarity learning applications [Wang *et al.*, 2019c].

Existing deep similarity predictors are trained in a distance learning fashion so that the features of same-class data points are close to each other in the learned embedding, while data features from other classes are further away. Consequently, most techniques distill this problem into optimizing a ranking objective that respects the relative ordinality of pairs, triplets, or even quadruplets [Law *et al.*, 2013] of training examples. These methods are characterized by the specificity of how the similarity model is trained, *e.g.*, data (pairs, triplets etc.) sampling [Wu *et al.*, 2017], sample weighting [Zheng *et al.*,

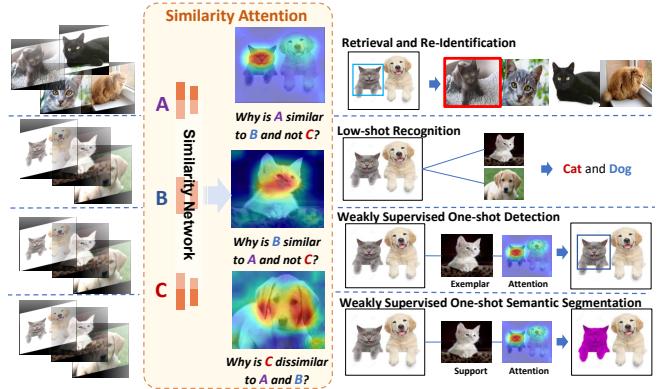


Figure 1. Proposed visual similarity explanation and its applications.

2019b], and adaptive ranking [Rippel *et al.*, 2016], among others. However, a key limitation of these approaches is their lack of decision reasoning, *i.e.*, explanations for why the model predicts the input set of images is similar or dissimilar. As we demonstrate in this work, our method not only offers model explainability, but such decision reasoning can also be infused into the model training process, in turn helping bootstrap and improve the generalizability of the trained similarity model.

Recent developments in CNN visualization have led to a surge of interest in visual explainability. Some methods [Li *et al.*, 2018a; Wang *et al.*, 2019b] enforce attention constraints using gradient-based attention [Selvaraju *et al.*, 2017], resulting in improved attention maps as well as downstream model performance. These techniques essentially ask: *where is the object in the image?* This limits their applicability to scenarios involving object categorization. On the other hand, we ask the question: *what makes image A similar to image B but dissimilar to image C?* (see Fig. 1). While existing works can explain classification models, their extensions to generating such visual similarity explanations is not trivial. A principled answer to this question will help explain models that predict visual similarity, which is what we address in our work.

To this end, we propose a new technique to generate CNN attention directly from similarity predictions. Note that this is substantially different from GradCAM-inspired [Selvaraju *et al.*, 2017] existing work [Li *et al.*, 2018a; Wang *et al.*, 2019b; Zheng *et al.*, 2019a] where an extra classification module is

needed to compute the network attention. Instead, our proposed method generates visual attention from feature vectors (produced by any CNN with fully-connected units) used to compute similarity, thereby resulting in a flexible and generic scheme that can be used in conjunction with any feature embedding network. Furthermore, we show that the resulting similarity attention can be modeled as the output of a differentiable operation, thereby enabling its use in model training as an explicit trainable constraint, which we empirically show improves model generalizability. A key feature of our proposed technique is its generality, evidenced by two characteristics we demonstrate. First, our design is not limited to a particular type of similarity learning architecture; we show applicability to and results with three different types of architectures: Siamese, triplet, and quadruplet. Next, we demonstrate the versatility of our framework in addressing problems different from image retrieval (*e.g.*, low-shot semantic segmentation) by exploiting its decision reasoning functionality to discover regions of interest. To summarize, our key contributions include:

- We present the first gradient-based similarity attention, to generate visual explanations from generic similarity metrics, equipping similarity models with explainability.
- Our proposed method only requires feature vectors to generate visual attention, thereby extensible to any feature embedding CNN model.
- We show how the proposed similarity attention can be formulated into trainable constraints, resulting in a new similarity mining learning objective and enabling similarity-attention-driven learning mechanisms for training similarity models with improved generalizability.
- We demonstrate the versatility of our proposed framework by a diverse set of experiments on a variety of tasks (*e.g.*, image retrieval, person re-id and low-shot semantic segmentation) and similarity model architectures.

2 Related Work

Our work is related to both the metric learning and visual explainability literature. In this section, we briefly review closely-related methods along these directions respectively.

Learning Distance Metrics. Metric learning approaches attempt to learn a discriminative feature space to minimize intra-class variations, while also maximizing the inter-class variance. Traditionally, this translated to optimizing learning objectives based on the Mahalanobis distance function or its variants. Much recent progress with CNNs has focused on developing novel objective functions or data sampling strategies [Wu *et al.*, 2017]. Substantial effort has also been expended in proposing new objective functions for learning the distance metric [Sohn, 2016][Song *et al.*, 2016], and proxy-NCA [Movshovitz-Attias *et al.*, 2017]. The goal of these and related objective functions is essentially to explore ways to penalize training data samples (pairs, triplets, quadruplets, or even distributions [Rippel *et al.*, 2016]) so as to learn a discriminative embedding. In this work, we take a different approach. Instead of just optimizing a distance objective, we explicitly consider and model network attention during training. This leads to two key innovations over existing work. First, we

equip our trained model with decision reasoning functionality. Second, by means of trainable attention, we guide the network to discover local image regions that contribute the most to the final decision, thereby improving model generalizability.

Learning Visual Explanations. Dramatic performance improvements of vision algorithms driven by black-box CNNs have led to a recent surge in attempts [Mahendran and Vedaldi, 2015; Zhou *et al.*, 2016; Selvaraju *et al.*, 2017] to interpret model decisions. Most CNN visual explanation techniques fall into either response-based or gradient-based categories. Class Activation Map (CAM) [Zhou *et al.*, 2016] used an additional fully-connected unit on top of the original deep model to generate attention maps, thereby requiring architectural modification during inference. Grad-CAM [Selvaraju *et al.*, 2017] solved this problem by generating attention maps using class-specific gradients of predictions. There has been works took a step forward, *e.g.* [Li *et al.*, 2018a; Wang *et al.*, 2019b; Zheng *et al.*, 2019a] use the attention maps to enforce trainable attention constraints, demonstrating improved model performance. These aforementioned gradient-based techniques all require a well-trained classifier for generating visual explanations and rely on application-specific assumptions. A few recent examples of attempts to visually explain similarity models include Plummer *et al.* [Plummer *et al.*, 2020] and Chen *et al.* [Chen *et al.*, 2020]. While Plummer *et al.* [Plummer *et al.*, 2020] needs attribute labels coupled with an attribute classification module and a saliency generator to generate explanations, and Chen *et al.* [Chen *et al.*, 2020] adopts a two-stage pipeline which first generates gradients by sampling training data tuples and then transfers gradient weight from training to testing by nearest neighbor search, our method is more *generic* that it does not need extra labels/additional learning modules, or training data access and weights transfer. Our proposed algorithm can generate similarity attention from any similarity measure, and additionally, can enforce trainable constraints using the generated similarity attention. Our design leads to a flexible technique and generalizable model that we show competitive results in areas ranging from metric learning to low-shot semantic segmentation.

3 Proposed Method

Given a set of N labeled images $\{(\mathbf{x}_i, y_i)\}, i = 1, \dots, N$ each belonging to one of k categories, where $\mathbf{x} \in \mathbb{R}^{H \times W \times c}$, and $y \in \{1, \dots, k\}$, we aim to learn a distance metric to measure the similarity between two images \mathbf{x}_1 and \mathbf{x}_2 . Our key innovation includes the design of a flexible technique to produce similarity model explanations by means of CNN attention, which we show can be used to enforce trainable constraints during model training. This leads to a model equipped with similarity explanation capability as well as improved model generalizability. In Section 3.1, we first briefly discuss the basics of existing similarity learning architectures followed by our proposed technique to learn similarity attention, and show how it can be easily integrated with existing networks. In Section 3.2, we discuss how the proposed mechanism facilitates principled attentive training of similarity models with our new similarity mining learning objective.

3.1 Similarity Attention

Traditional similarity predictors such as Siamese or triplet models are trained to respect the relative ordinality of distances between data points. For instance, given a training set of triplets $\{(\mathbf{x}_i^a, \mathbf{x}_i^p, \mathbf{x}_i^n)\}$, where $(\mathbf{x}_i^a, \mathbf{x}_i^p)$ have the same categorical label while $(\mathbf{x}_i^a, \mathbf{x}_i^n)$ belong to different classes, a triplet similarity predictor learns a d -dimensional feature embedding of the input \mathbf{x} , $f(\mathbf{x}) \in \mathbb{R}^d$, such that the distance between $f(\mathbf{x}_i^a)$ and $f(\mathbf{x}_i^n)$ is larger than that between $f(\mathbf{x}_i^a)$ and $f(\mathbf{x}_i^p)$ (within a predefined margin α).

Starting from such a baseline predictor (we choose the triplet model for all discussion here, but later show variants with Siamese and quadruplet models as well), our key insight is that we can use the similarity scores from the predictor to generate visual explanations, in the form of visual attention maps [Selvaraju *et al.*, 2017], for why the current input triplet satisfies the triplet criterion w.r.t the learned feature embedding $f(\mathbf{x})$. As an example of our final result, see Fig. 1, where we note our model is able to highlight common (cat) face region in the anchor (A) and the positive (B) image, whereas we highlight the corresponding face and ears region for the dog image (negative, C), illustrating why this current triplet satisfies the triplet criterion. This is what we refer to by *similarity attention*: the ability of the similarity predictor to automatically discover local regions in the input that contribute the most to the final decision (in this case, satisfying the triplet condition) and visualize these regions by means of attention maps.

Note that our idea of generating network attention from the similarity score is different from existing work [Selvaraju *et al.*, 2017; Zheng *et al.*, 2019a], in which an extra classification module and the classification probabilities are used to obtain attention maps. In our case, we are not limited by this requirement of needing a classification module. Instead, as we discuss below, we compute a similarity score directly from the feature vectors (*e.g.*, $f(\mathbf{x}_i^a)$, $f(\mathbf{x}_i^p)$, and $f(\mathbf{x}_i^n)$), which is then used to compute gradients and obtain the attention map. A crucial advantage with our method is that this results in a flexible and generic scheme, that can be used to visually explain virtually any feature embedding network.

An illustration of our proposed similarity attention generation technique is shown in Figure 2. Given a triplet sample $(\mathbf{x}^a, \mathbf{x}^p, \mathbf{x}^n)$, we first extract feature vectors \mathbf{f}^a , \mathbf{f}^p , and \mathbf{f}^n respectively. All the feature vectors are normalized to have l_2 norm equal to 1. Ideally, a perfectly trained triplet similarity model must result in \mathbf{f}^a , \mathbf{f}^p , and \mathbf{f}^n satisfying the triplet criterion. Under this scenario, local differences between the images in the image space will roughly correspond to proportional differences in the feature space as well. Consequently, there must exist some dimensions in the feature space that contribute the most to this particular triplet satisfying the triplet criterion, and we seek to identify these elements in order to compute the attention maps. To this end, we compute the absolute differences and construct the weight vectors \mathbf{w}^p and \mathbf{w}^n as $\mathbf{w}^p = \mathbf{1} - |\mathbf{f}^a - \mathbf{f}^p|$ and $\mathbf{w}^n = |\mathbf{f}^a - \mathbf{f}^n|$.

With \mathbf{w}^p , we seek to highlight the feature dimensions that have a small absolute difference value (*e.g.*, for those dimensions t , \mathbf{w}_t^p will be closer to 1), whereas with \mathbf{w}^n we seek to highlight the feature dimensions with large absolute differences. Given \mathbf{w}^p and \mathbf{w}^n , we construct a single weight vector

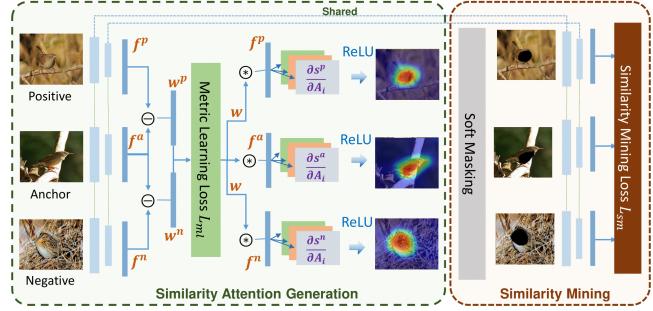


Figure 2. Similarity attention and similarity mining techniques.

$\mathbf{w} = \mathbf{w}^p \odot \mathbf{w}^n$ (\odot denotes element-wise product operation). With \mathbf{w} , we will obtain a higher weight with feature dimensions that have a high value in both \mathbf{w}^p and \mathbf{w}^n . To further understand this intuition, let us consider a simple example. If the first feature dimension $f^a(1) = 0.80$ and $f^p(1) = 0.78$, then this first dimension is important for the anchor to be close to the positive. In this case, the first dimension of the corresponding weight vector $w^p(1) = (1 - |0.80 - 0.78|) = 0.98$, which is a high value, quantifying the importance of this particular feature dimension for the anchor and positive to be close. Given these high-value dimensions, we identify all such important dimensions common across both \mathbf{w}^p and \mathbf{w}^n with the single weight vector \mathbf{w} . In other words, we focus on elements that contribute the most to (a) the positive feature pair being close, and (b) the negative feature pair being further away. This way, we identify dimensions in the feature space that contribute the most to \mathbf{f}^a , \mathbf{f}^p , and \mathbf{f}^n satisfying the triplet criterion. We now use these feature dimensions to compute network attention for the current image triplet $(\mathbf{x}^a, \mathbf{x}^p, \mathbf{x}^n)$.

Given \mathbf{w} , we compute the dot product of \mathbf{w} with \mathbf{f}^a , \mathbf{f}^p , and \mathbf{f}^n to get the sample scores $s^a = \mathbf{w}^T \mathbf{f}^a$, $s^p = \mathbf{w}^T \mathbf{f}^p$, and $s^n = \mathbf{w}^T \mathbf{f}^n$ for each image $(\mathbf{x}^a, \mathbf{x}^p, \mathbf{x}^n)$ respectively. We then compute the gradients of these sample scores with respect to the image's convolutional feature maps to get the attention map. Specifically, given a score s^i , $i \in \{a, p, n\}$, the attention map $\mathbf{M}^i \in \mathbb{R}^{m \times n}$ is determined as:

$$\mathbf{M}^i = \text{ReLU} \left(\sum_k \alpha_k \mathbf{A}_k \right) \quad (1)$$

where $\mathbf{A}_k \in \mathbb{R}^{m \times n}$ is the k^{th} ($k = 1, \dots, c$) convolutional feature channel (from one of the intermediate layers) of the convolutional feature map $\mathbf{A} \in \mathbb{R}^{m \times n \times c}$ and $\alpha_k = \text{GAP} \left(\frac{\partial s^i}{\partial \mathbf{A}_k} \right)$. The GAP operation is the same global average pooling operation described in [Selvaraju *et al.*, 2017].

Extensions to other architectures

Our proposed technique to generate similarity attention is not limited to triplet CNNs and is extensible to other architectures as well. Here, we describe how to generate our proposed similarity attention with Siamese and quadruplet models.

For a Siamese model, the inputs are pairs $(\mathbf{x}^1, \mathbf{x}^2)$. Given their feature vectors \mathbf{f}^1 and \mathbf{f}^2 , we compute the weight vector \mathbf{w} in the same way as the triplet scenario. If \mathbf{x}^1 and \mathbf{x}^2 belong

to the same class, $\mathbf{w} = \mathbf{1} - |\mathbf{f}^1 - \mathbf{f}^2|$. If they belong to different classes, $\mathbf{w} = |\mathbf{f}^1 - \mathbf{f}^2|$. With \mathbf{w} , we compute the sample scores $s^1 = \mathbf{w}^T \mathbf{f}^1$ and $s^2 = \mathbf{w}^T \mathbf{f}^2$, and use Equation 1 to compute attention maps \mathbf{M}^1 and \mathbf{M}^2 for \mathbf{x}^1 and \mathbf{x}^2 respectively. For a quadruplet model, the inputs are quadruplets $(\mathbf{x}^a, \mathbf{x}^p, \mathbf{x}^{n1}, \mathbf{x}^{n2})$, where \mathbf{x}^p is the positive sample and \mathbf{x}^{n1} and \mathbf{x}^{n2} are negative samples with respect to \mathbf{x}^a . Here, we compute the three difference feature vectors $\mathbf{f}^1 = |\mathbf{f}^a - \mathbf{f}^p|$, $\mathbf{f}^2 = |\mathbf{f}^a - \mathbf{f}^{n1}|$, and $\mathbf{f}^3 = |\mathbf{f}^a - \mathbf{f}^{n2}|$. Following the intuition described in the triplet case, we get the difference weight vectors as $\mathbf{w}^1 = \mathbf{1} - \mathbf{f}^1$ for the positive pair and $\mathbf{w}^2 = \mathbf{f}^2$ and $\mathbf{w}^3 = \mathbf{f}^3$ for the two negative pairs. The overall weight vector \mathbf{w} is then computed as the element-wise product of the three individual weight vectors: $\mathbf{w} = \mathbf{w}^1 \odot \mathbf{w}^2 \odot \mathbf{w}^3$. Given \mathbf{w} , we compute the sample scores $s^a = \mathbf{w}^T \mathbf{f}^a$, $s^p = \mathbf{w}^T \mathbf{f}^p$, $s^{n1} = \mathbf{w}^T \mathbf{f}^{n1}$, and $s^{n2} = \mathbf{w}^T \mathbf{f}^{n2}$, and use Equation 1 to obtain the four attention maps \mathbf{M}^a , \mathbf{M}^p , \mathbf{M}^{n1} , and \mathbf{M}^{n2} .

3.2 Learning with Similarity Mining

With our proposed mechanism to compute similarity attention, one can generate attention maps, as illustrated in Fig. 1, to explain why the similarity model predicted that the data sample satisfies the similarity criterion. However, we note all operations leading up to Equation 1, where we compute the similarity attention, are differentiable and we can use the generated attention maps to further bootstrap the training process. As we show later, this helps improve downstream model performance and generalizability. To this end, we describe a new learning objective, *similarity mining*, that enables such similarity-attention-driven training of similarity models.

The goal of similarity mining is to facilitate the complete discovery of local image regions that the model deems necessary to satisfy the similarity criterion. To this end, given the three attention maps $\mathbf{M}^i, i \in \{a, p, n\}$ (triplet case), we upsample them to be the same size as the input image and perform soft-masking, producing masked images that exclude pixels corresponding to high-response regions in the attention maps. This is realized as: $\hat{\mathbf{x}} = \mathbf{x} * (\mathbf{1} - \Sigma(\mathbf{M}))$, where $\Sigma(\mathbf{Z}) = \text{sigmoid}(\alpha(\mathbf{Z} - \beta))$ (all element-wise operations and α and β are pre-set, by cross validation, constants). These masked images are then fed back to the same encoder of the triplet model to obtain the feature vectors \mathbf{f}^{*a} , \mathbf{f}^{*p} , and \mathbf{f}^{*n} . Our proposed similarity mining loss L_{sm} , can then be expressed as:

$$L_{sm} = \left| \|\mathbf{f}^{*a} - \mathbf{f}^{*p}\| - \|\mathbf{f}^{*a} - \mathbf{f}^{*n}\| \right| \quad (2)$$

where $\|\mathbf{t}\|$ represents the Euclidean norm of the vector \mathbf{t} . The intuition here is that by minimizing L_{sm} , the model has difficulties in predicting whether the input triplet would satisfy the triplet condition. This is because as L_{sm} gets smaller, the model will have exhaustively discovered all possible local regions in the triplet, and erasing these regions (via soft-masking above) will leave no relevant features available for the model to predict that the triplet satisfies the criterion.

Extensions to Other Architectures

Like similarity attention, similarity mining is also extensible to other similarity learning architectures. For a Siamese similarity model, we consider only the positive pairs when enforcing

the similarity mining objective. Given the two attention maps \mathbf{M}^1 and \mathbf{M}^2 , we perform the soft-masking operation described above to obtain the masked images, resulting in corresponding features \mathbf{f}^{*1} and \mathbf{f}^{*2} . The similarity mining objective then attempts to maximize the distance between \mathbf{f}^{*1} and \mathbf{f}^{*2} , *i.e.*, $L_{sm} = -|\mathbf{f}^{*1} - \mathbf{f}^{*2}|$. Like the triplet case, the intuition of L_{sm} here is that it seeks to get the model to a state where after erasing, the model can no longer predict that the data pair belongs to the same class. This is because as L_{sm} gets smaller, the model will have exhaustively discovered all corresponding regions that are responsible for the data pair to be predicted as similar, *i.e.*, low feature space distance), and erasing these regions (via soft-masking) will result in a larger feature space distance between the positive samples.

For a quadruplet similarity model, using the four attention maps, we compute the feature vectors \mathbf{f}^{*a} , \mathbf{f}^{*p} , \mathbf{f}^{*n1} , and \mathbf{f}^{*n2} using the same masking strategy above. We then consider the two triplets $T_1 = (\mathbf{f}^{*a}, \mathbf{f}^{*p}, \mathbf{f}^{*n1})$ and $T_2 = (\mathbf{f}^{*a}, \mathbf{f}^{*p}, \mathbf{f}^{*n2})$ in constructing the similarity mining objective as $L_{sm} = L_{sm}^{T_1} + L_{sm}^{T_2}$, where $L_{sm}^{T_1}$ and $L_{sm}^{T_2}$ correspond to Equation 2 evaluated for T_1 and T_2 respectively.

3.3 Overall Training Objective

We train similarity models with both the traditional similarity/metric learning objective L_{ml} (*e.g.*, contrastive, triplet, etc.) as well as our proposed similarity mining objective L_{sm} . Our overall training objective L is:

$$L = L_{ml} + \gamma L_{sm} \quad (3)$$

where γ is a weight factor controlling the relative importance of L_{ml} and L_{sm} . Fig. 2 summarizes our training pipeline.

4 Experiments and Results

We conduct experiments on three different tasks: image retrieval (Sec. 4.1), person re-identification (Sec. 4.2), and one-shot semantic segmentation (Sec. 4.3) to demonstrate the efficacy and generality of our proposed framework. We use a pretrained ResNet50 as our base architecture and implement all our code in Pytorch.

4.1 Image Retrieval

We conduct experiments on the CUB200 (“CUB”) [Wah *et al.*, 2011], Cars-196 (“Cars”) [Krause *et al.*, 2013] and Stanford Online Products (“SOP”) [Song *et al.*, 2016] datasets, following the protocol of Wang *et al.* [Wang *et al.*, 2019c], and reporting performance using the standard Recall@K (R-K) metric [Wang *et al.*, 2019c]. We first show ablation results to demonstrate performance gains achieved by the proposed similarity attention and similarity mining techniques. Here, we also empirically evaluate our proposed technique with three different similarity learning architectures to demonstrate its generality. In Table 1, we show both baseline (trained only with L_{ml}) and our results with the Siamese, triplet, and quadruplet architectures (trained with $L_{ml} + \gamma L_{sm}$). As can be noted from these numbers, our method consistently improves the baseline performance across all three architectures. Since the triplet model gives the best performance among the three architectures considered in Table 1, for all subsequent

experiments, we only report results with the triplet variant. We next compare the performance of our proposed method with competing, state-of-the-art metric learning methods in Table 2. We note our proposed method is quite competitive, with R-1 performance improvement of 2.6% on CUB, matching (with DeML) R-1 and slightly better R-2 performance on Cars, and very close R-1 and slightly better R-1k performance (w.r.t. MS [Wang *et al.*, 2019c]) on SOP.

Arch.	Type	R-1	R-2	R-4
Siamese	Baseline	65.9	77.5	85.8
	SAM	67.7	77.8	85.5
Triplet	Baseline	66.4	78.1	85.6
	SAM	68.3	78.9	86.5
Quadruplet	Baseline	64.7	75.6	85.2
	SAM	66.4	77.0	85.2

Table 1. Ablation study on CUB dataset. All numbers in %.

	CUB		Cars		SOP	
	R-1	R-2	R-1	R-2	R-1	R-1k
Lifted [Song <i>et al.</i> , 2016]	47.2	58.9	49.0	60.3	62.1	97.4
N-pair [Sohn, 2016]	51.0	63.3	71.1	79.7	67.7	97.8
P-NCA [Movshovitz-Attias <i>et al.</i> , 2017]	49.2	61.9	73.2	82.4	73.7	-
HDC [Yuan <i>et al.</i> , 2017]	53.6	65.7	73.7	83.2	69.5	97.7
BIER [Opitz <i>et al.</i> , 2017]	55.3	67.2	78.0	85.8	72.7	98.0
ABE [Kim <i>et al.</i> , 2018]	58.6	69.9	82.7	88.8	74.7	98.0
MS [Wang <i>et al.</i> , 2019c]	65.7	77.0	84.1	90.4	78.2	98.7
HDML [Zheng <i>et al.</i> , 2019b]	53.7	65.7	79.1	89.7	68.7	-
DeML [Chen and Deng, 2019]	65.4	75.3	86.3	91.2	76.1	98.1
GroupLoss [Elezi <i>et al.</i> , 2020]	66.9	77.1	88.0	92.5	76.3	-
MS+SFT [Zhu <i>et al.</i> , 2020]	66.8	77.5	84.5	90.6	73.4	-
DRO-KL _M [Qi <i>et al.</i> , 2020]	67.7	78.0	86.4	91.9	-	-
SAM	68.3	78.9	86.3	91.4	77.9	98.8

Table 2. Results on CUB, CARS, and SOP. All numbers in %.

In addition to obtaining superior quantitative performance, another key difference between our method and competing algorithms is explainability. With our proposed similarity attention mechanism, we can now visualize, by means of similarity attention maps, the model’s decision reasoning. In Figures 4(a) and (b), we show examples of attention maps generated with our method on CUB and Cars testing data with both within- and cross-domain training data respectively. As shown in these figures, our proposed method is generally able to highlight intuitively satisfying correspondence regions across the images in each triplet. For example, in Fig. 4(a) (left 1), the beak color is what makes the second bird image similar, and the third bird image dissimilar, to the first (anchor) bird image. In Figures 4(a) and (b) (right), we show inter-dataset (cross-domain) results to demonstrate model generalizability. We note that, despite not being trained on relevant data, our model trained with similarity attention is able to discover local regions contributing to the final decision.

While one may add a classification module to any similarity CNN (e.g., Siamese), and then apply GradCAM [Selvaraju *et al.*, 2017] to generate class-specific visual explanations, we argue that GradCAM is specifically designed for image classification tasks, which would generate saliency maps for

each image (of the input pair) independently and does not ensure faithful explanation of the underlying similarity of the *pair* of images. Concretely, using GradCAM in such an independent fashion may fail to find explicit correspondences between the pair of input images as we shown in Figure 3, since it is designed to highlight regions that contribute to that individual image’s classification activations.

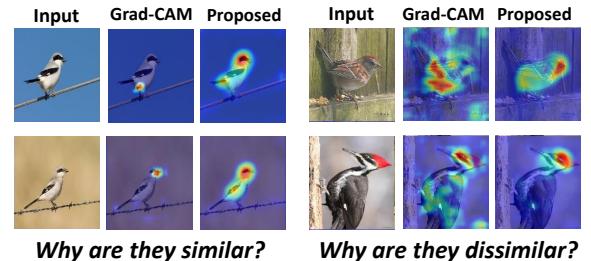


Figure 3. GradCAM (on adapted similarity CNNs with classification head) vs. proposed method. The proposed method is able to highlight corresponding regions more clearly when compared to GradCAM.

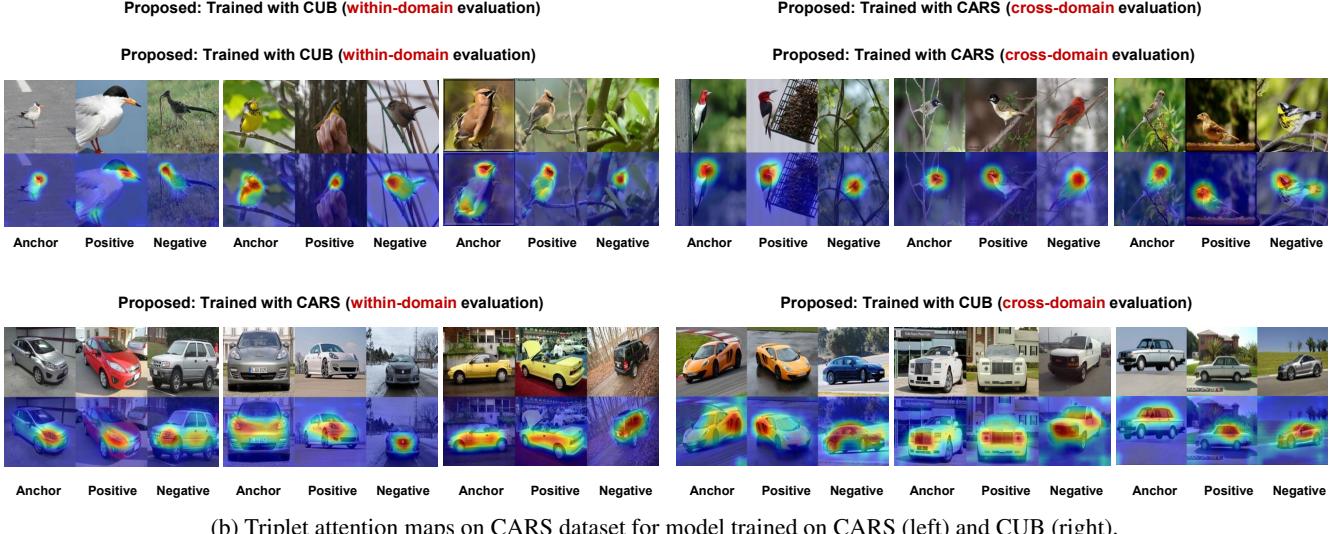
4.2 Person Re-Identification

We conduct experiments on person re-id to further prove the efficacy of our proposed framework. We evaluated on the CUHK03-NP dataset (“CUHK”) [Zhong *et al.*, 2017] and DukeMTMC-reid (“Duke”) [Ristani *et al.*, 2016] datasets, following the protocol in Sun *et al.* [Sun *et al.*, 2018].

We use the baseline architecture of Sun *et al.* [Sun *et al.*, 2018] and integrate our proposed similarity learning objective of Equation 3. We set $\gamma = 0.2$ and train the model for 40 epochs with the Adam optimizer. We summarize our results in Table 3, where we note our method results in about 3% rank-1 performance improvement on CUHK and very close performance (88.5% rank-1) to the best performing method (MGN) on Duke. We note that some of these competing methods have re-id specific design choices (e.g., upright pose assumption for attention consistency in CASN [Zheng *et al.*, 2019a], hard attention in HA-CNN [Li *et al.*, 2018b], attentive feature refinement and alignment in DuATM [Si *et al.*, 2018]). On the other hand, we make no such assumptions, however, is able to achieve competitive performance.

	CUHK		Duke	
	R-1	mAP	R-1	mAP
SVDNet [Sun <i>et al.</i> , 2017]	41.5	37.3	76.7	56.8
HA-CNN [Li <i>et al.</i> , 2018b]	41.7	38.6	80.5	63.8
DuATM [Si <i>et al.</i> , 2018]	-	-	81.8	64.6
PCB+RPP [Sun <i>et al.</i> , 2018]	63.7	57.5	83.3	69.2
MGN [Wang <i>et al.</i> , 2018]	66.8	66.0	88.7	78.4
CASN (PCB) [Zheng <i>et al.</i> , 2019a]	71.5	64.4	87.7	73.7
Proposed	74.5	67.5	88.5	75.8

Table 3. Re-Id results on CUHK and Duke (numbers in %).



(b) Triplet attention maps on CARS dataset for model trained on CARS (left) and CUB (right).

Figure 4. Triplet attention maps on (a) CUB and (b) CARS with our proposed method for models trained with CUB and CARS.

4.3 Weakly Supervised One-Shot Semantic Segmentation

In the one-shot semantic segmentation task, we are given a test image and a pixel-level semantically labeled support image, and we are to semantically segment the test image. Given that we learn similarity predictors, we can use our model to establish correspondences between the test and the support images. With the explainability of our method, the resulting similarity attention maps we generate can be used as cues to perform semantic segmentation. We use the PASCAL-5ⁱ dataset (“Pascal”) [Shaban *et al.*, 2017] for all experiments, following the same protocol as Shaban *et al.* [Shaban *et al.*, 2017]. A visualization of the proposed weakly-supervised one-shot segmentation workflow is shown in Figure 5.

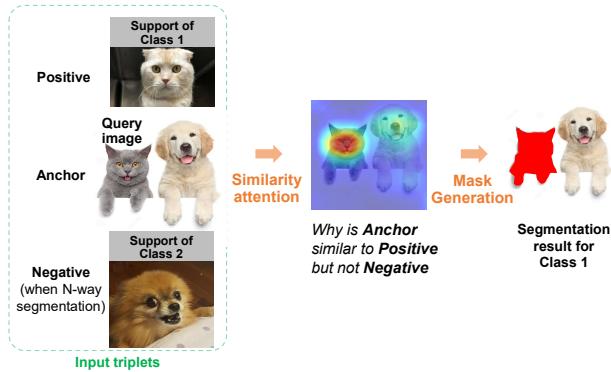
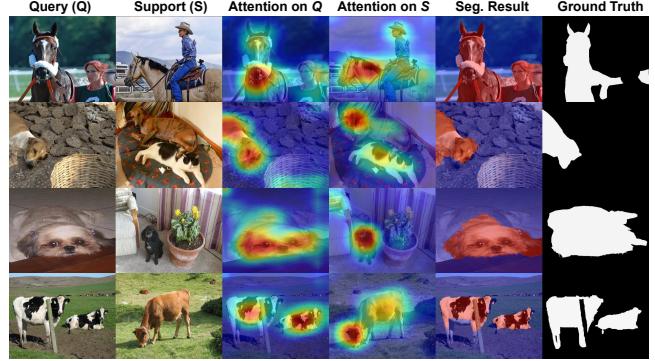


Figure 5. Our weakly-supervised one-shot segmentation workflow.

Given a test image and the corresponding support image, we first use our trained model to generate two similarity attention maps for each image. We then use the attention map for the test image to generate the final segmentation mask using the GrabCut [Rother *et al.*, 2004] algorithm. We call this the “1-way 1-shot” experiment. In the “2-way 1-shot” experiment,

Figure 6. One-shot segmentation results from the PASCAL-5*i*.

the test image has two objects of different classes and we are given two support images, image 1 and 2 from class 1 and 2 respectively. In this case, to generate results for object class 1, we use the support image 1 as the positive image and support image 2 as negative. Similarly, to generate results for object class 2, we use support image 2 as the positive image and support image 1 as negative. The “2-way 5-shot” experiment is similar; the only difference is we now have five support images for each of the two classes (instead of one image as above). We first show some qualitative results in Fig. 6 and 7 (left to right: test image, support image, test attention map, support image attention map, predicted segmentation mask, ground truth mask). In the third row of Fig. 6, we see that, in the test attention map, our method is able to capture the “dog” region in the test image despite the presence of a “cat” in the support image, helping generate the final segmentation result. In Fig. 7 first row, we see we are able to segment out both the person and the bike following the person and bike categories present in the two support images, helping generate a reasonably accurate final segmentation result. We also show the 1-way and 2-way meanIOU results in Table 4 (following

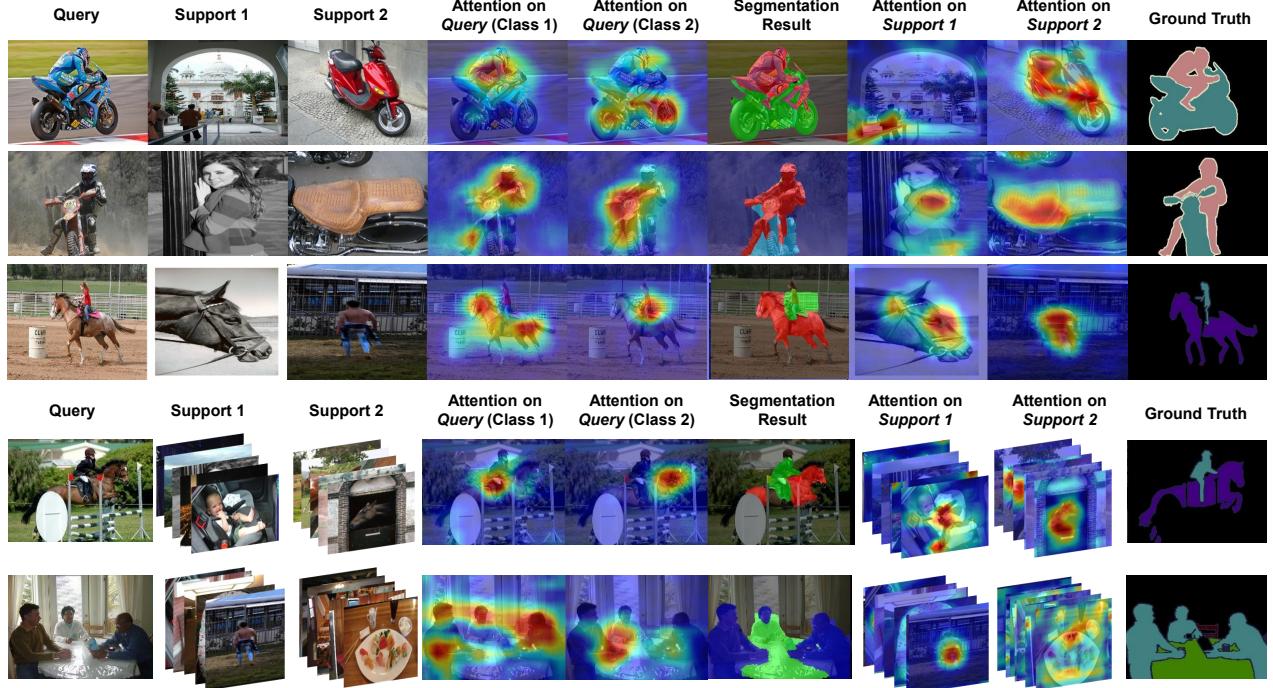


Figure 7. Qualitative one/five-shot segmentation results from the PASCAL-5i dataset.

the protocol of [Shaban *et al.*, 2017]) and Table 5 (following the protocol of [Dong and Xing, 2018]). Here, we highlight several aspects. First, all these competing methods are specifically trained towards the one-shot segmentation task, whereas our model is trained for metric learning. Second, they use the support image label mask both during training and inference, whereas our method does not use this label data. Finally, they are trained on Pascal, *i.e.*, relevant data, whereas our model was trained on CUB and Cars, data that is irrelevant in this context. Despite these seemingly disadvantageous factors, our method performs better than others in some cases and for the overall mean in the 1-way experiment and substantially outperforms competing methods in the 5-way experiments. Finally, we also substantially outperform the recently published PAN-init [Wang *et al.*, 2019a] which also does not train on the Pascal data (so this is closer to our experimental setup), while however using the support mask information during inference. These results demonstrate the potential of our proposed method in training similarity predictors that can generalize to data unseen during training and also to tasks for which the models were not originally trained.

Methods	Label	5 ⁰	5 ¹	5 ²	5 ³	Mean
OSVOS [Caelles <i>et al.</i> , 2017]	Yes	24.9	38.8	36.5	30.1	32.6
OSLSM [Shaban <i>et al.</i> , 2017]	Yes	33.6	55.3	40.9	33.5	40.8
co-FCN [Rakelly <i>et al.</i> , 2018]	Yes	36.7	50.6	44.9	32.4	41.1
PAN-init [Wang <i>et al.</i> , 2019a]	Yes	30.8	40.7	38.3	31.4	35.3
SAM	No	37.9	50.3	44.4	33.8	41.6

Table 4. 1-way 1-shot binary-IOU results (%) on PASCAL-5ⁱ.

Methods	Label	1-shot	5-shot
PL [Dong and Xing, 2018]	Yes	39.7	40.3
PL+SEG [Dong and Xing, 2018]	Yes	41.9	42.6
PL+SEG+PT [Dong and Xing, 2018]	Yes	42.7	43.7
SAM	No	56.9	60.1

Table 5. 2-way 1/5-shot binary-IOU results (%) on PASCAL-5ⁱ.

5 Summary and Future Work

We presented new techniques to explain and visualize, with gradient-based attention, predictions of similarity models. We showed our resulting *similarity attention* is generic and applicable to many commonly used similarity architectures. We presented a new paradigm for learning similarity functions with our *similarity mining* learning objective, resulting in improved downstream model performance. We also demonstrated the versatility of our framework in learning models for a variety of unrelated applications, *e.g.*, image retrieval (including re-id) and low-shot semantic segmentation, where one can easily extend this approach to generate explanations for set-to-set matching problems as well.

Acknowledgments

This material is based upon work supported by the U.S. Department of Homeland Security, Science and Technology Directorate, Office of University Programs, under Grant Award 18STEXP00001-03-02, Formerly 2013-ST-061-ED0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied,

of the U.S. Department of Homeland Security.

References

[Caelles *et al.*, 2017] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *CVPR*, 2017.

[Chen and Deng, 2019] B. Chen and W. Deng. Hybrid-attention based decoupled metric learning for zero-shot image retrieval. In *CVPR*, 2019.

[Chen *et al.*, 2020] L. Chen, J. Chen, H. Hajimirsadeghi, and G. Mori. Adapting grad-cam for embedding networks. In *WACV*, 2020.

[Dong and Xing, 2018] Nanqing Dong and Eric P. Xing. Few-shot semantic segmentation with prototype learning. In *BMVC*, 2018.

[Elezi *et al.*, 2020] Ismail Elezi, Sebastiano Vascon, Alessandro Torcinovich, Marcello Pelillo, and Laura Leal-Taixé. The group loss for deep metric learning. In *ECCV*, 2020.

[Kim *et al.*, 2018] Wonsik Kim, Bhavya Goyal, Kunal Chawla, Jungmin Lee, and Keunjoo Kwon. Attention-based ensemble for deep metric learning. In *ECCV*, 2018.

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

[Law *et al.*, 2013] Marc T Law, Nicolas Thome, and Matthieu Cord. Quadruplet-wise image similarity learning. In *ICCV*, 2013.

[Li *et al.*, 2018a] Kunpeng Li, Ziyan Wu, Kuan-Chuan Peng, Jan Ernst, and Yun Fu. Tell me where to look: Guided attention inference network. In *CVPR*, 2018.

[Li *et al.*, 2018b] Wei Li, Xiatian Zhu, and Shaogang Gong. Harmonious attention network for person re-identification. In *CVPR*, 2018.

[Mahendran and Vedaldi, 2015] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *CVPR*, 2015.

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

[Opitz *et al.*, 2017] Michael Opitz, Georg Waltner, Horst Possegger, and Horst Bischof. BIER — boosting independent embeddings robustly. In *ICCV*, 2017.

[Plummer *et al.*, 2020] Bryan Plummer, Mariya Vasileva, Vitali Pet- siuk, Kate Saenko, and David Forsyth. Why do these match? explaining the behavior of image similarity models. In *ECCV*, 2020.

[Qi *et al.*, 2020] Qi Qi, Yan Yan, Zixuan Wu, Xiaoyu Wang, and Tianbao Yang. A simple and effective framework for pairwise deep metric learning. In *ECCV*, 2020.

[Rakelly *et al.*, 2018] Kate Rakelly, Evan Shelhamer, Trevor Darrell, Alyosha Efros, and Sergey Levine. Conditional networks for few-shot semantic segmentation. In *ICLR Workshops*, 2018.

[Rippel *et al.*, 2016] Oren Rippel, Manohar Paluri, Piotr Dollar, and Lubomir Bourdev. Metric learning with adaptive density discrimination. In *ICLR*, 2016.

[Ristani *et al.*, 2016] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *ECCVW*, 2016.

[Rother *et al.*, 2004] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM transactions on graphics (TOG)*, volume 23, pages 309–314, 2004.

[Selvaraju *et al.*, 2017] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017.

[Shaban *et al.*, 2017] Amirreza Shaban, Shrav Bansal, Zhen Liu, Irfan Essa, and Byron Boots. One-shot learning for semantic segmentation. In *BMVC*, 2017.

[Si *et al.*, 2018] Jianlou Si, Honggang Zhang, Chun-Guang Li, Jason Kuen, Xiangfei Kong, Alex C Kot, and Gang Wang. Dual attention matching network for context-aware feature sequence based person re-identification. In *CVPR*, 2018.

[Sohn, 2016] K. Sohn. Improved deep metric learning with multi-class N-pair loss objective. In *NIPS*. 2016.

[Song *et al.*, 2016] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *CVPR*, 2016.

[Sun *et al.*, 2017] Yifan Sun, Liang Zheng, Weijian Deng, and Shengjin Wang. SVDNet for pedestrian retrieval. In *ICCV*, 2017.

[Sun *et al.*, 2018] Yifan Sun, Liang Zheng, Yi Yang, Qi Tian, and Shengjin Wang. Beyond part models: Person retrieval with refined part pooling. In *ECCV*, 2018.

[Wah *et al.*, 2011] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[Wang *et al.*, 2018] Guanshuo Wang, Yufeng Yuan, Xiong Chen, Jiwei Li, and Xi Zhou. Learning Discriminative Features with Multiple Granularities for Person Re-Identification. In *ACM MM*, 2018.

[Wang *et al.*, 2019a] K. Wang, J. H. Liew, Y. Zou, D. Zhou, and J. Feng. Panet: Few-shot image semantic segmentation with prototype alignment. In *ICCV*, 2019.

[Wang *et al.*, 2019b] L. Wang, Z. Wu, S. Karanam, K. Peng, R. V. Singh, and B. D. L. Metaxas. Sharpen focus: Learning with attention separability and consistency. In *ICCV*, 2019.

[Wang *et al.*, 2019c] Xun Wang, Xintong Han, Weiling Huang, Dengke Dong, and Matthew R. Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *CVPR*, 2019.

[Wu *et al.*, 2017] Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krahenbuhl. Sampling matters in deep embedding learning. In *ICCV*, 2017.

[Yuan *et al.*, 2017] Yuhui Yuan, Kuiyuan Yang, and Chao Zhang. Hard-aware deeply cascaded embedding. In *ICCV*, 2017.

[Zheng *et al.*, 2019a] M. Zheng, S. Karanam, z. Wu, and R. J. Radke. Re-identification with consistent attentive siamese networks. In *CVPR*, 2019.

[Zheng *et al.*, 2019b] W. Zheng, Z. Chen, J. Lu, and J. Zhou. Hardness-aware deep metric learning. In *CVPR*, 2019.

[Zhong *et al.*, 2017] Zhun Zhong, Liang Zheng, Donglin Cao, and Shaozi Li. Re-ranking person re-identification with k-reciprocal encoding. In *CVPR*, 2017.

[Zhou *et al.*, 2016] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016.

[Zhu *et al.*, 2020] Yuke Zhu, Yan Bai, and Yichen Wei. Spherical feature transform for deep metric learning. In *ECCV*, 2020.