RankMI: A Mutual Information Maximizing Ranking Loss

Mete Kemertas, Leila Pishdad, Konstantinos G. Derpanis, and Afsaneh Fazly
Samsung AI Centre Toronto
{mete.kemertas,leila.p,k.derpanis,a.fazly}@samsung.com

Abstract

We introduce an information-theoretic loss function, RankMI, and an associated training algorithm for deep representation learning for image retrieval. Our proposed framework consists of alternating updates to a network that estimates the divergence between distance distributions of matching and non-matching pairs of learned embeddings, and an embedding network that maximizes this estimate via sampled negatives. In addition, under this information-theoretic lens we draw connections between RankMI and commonly-used ranking losses, e.g., triplet loss. We extensively evaluate RankMI on several standard image retrieval datasets, namely, CUB-200-2011, CARS-196, and Stanford Online Products. Our method achieves competitive results or significant improvements over previous reported results on all datasets.

1. Introduction

Deep representation learning is fundamental for many downstream computer vision applications, including image retrieval and visual search [39, 14, 29, 27, 41], face re-identification [35, 6], 3D object retrieval [16], image captioning [36, 19], and cross-modal learning and retrieval [22, 13, 9, 25]. Most such tasks use deep neural networks to map their input (e.g., images, 3D shapes, text or audio captions, etc.) into an embedding space. The objective is to learn representations that yield high similarity for semantically related or *matching* items (e.g., same-category images/objects, faces of the same person, and paired image–captions) and low similarity for semantically unrelated or *non-matching* ones (e.g., images/objects of different types, faces of different people, and non-matching image–captions). A common practice is to formulate representation learning as a retrieval (or ranking) problem, and train a network using matching and non-matching query–value pairs as positive and negative samples, respectively.

A variety of loss functions, embedding ensembling, and sampling methods have been proposed in the literature, all seeking to learn representations that move sample positive points closer to, and sample negative points farther from, a target query. The different loss functions differ in the specifics of how they achieve this goal. For instance, several works impose a margin between positive and negative distances [43, 6, 27, 42]. Others learn a threshold on the distance scores [27, 40], or directly minimize the likelihood of negative points having a lower distance than positive points [41]. To further boost performance, ensembling methods have been proposed that combine embedding vectors from different layers from a single network or from different networks [45, 31, 31, 20]. Complementary efforts have been placed on sampling strategies that focus on finding infor-

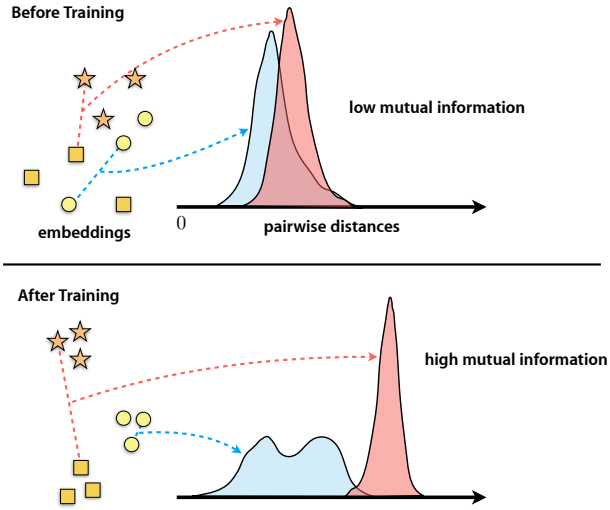

Figure 1. We propose RankMI, a novel information-theoretic ranking loss function. (Top) Before training, semantically related (matching) items do not necessarily have low distance, and unrelated (non-matching) ones do not have high distance. This is reflected in the high overlap between the distance score distributions for the matching (same-category), shown in blue, and the non-matching (different-category), shown in red. (Bottom) After optimizing our mutual information-based objective, the distributions are well separated, and by extension the embedding space is well-organized with matching items having low distance and non-matching ones having high distance. At no point are the distributions themselves modelled explicitly.

mative negative examples [35, 38, 3], weighting examples according to their informativeness [14, 27], or generating synthetic or proxy examples [8, 29, 46]. In this paper, our focus is primarily on the loss function.

Inspired by recent information-theoretic approaches to deep representation learning [2, 17], we present a novel view of the retrieval problem. In particular, we propose a retrieval objective that finds an image (value) that shares the highest amount of *information* with a given (image) query. We thus seek to learn (image) representations that maximize the mutual information (MI) among *instances* of the same category. Figure 1 depicts this objective: Prior to training, the distance score distribution of same-category (blue) pairs has a high overlap with that of pairs from different categories. By seeking to maximize the MI among same-category items, we learn representations that have high proximity for same-category items and low proximity for items from different categories, hence supporting accurate search and retrieval. To achieve this objective, we propose a novel loss function, called RankMI, that approximates the MI between query and value (images) by estimating a tight lower bound on the divergence between their joint probability and the product of marginals.

Contributions. In this paper, we make the following three main contributions. First, we propose a novel loss function, RankMI, which optimizes a theoretically-grounded objective, as well as an associated training algorithm. Second, under this information-theoretic lens, we draw connections to commonly used ranking losses. Finally, we present extensive evaluation of RankMI on standard image retrieval benchmarks, CUB-200-2011 [44], CARS-196 [23], and Stanford Online Products [39]. Our method achieves competitive results or significant improvements over previous reported results on all datasets.

2. Related work

A variety of loss functions, sampling techniques, and ensembling strategies have been proposed in the literature for deep representation learning. Here, we elaborate on those most relevant to our work, and establish the relevance of our main technical contributions. Note that parallel research dedicated to landmark retrieval (e.g., [1, 34, 33]) has also considered image representations, loss functions, and sampling strategies. These works are beyond the scope of the following survey.

Ranking losses. An important component of deep representation learning is the loss function that expresses the learning objective in the context of the target application. Triplet loss [43] is one of the most commonly used loss functions for retrieval tasks. This loss considers a set of data triplets, where each triplet consists of data elements termed the anchor, positive, and negative. A positive example shares the same class as the anchor, whereas the negative example differs. The goal of training is to ensure that the distance of the negative pair is higher than that of the positive (plus a margin), for all queries (within a batch). Quadruplet loss [6, 24] extends the triplet loss to additionally force a margin between sampled positives and negatives that do not share an anchor. Several other losses also extend the triplet loss: Angular loss [42] considers the relation between all three elements in a triplet, whereas n-pair loss [37] allows joint comparison with more than one negative example by taking one sample from every negative class. Other loss functions, such as [40] and [27], explicitly increase (decrease) the distance of positives (negatives) to be lower (higher) than a learnable threshold plus a fixed or learnable margin. Histogram loss [41] directly minimizes the empirically-estimated probability of a negative sample having a lower distance than a positive one, thus avoiding the need for tuning hyper-parameters such as threshold or margin. Unlike histogram loss, in the deep variational metric learning framework [26] the intra-class variations are modeled assuming a Gaussian latent space.

Our MI-based loss function presents an intuitive view of the ranking problem. Moreover, by drawing on recent advances in neural MI estimation (see below for related work on MI), RankMI estimates MI without having to directly compute the distance distributions. In addition, via the use of a neural network that collects dataset-level statistics, RankMI seamlessly incorporates global properties of the data distributions into the representation learning network. Next, we discuss several sampling techniques and other strategies that have been proposed to improve the effectiveness of these loss functions.

Beyond losses. Optimization of deep network parameters relies on the local gradients within a mini-batch, hence ignoring the global data distributions. To alleviate this problem, a significant body of research [39, 16] has explored ways of effectively and efficiently finding informative samples to learn from. Earlier work suggested that the use of semi-hard negative samples (i.e., samples that trigger false alarm) improves performance [35, 38, 3]. More recent work suggests strategies for assigning weights to sample points based on their informativeness [14, 27], generating synthetic adversarial hard negatives from easy negatives [8], or learning proxies for hard negatives across an entire dataset [29, 46]. Although sampling strategies succeed at choosing informative examples to learn from, they still work within a mini-batch and do not take a global view of the data. Some recent work explicitly incorporate global information in the form of class-level similarities [10, 5]. However, these studies rely on the existence of a hierarchy over classes. Yet others have pursued ensembling schemes [45, 31, 20] which combine embeddings taken from various layers of the same

network or from entirely different networks.

These efforts are orthogonal to our contributions, since RankMI does not impose constraints on the sampling procedure. For our evaluation, we use distance-weighted sampling [27], since it has been shown to outperform other commonly-used negative sampling strategies. Also, we do not resort to embedding ensembling.

Mutual information and deep learning. RankMI draws on recent information-theoretic approaches in deep learning that estimate the divergence between two probability distributions and/or mutual information between two random variables using neural network estimators. Nowozin et al. [30] show that one can recover the original generative adversarial network (GAN) training loss by Goodfellow et al. [12] as a minimization of the estimated divergence between the generated and true data distributions. Belghazi et al. [2] propose Mutual Information Neural Estimation (MINE) and show that the application of divergence estimation and subsequent minimization or maximization techniques extends beyond training GANs. In particular, MINE shows that one can design algorithms to consistently estimate the mutual information between two random variables via a neural network. While these developments have been used for training GANs and unsupervised representation learning [17], to the best of our knowledge, they have not been previously considered in the context of ranking and retrieval tasks, as done in our work.

In closely related work, Cakir et al. [4] learn binary hash codes. For learning, they also use mutual information to quantify the separation of distributions of positive and negative pairings. A major difference between Cakir et al. and our work is that their formulation is specialized to learning binary encodings with a particular distance measure (i.e., Hamming distance). Furthermore, their loss requires the explicit (quantized) modeling of the distance distributions, whereas we leverage variational functions and avoid such explicit modeling altogether.

3. Technical approach

Here, we present details of our information-theoretic loss function, RankMI. Section 3.1 provides background material on mutual information estimation using neural networks. We describe our RankMI loss in Section 3.2, and our training algorithm in Section 3.3. Finally, in Section 3.4, we draw connections between RankMI and commonly-used ranking losses, e.g., triplet loss.

3.1. Preliminaries

The mutual information, $I(X;Y)$, between two random variables, X and Y, can be expressed as the following KL-divergence:

$$I(X;Y) = D_{\mathrm{KL}}(\mathbb{J} \parallel \mathbb{M}), \tag{1}$$

where $\mathbb{J}$ is the joint probability distribution between X and Y, and $\mathbb{M}$ is their product of marginals. On the basis of this connection, as well as established lower bounds on KL-divergence (KLD) [7], Belghazi et al. [2] proposed the Mutual Information Neural Estimator (MINE), which uses a neural network to estimate a tight lower bound on MI. Importantly, MINE draws on prior work by Nowozin et al. [30] who proposed techniques for using neural networks to estimate a family of divergences, including KL-divergence as well as Jensen-Shannon Divergence (JSD). More recently, Hjelm et al. [17] showed that the JSD and the KLD between the joint distribution of two random variables, and their product of marginals, have an approximately monotonic relationship. Hjelm et al. [17] used this insight for deep representation learning, where they maximize MI (between local and global representations of an image) by simultaneously estimating and maximizing a lower bound on JSD instead of KLD, as JSD demonstrated favourable properties in optimization. In particular, they used a dual representation of JSD to establish a lower bound on MI, via a variational function T_ϕ, as follows:

$$\hat{I}^{(\mathrm{JSD})}_{\phi}(X,Y) \geq \sup_{\phi \in \Phi} \Big\{ \mathbb{E}_{\mathbb{J}}[T_\phi(x,y)] - \mathbb{E}_{\mathbb{M}}[-\log(2 - e^{T_\phi(x,y)})] \Big\}, \tag{2}$$

where T_ϕ is a function of the following form [30]:

$$T_\phi(x,y) = \log(2) - \log(1 + e^{-V_\phi(x,y)}). \tag{3}$$

Here, $V_\phi(x,y) : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ denotes an almost everywhere differentiable function with parameters ϕ. This ensures that $T_\phi(x,y) < \log(2)$ for any value of V_ϕ, and consequently, the second term in (2) is finite as required.

We draw on [17] to propose a ranking loss function that maximizes MI by maximizing a lower bound on JSD, as in (2). Specifically, we use a neural network to learn the variational function T_ϕ, which we define as a function of a distance measure over the learned embedding space.

3.2. RankMI loss

Let $z_i = f_\theta(x_i)$ be an image embedding computed over an image, x_i, via a deep neural network, $f_\theta : \mathbb{R}^n \rightarrow \mathbb{R}^d$, referred to as the *embedding network*. The purpose of our loss function is to learn parameters θ such that images that share the same class label, c, are mapped in close proximity in the embedding space and therefore can be retrieved based on distance. Given a batch of B images, our proposed loss function is computed over two sets of paired image embeddings $\mathcal{P} = \{(z_i, z_j) \mid c_i = c_j\}$ and $\mathcal{N} = \{(z_i, z_j) \mid c_i \neq c_j\}$ for $1 \leq i, j \leq B$ and $i \neq j$.

Our sampling procedure for a positive (matching) pair of images (x_i, x_j) consists of initially sampling a class label, then sampling two images independently given the same

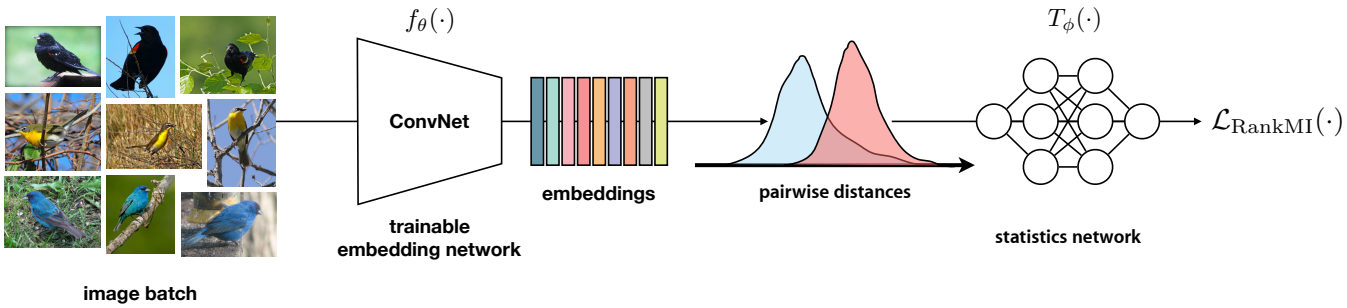

Figure 2. Overview of RankMI training. (left-to-right) Given an image batch, the images are first mapped by a ConvNet, $f_\theta(\cdot)$, to their respective embeddings. Next, matching and non-matching pairwise distances of the embeddings are computed, here illustrated as matching (blue) and non-matching distributions (red). Note, the distributions themselves are not modelled explicitly. The statistics network, $T_\phi(\cdot)$, non-linearly scales distance measurements to compute an estimate of mutual information between image embeddings that share the same class label. The output of the statistics network is passed to our RankMI loss, $\mathcal{L}_{\text{RankMI}}(\cdot)$. Training consists of alternating stochastic gradient descent (SGD) updates of the statistics and the embedding network parameters.

class label. Under this conditional independence assumption, we obtain their joint distribution:

$$\begin{aligned} p(x_i, x_j) &= \sum_{k\in\mathcal{C}} p(c{=}k)p(x_i|c{=}k)p(x_j|c{=}k) \\ &= \sum_{k\in\mathcal{C}} p(c{=}k)p(x_i, x_j|c{=}k) \\ &= \sum_{k\in\mathcal{C}} p(x_i, x_j, c{=}k). \end{aligned} \tag{4}$$

Secondly, for a large number of classes, $||\mathcal{C}||$, and high entropy $p(c)$, which is often the case in retrieval tasks, the sampling procedure for negative pairs closely approximates sampling from the product of marginals:

$$\begin{aligned} &p(x_i)p(x_j) \approx \\ &\sum_{k\in\mathcal{C}} \sum_{\substack{k'\in\mathcal{C} \\ k'\neq k}} p(x_i|c{=}k)p(c{=}k)p(x_j|c{=}k')p(c{=}k'). \end{aligned} \tag{5}$$

A justification for this approximation is provided in the supplementary materials.

Therefore, using sample positive pairs, $\mathcal{P}$, and sample negative pairs, $\mathcal{N}$, in a mini-batch, we can estimate the expectations in (2). Then, we can construct our loss function to maximize the lower bound on the mutual information between the representations (z_i, z_j) of images depicting shared content (e.g., the same product, the same bird species), as in:

$$\begin{aligned} \mathcal{L}_{\text{RankMI}} = &- \frac{1}{||\mathcal{P}||} \sum_{(z_i,z_j)\in\mathcal{P}} T_\phi(z_i, z_j) \\ &- \frac{1}{||\mathcal{N}||} \sum_{(z_i,z_j)\in\mathcal{N}} \log(2 - e^{T_\phi(z_i,z_j)}). \end{aligned} \tag{6}$$

Based on (3), we define the *statistics network*, T_ϕ, as:

$$T_\phi(z_i, z_j) \coloneqq \log(2) - \log(1 + e^{-V_\phi(d_{ij})}), \tag{7}$$

where $V_\phi : \mathbb{R} \to \mathbb{R}$ is a simple multi-layer perceptron (MLP), and d_{ij} is the distance between embeddings (z_i, z_j), e.g., the normalized L_2 distance. Crucially, defining T_ϕ as a function of d_{ij} allows us to connect mutual information to the distance in the embedding space. A discussion is provided in the supplementary materials about the conditions under which our RankMI loss function, (6) and (3), can provide tight estimates of JSD.

Figure 2 provides an overview of our framework. Along with an embedding network for learning d-dimensional feature vectors, z_i, we train a statistics network, T_ϕ, that captures the statistics of the distances between the vectors. Without explicitly modelling the distributions of positive and negative pairs, we optimize a variational function, T_ϕ, which enables estimating their divergence. Once this estimator is available, we use it to provide training signals to the embedding network. This procedure does not necessitate prior assumptions on the distance distributions, allowing us to learn variational functions optimized to separate arbitrarily complex distributions, such as those shown in Figure 1.

Statistics network. Here, we describe considerations for the design of the statistics network, $T_\phi(\cdot)$, such that it satisfies the requirement of ranking positive items closer than negative items for a given query. Let $p(d_{ij}^+|\theta_t)$ and $p(d_{ij}^-|\theta_t)$ be the conditional density functions associated with positive pair distances, d_{ij}^+, and negative pair distances, d_{ij}^-, respectively, given embedding network parameters θ_t at timestep t. A necessary property of any ranking loss is that gradient updates should move positive pairs closer, and push negatives farther in the embedding space. In our RankMI formulation, we express this requirement as follows:

$$\text{sgn}(\nabla_\theta \mathcal{L}_{\text{RankMI}}) = \text{sgn}(\nabla_\theta d_{ij}^+) \tag{8a}$$

$$\text{sgn}(\nabla_\theta \mathcal{L}_{\text{RankMI}}) = -\text{sgn}(\nabla_\theta d_{ij}^-), \tag{8b}$$

$\forall (i,j) \in \{i,j \mid p(d_{ij}^+|\theta_t) \neq 0 \text{ or } p(d_{ij}^-|\theta_t) \neq 0\}$. This

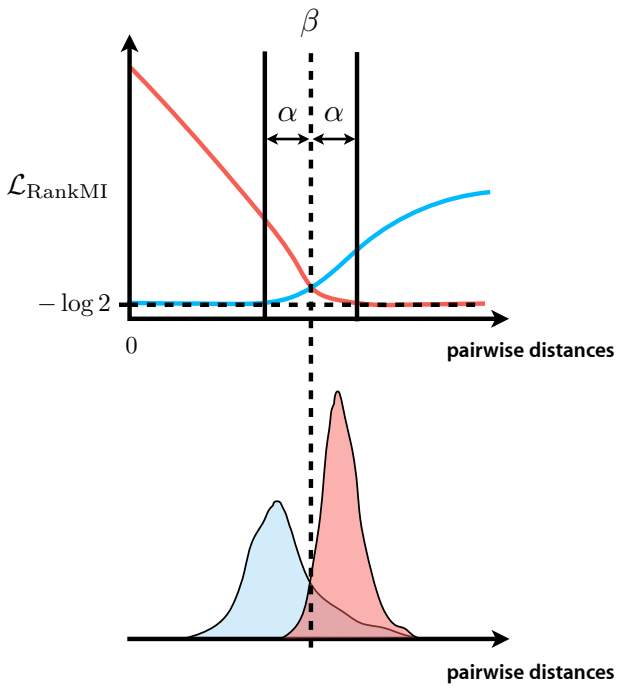

Figure 3. Training loss snapshot. (Top) RankMI loss as defined by T_ϕ. The dashed line marks β estimated via Newton's method. (Bottom) Positive and negative pair distance distributions. β marks the point where $p(d_{ij}^+|\theta) = p(d_{ij}^-|\theta)$.

requirement is satisfied, if the following holds for V_ϕ:

$$\frac{\partial V_\phi}{\partial d_{ij}} < 0. \tag{9}$$

That is, if V_ϕ is decreasing around the neighbourhood of distances for positive and negative pairs at a given timestep during training, the RankMI loss minimizes the positive pair distances, and maximizes the negative pair distances. Intuitively, when (9) holds, mutual information and distance are not only connected, but they also have a monotonic relationship (higher MI corresponds to lower distance).

In practice, this requirement on V_ϕ is generally not violated during SGD, since with initialization of f_θ with pretrained weights, the distributions are already separated in the desired direction at $t=0$. We empirically observe that V_ϕ naturally converges to a decreasing function early in training. However, to better facilitate this property, we add a residual connection from the input of V_ϕ to its output, such that it is of the form:

$$V_\phi(x) := \widetilde{V}_\phi(x) - x, \tag{10}$$

thus making $\partial V_\phi / \partial x \approx -1$ at $t=0$ (with standard weight initialization of $\widetilde{V}_\phi$).

One can introduce soft constraints on V_ϕ to ensure (9) holds. We do not adopt this approach in our experiments but present it in the supplementary material.

Algorithm 1 RankMI training algorithm. Please see Section 3.3 for details. $x[m]$ denotes an indexing operation on a tensor x via a binary mask m of the same shape.

Require: θ_0, ϕ_0: Initial network parameters
Require: lr_θ, lr_ϕ: Learning rates
Require: β_0: Initial value of β
Require: α: Fixed margin (See Figure 3)
Require: k: Alternating gradient descent ratio
Require: B: Mini-batch size

$t \leftarrow 0$
$\beta_t \leftarrow estimateBeta(\phi_0, \beta_0)$
while stoppingCriterionNotMet **do**
  for $x_{1:B}, c_{1:B}$ **in** $dataset$ **do**
    $z_{1:B} \leftarrow f_\theta(x_{1:B})$
    $d_{ij} \leftarrow \|z_i - z_j\|_2 \ \ \forall (i,j) \in \{1, 2, \ldots, B\}$
    $d_{ij}^+ \leftarrow d_{ij}[c_i == c_j]$
    $d_{ij}^- \leftarrow d_{ij}[c_i \neq c_j]$
    if $t \bmod (k+1) \neq 0$ **then**
      ▷ Update statistics network, V_ϕ
      $loss \leftarrow \mathcal{L}_{RankMI}(d_{ij}^+, d_{ij}^-, \phi_t)$
      $\nabla\phi_t \leftarrow \nabla_\phi loss$
      $\phi_{t+1} \leftarrow \phi_t - lr_\phi * \nabla\phi_t$
      $\beta_{t+1} \leftarrow estimateBeta(\phi_{t+1}, \beta_t)$
      ▷ Do not update embedding network, f_θ
      $\theta_{t+1} \leftarrow \theta_t$
    else
      ▷ Update embedding network, f_θ
      $d_{ij}^+ \leftarrow d_{ij}^+[d_{ij}^+ > \beta_t - \alpha]$
      $d_{ij}^- \leftarrow d_{ij}^-[d_{ij}^- < \beta_t + \alpha]$
      $d_{ij}^- \leftarrow negSampling(d_{ij}^-)$
      $loss \leftarrow \mathcal{L}_{RankMI}(d_{ij}^+, d_{ij}^-, \phi_t)$
      $\nabla\theta_t \leftarrow \nabla_\theta loss$
      $\theta_{t+1} \leftarrow \theta_t - lr_\theta * \nabla\theta_t$
      ▷ Do not update statistics network, V_ϕ
      $\phi_{t+1} \leftarrow \phi_t$
      $\beta_{t+1} \leftarrow \beta_t$
    $t \leftarrow t + 1$
return θ_t

3.3. Training

Sampling is important for deep representation learning, regardless of the loss function being used [27]. We view the use of margins as an additional strategy for improving sampling, since margins are used to drop easy positives and negatives from a batch, and focus on the harder, margin-violating samples. Therefore, we design our training algorithm, Algorithm 1, to easily incorporate different margin enforcement and negative sampling schemes.

Figure 3 depicts a snapshot of the RankMI loss learned as a function of pairwise distances. Here, the blue curve represents the loss incurred by a positive pair calculated as

$T_\phi(z_i, z_j)$, the first component in (6). The loss incurred by a negative pair is shown by the red curve, and is calculated as $\log(2 - e^{T_\phi(z_i,z_j)})$, the second component in (6). β is the distance score for which the two curves intersect, that is where positive and negative pairs incur equal loss.

We observe that, analytically, $T_\phi(z_i, z_j) = 0$ is a unique solution that makes the two terms of $\mathcal{L}_{\text{RankMI}}$ equal for the same d_{ij} value. Further, solving for V_ϕ in (7), we get $V_\phi(\beta) = 0$. Motivated by this insight we use a root-finding algorithm, such as Newton's method, to closely approximate β_t given the current parameters ϕ_t at training step t. This step is repeated every time parameters ϕ are updated as shown in Line 14 of Algorithm 1, and adds negligible computational overhead. For details on this procedure, please see supplementary materials.

Once β is found, we can easily incorporate margins α into our training algorithm. Figure 3 highlights the effect of this margin on RankMI loss. In particular, we can drop negative pairs if $d^-_{ij} > \beta + \alpha$, and drop positive pairs if $d_{ij} < \beta - \alpha$. As outlined in Algorithm 1, our training procedure alternates between two phases: updates to the statistics network, and updates to the embedding network. For k steps, we use all positive and negative pairs available in the batch to tighten the divergence lower bound estimated via parameters ϕ and (2). Importantly, we use all available samples for this phase, since using more samples improves the approximation to expectations in (2). Then, we perform a single update on the embedding network, after filtering out samples that are not margin-violating and employing any negative sampling procedure, such as distance-weighted (see Algorithm 1, Lines 19-21). This procedure allows us to leverage the strength of mutual information neural estimators, without sacrificing the ability to employ sophisticated negative sampling strategies.

3.4. Connections of RankMI to other losses

Here, we draw connections between RankMI, and two common ranking losses, the triplet [43] and quadruplet losses [6, 24]. To enable direct comparison, we reformulate both triplet and quadruplet losses using a common notation that captures a spectrum of sampling strategies.

We reformulate triplet loss as follows:

$$\mathcal{L}_{\text{trp}} = \sum_{i=1}^{N} \mathbb{E}_{\mathbb{Q}}[d^+_{ij} - d^-_{ik} + \eta]_+, \tag{11}$$

where N is the number of dataset samples, $\mathbb{Q}$ describes the distribution for sampling a pair of points (j, k) for i to form a triplet (i, j, k), and $[\cdot]_+$ the $\max(0, \cdot)$ operator.

Similarly, we reformulate quadruplet loss as follows:

$$\mathcal{L}_{\text{quad}} = \mathcal{L}_{\text{trp}} + \sum_{i=1}^{N} \mathbb{E}_{\mathbb{R}}[d^+_{ij} - d^-_{kl} + \gamma]_+. \tag{12}$$

Here, $\mathbb{R}$ describes a distribution for sampling (j, k, l) to form quadruplets of positive pairs (i, j) and negative pairs (k, l). We observe that if $\mathbb{R}$ is defined such that $i = k$, the quadruplet term recovers triplet loss as defined in (11). In that sense, we characterize the difference between triplet and quadruplet losses as a difference between the sampling strategies, or the distributions $\mathbb{Q}$ and $\mathbb{R}$.

We observe that if V in the Jensen-Shannon lower bound, (3), is *fixed* to be the following function,

$$V(x) = \begin{cases} \log(e^{-x} - 1) & x < 0 \\ \text{undefined} & x = 0 \\ -\log(e^{x} - 1) & x > 0 \end{cases}, \tag{13}$$

then the value being optimized (ignoring constant terms), as in (6), reduces to the quadruplet term in (12) with a sufficiently large margin γ that retains all quadruplets. The margin γ is used to discard less informative quadruplets. However, instead of a fixed α, RankMI could also incorporate margins stochastically based on random pairings of positive and negative pairs. Therefore, we view triplet as a special case of quadruplet loss with an anchor-based sampling strategy, and quadruplet loss a special case of RankMI with a fixed V. Moreover, we can learn to estimate tighter bounds than those that can be estimated with the fixed function V in (13). This ability to train on tighter bounds is the source of our substantial performance improvements.

Given the similarity between the triplet and quadruplet losses to other losses, we conjecture that other losses may be recast and understood under this common information-theoretic framework. We reserve this for future work.

4. Empirical evaluation

In this section, we present extensive evaluation and comparisons of our metric learning method on the image retrieval and clustering tasks.

Datasets. We conduct experiments on three standard datasets: CUB200-2011 [44], CARS-196 [23], and Stanford Online Products [39]. Our evaluation setup follows the one established in Song et al. [39]. CUB200-2011 contains $11,788$ images of birds depicting 200 species. The first 100 bird species ($5,864$ images) are used for training and the rest are reserved for evaluation. CARS-196 contains $16,185$ images of cars depicting 196 car models. The first 98 car models ($8,054$ images) are used for training and the rest for evaluation. Stanford Online Products, the largest among our datasets, contains $120,053$ images depicting $22,634$ product categories. The first $11,318$ product categories ($59,551$ images) are used for training and the remaining for testing. Note, the evaluation conducted in previous works vary by whether cropped imagery is used based on the provided bounding boxes in the datasets. We

evaluate our method and compare with previous work using the original images without cropping.

Evaluation metrics. To evaluate image retrieval, we use the standard Recall@k metric [39], computed as the percentage of queries that have at least one example from the same category in the k nearest neighbours. For our clustering evaluation, we use the Normalized Mutual Information (NMI) score [28]. The clusters are realized with K-means clustering. NMI is defined as the ratio of mutual information and mean entropy of the clusters and ground truth, $\text{NMI}(\Omega, \mathbb{C}) = 2I(\Omega; \mathbb{C})/(H(\Omega) + H(\mathbb{C}))$, where $I(\cdot, \cdot)$ and $H(\cdot)$ denote mutual information and entropy, respectively, $\Omega = \{\omega_1, \ldots, \omega_K\}$ the ground truth clusters, and $\mathbb{C} = \{c_1, \ldots, c_K\}$ the (K-means) cluster assignments.

4.1. Implementation details

Following previous work [27, 5], in all our experiments we use the standard ResNet-50 architecture [15] as the feature extractor, pre-trained on ImageNet [18]. Both margin-based [27] and FastAP [5] represent high performing (non-ensemble) baselines in our tables. ResNet-50 is followed by a dense layer to produce embeddings of the desired dimensionality. We train our models using Adam [21], and a batch size of 120 for all datasets. For the negative sampling procedure, we adopt distance-weighted sampling [27]. As in prior work, we use horizontal mirroring and random cropping for data augmentation during training. In all experiments, $\beta_0 = 1$, $\alpha = 0.2$, $lr_\phi = lr_\theta = 0.001$, and $k = 1$. Following common practice, the learning rate is divided by a factor of 100 for pre-trained convolutional filters. We use a weight decay multiplier equal to 0.0001 for all parameters in both networks. When sampling mini-batches, we sample at least m images for each class represented in the batch with $m = 5$ for CARS-196 and CUB200-2011, and $m = 2$ for Stanford Online Products. We set the dimensionality of the embedding space to a compact size of 128 for all datasets. These settings were chosen to keep our evaluation in line with previous work and thus isolate performance differences to our loss function and training algorithm. Our method is implemented in PyTorch [32].

Our network architecture for V_ϕ in (3) is as follows:

$$\begin{aligned} d_{ij} &\rightarrow \text{Linear}(1, H) \rightarrow \text{LeakyRelu}(0.1) \\ &\rightarrow (\text{Linear}(H, H) \rightarrow \text{LeakyRelu}(0.1)) \times L \\ &\rightarrow \text{Linear}(H, 1) \rightarrow V_\phi(d_{ij}), \end{aligned} \tag{14}$$

where $H = 128$ and $L = 2$ for all our experiments, and the network weights initialized with Xavier initialization [11].

4.2. Quantitative results

Tables 1, 2, and 3 compare image retrieval and clustering results of our method with previous reported results. We also compare to various embedding ensemble methods that realize embedding vectors by combining embeddings gathered from different layers from the same network or from different networks. As can be seen, we achieve state-of-the-art results on CUB-200-2011, compared to all baselines including the ensemble methods. On CARS-196, we generally improve upon the state of the art across all recalls, with the exception of the ensemble method ABE-8 [20]. On Stanford Online Products, we improve upon or perform competitively with most of the non-ensemble methods, except for FastAP [5]. Note that FastAP (batch $= 256$) realizes an additional performance boost via a heuristic that enables large-batch training. More generally, FastAP makes use of hierarchical class relationships for sampling. While appropriate for Stanford Products Online, such hierarchies are generally not available, and hence not applicable to all datasets, e.g., CUB200-2011 and CARS-196.

Methods		Recall@k 1	2	4	8	NMI
Triplet Semi-hard [35] 128O		42.6	55.0	66.4	77.2	55.4
LiftedStruct [39, 38] 64B		43.6	56.6	68.6	79.6	56.5
StructClustering [38] 64B		48.2	61.4	71.8	81.9	59.2
Proxy NCA [29] 64B		49.2	61.9	67.9	72.4	59.5
Binomial Deviance [41] 512G		50.3	61.9	72.6	82.4	-
N-pairs [37] 64G		51.0	63.3	74.3	83.2	60.4
DVML + Triplet$_2$ + DWS [26] 512G		52.7	65.1	75.5	84.3	61.4
Histogram [41] 512G		52.8	64.4	74.7	83.9	-
Angular Loss [42] 512G		53.6	65.0	75.3	83.7	61.0
HDML + N-pairs [46] 512G		53.7	65.7	76.7	85.7	62.6
HTL [10] 512G		57.1	68.8	78.7	86.5	-
Margin [27] 128R		63.6	74.4	83.1	90.0	69.0
Ensemble	HDC [45] 384G	53.6	65.7	77.0	85.6	-
	BIER [31] 512G	55.3	67.2	76.9	85.1	-
	ABE-8 [20] 512G	60.6	71.5	79.8	87.4	-
RankMI (Ours) 128R		66.7	77.2	85.1	91.0	71.3

Table 1. Recall@k and NMI on CUB200-2011 [44]. Baseline results are taken from the respective papers. The number after each citation denotes the embedding dimensionality. The letter after each embedding dimension indicates the embedding network used. The letters R, G, B, and O denote ResNet-50, GoogLeNet, BN-Inception and Other, respectively.

An ablation study is provided in the supplementary materials analyzing the sensitivity of RankMI to the statistics network depth/width, alternating gradient descent ratio (AGDR), embedding size, and batch size.

4.3. Qualitative results

Figure 4 shows example retrieval results of our learned embeddings on all three datasets. As can be seen, the datasets contain classes with subtle inter-class (e.g., same car make but different model) and large intra-class variations, such as object color, scene illumination, camera viewpoint, and background. Despite these challenges, we can successfully perform retrieval. Even the failure retrievals are reasonable, as the differences between the query and retrieval are difficult to tease out by visual inspection.

Methods		Recall@k				NMI
		1	2	4	8	
Triplet Semi-hard [35] 128O		51.5	63.8	73.5	82.4	53.4
LiftedStruct [39, 38] 64B		53.0	65.7	76.0	84.3	56.9
StructClustering [38] 64B		58.1	70.6	80.3	87.8	59.0
Angular Loss [42] 512G		71.3	80.7	87.0	91.8	62.4
N-pairs [37] 64G		71.1	79.7	86.5	91.6	64.0
Proxy NCA [29] 64B		73.2	82.4	86.4	88.7	64.9
Margin [27] 128R		79.6	86.5	91.9	95.1	69.1
HTL [10] 512G		81.4	88.0	92.7	95.7	-
DVML + Triplet$_2$ + DWS [26] 512G		82.0	88.4	93.3	96.3	67.6
Ensemble	HDC [45] 384G	73.7	83.2	89.5	93.8	-
	BIER [31] 512G	78.0	85.8	91.1	95.1	-
	ABE-8 [20] 512G	85.2	90.5	94.0	96.1	-
RankMI (Ours) 128R		83.3	89.8	93.8	96.5	69.4

Table 2. Recall@k and NMI on CARS-196 [23]. Baseline results are taken from the respective papers. The number after each citation denotes the embedding dimensionality. The letter after each embedding dimension indicates the embedding network used. The letters R, G, B, and O denote ResNet-50, GoogLeNet, BN-Inception and Other, respectively.

Methods		Recall@k				NMI
		1	10	100	1000	
LiftedStruct [39, 38] 64B		62.5	80.8	91.9	-	88.7
Histogram [41] 512G		63.9	81.7	92.2	97.7	-
Binomial Deviance [41] 512G		65.5	82.3	92.3	97.6	-
Triplet Semi-hard [35] 128O		66.7	82.4	91.9	-	89.5
StructClustering [38] 64B		67.0	83.7	93.2	-	89.5
N-pairs [37] 512G		67.7	83.8	93.0	97.8	88.1
Angular Loss [42] 512G		67.9	83.2	92.2	97.7	87.8
HDML + N-pairs [46] 512G		68.7	83.2	92.4	-	89.3
DVML + Triplet$_2$ + DWS [26] 512G		70.2	85.2	93.8	-	90.8
Margin [27] 128R		72.7	86.2	93.8	98.0	90.7
Proxy NCA [29] 64B		73.7	-	-	-	-
FastAP [5] 128R		73.8	88.0	94.9	98.3	-
HTL [10] 512G		74.8	88.3	94.8	98.4	-
FastAP [5] (batch = 96) 512R		75.8	89.1	95.4	98.5	-
FastAP [5] (batch = 256 512R		76.4	89.0	95.1	98.2	-
Ensemble	HDC [45] 384G	70.1	84.9	93.2	97.8	-
	BIER [31] 512G	72.7	86.5	94.0	98.0	-
	ABE-8 [20] 512G	76.3	88.4	94.8	98.2	-
RankMI (Ours) 128R		74.3	87.9	94.9	98.3	90.5

Table 3. Recall@k and NMI on Stanford Online Products [39]. Baseline results are taken from the respective papers. The number after each citation denotes the embedding dimensionality. The letter after each embedding dimension indicates the embedding network used. The letters R, G, B, and O denote ResNet-50, GoogLeNet, BN-Inception and Other, respectively.

5. Conclusions

We proposed RankMI, a new loss function and associated training algorithm for representation learning. Our approach is based on connections to variational divergence maximization and mutual information estimation. We showed that our loss function performs competitively or surpasses state-of-the-art results on standard image retrieval benchmarks. To do this, we use neural networks as function approximators to simultaneously estimate and maximize the divergence between the distance score distributions of matching and non-matching pairs to learn the ranking. Our carefully designed training algorithm can easily incorporate architectural improvements to the embedding network (e.g., ensemble of embeddings), and orthogonal improvements to the negative sampling procedure (e.g., distance-weighted sampling).

Figure 4. Example retrieval results. (left-to-right) query and five nearest neighbours. The three top, middle, and bottom rows correspond to examples from CUB200-2011, CARS-196, and Stanford Online Products, respectively. The number over each image indicates the class label. Blue and red outlines around the retrievals indicate correct and incorrect retrievals, respectively. The last row for each dataset contains incorrect retrievals.

Our approach draws on recent techniques for mutual information estimation, which suffer from high variance estimates. We conjecture that future developments to reduce the variance of these estimates should benefit our proposed approach by making optimization easier. Also, better function approximators (e.g., neural architecture search) for the statistics network may improve performance.

Acknowledgements

We thank Allan Jepson for his helpful discussions and insightful feedback throughout this project.

References

[1] Relja Arandjelovic, Petr Gronát, Akihiko Torii, Tomás Pajdla, and Josef Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1437–1451, 2018. 2

[2] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R Devon Hjelm. Mutual information neural estimation. In *International Conference on Machine Learning*, 2018. 2, 3

[3] Herbin S. Jurie F. Bucher, M. Hard negative mining for metric learning based zero-shot classification. In *European Conference on Computer Vision*, pages 524–531, 2016. 2

[4] Fatih Çakir, Kun He, Sarah Adel Bargal, and Stan Sclaroff. Hashing with mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 3

[5] Fatih Çakir, Kun He, Xide Xia, Brian Kulis, and Stan Sclaroff. Deep metric learning to rank. In *Conference on Computer Vision and Pattern Recognition*, pages 1861–1870, 2019. 2, 7, 8

[6] Weihua Chen, Xiaotang Chen, Jianguo Zhang, and Kaiqi Huang. Beyond triplet loss: A deep quadruplet network for person re-identification. In *Conference on Computer Vision and Pattern Recognition*, 2017. 1, 2, 6

[7] M. D. Donsker and S. R. S. Varadhan. Asymptotic evaluation of certain Markov process expectations for large time. IV. *Communications on Pure and Applied Mathematics*, 1983. 3

[8] Yueqi Duan, Wenzhao Zheng, Xudong Lin, Jiwen Lu, and Jie Zhou. Deep adversarial metric learning. In *Conference on Computer Vision and Pattern Recognition*, 2018. 2

[9] Fartash Faghri, David J. Fleet, Jamie Ryan Kiros, and Sanja Fidler. VSE++: Improving visual-semantic embeddings with hard negatives. In *British Machine Vision Conference*, 2018. 1

[10] Weifeng Ge, Weilin Huang, Dengke Dong, and Matthew R. Scott. Deep metric learning with hierarchical triplet loss. In *European Conference on Computer Vision*, 2018. 2, 7, 8

[11] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pages 249–256, 2010. 7

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Conference on Neural Information Processing Systems*, 2014. 3

[13] David Harwath, Adria Recasens, Didac Suris, Galen Chuang, Antonio Torralba, and James Glass. Jointly discovering visual objects and spoken words from raw sensory input. In *European Conference on Computer Vision*, 2018. 1

[14] Ben Harwood, Vijay Kumar B G, Gustavo Carneiro, Ian Reid, and Tom Drummond. Smart mining for deep metric learning. In *International Conference on Computer Vision*, 2017. 1, 2

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition*, 2016. 7

[16] Xinwei He, Yang Zhou, Zhichao Zhou, Song Bai, and Xiang Bai. Triplet-center loss for multi-view 3D object retrieval. In *Conference on Computer Vision and Pattern Recognition*, 2018. 1, 2

[17] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*, 2019. 2, 3

[18] R. Socher L.-J. Li K. Li J. Deng, W. Dong and L. FeiFei. Imagenet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition*, 2009. 7

[19] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39, 2017. 1

[20] Wonsik Kim, Bhavya Goyal, Kunal Chawla, Jungmin Lee, and Keunjoo Kwon. Attention-based ensemble for deep metric learning. In *European Conference on Computer Vision*, pages 760–777, 2018. 1, 2, 7, 8

[21] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. 7

[22] Ryan Kiros, Ruslan Salakhutdinov, and Richard S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, 2014. 1

[23] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3D object representations for fine-grained categorization. In *International Conference on Computer Vision Workshops*, pages 554–561, 2013. 2, 6, 8

[24] Marc T. Law, Nicolas Thome, and Matthieu Cord. Quadruplet-wise image similarity learning. In *International Conference on Computer Vision*, 2013. 2, 6

[25] Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. Stacked cross attention for image-text matching. In *European Conference on Computer Vision*, 2018. 1

[26] Xudong Lin, Yueqi Duan, Qiyuan Dong, Jiwen Lu, and Jie Zhou. Deep variational metric learning. In *European Conference on Computer Vision*, pages 714–729, 2018. 2, 7, 8

[27] R. Manmatha, Chao-Yuan Wu, Alexander J. Smola, and Philipp Krähenbühl. Sampling matters in deep embedding learning. In *International Conference on Computer Vision*, pages 2859–2867, 2017. 1, 2, 3, 5, 7, 8

[28] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, UK, 2008. 7

[29] Yair Movshovitz-Attias, Alexander Toshev, Thomas K. Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. In *International Conference on Computer Vision*, pages 360–368, 2017. 1, 2, 7, 8

[30] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-GAN: Training generative neural samplers using variational divergence minimization. In *Conference on Neural Information Processing Systems*, 2016. 3

[31] Michael Opitz, Georg Waltner, Horst Possegger, and Horst Bischof. Deep metric learning with BIER: boosting inde-

pendent embeddings robustly. *CoRR*, abs/1801.04815, 2018. 1, 2, 7, 8

[32] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Conference on Neural Information Processing Systems*, pages 8024–8035, 2019. 7

[33] Filip Radenovic, Giorgos Tolias, and Ondrej Chum. Fine-tuning CNN image retrieval with no human annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(7):1655–1668, 2019. 2

[34] Jérôme Revaud, Jon Almazán, Rafael S. Rezende, and César Roberto de Souza. Learning with average precision: Training image retrieval with a listwise loss. In *International Conference on Computer Vision*, pages 5106–5115, 2019. 2

[35] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. In *Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015. 1, 2, 7, 8

[36] Richard Socher, Andrej Karpathy, Quoc V. Le, Christopher D. Manning, and Andrew Y. Ng. Grounded compositional semantics for finding and describing images with sentences. *Transactions of the Association for Computational Linguistics*, 2, 2014. 1

[37] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Conference on Neural Information Processing Systems*, pages 1849–1857, 2016. 2, 7, 8

[38] Hyun Oh Song, Stefanie Jegelka, Vivek Rathod, and Kevin Murphy. Learnable structured clustering framework for deep metric learning. *CoRR*, abs/1612.01213, 2016. 2, 7, 8

[39] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *Conference on Computer Vision and Pattern Recognition*, pages 4004–4012, 2016. 1, 2, 6, 7, 8

[40] O. Tadmor, T. Rosenwein, S. Shalev-Shwartz, Y. Wexler, and A. Shashug. Learning a metric embedding for face recognition using the multibatch method. In *Conference on Neural Information Processing Systems*, 2016. 1, 2

[41] Evgeniya Ustinova and Victor Lempitsky. Learning deep embeddings with histogram loss. In *Conference on Neural Information Processing Systems*, 2016. 1, 2, 7, 8

[42] Jian Wang, Feng Zhou, Shilei Wen, Xiao Liu, and Yuanqing Lin. Deep metric learning with angular loss. In *International Conference on Computer Vision*, 2017. 1, 2, 7, 8

[43] Kilian Q. Weinberger and Lawrence K. Saul. Fast solvers and efficient implementations for distance metric learning. In *International Conference on Machine Learning*, 2008. 1, 2, 6

[44] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010. 2, 6, 7

[45] Yuhui Yuan, Kuiyuan Yang, and Chao Zhang. Hard-aware deeply cascaded embedding. In *International Conference on Computer Vision*, pages 814–823, 2017. 1, 2, 7, 8

[46] Wenzhao Zheng, Zhaaodong Chen, Jiwen Lu, and Jie Zhou. Hardness-aware deep metric learning. In *Conference on Computer Vision and Pattern Recognition*, 2019. 2, 7, 8