Potential Field Based Deep Metric Learning

Shubhang Bhatnagar Narendra Ahuja
University of Illinois Urbana-Champaign
{sb56, n-ahuja}@illinois.edu

Abstract

Deep metric learning (DML) involves training a network to learn a semantically meaningful representation space. Many current approaches mine n-tuples of examples and model interactions within each tuplets. We present a novel, compositional DML model that instead of in tuples, represents the influence of each example (embedding) by a continuous potential field, and superposes the fields to obtain their combined global potential field. We use attractive/repulsive potential fields to represent interactions among embeddings from images of the same/different classes. Contrary to typical learning methods, where mutual influence of samples is proportional to their distance, we enforce reduction in such influence with distance, leading to a decaying field. We show that such decay helps improve performance on real world datasets with large intra-class variations and label noise. Like other proxy-based methods, we also use proxies to succinctly represent sub-populations of examples. We evaluate our method on three standard DML benchmarks- Cars-196, CUB-200-2011, and SOP datasets where it outperforms state-of-the-art baselines.

1. Introduction

The goal of visual metric learning is to learn a representation function where common distance metrics like the ℓ_2 distance capture similarity between images. Semantic similarity is useful for applications such as image/video retrieval and search [3, 4, 21, 34], open set classification and segmentation [6, 14, 32, 37], few-shot learning [41, 45], person reidentification[7, 54] and face verification [10, 17, 31]. Deep neural networks have helped make significant advances in visual metric learning [40], as they are trained to learn a non-linear mapping of images onto a semantic space.

Existing Methods: Popular loss functions to train such networks help force tuplets of examples/samples/embeddings of the same class to be closer than those in different classes. Examples include the triplet loss [52], the contrastive loss [8] and their extensions [42, 51].

The utility of supervision provided by samples in a tuplet depends on samples' relative locations and labels. Most tuplet-based methods use sample mining strategies [15, 40] to choose a subset of informative tuplets to train the network, without which they suffer from lower performance and slower convergence [53]. Such strategies not only require the computationally demanding calculation of the relative hardness of all tuples formed (O(n^3) for triplets among n samples), but also incur a loss of valuable supervision by discarding interactions. Specifically, limiting interactions to only a subset of all possible tuplets leads to greater susceptibility to noise [25], overfitting on the training set [26], and reduced learning of intra-class variations and generalizable features [38].

Proxy-based methods, first proposed in [34] attempt to mitigate some of these shortcomings by instead using class representative 'proxies' as one or more hypothesized centers for each class. Computation of the loss is then in terms of sample-proxy distances instead of sample-sample distances, helping circumvent the problem of tuplet sampling. However, these methods have been found to learn less generalizable intra-class features due to the exclusive use of proxy-sample distances while ignoring sample-sample distances[39]. This effect is compounded when proxies, whose locations are themselves learned, end up far from sample embedding locations they are intended to represent.

Proposed Representation: To address these challenges, we propose Potential Field based Deep Metric learning (PFML), a metric learning framework that uses a continuous, potential field representation of all data samples motivated by potential fields used to model influence of charges in physics. Besides its analogical relevance, our motivation for using this representation comes from its previous, very successful uses, where it also helps overcome complexity in modeling interactions. These applications include planning paths of a robot avoiding obstacles [19] and planning an efficient strategy for a robot manipulator to grasp an object [20].

In PFML, we view each data sample as a charge, creating a field. The fields due to the individual samples are added (superposed) to obtain the global potential field. This global field thus constitutes a *compositional* representation, capturing the combined influence at a given location (or sample) as a weighted *addition* of the influences of the individual samples. Each sample generates two types of fields: (1) an attraction field to drive similar examples closer to it, and (2) a repulsion field that drives dissimilar examples away from it, each weakening with distance.

We also make use of proxies like previous work, but only to augment the potential field generated by the current batch of samples. This is in contrast to past work that completely replaces sample-sample interactions with sample-proxy interactions. Preserving sample-sample relations through the potential field helps learn better intra-class features.

Advantages of Potential Field: (1) Using the potential field representation enables us to model interactions among all sample embeddings, as opposed to modeling those among smaller number of samples (or proxies) as done in methods using point-tuplet based (e.g., triplet) loss. (2) Addition of potentials due to all points increases robustness to noise since the effect of noise on interactions among a smaller number of samples will have a larger variance while also (3) Considering all interactions improves the quality of features learned, unlike tuplet based methods which lose performance when all interactions are considered[53]. (4) This is because of a major feature that differentiates our potential field from past methods: The variation of strength of interaction between two points as the distance between them increases: instead of remaining *constant* or even becoming *stronger*, as is the case with most existing methods, e.g. [8, 21], in our model it becomes *weaker* with distance.[1] The decay property is helpful in several ways: (4a) It ensures the intuitive expectation that two distant positive samples are too different to be considered as variants of each other, helping treat them as different varieties (e.g., associated with different proxies). (4b) The decay property also significantly improves PFML performance for the specific type of noise affecting labels, e.g., due to annotation errors common in real-world datasets (Sec. 4.3). (4c) As a result of the decay, the learned proxies remain closer to (at smaller Wasserstein distance W_2 from) the sample embeddings they represent than for (e.g., current proxy-based) methods where interactions strengthen with distance [21, 34, 36]; Sec 3.4), thereby enhancing their desired role.

Contributions: The main contributions of the proposed approach are:

- We present PFML, a novel framework for metric learning with a compositional representation using a continuous, potential field to model all sample interactions instead of only interactions between a subset of all possible n-tuplets. This enables better feature learning, especially in the presence of annotation noise.
- PFML reverses the almost universally used model of inter-sample interactions strengthening with distance; interaction strength instead weakens with distance. This (1) helps improve robustness in the presence of label noise, and (2) we prove that it helps better align the distribution of proxies with the sample embeddings they represent.
- We evaluate PFML on three zero-shot image retrieval benchmarks: Cars-196, CUB-200-2011, and the SOP dataset where it outperforms current, state-of-the-art techniques in both the standard no-label-noise scenario, and with even greater margins (>7% R@1) in the more realistic cases of label noise.

2. Related Work

Broadly, DML may be divided into two types, ranking/pair-based and proxy-based [21, 51].

Ranking/Pair based DML: These methods utilize losses that derive information from relations between inter-sample distances in pairs/ tuples of sample embeddings. Examples include the classical contrastive loss [8] or the triplet loss[40, 52, 53], which extends the concept to take into account relative distances in a triplet. [42] and [43] generalize these constraints to include multiple dissimilar samples. Mining informative pairs/tuples from a mini-batch plays an important role in the effectiveness of pair-based methods. Several strategies [15, 40, 57] have been proposed to pick samples based on their relative difficulty, with hard and semi-hard negative (picking pair of inter-class examples close together) and easy positive (same class examples close together) [25, 55] being popular. PFML naturally emphasizes such local (dis) similarities by virtue of its decaying formulation without the complexity of mining tuplets. Also note that unlike PFML, none of these strategies optimize all interactions among a batch of samples.

Proxy-based DML: ProxyNCA [34] and its variants [23, 48] attempt to minimize the distance of a sample from the proxy of its class while maximizing the distance from proxies of other classes. Proxy Anchor [21] builds on these by proposing a modified loss that takes into account sample hardness. [10, 36] introduced the use of multiple proxies per class to better model sub-clusters.

Potential Field: Potential fields have been used in a variety of applications to model interactions of objects. Of interest here is their use in robot path planning [9, 19], shape representation [2, 18], image segmentation [1, 47] and object manipulation [20]. Here, they have been used to direct a robot towards its goal using an attraction field while avoiding obstacles using repulsion fields. Training supervision for DML shares a similar goal of moving sample embeddings towards other nearby samples of the same class, while avoiding dissimilar examples belonging to other classes.

[1] It remains constant over a small radius and decays outside; however, for brevity, we will disregard this detail in the rest of this paper, and refer to the influence as decaying with distance, or as the decay property.

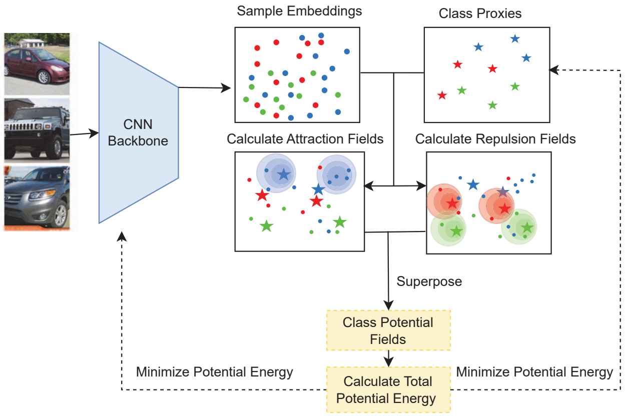
Sample Embeddings
Class Proxies
CNN Backbone
Calculate Attraction Fields
Calculate Repulsion Fields
Superpose
Class Potential Fields
Calculate Total Potential Energy
Minimize Potential Energy
Minimize Potential Energy

Figure 1. Overview of our Potential-field based DML pipeline. The process includes (1) Computing attraction and repulsion fields generated by each embedding and proxy, (2) Computing the class potential fields by superposition of individual fields (3) Evaluating total potential energy by summing up the potentials of embeddings and proxies under the class potential field and (4) Updating locations of sample embeddings (through network parameters) and proxies to minimize total potential energy through backprop.

3. Method

3.1. Problem Setup

The goal of deep metric learning is to define a semantic distance metric $d(\mathbf{x}_1, \mathbf{x}_2)$ for any pair of samples $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{D}$, where $\mathcal{D} = \{\mathbf{x}_i, y_i\}, i \in \{1....|\mathcal{D}|\}$ and labels $y_i \in \{1, \dots N\}$ represent N classes. The learned distance metric for any two points $d(\mathbf{x}_1, \mathbf{x}_2)$ should represent their semantic dissimilarity. Instead of directly learning the distance metric, a projector function f_θ is often represented by a neural network, parameterized by θ learned from data and distance $d(\mathbf{x}_1, \mathbf{x}_2) = \|f_\theta(\mathbf{x}_1), f_\theta(\mathbf{x}_2)\|_2$ is given by the Euclidean distance between the projections/embeddings. For simplicity, we will refer to the embeddings $f_\theta(\mathbf{x}_i)$ belonging to a batch $\mathcal{B}$ sampled from $\mathcal{D}$ as $\mathbf{z}_i$.

3.2. The Potential Field

With our potential field representation, embeddings need to be driven towards other nearby embeddings belonging to the same class, while also being driven away from embeddings of other classes. This is reminiscent of the behavior of an isolated system of electric charges, where dissimilar charges are drawn together while similar ones are repelled. Analogously, we define attractive and repulsive potentials created by a single embedding, with the net potential generated by a set of samples given by superposing individual potentials exerted by each embedding. The gradient of the net potential at a point yields the net force (on network update) on a sample embedding located at that point.

As samples of each class have a different set of similar and dissimilar points, we define a separate potential field Ψ_j, for each class $j \in \{1, \dots N\}$. The class potential field Ψ_j models the force applied on embeddings belonging to class j due to all embeddings. It is formed from a superposition of potentials belonging to sample embeddings from all classes, with embeddings from the same class exerting an attractive potential field (to pull similar points closer) while those from other classes exert a repulsive potential field (to push apart dissimilar points). We assume that each sample embedding z_i has a unit charge, so the net force on it is the gradient of the potential field Ψ_{y_i} of its class.We design the attractive and repulsive potential functions by taking motivation from both, electrostatic fields and metric learning literature, which we discuss in detail in the following subsections. Figure 1 provides an overview of our complete pipeline of generation and use of the potential field for metric learning .

3.2.1. Potential field exerted by a single point

By using forces on the embeddings, we wish the training process to move the embeddings to a state of equilibrium at the end of training in which all embeddings from a class are nearby. To have a potential function exhibiting this behavior, we ensure that individual potential field has the following two properties:

Property 1 - Extremum Property: Potential generated by an individual embedding achieves an extremum at the embedding (minimum for attraction field, maximum for repulsion). This ensures that similar points are pulled together and dissimilar ones are pushed apart on minimizing potential.

Property 2 - Decay Property: The magnitude of the gradient of the potential (force) due to an embedding decays with distance. Such a decay ensures two desired characteristics. (1) An embedding is not pulled towards a distant sample embedding belonging to the same class, which is likely to be too different to be considered as a variant of it; instead, the decayed influence helps treat the distant sample as significantly different variety of the class, associated with other nearby embeddings of the class. (2) Like for a single charge in the Extremum property, a superposition of potentials of different embeddings still yields local minima at or near sample embeddings. As shown in Sec 3.4, this helps proxies come close to such minima and thus help better model the data distribution.

Given these properties motivated by electrostatic fields, we select the attraction and repulsion potentials ψ_{att}, ψ_{rep} generated by an embedding $\mathbf{z}_i$, with their strength at location $\mathbf{r}$ defined as:

$$\psi_{att}(\mathbf{r}, \mathbf{z}_i) := \begin{cases} -\dfrac{1}{\delta^\alpha} & \text{if } \|\mathbf{r} - \mathbf{z}_i\|_2 < \delta \\ -\dfrac{1}{\|\mathbf{r} - \mathbf{z}_i\|_2^\alpha} & \text{otherwise.} \end{cases} \quad (1)$$

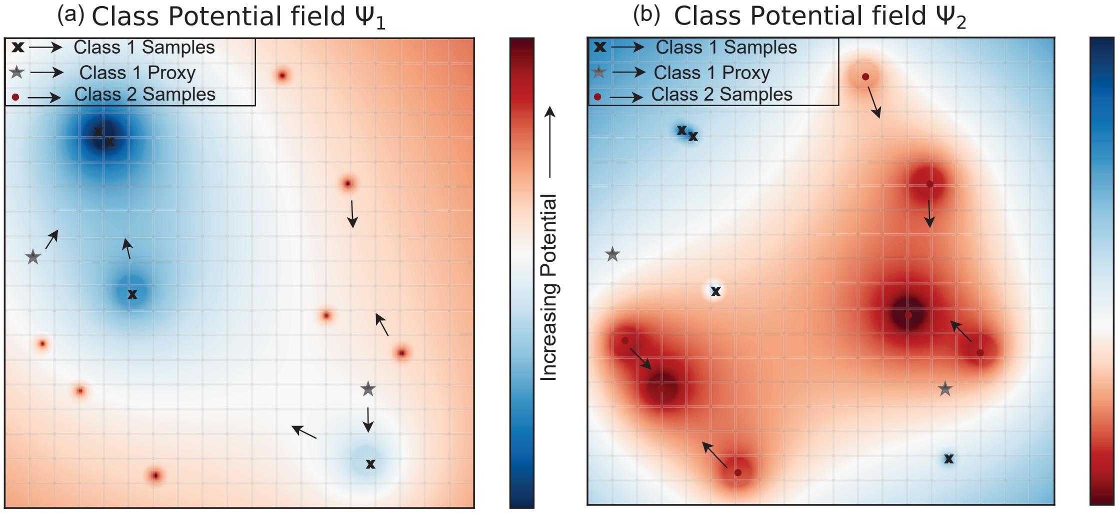

Figure 2. An example of the class potential fields Ψ_1 and Ψ_2 (Sec. 3.2.3), created by superposing the fields of individual embeddings (Sec 3.2.1) belonging to Classes 1 and 2. Arrows denote the gradient of the respective potentials representing the net force on them. (a) Ψ_1 draws samples/proxies of class 1 towards nearby samples/ proxies of class 1 while keeping them at least δ distance away from embeddings of class 2. Proxies at starred locations are drawn towards the nearest Class 1 embeddings which are potential minima, helping them better model the data distribution (Sec 3.4). (b) Ψ_2 draws embeddings of class 2 towards other nearby embeddings of class 2 instead of distant class 2 embeddings, which might be a significantly different variant of the class (or potentially a mislabeled data point), helping better feature learning (or improving noise robustness if mislabeled). Ψ_2 also keeps class 2 embeddings δ distance them away from embeddings of class 1.

$$\psi_{rep}(\mathbf{r}, \mathbf{z}_i) := \begin{cases} \dfrac{1}{\|\mathbf{r} - \mathbf{z}_i\|_2^{\alpha}} & \|\mathbf{r} - \mathbf{z}_i\|_2 < \delta \\ \dfrac{1}{\delta^{\alpha}} & \text{otherwise.} \end{cases} \tag{2}$$

Here, α is a hyperparameter that determines the rate of decay of the field (we evaluate it's effect in Sec. 4.5 instead of fixing $\alpha = 1$ as done in electrostatic fields). Note that the attraction potential is negative and increases with distance, implying that the force between the embedding and the point is attractive. The repulsion potential is positive and decreasing with distance, indicating repulsion.

A constant attraction potential inside the δ sphere (i.e., force = 0) ensures that sample embeddings within δ distance of one another do not continue to attract each other, thus preventing the network from embeddings from collapsing to a single point which harms generalizability of the features learned (analogous to the effect margins used in tuplet-based losses have).

Limiting the repulsion potential to a sphere of radius δ ensures that embeddings not belonging to the same class are driven to a distance $> \delta$. Since the aforementioned constant attraction potential ensures that δ is the maximum distance between samples z_i from the same class, this leads to connected components of δ spheres formed by nearby embeddings of the same class. Extending repulsion to distances $> \delta$ does not provide any useful additional supervisory information while forcing further unmotivated inter-class separation; as we empirically show in Sec. 4.5, such excessive separation does indeed harm performance.

3.2.2. Proxies to Represent Sample Populations

To represent the image embeddings not sampled in the current batch $\mathcal{B}$, we propose to use a set of M proxies per class for each of the N classes, $\mathbf{p}_{j,k}, k \in \{1, \ldots M\}, j \in \{1, \ldots N\}$, and augment the potential field generated by the embeddings in the batch by those generated by the proxies. The proxies serve as stand-ins for the larger numbers of out-of-the-batch embeddings for calculating the attraction and repulsion fields, while also being affected by the field themselves like all embeddings. The proxies, being learnable parameters with high learning rates, tend to gravitate toward the nearest class modes of the potential field (and hence the class data distribution). The number of proxies M chosen should be large enough to capture the class-modes.

3.2.3. Class Potential Field

Given a batch $\mathcal{B}$ of samples and all proxies, we calculate the class potential field $\Psi_j(\mathbf{r})$ at a location $\mathbf{r}$ due to a class j as a superposition of their attraction and repulsion potentials:

$$\Psi_j(\mathbf{r}) = \Psi_{j,att}(\mathbf{r}) + \Psi_{j,rep}(\mathbf{r}), \text{ where} \tag{3}$$

$$\Psi_{j,att}(\mathbf{r}) = \sum_{i=1}^{\|\mathcal{B}\|, y_i = j} \psi_{att}(\mathbf{r}, \mathbf{z}_i) + \sum_{k=1}^{M} \psi_{att}(\mathbf{r}, \mathbf{p}_{j,k}) \tag{4}$$

$$\Psi_{j,rep}(\mathbf{r}) = \sum_{i=1}^{\|\mathcal{B}\|, y_i \neq j} \psi_{rep}(\mathbf{r}, \mathbf{z}_i) + \sum_{\gamma=1,\gamma\neq j}^{N} \sum_{k=1}^{M} \psi_{rep}(\mathbf{r}, \mathbf{p}_{\gamma,k}) \tag{5}$$

Figure 2 visualizes the class potential fields Ψ_1 and Ψ_2 for a given set of embeddings belonging to two classes.

3.3. Training

Using gradient descent, the network is trained to minimize the total potential energy $\mathcal{U}$ of all proxies and embeddings in the batch $\mathbf{z}_i \in \mathcal{B}$ by applying a force on embeddings and proxies in the direction of $-\nabla\Psi_{y_i}(\mathbf{z}_i)$. $\mathcal{U}$ given by :

$$\mathcal{U} = \sum_{i=1}^{\|\mathcal{B}\|} \Psi_{y_i}(\mathbf{z}_i) + \sum_{j=1}^{N} \sum_{k=1}^{M} \Psi_j(\mathbf{p}_{j,k}) \tag{6}$$

3.4. Why decay interaction strength with distance ?

PFML uses a potential field where the gradient (interaction strength) decays with distance, which is unlike previous approaches. This leads to the two benefits described in Sec 3.2.1, which we discuss in more detail here.

3.4.1. Robustness when learning with datasets having large intra-class variations and noisy labels

The decay property helps PFML model all interactions even when using datasets with large intra-class variations. As seen in Figure 2, PFML draws embeddings towards other nearby embeddings from the same class rather than distant class embeddings that might be too different to be considered a variation of it. It also helps PFML become significantly more robust to the effect of incorrectly labeled examples as compared to existing methods. Recent work [26, 30] shows that such examples are common in metric learning benchmarks. The decay de-emphasizes the contribution of such mislabeled examples, that are typically located away from correctly labeled samples. We empirically confirm this in Sec. 4.3, demonstrating that methods not adhering to the decay property (i.e., all previous tuplet-based and proxy-based methods) experience significantly greater performance degradation due to label noise compared to PFML.

3.4.2. Better alignment of proxies with data distribution

The decaying interaction strength helps PFML better align the distribution of proxies with the underlying sample data distribution (of the complete dataset) that they represent. To analyze this, we compare PFML with the closely related contrastive loss [8] that also models pairwise interactions like PFML (defined in Eq. 1 & 2), and hence can be analyzed similarly by using a potential field. Specifically, we introduce Contrastive Potential based Metric Learning (CPML) by simply extending the contrastive loss by using multiple proxies per class, and subsequently analyzing the resulting loss function through a potential field framework. In CPML, strength of interaction is directly proportional to the square of the distance from it ($\|\mathbf{r}-\mathbf{r}_0\|_2^2$) like in the original contrastive loss . For fair comparison, we keep all other details (use of δ, proxies) as the same, with its associated potential fields denoted by Ψ^*_{att} , Ψ^*_{rep} (formally defined in Supplement Sec 1). We analyze this theoretically using the following proposition and its corollary:

Proposition 1:*Let $Z = \{z_1 \ldots z_n\}$ be a set of sample embeddings belonging to a class, then there exists a $0 < \delta < \dfrac{\min_{i,j} \|z_i - z_j\|_2}{2\left(1+\frac{1}{n}\right)}$ and points $\mathbf{z}_{min,i}$ within δ distance from each embedding $z_i,\ i \in \{1 \ldots n\}$ such that the attractive potential field Ψ_{att} defined using Z, δ has a minimum at each $\mathbf{z}_{min,i}$. The field Ψ^*_{att} defined by them does not achieve a minimum at points within δ distance from all z_i.*

Proposition 1 is a consequence of the potential field of an embedding dominating in its vicinity over potential fields of other distant embeddings. Its proof relies on the Decay property, which is not satisfied by Ψ^* and can be found in Supp. Sec 1. A consequence of this is:

Corollary 1: *Let $Z = \{z_1 \ldots z_n\}$ be a set of sample embeddings from a class exerting an attraction field on a set of proxies $P = \{\boldsymbol{p}_1 \ldots \boldsymbol{p}_m\}$. Consider the equilibrium distribution P_{eq} of proxies minimizing the potential energy. If the potential field is defined by Ψ_{att}, then the Wasserstein distance W_2 between P_{eq} and the subset of data they represent is lower than when the potential field is defined by Ψ^*_{att}.*

The proof for Corollary 1 relies on Proposition 1 and can be found in the Supplement (Sec. 2).

Discussion: As proved in Corollary 1, CPML leads to proxies having a larger distance from the data distribution they represent (measured by W_2) than our proposed PFML, making the use of proxies in such a framework less effective. This leads to the proxies forming poorer stand-ins for the sample embeddings they represent, reducing the quality of features learned (verified in Sec. 4). This is caused by CPML not following the decay property (which is its only difference in design wrt PFML). Other proxy-based losses like the Proxy NCA [34], Soft Triplet [36], and Proxy Anchor [21] which perform similarly as CPML (see Tab. 1), also share with it the property of force increasing with distance (violating the decay property). Due to this, we expect them to similarly have a poorer alignment of proxies with underlying data distribution as compared to PFML (Applying an analysis similar to the one in Proposition 1 to them is considerably more challenging due to the absence of a potential field structure.). We verify this empirically in Sec. 4.4 by measuring the average distance between proxies and the closest data points (W_2 distance) for these methods.

3.5. Advantages in Learning Fine-grained Features

In contrast with all previous proxy-based methods, PFML (1) Models sample-sample interactions directly using the potential field, gaining fine-grained supervision, and (2) Makes use of an explicit margin-like δ for proxies, which has been shown to help zero-shot generalization and fine-grained feature learning in tuplet-based methods[53]. Turning off the interaction within the δ radius also prevents a bias towards learning isotropic distributions as is present in other proxy-based methods [39], helping us obtain better intra-class resolution.

4. Experiments and Results

4.1. Setup

Datasets: We empirically compare PFML with current state-of-the-art baselines on three public benchmark datasets **(1)** The Cars-196 dataset [24] which includes 16,185 images from 196 car model categories, **(2)** CUB-200-2011 dataset [49] consisting of 11,788 images from 200 bird species, and **(3)** Stanford Online Products (SOP) dataset [43] containing 120,053 images of 22,634 different products sold online. The training and testing splits for the three datasets follow standard zero-shot retrieval-based evaluation protocol [21, 51, 61] of using the first half of the classes for training and the second half for testing.

Backbone: To enable standardized comparison with a wide variety of methods, we evaluate our method using the Inception with batch normalization (IBN) [46], ResNet-50 [16], ViT [11] and DINO [5] backbones commonly used by DML methods. ImageNet pre-training is used for initialization, and only the last fully connected layer is changed to fit the dimension of the embedding needed. We follow past work [21, 36, 48] and ℓ_2 normalize the final output. An input image size of 224×224 is used for all experiments.

Training parameters: We train the backbone for 200 epochs on each dataset. We use the Adam optimizer with the learning rate chosen as $5e^{-4}$. We scale up the learning rate of our proxies by 100 times following [21, 34, 36] for better convergence. We choose values of δ between $[0.1, 0.3]$, α between $\{0, 6\}$ via cross-validation. We use $M = 15$ for the CUB-200 and Cars-196 datasets, while $M = 2$ is used for the SOP dataset. All experiments are performed on a single NVIDIA A4000 GPU.

Evaluation Settings: We follow [21, 22, 29, 51] in using 224×224 sized center crops from 256×256 images for evaluation. We measure image retrieval performance using Recall@K, which computes the percentage of samples which have a valid similar neighbor (belonging to same class) among its K nearest neighbors.

4.2. Image Retrieval Performance

As seen in Table 1, our method outperforms all other methods on Image Retrieval on the CUB-200-2011, Cars-196 and SOP datasets as measured by the Recall@K metric. It consistently outperforms other methods irrespective of the backbone used, significantly outperforming the best-performing proxy-based methods (which similar to us, only change the loss function) like ProxyAnchor[21], ProxyGML[61] and SoftTriplet [36] in terms of Recall@1 (R@1), by more than 5% on Cars-196, 3.7% on CUB-200-2011 and 1.5% on the SOP dataset. It also outperforms the significantly more complex, current state-of-the-art HIST[28] by 3.1%, 2% and 1.5% in terms of R@1 when using the Resnet-50 backbone on the Cars-196, CUB-200-2011 and SOP datasets respectively. It also significantly outperforms the best tuplet-based loss: the Multi-Similarity loss by 6%, 5.8% and 2.4% in terms of R@1 when using the BN Inception backbone on the Cars-196, CUB-200 and SOP datasets respectively. Additionally, our method outperforms CPML (Sec. 3.4.2, a simple combination of the classical contrastive loss with proxies) by 7.5%, 5.1%, and 3.5% on CUB200, Cars and SOP datasets. Note that our performance gains are larger than current SOTA compared to improvements reported by previous efforts on the same benchmarks, despite our intuitive and simpler method.

We also evaluate our method 1) Using the protocol and metrics proposed in [35] and (2) When used to learn smaller 128-dim embeddings for more efficient retrieval. Our method still outperforms all SOTA methods in both cases. Details of these experiments are in Supp. Sec. 3. We also show qualitative retrieval results of our method in Supp. Sec 6.

4.3. Image Retrieval in Presence of Label Noise

We evaluate the robustness of our method vs. state-of-the-art baselines in the realistic scenario of label noise (e.g., due to mislabeling, which is very common, particularly in fine-grained datasets such as those used for DML). Specifically, we randomly corrupt labels of 20% of samples in the training set. As seen in the results in Table 2, this leads to a significant decline in the performance of all methods. But our method is the least affected by the noise. Other methods like the triplet [15] or ProxyAnchor[21] suffer much greater declines because they are affected more by distant points represented by the mislabeled examples. Our method significantly outperforms Proxy Anchor [21] (the second-best method) by 6% and 7.6% in terms of R@1 on the CUB-200-2011 and Cars-196 datasets.

4.4. Distance Between Proxies and Sample Embeddings

We measure the average (over a complete epoch) Wasserstein distance W_2 between proxies and the subset of training

Benchmarks →	**CUB-200-2011**			**Cars-196**			**SOP**		
Methods ↓ (Chronological)	R@1	R@2	R@4	R@1	R@2	R@4	R@1	R@10	R@100
ResNet50 (512 dim)									
ESPHN [55]	64.9	75.3	83.5	82.7	89.3	93.0	78.3	90.7	96.3
N.Softmax [58]	61.3	73.9	83.5	84.2	90.4	94.4	78.2	90.6	96.2
DiVA [33]	69.2	79.3	-	87.6	92.9	-	79.6	91.2	-
Proxy NCA++ [48]	64.7	-	-	-	85.1	-	- 79.6	-	-
Proxy Anchor [21]	69.7	80.0	87.0	87.7	92.9	95.8	-	-	-
DCML-MDW [59]	68.4	77.9	86.1	85.2	91.8	96.0	79.8	90.8	95.8
MS+DAS [29]	69.2	79.2	87.1	87.8	93.1	96.0	80.6	91.8	96.7
HIST [28]	71.4	81.1	88.1	89.6	93.9	96.4	81.4	92.0	96.7
HIER[22]	70.1	79.4	86.9	88.2	93.0	95.6	80.2	91.5	96.6
HSE-PA [56]	70.6	80.1	87.1	89.6	93.8	96.0	80.0	91.4	96.3
CPML (Sec. 3.4)	68.3	78.7	86.2	85.2	91.5	95.2	79.4	90.7	96.1
Potential Field (Ours)	**73.4 ± 0.3**	**82.4± 0.1**	**88.8± 0.1**	**92.7± 0.3**	**95.5 ± 0.1**	**97.6± 0.1**	**82.9 ± 0.2**	**92.5 ± 0.2**	**96.8 ± 0.1**
BN Inception (512 dim)									
HTL [13]	57.1	68.8	78.7	81.4	88.0	92.7	74.8	88.3	94.8
MultiSimilarity [51]	65.7	77.0	86.3	84.1	90.4	94.0	78.2	90.5	96.0
SoftTriple [36]	65.4	76.4	84.5	84.5	90.7	94.5	78.3	90.3	95.9
CircleLoss [44]	66.7	77.4	86.2	83.4	89.8	94.1	78.3	90.5	96.1
DiVA [33]	66.8	77.7	-	84.1	90.7	-	78.1	90.6	-
ProxyGML [61]	66.6	77.6	86.4	85.5	91.8	95.3	78.0	90.6	96.2
Proxy Anchor [21]	68.4	79.2	86.8	86.1	91.7	95.0	79.1	90.8	96.2
DRML-PA [60]	68.7	78.6	86.3	86.9	92.1	95.2	71.5	85.2	93.0
MS+DAS [29]	67.1	78.11	86.4	85.7	91.6	95.3	78.2	90.3	96.0
HIST [28]	69.7	80.0	87.3	87.4	92.5	95.4	79.6	91.0	96.2
DFML-PA [50]	69.3	-	-	88.4	-	-	-	-	-
HSE-M [56]	67.6	78.0	85.8	82.0	88.9	93.3	-	-	-
PA+niV [23]	69.5	80.0	-	86.4	92.0	-	79.2	90.4	-
Potential Field (Ours)	**71.5± 0.3**	**81.2±0.2**	**88.3±0.2**	**90.1±0.2**	**93.9±0.1**	**96.3±0.1**	**80.6 ± 0.3**	**91.8 ± 0.1**	**96.4 ± 0.1**
DINO (384 dim)									
DINO [5]	70.8	81.1	88.8	42.9	53.9	64.2	63.4	78.1	88.3
Hyp [12]	80.9	87.6	92.4	89.2	94.1	96.7	85.1	94.4	97.8
HIER [22]	81.1	88.2	93.3	91.3	95.2	97.1	85.7	94.6	97.8
Potential Field (Ours)	**83.1 ± 0.3**	**89.3 ± 0.2**	**94.2 ± 0.1**	**94.7 ± 0.1**	**96.5 ± 0.1**	**97.8 ± 0.1**	**86.5 ± 0.3**	**95.1 ± 0.3**	**98.0 ± 0.2**
ViT (384 dim)									
ViT-S [11]	83.1	90.4	94.4	47.8	60.2	72.2	62.1	77.7	89.0
Hyp [12]	85.6	91.4	94.8	86.5	92.1	95.3	85.9	94.9	98.1
HIER [22]	85.7	91.3	94.4	88.3	93.2	96.1	86.1	95.0	98.0
Potential Field (Ours)	**87.8 ± 0.2**	**92.6 ± 0.2**	**95.7 ± 0.1**	**91.5 ± 0.3**	**95.2 ± 0.2**	**97.4 ± 0.1**	**88.2 ± 0.1**	**95.7 ± 0.1**	**98.6 ± 0.1**

Table 1. Comparison of the Recall@K (%) achieved by our method on the CUB-200-2011, Cars-196 and SOP datasets with state-of-the-art baselines under standard settings. The table reports the average performance and standard deviations of our method over 5 runs.

	CUB-200-2011		**Cars-196**	
Methods	R@1	R@2	R@1	R@2
Triplet[52]	55.1	68.7	67.5	77.9
MS [51]	58.9	71.8	70.4	79.8
PNCA[34]	60.1	74.7	74.3	82.4
PA [21]	60.7	75.1	76.9	83.1
HIST[27]	59.7	74.6	72.9	81.8
Ours	**66.7 ± 0.6**	**76.9± 0.3**	**84.5± 0.5**	**88.6± 0.3**

Table 2. A comparison of Recall@K (%) achieved by our method with SOTA baselines on CUB-200-2011 and Cars-196 datasets with label noise when using a Resnet-50 backbone (512-dim). Performance and std. deviations reported are over 5 runs.

Method	**Average W_2**
S.Triplet [36]	0.24 ± 0.04
PNCA [21]	0.36 ± 0.02
PA [21]	0.38 ± 0.03
CPML (Sec. 3.4)	0.26 ± 0.03
Ours	**0.16 ± 0.03**

Table 3. Average Wasserstein distance between proxies and the subset of data closest to them for different methods on CUB-200-2011 dataset. Mean and std. deviations are over 5 runs.

data represented by/closest to them [2] for different proxy-based methods. We train networks on the CUB-200-2011 using the settings in Section 4.1 and use the *same number*

[2] For methods using a single proxy per class, this equals the distance between a proxy and the closest embedding belonging to its class

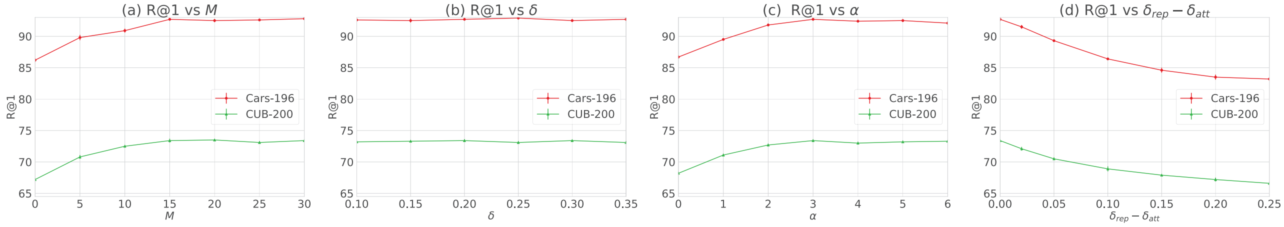

Figure 3. Variation in Recall@1 with M, δ, α and $\delta_{rep} - \delta_{att}$ on the Cars-196 and CUB-200-2011 datasets. Error bars represent std. deviations over 5 runs.

of proxies for all methods utilizing multiple proxies. As seen in Table 3, our method shows 33.3% lower W_2 distance between the proxies and the subset of data than the second best SoftTriplet [36] loss, empirically validating our discussion in Sec. 3.4.2.

4.5. Ablation studies

In all experiments, we use the parameters mentioned in Section 4.1 with a ResNet-50 backbone on both the Cars-196 and CUB-200-2011 datasets.

Effect of proxies and their number: Proxies augment the potential field, and the number of proxies per class M decides the granularity at which the proxies model the distribution of sample embeddings. We study their effect on performance by varying M between $[0, 30]$. From the results plotted in Figure 3(a) we observe that (1) Using proxies to augment the field boosts performance (vs using no-proxies) and (2) Network performance remains stable for a wide range of M, decreasing slightly for small values of M (still outperforming most other proxy-based methods). This behavior is expected as using potential field created using a very small M would not be able to effectively model underlying sample distribution, leading to worse supervision. We also observe that using a larger number of proxies per class does not significantly affect training times, as bulk of computation is dominated by the network. For further details, please refer to Supp. Sec. 5.

Effect of varying δ **:** We conduct an empirical study to determine the effect of varying the radius parameter δ on performance by varying it over $[0.1, 0.35]$. As seen in Figure 3(b), we observe that performance remains relatively stable for a wide range of δ values, demonstrating our method's robustness to it.

Effect of potential decay α: The decay parameter α determines the rate at which the potential field of an embedding decays with distance. We study its effect by varying it over $[0, 6]$. As seen in Figure 3(c), performance is worse when the potential field does not decay ($\alpha = 0$). A field that decays strongly $\alpha \in [3, 6]$ seems to perform better than one with a much milder decay ($\alpha \in [1, 2]$). The likely reason for this trend is that when the potential field has no decay /decays mildly, the force on an embedding would still be dominated by a large number of distant sample embeddings rather than those in its vicinity.

Restricting repulsion to a distance of δ:Unlike most proxy-based methods, our design of potential field Ψ_{rep} limits the range of repulsion to within δ distance from embeddings or proxies. We examine extending this range of repulsion beyond the δ radius by having $\delta_{rep} > \delta_{att}$ where δ_{att} and δ_{rep} represent the values of δ used for attraction (Eq.1) and repulsion (Eq.2) fields. Fixing $\delta_{att} = 0.2$, we increase δ_{rep} to study the effect. As shown in Figure 3(d), his reduces performance, validating our design. This is due to individual potentials repelling dissimilar points to a distance of at least δ_{rep}, greater than the distance δ_{att} within which all points of the same class are attracted to. Using $\delta_{rep} > \delta_{att}$ enforces excessive separation between dissimilar examples, compressing intra-class features more than needed.

5. Limitations

PFML relies on proxies to augment potential fields in each batch, hence sharing the main limitation of proxy-based methods: with a large number of classes, the computational cost during training increases due to linear scaling of proxies with class count. Despite this, proxy-based methods have been widely used in applications with a large number of classes like face recognition (e.g. [10] on 500,000 classes), likely because (1) proxies only affect training, not inference complexity and (2) The parameters/compute complexity added by the proxies is relatively low for the performance gains they offer, especially when such compute complexity is compared to the forward and backward passes of the neural network that consume bulk of compute in DML.

6. Conclusion

PFML introduces a novel paradigm to capture interactions between all samples through a continuous potential field, where strength of sample influence decreases with distance. This reversal of the common model of increasing influence with distance improves robustness to label noise and better aligns proxy distributions with sample embeddings. The overall design of our method is validated by its performance on standard image retrieval benchmarks, where it outperforms state-of-the-art methods both in the standard noise-free setting and in the presence of label noise unavoidable in real-world datasets.

References

[1] Narendra Ahuja. A transform for multiscale image segmentation by integrated edge and region detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18 (12):1211–1235, 1996. 2

[2] N. Ahuja and Jen-Hui Chuang. Shape representation using a generalized potential field model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):169–176, 1997. 2

[3] Shubhang Bhatnagar and Narendra Ahuja. Piecewise-linear manifolds for deep metric learning. In *Conference on Parsimony and Learning (Proceedings Track)*, 2023. 1

[4] Biagio Brattoli, Joseph Tighe, Fedor Zhdanov, Pietro Perona, and Krzysztof Chalupka. Rethinking zero-shot video classification: End-to-end training for realistic applications. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 4612–4622. Computer Vision Foundation / IEEE, 2020. 1

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 6, 7

[6] Jun Cen, Peng Yun, Junhao Cai, Michael Yu Wang, and Ming Liu. Deep metric learning for open world semantic segmentation. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15313–15322, 2021. 1

[7] Weihua Chen, Xiaotang Chen, Jianguo Zhang, and Kaiqi Huang. Beyond triplet loss: A deep quadruplet network for person re-identification. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 1320–1329. IEEE Computer Society, 2017. 1

[8] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, pages 539–546. IEEE, 2005. 1, 2, 5

[9] Jen-Hui Chuang and N. Ahuja. An analytically tractable potential field model of free space and its application in obstacle avoidance. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 28(5):729–736, 1998. 2

[10] J. Deng, J. Guo, N. Xue, and S. Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4685–4694, 2019. 1, 2, 8

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 6, 7

[12] Aleksandr Ermolov, Leyla Mirvakhabova, Valentin Khrulkov, Nicu Sebe, and Ivan Oseledets. Hyperbolic vision transformers: Combining improvements in metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7409–7419, 2022. 7

[13] Weifeng Ge, Weilin Huang, Dengke Dong, and Matthew R. Scott. Deep metric learning with hierarchical triplet loss. In *Computer Vision – ECCV 2018*, pages 272–288, Cham, 2018. Springer International Publishing. 7

[14] Matheus Gutoski, André Eugênio Lazzaretti, and Heitor Silvério Lopes. Deep metric learning for open-set human action recognition in videos. *Neural Comput. Appl.*, 33(4): 1207–1220, 2021. 1

[15] Ben Harwood, Vijay Kumar BG, Gustavo Carneiro, Ian Reid, and Tom Drummond. Smart mining for deep metric learning. In *Proceedings of the IEEE international conference on computer vision*, pages 2821–2829, 2017. 1, 2, 6

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[17] J. Hu, J. Lu, and Y. Tan. Discriminative deep metric learning for face verification in the wild. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 1

[18] Y.K. Hwang and N. Ahuja. Path planning using a potential field representation. In *Proceedings. 1988 IEEE International Conference on Robotics and Automation*, pages 648–649 vol.1, 1988. 2

[19] Y.K. Hwang and N. Ahuja. A potential field approach to path planning. *IEEE Transactions on Robotics and Automation*, 8 (1):23–32, 1992. 1, 2

[20] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, pages 500–505, 1985. 1, 2

[21] Sungyeon Kim, Dongwon Kim, Minsu Cho, and Suha Kwak. Proxy anchor loss for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2, 5, 6, 7

[22] Sungyeon Kim, Boseung Jeong, and Suha Kwak. Hier: Metric learning beyond class labels via hierarchical regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19903–19912, 2023. 6, 7

[23] M. Kirchhof, K. Roth, Z. Akata, and E. Kasneci. A non-isotropic probabilistic take on proxy-based deep metric learning. In *Computer Vision - ECCV 2022 - 17th European Conference, Proceedings, Part XXVI*, pages 435–454. Springer, 2022. 2, 7

[24] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3D object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013. 6

[25] Elad Levi, Tete Xiao, Xiaolong Wang, and Trevor Darrell. Rethinking preventing class-collapsing in metric learning with margin-based losses. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10316–10325, 2021. 1, 2

[26] Christopher Liao, Theodoros Tsiligkaridis, and Brian Kulis. Supervised metric learning to rank for retrieval via contextual similarity optimization. In *International Conference on Machine Learning*, pages 20906–20938. PMLR, 2023. 1, 5

[27] Jongin Lim, Sangdoo Yun, Seulki Park, and Jin Young Choi. Hypergraph-induced semantic tuplet loss for deep metric learning code. https://github.com/ljin0429/HIST, 2022. 7

[28] Jongin Lim, Sangdoo Yun, Seulki Park, and Jin Young Choi. Hypergraph-induced semantic tuplet loss for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 212–222, 2022. 6, 7

[29] Lizhao Liu, Shangxin Huang, Zhuangwei Zhuang, Ran Yang, Mingkui Tan, and Yaowei Wang. Das: Densely-anchored sampling for deep metric learning. In *European Conference on Computer Vision*, pages 399–417. Springer, 2022. 6, 7

[30] Ruofan Liu, Yun Lin, XIANGLIN YANG, and Jin Song Dong. Debugging and explaining metric learning approaches: An influence function based perspective. In *Advances in Neural Information Processing Systems*, 2022. 5

[31] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1

[32] Benjamin J. Meyer and Tom Drummond. The importance of metric learning for robotic vision: Open set recognition and active learning. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 2924–2931, 2019. 1

[33] Timo Milbich, Karsten Roth, Homanga Bharadhwaj, Samarth Sinha, Yoshua Bengio, Björn Ommer, and Joseph Paul Cohen. Diva: Diverse visual feature aggregation for deep metric learning. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII*, page 590–607, Berlin, Heidelberg, 2020. Springer-Verlag. 7

[34] Yair Movshovitz-Attias, Alexander Toshev, Thomas K Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. In *Proceedings of the IEEE international conference on computer vision*, pages 360–368, 2017. 1, 2, 5, 6, 7

[35] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. A metric learning reality check. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16*, pages 681–699. Springer, 2020. 6

[36] Qi Qian, Lei Shang, Baigui Sun, Juhua Hu, Hao Li, and Rong Jin. Softtriple loss: Deep metric learning without triplet sampling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 2, 5, 6, 7, 8

[37] Samyak Rawlekar, Shubhang Bhatnagar, Vishnuvardhan Pogunulu Srinivasulu, and Narendra Ahuja. Improving multi-label recognition using class co-occurrence probabilities. In *International Conference on Pattern Recognition*, pages 424–439. Springer, 2025. 1

[38] Karsten Roth, Timo Milbich, Samarth Sinha, Prateek Gupta, Björn Ommer, and Joseph Paul Cohen. Revisiting training strategies and generalization performance in deep metric learning. In *Proceedings of the 37th International Conference on Machine Learning*. JMLR.org, 2020. 1

[39] Karsten Roth, Oriol Vinyals, and Zeynep Akata. Non-isotropy regularization for proxy-based deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7420–7430, 2022. 1, 6

[40] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015. 1, 2

[41] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017. 1

[42] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2016. 1, 2

[43] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 6

[44] Yifan Sun, Changmao Cheng, Yuhan Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei. Circle loss: A unified perspective of pair similarity optimization. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6397–6406, 2020. 7

[45] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H. S. Torr, and Timothy M. Hospedales. Learning to compare: Relation network for few-shot learning. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2017. 1

[46] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 6

[47] Mark Tabb and Narendra Ahuja. Multiscale image segmentation by integrated edge and region detection. *IEEE transactions on image processing*, 6(5):642–655, 1997. 2

[48] Eu Wern Teh, Terrance DeVries, and Graham W. Taylor. Proxynca++: Revisiting and revitalizing proxy neighborhood component analysis. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXIV*, pages 448–464. Springer, 2020. 2, 6, 7

[49] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 6

[50] Chengkun Wang, Wenzhao Zheng, Junlong Li, Jie Zhou, and Jiwen Lu. Deep factorized metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7672–7682, 2023. 7

[51] Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R. Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 6, 7

[52] Kilian Q Weinberger, John Blitzer, and Lawrence Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in Neural Information Processing Systems*. MIT Press, 2005. 1, 2, 7

[53] Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krahenbuhl. Sampling matters in deep embedding learning. In *Proceedings of the IEEE international conference on computer vision*, pages 2840–2848, 2017. 1, 2, 6

[54] Tong Xiao, Shuang Li, Bochao Wang, Liang Lin, and Xiaogang Wang. Joint detection and identification feature learning for person search. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3415–3424, 2017. 1

[55] Hong Xuan, Abby Stylianou, and Robert Pless. Improved embeddings with easy positive triplet mining. *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 2463–2471, 2019. 2, 7

[56] Bailin Yang, Haoqiang Sun, Frederick WB Li, Zheng Chen, Jianlu Cai, and Chao Song. Hse: Hybrid species embedding for deep metric learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11047–11057, 2023. 7

[57] Yuhui Yuan, Kuiyuan Yang, and Chao Zhang. Hard-aware deeply cascaded embedding. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 2

[58] Andrew Zhai and Hao-Yu Wu. Classification is a strong baseline for deep metric learning. In *British Machine Vision Conference*, 2018. 7

[59] Wenzhao Zheng, Chengkun Wang, Jiwen Lu, and Jie Zhou. Deep compositional metric learning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9316–9325, 2021. 7

[60] Wenzhao Zheng, Borui Zhang, Jiwen Lu, and Jie Zhou. Deep relational metric learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12065–12074, 2021. 7

[61] Yuehua Zhu, Muli Yang, Cheng Deng, and Wei Liu. Fewer is more: A deep graph metric learning perspective using fewer proxies. In *Advances in Neural Information Processing Systems*, pages 17792–17803. Curran Associates, Inc., 2020. 6, 7