Deep Metric Learning with Chance Constraints

Yeti Z. Gürbüz[†,1]   Oğul Can[1, 2]   A. Aydın Alatan[3]

[1]MetaDialog   [2]Cerebrate AI   [3]OGAM and METU

Abstract

Deep metric learning (DML) aims to minimize empirical expected loss of the pairwise intra-/inter- class proximity violations in the embedding space. We relate DML to feasibility problem of finite chance constraints. We show that minimizer of proxy-based DML satisfies certain chance constraints, and that the worst case generalization performance of the proxy-based methods can be characterized by the radius of the smallest ball around a class proxy to cover the entire domain of the corresponding class samples, suggesting multiple proxies per class helps performance. To provide a scalable algorithm as well as exploiting more proxies, we consider the chance constraints implied by the minimizers of proxy-based DML instances and reformulate DML as finding a feasible point in intersection of such constraints, resulting in a problem to be approximately solved by iterative projections. Simply put, we repeatedly train a regularized proxy-based loss and re-initialize the proxies with the embeddings of the deliberately selected new samples. We applied our method with 4 well-accepted DML losses and show the effectiveness with extensive evaluations on 4 popular DML benchmarks. Code is available at: https://github.com/yetigurbuz/ccp-dml

1. Introduction

Deep metric learning (DML) poses distance metric problem as learning the parameters of an embedding function so that the semantically similar samples are embedded to the small vicinity in the representation space as the dissimilar ones are placed relatively apart in the Euclidean sense. The typical embedding function is implemented as convolutional neural networks (CNN) for visual tasks and the parameters are learned through minimizing the empirical expected loss with possibly deliberately selected mini-batch gradient updates [36, 44]. The loss terms in the empirical loss penalize violations of the desired intra- and inter-class proximity constrains. Large-scale problems (in terms of $\#classes$) suffer from the noisy estimation of the expected loss with mini-
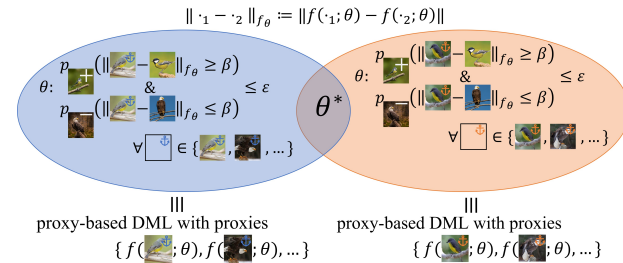

Figure 1. Simple illustration of our chance constrained DML formulation over two sets where each set's elements are the embedding function parameters θ that yield an embedding space in which the distances to anchor samples assess the semantic dissimilarity with high probability (*i.e.*, satisfying chance constraints). We consider the parameters of the desired embedding function $f(\cdot; \theta^*)$ to lie in the intersection of such sets. We show that to each such set, there corresponds a proxy-based DML solution. Hence, we solve DML as a set intersection problem via solving multiple proxy-based DML problems.

batches [36, 48, 57]. Recently, augmenting the mini-batches with virtual embeddings called *proxies* is shown to better approximate empirical loss in large-scale problems [36, 57] owing to pseudo-global consideration of the dataset during loss computation. These advances raise a critical question: *"How does increasing proxies help?"* which is partially addressed empirically with the methods exploiting multiple proxies per class [40, 57, 72].

Characterizing generalization performance of proxy-based DML can be a decisive step towards theoretically addressing that question. To this end, we approach DML differently by posing it as a feasibility problem. In particular, we consider a chance constraint [3] for the desired embedding function and relate it to the typical expected loss of DML. Using such a relation, we provide an upper bound to the generalization error of proxy-based DML. Aligned with the literature, the form of the bound suggests possible room for the improvement on the generalization performance if more and diverse proxies are considered per class. However, straightforward increase of the proxies may not help; since, $i)$ proxies of the same class tend to merge [40] and $ii)$ memory is prohibitive to arbitrarily increase the proxies.

[†]Affiliated with OGAM-METU during the research.

To alleviate these limitations, we relate the minimizer of the proxy-based DML to a feasible point of some chance constraints, and reformulate DML as finding a point at the intersection of the sets that the proxies imply. We provide a scalable algorithm using iterative projections to the individual sets to solve the problem. Each projection corresponds to a regularized proxy-based DML. Hence, we inherently increase the number of diverse proxies included in the problem. We empirically study the implications of our formulation and show its effectiveness by applying our method on 4 DML losses and testing on 4 DML benchmarks. Results show consistent and state-of-the-art (SOTA) performance in improving the baselines.

2. Related Work

We discuss the works which are most related to ours. Briefly, our contributions include that i) we reformulate DML as a chance constrained feasibility problem, ii) we characterize generalization of proxy-based DML by expanding on the discussions of the works studying generalization bounds, iii) we write the feasibility problem, *i.e.*, DML, as a set intersection problem to be solved by iterative projections using proxy-based DML, and iv) we effectively utilize arbitrary number of proxies per class.

DML. Primary momentum in DML includes i) tailoring the loss terms [36] to impose the desired intra- and inter-class proximity constraints in the representation space, ii) pair mining [44, and the references therein] to increase diversity in the loss computation or to reduce noise [32], and iii) synthesizing informative samples with generative models [8, 31, 67, 68] or via interpolation [11, 25, 54]. To improve embedding quality, detaching class-discriminative and class-shared features [14, 31, 42], intra-batch feature aggregation [30, 49], ranking surrogates [39], and further regularization terms [19, 23, 45, 65] are utilized. Going beyond of a single model and loss, ensemble [22, 47, 62, 69, 70] and multi-task based approaches [9, 34, 43] are also used. Different to them, we approach DML from a unique perspective, redefining it as a set intersection problem with chance constraints.

Ranking losses in DML. Typical DML objective enforces distance ranking constraints among the samples in the embedding space via hinge losses penalizing ranking violations. The contrastive [15, 17], triplet [48, 58], and generalized contrastive with margin [59] losses are the simplest forms of the pairwise distance ranking based losses. Proceeding approaches utilize smoothed versions of these losses by replacing hinge loss with log-sum-exp [37, 56] or soft-max [51, 64] expressions, which inherently employ ranking among more samples via soft-batch-mining. Until very recently, log-sum-exp based ranking loss has been revamped with Bayesian perspective [20]. Likewise, we rediscover contrastive loss as a surrogate loss for our chance constrained DML formulation. Different to existing work,

our formulation gives algorithmic implications to solve the DML as a set intersection problem which indeed can be solved efficiently with proxy-based DML.

Proxy-based DML. Proxy-based methods consider augmenting the mini-batch with more samples for less noisy estimate of the expected loss and circumvent the costly embedding computation to include more samples in the mini-batches. Proxies typically are vectors representing embeddings of the class centers [6, 21, 35, 53] and are trained along with the embedding function parameters. Non-trainable proxies are also exploited in [26, 57] to gradually augment mini-batch with previously computed embeddings. In proxy-based DML, the pairwise distances are computed between the proxies and the mini-batch samples. Thus, pseudo-global dataset geometry is considered during loss computation. To better represent global geometry, multiple proxies per class are considered in [40, 41, 72] and a hierarchical structure is imposed to proxies in [63] where the former two [40, 41] build on improving R@1 (immediate neighbourhood) by fine-grained clustering of class samples to overlook intraclass variances. In our analysis, we also align with increasing the proxies. Our work differs in that i) we build on reducing the probability of proximity violations (*i.e.*, improving MAP@R) and ii) we progressively increase the proxies by relating the proxy-based DML instances.

DML as constrained optimization. Pioneer metric learning approaches [58, 60] consider sample-driven proximity constraints to formulate the problem and exploit alternating projections to perform projected gradient ascent. Recently, sample-driven constraints are reconsidered in [5] for a reformulation of DML as a set intersection. Sharing the set intersection concept, we approach the problem with a different perspective using chance constraints rather than sample-driven constraints, which enables us to formally develop a method that does not suffer from the poor scalability of exploiting class representatives unlike the method proposed in [5] does. What makes our method unique is the theoretically sound way we connect the proxy-based DML and set intersection concepts to arbitrarily increase the number of class representatives exploited in the problem.

Characterizing generalization bounds. Notion of robustness in learning algorithms is studied in [61] and generalization error bounds of several techniques are derived accordingly. This study is extended to metric learning setting in [2]. These works study the deviation between the expected loss and the empirical loss over the whole dataset. Differently in [50], deviation between two empirical losses, *core-set loss*, is studied to characterize generalization loss when a subset of the training data is exploited. Generalization bound for metric learning is further studied in [7, 29] to analyze and suggest training strategies. Our work expands on the theories in the aforementioned works to characterize and improve generalization bound for proxy-based DML.

3. Notation and Problem Definition

In typical DML, we consider the set $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ with elements $z = (x \in \mathcal{X}, y \in \mathcal{Y})$ where $\mathcal{X}$ is a compact space and $\mathcal{Y} = \{1, \dots, C\}$ is a finite label set. We will use x (or y) to denote data (or label) component of z. We have $p_{\mathcal{Z}}$, an unknown probability distribution over $\mathcal{Z}$. Indicator of the two samples, z_i and z_j, belonging to the same class is denoted as $y_{ij} \in \{-1, 1\}$ where $y_{ij} = 1$ if $y_i = y_j$.

We are interested in finding the parameters θ of an embedding function $f(\cdot; \theta) : \mathcal{X} \to I\!\!R^D$ so that the parametric distance, $\|x_i - x_j\|_{f_\theta} := \|f(x_i; \theta) - f(x_j; \theta)\|_2$, is small only whenever $y_i = y_j$. For any pair $(z_i, z_j) \sim p_{\mathcal{Z}}$ and embedding function $f(\cdot; \theta)$, we associate a loss $\ell(z_i, z_j; \theta)$ penalizing proximity violations in the embedding image. We omit f dependency in the ℓ notation for simplicity. We are to consider minimization of the expected loss:

$$\theta^* = \arg\min_\theta \mathbb{E}_{z_i, z_j}[\ell(z_i, z_j; \theta)] \qquad (3.1)$$

In practice, we are given a dataset of n instances sampled *i.i.d.* from $\mathcal{Z}$ as $\{z_i\}_{i \in [n]} \sim p_{\mathcal{Z}}$ where $[n] = \{1, \dots, n\}$, and an algorithm $\mathcal{A}_{s_1 \times s_2}$ which outputs parameters θ minimizing empirical expected loss with a training error $e(\mathcal{A}_{s_1 \times s_2})$ for a given set $\{(z_i, z_j)\}_{i,j \in s_1 \times s_2}$ of pairs from the dataset, where $s_k = \{s_k(l) \in [n]\}_{l \in [n_k]} \subseteq [n]$ is a pool of indexes chosen from the dataset, $[n]$. In other words,

$$\mathcal{A}_{s_1 \times s_2} := \arg\min_\theta \frac{1}{|s_1||s_2|} \sum_{i \in s_1} \sum_{j \in s_2} \ell(z_i, z_j; \theta), \quad (3.2)$$

and we formally define DML as $\mathcal{A}_{[n] \times [n]}$, *i.e.*, minimizing empirical expected loss with all possible pairs. We consider improving the generalization error of $\mathcal{A}_{s_1 \times s_2}$ which is:

$$\mathcal{L}(\mathcal{A}_{s_1 \times s_2}) = \mathbb{E}_{z_i, z_j}[\ell(z_i, z_j; \mathcal{A}_{s_1 \times s_2})]. \qquad (3.3)$$

4. Method

We will iteratively solve multiple proxy-based DML problems. At each problem, we re-initialize the class proxies by samples from the dataset. We relate the problems by regularizing the learned parameters to be in the close vicinity of the previous ones. In the following sections, we provide theoretical foundation behind the motivation of our method. We defer all the upcoming proofs to appendix [13].

We start with reformulating DML with a chance constraint. We will introduce two propositions that allow us to decompose the chance constraint into finite chance constraints. We also show minimizer of proxy-based DML satisfies some chance constraints. Hence, we link DML to finding a point in the intersection of finite sets, which we solve using iterative projections that correspond to regularized proxy-based DML problem instances.

In the formulations throughout the paper, we rely on Lipschitz continuity of the loss function for which we refer to Lemma 4.4. Our approach focuses on enhancing the
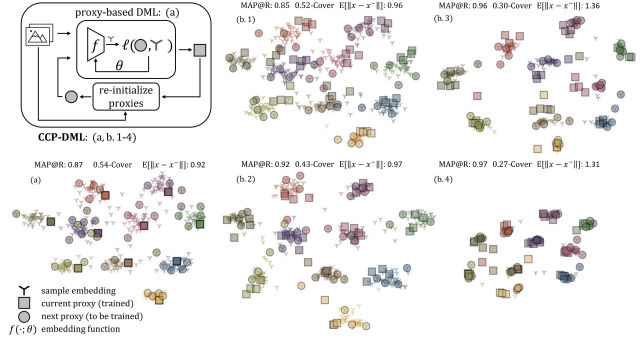

Figure 2. Illustration of our method (CCP, Algorithm 1) and the geometry of the embedding space in MNIST dataset: Boxes represent the converged proxies, while circles represent the next proxies resulting from K-Center (Algorithm 2). (a) In proxy-based DML (before our method), proxies coalesce into one. (b) With CCP (through iterations 1-4), diverse proxies are obtained, resulting in a reduced covering radius. Magnified version is in suppl. [13].

generalization performance in the seen domain, with implications for the crucial goal of generalizing to unseen classes in DML. In DML models, the embedding vector is derived by globally averaging local CNN features, which act as visual words [12, 71]. By prioritizing improved generalization during training, we can transfer the semantic knowledge captured by these visual words to effectively represent samples from unseen classes. Furthermore, our empirical studies provide strong evidence supporting the effectiveness of the proposed formulations and their implications for DML.

4.1. Chance Constrained Formulation of DML

We consider the solution of the following chance constrained feasibility problem:

$$\min_\theta 0^\mathsf{T}\theta \ \text{ s.to } \ p_{z_i, z_j}(y_{ij}(\|x_i - x_j\|_{f_\theta} - \beta) \geqslant 0) \leqslant \varepsilon \ (4.1)$$

with some small ε. In essence, we want the probability of observing two samples of the same (different) class being apart (close) more than β in the embedding space being low. We write that probability as expected violation, $\mathbb{E}_{z_i, z_j}[\mathbb{1}(y_{ij}(\|x_i - x_j\|_{f_\theta} - \beta) \geqslant 0)]$ where $\mathbb{1}(\cdot)$ being indicator function, and bound it for $\beta \geqslant \alpha > 0$ as:

$$\begin{aligned} &p_{z_i, z_j}(y_{ij}(\|x_i - x_j\|_{f_\theta} - \beta) \geqslant 0) \\ &\qquad \leqslant \frac{1}{\alpha} \mathbb{E}_{z_i, z_j}[(y_{ij}(\|x_i - x_j\|_{f_\theta} - \beta) + \alpha)_+], \end{aligned} \qquad (4.2)$$

using Markov's inequality where $(u)_+ = \max\{0, u\}$. Note that to each value of the expectation $e(\theta)$ there corresponds an $\varepsilon = e(\theta)/\alpha$ which the chance constraint satisfies. Hence, we use the expectation as the surrogate of the penalty term for the chance constraint and can redefine the aforementioned feasibility problem as the expected loss minimization in (3.1) with $\ell(z_i, z_j; \theta) = (y_{ij}(\|x_i - x_j\|_{f_\theta} - \beta) + \alpha)_+$. In particular,

we end up with minimization of the expected *contrastive loss with positive margin* [59].

We now consider the relaxed feasibility problem in which we consider m chance constraints conditioned on given m samples $\mathcal{S}=\{z_i\}_{i\in[m]} \sim p_{\mathcal{Z}}$, say *anchor samples*. To be more precise, we want to find $\theta \in \mathcal{C}_{\mathcal{S}}$ with:

$$\mathcal{C}_{\mathcal{S}}=\{\theta \mid p_{z_j}(y_{ij}(\|x_i\text{-}x_j\|_{f_\theta}-\beta)\geqslant 0) \leqslant \varepsilon, \, \forall i\in[m]\}, \quad (4.3)$$

where $[m]$ indexes the samples in $\mathcal{S}$. Using expectation bounds as in (4.2), the unconstrained problem becomes:

$$\theta^* = \arg\min_\theta \frac{1}{m}\sum_{i\in[m]} \mathbb{E}_{z_j}[\ell(z_i,z_j;\theta)]. \quad (4.4)$$

We are particularly interested in the problem of the form in (4.4) owing to its relation to proxy-based methods to characterize their generalization. Prior to delving into such a relation, we first bound the deviation from the actual expectation in (3.1) when we solve the problem in (4.4) instead.

Proposition 4.1. *Given $\mathcal{S}=\{z_i\}_{i\in[m]} \overset{i.i.d.}{\sim} p_{\mathcal{Z}}$ such that $\forall k\in\mathcal{Y}$ $\{x_i|y_i=k\}$ is $\delta_{\mathcal{S}}$-cover[1] of $\mathcal{X}$, $\ell(z_i,z_j;\theta)$ is ζ-Lipschitz in x_i, x_j for all y_i, y_j and θ, and bounded by L; then with probability at least $1-\gamma$,*

$$\left| \mathbb{E}_{z_i,z_j}[\ell(z_i,z_j;\theta)] - \frac{1}{m}\sum_{i\in[m]} \mathbb{E}_{z_j}[\ell(z_i,z_j;\theta)]\right|$$
$$\leqslant \mathcal{O}(\zeta\,\delta_{\mathcal{S}}) + \mathcal{O}(L\sqrt{\log\tfrac{1}{\gamma}/m}).$$

Proposition 4.1 gives an upper bound which is controlled by the diversity of the anchor samples defining the relaxed problem. Theoretically, such a controlled bound allows DML to be formulated as a feasibility problem of finite sets for some accepted error tolerance. In practice, the best we can do is using all the samples in the dataset as the anchor samples when defining $\mathcal{C}_{\mathcal{S}}$ in (4.3). Granted that the minimization in (4.4) with the empirical loss boils down to the classical DML in (3.2), it has different stochastic optimization procedure. The relaxed problem suggests sampling batch of instances rather than pairs, which yields less noisy gradient estimates with the same batch budget.

4.2. Reducing Chance Constraints

The loss terms conditioned on anchor samples in (4.4) are computationally prohibitive in large-scale problems. Thus, we are interested in reducing the chance constraints, *i.e.*, anchor samples. To this end, proxy-based methods are quite related in that a proxy-based DML constitutes a superset of the feasible region of the primary DML problem in (4.1) as we will show shortly.

Proxy-based methods use parametric vectors $\{\rho_i\}_{i\in[C]}$ to represent embedding of the class centers and minimize the

pair losses with respect to those centers. Formally, given a dataset $\{z_i\}_{i\in[n]} \sim p_{\mathcal{Z}}$, proxy-based methods consider the following problem:

$$\min_{\theta,\rho} \frac{1}{n\,C} \sum_{i\in[C]} \sum_{j\in[n]} \hat{\ell}(\rho_i,z_j;\theta), \quad (4.5)$$

where $\hat{\ell}(\rho_i,z_j;\theta)$ is a loss term in which the pairwise distance is computed as $\|\rho_i - f(x_j;\theta)\|_2$. We can associate an algorithm $\mathcal{A}_{s\times[n]}$ defined in (3.2) to the minimizer of (4.5) with $e(\mathcal{A}_{s\times[n]})$ training error where $s = \{s(i)\in[n] \mid f(x_{s(i)};\mathcal{A}_{s\times[n]}) = \rho_i\}_{i\in[C]}$. In other words, to each proxy, we associate a dataset sample whose embedding matches that proxy, assuming such sample exists. Hence, the minimizer of the proxy-based methods can be reformulated as the following feasibility problem:

$$\min_\theta 0^{\mathsf{T}}\theta \text{ s. to } p_{z_j}(y_{ij}(\|x_i\text{-}x_j\|_{f_\theta}-\beta)\geqslant 0) \leqslant \varepsilon, \, \forall i\in s, \quad (4.6)$$

where s, as explained above, indexes C-many dataset samples corresponding to proxies, and $\varepsilon=\frac{1}{\alpha}\mathcal{L}(\mathcal{A}_{s\times[n]})$ from the expression in (4.2). $\mathcal{L}(\mathcal{A}_{s\times[n]})$ defined in (3.3) is shown to be bounded in [2], hence so is ε. Reformulation of proxy-based DML defines the feasibility problem in (4.3) with one sample per class.

We now consider more general case where we use m samples per class from the dataset $\{z_i\}_{i\in[n]}\sim p_{\mathcal{Z}}$ to define the feasibility problem. We have m-many disjoint 1-per-class sets $s = \cup_{k\in[m]}s_k$, where $s_k = \{s_k(i)\in[n] \mid y_{s_k(i)}=i\}_{i\in[C]}$ with $\cap_{k\in[m]}s_k = \emptyset$. We define the problem:

$$\min_{\theta\in\cap_k \mathcal{C}_{s_k}} 0^{\mathsf{T}}\theta \quad \text{where} \quad \mathcal{C}_{s_k} = \{\theta \mid \forall i \in s_k,$$
$$p_{z_j}(y_{ij}\|x_i - x_j\|_{f_\theta} - \beta) \geqslant 0) \leqslant \varepsilon\}. \quad (4.7)$$

Solving the problem by minimizing the empirical expectation bounds in (4.4), we end up with an algorithm $\mathcal{A}_{s\times[n]}$ in which we are minimizing expected loss over a subset of all possible pairs. We want to characterize the generalization performance of the algorithm $\mathcal{A}_{s\times[n]}$. We consider the following bound from [50] for the generalization error:

$$\mathbb{E}_{z_i,z_j}[\ell(z_i,z_j;\mathcal{A}_{s\times[n]})] \leqslant \left| \frac{1}{|s|\,n}\sum_{\substack{i,j\in s\times[n]}} \ell(z_i,z_j;\mathcal{A}_{s\times[n]})\right|_{(\mathcal{L}_1)}$$

$$+ \left| \mathbb{E}_{z_i,z_j}[\ell(z_i,z_j;\mathcal{A}_{s\times[n]})] - \frac{1}{n^2}\sum_{\substack{i,j\in[n]\times[n]}} \ell(z_i,z_j;\mathcal{A}_{s\times[n]})\right|_{(\mathcal{L}_2)}$$

$$+ \left| \frac{1}{n^2}\sum_{\substack{i,j\in[n]\times[n]}} \ell(z_i,z_j;\mathcal{A}_{s\times[n]}) - \frac{1}{|s|\,n}\sum_{\substack{i,j\in s\times[n]}} \ell(z_i,z_j;\mathcal{A}_{s\times[n]})\right|_{(\mathcal{L}_3)}$$
$$(4.8)$$

where the bound is controlled by $(\mathcal{L}_1)$ training loss (*i.e.*, $e(\mathcal{A}_{s\times[n]})$), $(\mathcal{L}_2)$ the deviation between expected loss and empirical loss over all possible pairs, and $(\mathcal{L}_3)$ the deviation between empirical loss over all possible pairs and empirical loss over the subset of pairs defining the algorithm $\mathcal{A}_{s\times[n]}$. It

---

[1] $\mathcal{S} \subset \mathcal{S}'$ is $\delta_{\mathcal{S}}$-cover of $\mathcal{S}'$ if $\forall z'\in\mathcal{S}', \exists z\in\mathcal{S} : \|z - z'\|_2 \leqslant \delta_{\mathcal{S}}$.

is widely observed that high capacity CNNs can reach very small training error. Moreover, $\mathcal{L}_2$ is proved to be bounded in [2] and is independent of $\mathcal{A}$. Thus, $\mathcal{L}_3$ characterizes the generalization performance of using the subset of pairs over exploiting all possible pairs.

Proposition 4.2. *Given $\{z_i\}_{i\in[n]} \overset{i.i.d.}{\sim} p_{\mathcal{Z}}$ and a set $s \subset [n]$. If $s = \cup_k s'_k$ with s'_k is the δ_s-cover of $\{i \in [n] \mid y_i = k\}$ (i.e., the samples in class k), $\ell(z_i, z_j; \theta)$ is ζ-Lipschitz in x_i, x_j for all y_i, y_j and θ, and bounded by L, $e(\mathcal{A}_{s\times[n]})$ training error; then with probability at least $1 - \gamma$ we have:*

$$\left| \tfrac{1}{n^2} \sum_{\substack{i,j\in[n]\times[n]}} \ell(z_i, z_j; \mathcal{A}_{s\times[n]}) - \tfrac{1}{|s|\, n} \sum_{\substack{i,j\in s\times[n]}} \ell(z_i, z_j; \mathcal{A}_{s\times[n]}) \right|$$

$$\leqslant \mathcal{O}(\zeta\, \delta_s) + \mathcal{O}(e(\mathcal{A}_{s\times[n]})) + \mathcal{O}(L\sqrt{\tfrac{\log 1/\gamma}{n}}).$$

Corollary 4.3. *Generalization of the proxy-based methods can be limited by the maximum of distances between the proxies and the corresponding class samples in the dataset.*

Proposition 4.2 implies that increasing the number of chance constraints with more anchor samples in the feasible point problem formulation of DML improves the generalization error bound as long as the included samples improve the covering radius of the dataset. In other words, including more anchor samples do not improve the bound unless the covering radius is decreased. Similarly, Corollary 4.3 informally suggests possible improvement on the generalization error bound of the proxy-based methods if we manage to introduce more proxies which are spread over the dataset once trained. In practice introducing more proxies generally does not help the performance since they eventually coalesce into a single point [40]. Besides, the computation resource limits the number of proxies to be included in the formulation. In the next section, we develop an approach to alleviate these problems.

4.3. Solving the Feasibility Problem

We now introduce our chance constrained programming (CCP) method, outlined in Algorithm 1, exploiting proxy-based training together with satisfying arbitrarily increased chance constraints. In short, we repeatedly solve a proxy-DML and improve the solution by re-initializing the proxies with the new samples reducing the covering radius.

We consider the problem in (4.7) as finding a point in the intersection of the sets. In particular, given dataset $\{z_i\}_{i\in[n]} \sim p_{\mathcal{Z}}$, we have m many 1-per-class sets $s_k = \{s_k(i) \in [n] \mid y_{s_k(i)} = i\}_{i\in[C]}$ to define the constraint set as $\mathcal{C}_s = \cap_{k\in[m]}\mathcal{C}_{s_k}$. If the sets were closed and convex, the problem would be solvable by iterative projection methods [1, 4]. Nevertheless, it is not uncommon to perform iterative projection methods to non-convex set intersection problems [38, 52]. Hence, we propose to solve the problem approximately by performing iterative projections onto the

---

Algorithm 1 CCP DML

---

initialize θ^* randomly, given $\{z_i\}_{i\in[n]} \sim p_{\mathcal{Z}}$ dataset
initialize ρ^* with random samples, set budget b
repeat
  $\rho \leftarrow GreedyKCenterProxy(\rho^*, b, f(\cdot; \theta^*))$
  repeat
    sample $s = \{j \in [n]\}_{j\in[m]} \sim [n]$ a batch
    $g_\theta \leftarrow \lambda\,(\theta^* - \theta) + \nabla_\theta \tfrac{1}{m|\rho|} \sum_{\rho\times s} \ell(\rho_i, z_j; \theta)$
    $g_\rho \leftarrow \nabla_\rho \tfrac{1}{m|\rho|} \sum_{\rho\times s} \ell(\rho_i, z_j; \theta)$
    $(\theta, \rho) \leftarrow ApplyGradient(\theta, \rho, g_\theta, g_\rho)$
  until convergence
  $\theta^* \leftarrow \theta, \rho^* \leftarrow \rho$
until convergence

---

feasible sets $\mathcal{C}_{s_k}$ defined by s_k. At each iteration k we solve the following projection problem given $\theta^{(k\text{-}1)}$:

$$\theta^{(k)} = \underset{\theta\in\mathcal{C}_{s_k}}{\arg\min} \; \tfrac{1}{2}\|\theta^{(k\text{-}1)} - \theta\|_2^2, \qquad (4.9)$$

where C_{s_k} is defined in (4.7). Using expectation bounds as the surrogate of the penalty terms for the chance constraints as we do in § 4.1, we have:

$$\theta^{(k)} = \underset{\theta}{\arg\min} \; \tfrac{\lambda}{2}\|\theta^{(k\text{-}1)} - \theta\|_2^2 + \tfrac{1}{C} \sum_{i\in[C]} \mathbb{E}_{z_j}[\ell(z_{s_k(i)}, z_j; \theta)],$$
$$(4.10)$$

where λ is a hyperparameter for the projection regularization. We can minimize the resultant loss by using batch stochastic gradient approaches. However, the batch should be augmented by C many anchor samples to compute the loss, which becomes prohibitive for large-scale problems. To alleviate costly embedding computation of C many samples, we propose to use proxies ρ_i in place of the embedding of the samples $z_{s_k(i)}$. Namely, at each iteration k, we initialize $\rho_i = f(z_{s_k(i)}; \theta^{(k\text{-}1)})$ and solve:

$$\theta^{(k)}, \rho^* = \underset{\theta,\rho}{\arg\min} \; \tfrac{\lambda}{2}\|\theta^{(k\text{-}1)} - \theta\|_2^2 + \tfrac{1}{C} \sum_{i\in[C]} \mathbb{E}_{z_j}[\ell(\rho_i, z_j; \theta)],$$
$$(4.11)$$

where the resultant problem we solve at each iteration corresponds to a proxy-based DML. Any pairwise distance based loss can replace $\ell(\cdot)$ with anchor samples being class proxies. *I.e.*, we repurpose existing objectives with a regularization term in an iterative manner. Although we set up the formulation using single proxy per class, extending it to accommodate multiple proxies is a straightforward process.

Theoretically, we should cycle through the sets until convergence to solve $\theta \in \cap_{k\in[m]}\mathcal{C}_{s_k}$. Thus, we must pick anchor samples for each set to initialize proxies. The updates of the proxies are not guaranteed to mimic the actual updates of the corresponding anchor samples. With that being said, we will still have a solution, as (4.6) suggests, to feasibility

of some chance constraints as long as the converged proxies ρ^* are diverse. We empirically observe that the proxies initialized with diverse samples converge to embedding of distinct samples (Fig. 2). Hence, on one hand, we have solutions to different constraint sets as long as we re-initialize the proxies with new samples and solve proxy-based DML problems. On the other hand, Proposition 4.2 implies that generalization is improved as long as we end up with converged proxies reducing the covering radius. Therefore, the theory suggests a set intersection mechanism to reduce the covering radius yet allows a greedy algorithm via iterative projections to select (*i.e.*, initialize) the next proxies on the fly instead of explicitly defining the sets we will iterate on. Such a result is useful especially for the cases where the dataset is stochastically extended with random data augmentations which obstruct explicit set forming.

Proxy selection. We can simply use random sampling for anchor samples to initialize proxies since we eventually observe informative samples reducing the covering radius through the iterations. We can as well explicitly mine samples that possibly help with reducing the covering radius. Thus, we also exploit clever selection of proxies as outlined in Algorithm 2. Given a budget b, we sample b many instances per class and compute their embeddings to form a pool. We then select the samples that reduce the covering radius most once added to proxy set. This selection is equivalent to K-Center problem as formulated in [50]. Such a selection of proxies helps converged proxies to be diverse. $b = 1$ reduces to random sampling. In both, we inherently increase the number of anchor samples defining the problem and hence reducing the covering radius.

Algorithm 2 Greedy K-Center Proxy

---

input: proxy set ρ, sampling budget b and $f(\cdot; \theta)$
repeat for each class c
    $s_c \leftarrow \{x_i \mid y_i = c\}_{i \in [b]}$, b-sample-per-class
    initialize $r_c \leftarrow \{\}$, $p \leftarrow f(s_c; \theta)$
    repeat
        $q \leftarrow \arg\max_{u \in p \backslash r_c} \min_{v \in \rho_c \cup r_c} \|u - v\|_2$
        $r_c \leftarrow \{q\} \cup r_c$
    until $|r_c| = |\rho_c|$
return $\cup_c r_c$

4.4. Implementation Details

Embedding function. For the embedding function $f(\cdot; \theta)$ we use ImageNet [46] pretrained CNNs with ReLU activation, max- and average-pooling. We exploit architectures until the output of the global average pooling layer. We add a fully connected layer (*i.e.*, linear transform) to the output of the global average pooling layer to obtain the embedding vectors. We state the following lemma to prove our loss is Lipschitz continuous.

Lemma 4.4. *Generalized contrastive loss defined as* $\ell(z_i, z_j; \theta) := (y_{ij}(\|x_i - x_j\|_{f_\theta} - \beta) + \alpha)_+$ *is* $\sqrt{2}\omega^L$-*Lipschitz in* x_i *and* x_j *for all* y_i, y_j, θ *for the embedding function* $f(\cdot; \theta)$ *being L-layer CNN (with ReLU, max-pool, average-pool) with a fully connected layer at the end, where* ω *is the maximum sum of the input weights per neuron.*

ω can be made arbitrarily small with weight regularization, which is commonly used [57]. SOTA methods widely use $\ell 2$ normalization on the embeddings. For normalization, we apply $\hat{v} = v/\|v\|_2$ if $\|v\|_2 \geqslant 1$ or identity otherwise (*i.e.*, $\hat{v} = v$ if $\|v\|_2 \leqslant 1$). Unlike $\ell 2$ normalization, such a transform is Lipschitz continuous, hence so are our loss.

Solving projections. Performing a projection defined in (4.11) involves a minimization problem. We monitor MAP@R validation accuracy and use early stopping patience of 3 to pass the next projection.

5. Experimental Work

5.1. Setup

We follow the suggestions of recent work [10, 36, 44] explicitly studying the fair evaluation strategies for DML in order to minimize the confounding of the factors other than our method. Specifically, we mostly follow the MLRC procedures proposed in [36] to provide fair and unbiased evaluation of our method as well as comparisons with the other methods. We offer detailed experimental setup information in the supplementary material [13, § 2.1] for reproducibility.
Backbone. BN-Inception [18] with 128D embedding.
Datasets. CUB [55], Cars [27], In-shop [33], SOP [37] with MLRC [36] data augmentation.
Training. Adam [24] optimizer with 10^{-5} learning rate, 10^{-4} weight decay, 32 batch size (4 per class), 4-fold: 4 models (1 for each $3/4$ train set partition).
Evaluation. Average performance (Separated-128D) with mean average precision (MAP@R) at R where R is defined for each query and is the total number of its true references.
Losses with CCP. *C1-CCP*: Contrastive [15], *C2-CCP*: Contrastive with positive margin [59], *MS-CCP*: Multi-similarity (MS) [56], *Triplet-CCP*: Triplet [48].
Compared methods and fairness. We compare our method against proxy-based SoftTriple [40], ProxyAnchor [21] and ProxyNCA++ [53] methods as well as XBM [57]. Our experiments cover wide range of the DML losses since ProxyAnchor is indeed proxy-based MS loss except for missing a margin term, similarly ProxyNCA is $\log \Sigma \exp$-approximation of proxy-based Triplet with hard negative mining, and for single proxy case SoftTriple $\equiv$ ProxyNCA.
CCP hyperparameters. We introduce 3 new hyperparameters to a typical DML: λ, #$proxy$ (proxy per class), b (pool size). We optimize λ-#$proxy$ with Bayesian search (details are in supplementary material [13, § 1.2]) and b-#$proxy$ with grid search (Fig. 5). Based on our analysis, we set $\lambda = 2 \cdot 10^{-4}$,

Table 1. Conventional evaluation with BN-Inception. Red: the best. Blue: the second best. Bold: previous SOTA.

Backbone →	BN-Inception-512D			
Dataset →	CUB	Cars196	SOP	In-shop
Method ↓	R@1	R@1	R@1	R@1
SoftTriple-L [40]	65.40	84.50	78.60	-
C1-XBM-L [57]	65.80	82.00	79.50	89.90
ProxyAnchor [21]	68.40	86.10	79.10	**91.50**
DiVA [34]	66.80	84.10	78.10	-
ProxyFewer [72]	66.60	85.50	78.00	-
PROFS [5]	66.00	86.30	78.70	-
Margin-S2SD [43]	**68.50**	**87.30**	79.30	-
HIST [30]	**69.70**	**87.40**	**79.60**	-
C1-CCP-L	67.74	83.74	79.86	90.98
C2-CCP-L	69.87	83.90	80.01	91.72
MS-CCP-L	69.09	86.01	79.75	91.84

Table 2. Conventional evaluation with ResNet50. Red: the best. Blue: the second best. Bold: previous SOTA. [†]*Results from LIBC [49].*

Backbone →	ResNet50			
Dataset →	CUB	Cars196	SOP	In-shop
Method ↓	R@1	R@1	R@1	R@1
C1-XBM[128] [57]	-	-	80.60	91.30
ProxyAnchor[512] [21]	69.70	87.70	80.00[†]	92.10[†]
DiVA[512] [34]	69.20	87.60	79.60	**-**
ProxyNCA++[512] [53]	66.30	85.40	80.20	88.60
PROFS[512] [5]	64.90	81.30	76.90	-
Margin-S2SD[512] [43]	69.00	89.50	81.20	-
LIBC[512] [49]	70.30	88.10	81.40	92.80
ProxyAnchor-DIML[128] [66]	66.46	86.13	79.22	-
MS+Metrix[512] [54]	71.40	89.60	81.00	92.20
HIST[512] [30]	71.40	89.60	81.40	-
HPL-PA[512] [63]	-	-	80.04	92.46
C1-CCP-L[512]	69.87	87.12	81.74	92.07
C2-CCP-L[512]	71.04	85.93	81.66	92.46
MS-CCP-L[512]	70.37	89.02	81.59	92.71

$\#proxy=8$, $b=12$ for CUB and Cars. For SOP and In-shop, we reduce $\#proxy=4$ and $b=7$ owing to relatively less number of samples per class in the dataset.

Conventional evaluation. We additionally follow the relatively old-fashioned *conventional* procedure [37] for the evaluation of our method. We use BN-Inception [18] and ResNet50 [16] backbones with 512D embeddings. We use global max pooling as well as global average pooling, likewise the recent approaches [21,53,54,57]. We use batch size of 128 for BN-Inception and 96 for ResNet50.

5.2. Results

MLRC. We present the tabulated MLRC evaluation results in in supplementary material [13, Tables 1 and 2] and summarize MAP@R rankings with 128D embeddings in Fig. 3. We use Method-S/L naming convention to denote memory size in XBM, and the proxy per class in SoftTriple
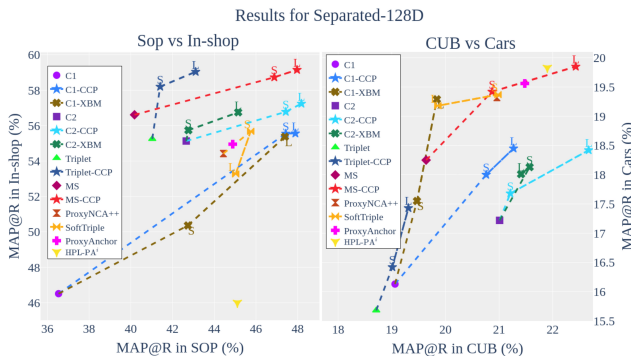

Figure 3. Summary of relative improvements for MLRC evaluation. [i]*In-shop result is not available for HPL-PA [63].*

and CCP where S denotes 1, and L denotes 4(10) for Soft-Triple and 4(8) for CCP in In-shop, SOP (CUB, Cars196). For fairness, we match XBM memory size and the number of proxies in CCP. We observe that CCP consistently outperforms the associated baseline methods on each dataset. Contrastive loss' compelling performance with CCP is important to support the implications of our formulation. Moreover, performance improvements on the losses which do not directly fit in our formulation show the broader applicability of our method to the pairwise distance based losses. Additionally, CCP framework outperforms not only the related proxy-based methods but also every single benchmarked approach in [36]. When compared with SoftTriple and XBM (*i.e.*, multiple proxy methods), CCP outperforms them by large margin especially in the cases where less number of proxies are used (*i.e.*, *method*-S comparisons in Fig. 3). We observe especially in large-scale datasets (SOP & In-shop) that even single proxy per class brings substantial performance improvement with CCP. Finally, outperforming hierarchical proxy-based loss [63] further supports CCP's superior embedding geometry.

Conventional. We provide R@1 results in Tables 1 and 2 for the comparison with SOTA. We observe that our method outperforms SOTA in most cases and performs on par with or slightly worse in a few. Predominantly, our method has superior performance on large-scale datasets, especially compared to PROFS [5] which suffers from the poor scalability of exploiting class representatives.

5.3. Ablations

We include the analyses for the implications of our formulation and the effects of the hyperparameters. We defer computational analysis to supplementary material [13, § 1.2].
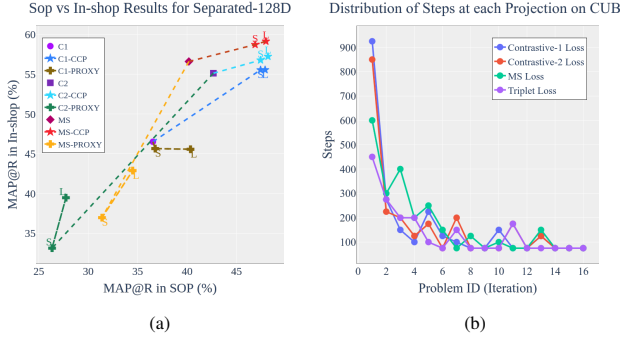
Figure 4. Impact of alternating proxies (a) and typical distribution of the steps per projection problem (b).
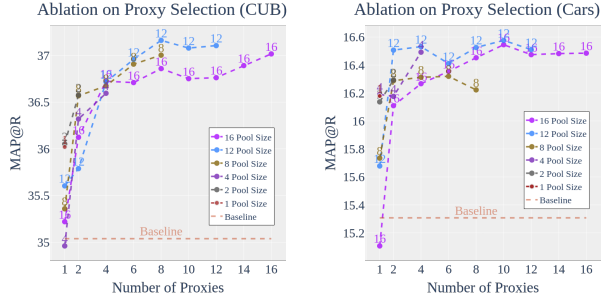

Figure 5. Analysis of the relation between the number of proxies and the pool size used for the proxy selection on CUB (left) and Cars (right) dataset with C2-CCP.

Proof of the concept. We evaluate our method using ResNet20V2 [16] on MNIST [28] dataset with 2D embeddings to show the implications of our formulation. In Fig. 2, we provide the distribution of the samples in the embedding space. We use 4 proxies per class and pool size $b=16$. We observe that when single proxy-based method is converged (Fig. 2-(a)), the class proxies collapse to a single point. Once we continue training with proposed approach (Fig. 2-(b)), the covering radius decreases, leading to performance improvement. We as well observe that diverse samples result in diverse proxies. In supplementary material [13, § 1.2], we extend this study and further provide the visualization of the validation data in CUB dataset to see how reducing the covering radius in training transfers to the test domain. We additionally experiment the case where we use samples instead of proxies. Though it is not practically applicable to large-scale problems, it is important to see whether our intuitions about alternating proxies in place of samples hold. We obtain 98.06% MAP@R performance with sample-based training against 97.21% MAP@R performance of proxy-based training. This empirical result supports our motivation on using the proxies in place of samples.

Effect of alternating problems. We provide results on MNIST in Fig. 2 to show the effect of solving alternating problems instead of single proxy-based DML. We additionally evaluate the baseline losses through solving only a single proxy-DML (*Loss*-Proxy) to show (Fig. 4-(a)) that our performance increase is not solely coming from augmentation of proxies in the problem. We clearly observe that alternating proxies helps performance as our formulation suggests. Moreover, we also provide a typical distribution of the steps per proxy-based projection problem in Fig. 4-(b) to show that we are not greedy on alternating the proxies just to provide more examples. We do have some relatively small steps, implying the selected proxies are not informative enough to change the geometry of the embedding space.

Effect of proxy selection. We analyze the relation between the number of proxies and the pool size used for the

proxy selection with C2-CCP. The related plots are in Fig. 5. We observe that both increasing the number of proxies and the pool size for proxy selection help performance. We interestingly see that for single proxy case, increasing the pool size gives no better results than random selection. Owing to our greedy proxy selection, we do consider the past geometry no earlier than single step. Thus, in the single proxy case, we are prone to oscillate between similar samples for proxy selection. On the other hand, selecting the samples that reduce the covering radius most brings better generalization over random selection. That said, random sampling in proxy selection (*i.e.*, pool size = #proxy) still works well since random sampling indeed can provide diversity in the samples as well. Such a result supports that the key to our method is alternating the proxies with new samples. As long as we re-initialize the proxies with new samples, we will have some diverse proxies through the iterations. To this end, we use Greedy K-Center to pick the samples in a clever way to reduce the covering radius as much as we can (*analogous to mining in batch construction*).

6. Conclusion

Bringing a different perspective to DML formulation, we formulate DML as a chance constrained optimization problem and theoretically show that a contrastive loss based DML objective is a surrogate for the chance constraints. We rigorously convert the initial problem formulation into another form enabling expressing DML as a set intersection problem. The theory suggests a set intersection mechanism yet allows a greedy algorithm via iterative projections. To this end, we also relate the solution of a proxy-based DML approach to one of the supersets to be intersected to obtain the desired solution. As a result, we formally develop a proxy-based method that inherently employs arbitrary number of proxies for better generalization, realizing the mechanism suggested by the theory with a simple, yet effective, algorithm. Supporting our claims, extensive evaluations on 4 DML benchmarks with 4 DML losses showed the effectiveness of our method.

References

[1] Heinz H Bauschke and Adrian S Lewis. Dykstras algorithm with bregman projections: A convergence proof. *Optimization*, 48(4):409–427, 2000. 5

[2] Aurélien Bellet and Amaury Habrard. Robustness and generalization for metric learning. *Neurocomputing*, 151:259–267, 2015. 2, 4, 5

[3] John R Birge and Francois Louveaux. *Introduction to stochastic programming*. Springer Science & Business Media, 2011. 1

[4] Lev M Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR computational mathematics and mathematical physics*, 7(3):200–217, 1967. 5

[5] Oğul Can, Yeti Z Gürbüz, and A Aydın Alatan. Deep metric learning with alternating projections onto feasible sets. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 1264–1268. IEEE, 2021. 2, 7

[6] Binghui Chen, Weihong Deng, and Haifeng Shen. Virtual class enhanced discriminative embedding learning. *Advances in Neural Information Processing Systems*, 31:1942–1952, 2018. 2

[7] Mingzhi Dong, Xiaochen Yang, Rui Zhu, Yujiang Wang, and Jing-Hao Xue. Generalization bound of gradient descent for non-convex metric learning. *Advances in Neural Information Processing Systems*, 33:9794–9805, 2020. 2

[8] Yueqi Duan, Wenzhao Zheng, Xudong Lin, Jiwen Lu, and Jie Zhou. Deep adversarial metric learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2780–2789, 2018. 2

[9] Mohammad K Ebrahimpour, Gang Qian, and Allison Beach. Multi-head deep metric learning using global and local representations. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3031–3040, 2022. 2

[10] Istvan Fehervari, Avinash Ravichandran, and Srikar Appalaraju. Unbiased evaluation of deep metric learning algorithms. *arXiv preprint arXiv:1911.12528*, 2019. 6

[11] Geonmo Gu, Byungsoo Ko, and Han-Gyu Kim. Proxy synthesis: Learning with synthetic classes for deep metric learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1460–1468, 2021. 2

[12] Yeti Z Gürbüz and A Aydın Alatan. A novel bovw mimicking end-to-end trainable cnn classification framework using optimal transport theory. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 3053–3057. IEEE, 2019. 3

[13] Yeti Z Gurbuz, Ogul Can, and A Aydın Alatan. Supplemental material for 'deep metric learning with chance constraint'. [https://github.com/yetigurbuz/ccp-dml], 2023. 3, 6, 7, 8

[14] Yeti Z Gurbuz, Ozan Sener, and A Aydın Alatan. Generalized sum pooling for metric learning. *arXiv preprint arXiv:2308.09228*, 2023. 2

[15] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006. 2, 6

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016. 7, 8

[17] Junlin Hu, Jiwen Lu, and Yap-Peng Tan. Discriminative deep metric learning for face verification in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1875–1882, 2014. 2

[18] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015. 6, 7

[19] Pierre Jacob, David Picard, Aymeric Histace, and Edouard Klein. Metric learning with horde: High-order regularizer for deep embeddings. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 2

[20] Shichao Kan, Zhiquan He, Yigang Cen, Yang Li, Vladimir Mladenovic, and Zhihai He. Contrastive bayesian analysis for deep metric learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 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*, pages 3238–3247, 2020. 2, 6, 7

[22] Wonsik Kim, Bhavya Goyal, Kunal Chawla, Jungmin Lee, and Keunjoo Kwon. Attention-based ensemble for deep metric learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 736–751, 2018. 2

[23] Yonghyun Kim and Wonpyo Park. Multi-level distance regularization for deep metric learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1827–1835, 2021. 2

[24] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[25] Byungsoo Ko and Geonmo Gu. Embedding expansion: Augmentation in embedding space for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7255–7264, 2020. 2

[26] Byungsoo Ko, Geonmo Gu, and Han-Gyu Kim. Learning with memory-based virtual classes for deep metric learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11792–11801, 2021. 2

[27] Andreas Krause and Daniel Golovin. Submodular function maximization. In *Tractability: Practical Approaches to Hard Problems*, pages 71–104. Cambridge University Press, 2014. 6

[28] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/, 2010. 8

[29] Yunwen Lei, Mingrui Liu, and Yiming Ying. Generalization guarantee of sgd for pairwise learning. *Advances in Neural Information Processing Systems*, 34, 2021. 2

[30] 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. 2, 7

[31] Xudong Lin, Yueqi Duan, Qiyuan Dong, Jiwen Lu, and Jie Zhou. Deep variational metric learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 689–704, 2018. 2

[32] Chang Liu, Han Yu, Boyang Li, Zhiqi Shen, Zhanning Gao, Peiran Ren, Xuansong Xie, Lizhen Cui, and Chunyan Miao. Noise-resistant deep metric learning with ranking-based instance selection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6811–6820, 2021. 2

[33] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1096–1104, 2016. 6

[34] 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 *European Conference on Computer Vision*, pages 590–607. Springer, 2020. 2, 7

[35] 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. 2

[36] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. A metric learning reality check. In *European Conference on Computer Vision*, pages 681–699. Springer, 2020. 1, 2, 6, 7

[37] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2, 6, 7

[38] CH Pang. Nonconvex set intersection problems: From projection methods to the newton method for super-regular sets. *arXiv preprint arXiv:1506.08246*, 2015. 5

[39] Yash Patel, Giorgos Tolias, and Jiří Matas. Recall@ k surrogate loss with large batches and similarity mixup. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7502–7511, 2022. 2

[40] Qi Qian, Lei Shang, Baigui Sun, Juhua Hu, Hao Li, and Rong Jin. Softtriple loss: Deep metric learning without triplet sampling. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 1, 2, 5, 6, 7

[41] Oren Rippel, Manohar Paluri, Piotr Dollar, and Lubomir Bourdev. Metric learning with adaptive density discrimination. *International Conference on Learning Representations*, 2016. 2

[42] Karsten Roth, Biagio Brattoli, and Bjorn Ommer. Mic: Mining interclass characteristics for improved metric learning. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 2

[43] Karsten Roth, Timo Milbich, Bjorn Ommer, Joseph Paul Cohen, and Marzyeh Ghassemi. S2sd: Simultaneous similarity-based self-distillation for deep metric learning. In *ICML 2021:*

[44] Karsten Roth, Timo Milbich, Samarth Sinha, Prateek Gupta, Bjorn Ommer, and Joseph Paul Cohen. Revisiting training strategies and generalization performance in deep metric learning. In *International Conference on Machine Learning*, pages 8242–8252. PMLR, 2020. 1, 2, 6

[45] 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. 2

[46] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 6

[47] Artsiom Sanakoyeu, Vadim Tschernezki, Uta Buchler, and Bjorn Ommer. Divide and conquer the embedding space for metric learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[48] 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, 6

[49] Jenny Seidenschwarz, Ismail Elezi, and Laura Leal-Taixé. Learning intra-batch connections for deep metric learning. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 9410–9421. PMLR, 2021. 2, 7

[50] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018. 2, 4, 6

[51] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in Neural Information Processing Systems*, pages 1857–1865, 2016. 2

[52] Justin Solomon, Fernando De Goes, Gabriel Peyré, Marco Cuturi, Adrian Butscher, Andy Nguyen, Tao Du, and Leonidas Guibas. Convolutional wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Transactions on Graphics (TOG)*, 34(4):66, 2015. 5

[53] Eu Wern Teh, Terrance DeVries, and Graham W Taylor. Proxynca++: Revisiting and revitalizing proxy neighborhood component analysis. In *European Conference on Computer Vision (ECCV)*. Springer, 2020. 2, 6, 7

[54] Shashanka Venkataramanan, Bill Psomas, Ewa Kijak, laurent amsaleg, Konstantinos Karantzalos, and Yannis Avrithis. It takes two to tango: Mixup for deep metric learning. In *International Conference on Learning Representations*, 2022. 2, 7

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

[56] Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R. Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2, 6

[57] Xun Wang, Haozhi Zhang, Weilin Huang, and Matthew R Scott. Cross-batch memory for embedding learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6388–6397, 2020. 1, 2, 6, 7

[58] Kilian Q Weinberger, John Blitzer, and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in neural information processing systems*, pages 1473–1480, 2006. 2

[59] 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. 2, 4, 6

[60] Eric P Xing, Michael I Jordan, Stuart J Russell, and Andrew Y Ng. Distance metric learning with application to clustering with side-information. In *Advances in neural information processing systems*, pages 521–528, 2003. 2

[61] Huan Xu and Shie Mannor. Robustness and generalization. *Machine learning*, 86(3):391–423, 2012. 2

[62] Hong Xuan, Richard Souvenir, and Robert Pless. Deep randomized ensembles for metric learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 723–734, 2018. 2

[63] Zhibo Yang, Muhammet Bastan, Xinliang Zhu, Douglas Gray, and Dimitris Samaras. Hierarchical proxy-based loss for deep metric learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1859–1868, 2022. 2, 7

[64] Baosheng Yu and Dacheng Tao. Deep metric learning with tuplet margin loss. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 2

[65] Dingyi Zhang, Yingming Li, and Zhongfei Zhang. Deep metric learning with spherical embedding. *Advances in Neural Information Processing Systems*, 33, 2020. 2

[66] Wenliang Zhao, Yongming Rao, Ziyi Wang, Jiwen Lu, and Jie Zhou. Towards interpretable deep metric learning with structural matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9887–9896, 2021. 7

[67] Yiru Zhao, Zhongming Jin, Guo-jun Qi, Hongtao Lu, and Xian-sheng Hua. An adversarial approach to hard triplet generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 501–517, 2018. 2

[68] Wenzhao Zheng, Zhaodong Chen, Jiwen Lu, and Jie Zhou. Hardness-aware deep metric learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[69] Wenzhao Zheng, Chengkun Wang, Jiwen Lu, and Jie Zhou. Deep compositional metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9320–9329, 2021. 2

[70] 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. 2

[71] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016. 3

[72] Yuehua Zhu, Muli Yang, Cheng Deng, and Wei Liu. Fewer is more: A deep graph metric learning perspective using fewer proxies. *Advances in Neural Information Processing Systems*, 33:17792–17803, 2020. 1, 2, 7