Revisiting Training Strategies and Generalization Performance in Deep Metric Learning

Karsten Roth [*] [1] [2] **Timo Milbich** [*] [2] **Samarth Sinha** [1] [3] **Prateek Gupta** [1] [4] **Björn Ommer** [2] **Joseph Paul Cohen** [1]

Abstract

Deep Metric Learning (DML) is arguably one of the most influential lines of research for learning visual similarities with many proposed approaches every year. Although the field benefits from the rapid progress, the divergence in training protocols, architectures, and parameter choices make an unbiased comparison difficult. To provide a consistent reference point, we revisit the most widely used DML objective functions and conduct a study of the crucial parameter choices as well as the commonly neglected mini-batch sampling process. Under consistent comparison, DML objectives show much higher saturation than indicated by literature. Further based on our analysis, we uncover a correlation between the embedding space density and compression to the generalization performance of DML models. Exploiting these insights, we propose a simple, yet effective, training regularization to reliably boost the performance of ranking-based DML models on various standard benchmark datasets. Code and a publicly accessible WandB-repo are available at https://github.com/Confusezius/Revisiting_Deep_Metric_Learning_PyTorch.

1. Introduction

Learning visual similarity is important for a wide range of vision tasks, such as image clustering (Bouchacourt et al., 2018), face detection (Schroff et al., 2015) or image retrieval (Wu et al., 2017). Measuring similarity requires learning an embedding space which captures images and reasonably reflects similarities using a defined distance metric. One of the most adopted classes of algorithms for this task is Deep Metric Learning (DML) which leverages deep neural networks to learn such a distance preserving embedding.

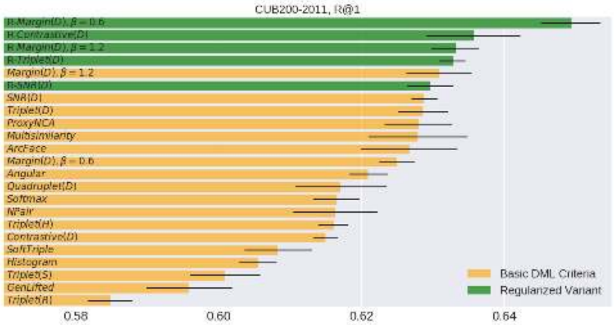


Figure 1. Mean recall performance and standard deviation of various DML objectives trained with (green) and without (orange) our proposed regularization. For all benchmarks, see appendix.

Due to the growing interest in DML, a large corpus of literature has been proposed contributing to its success. However, as recent DML approaches explore more diverse research directions such as architectures (Xuan et al., 2018; Jacob et al., 2019), objectives functions (Wang et al., 2019b; Yuan et al., 2019) and additional training tasks (Roth et al., 2019; Lin et al., 2018), an unbiased comparison of their results becomes more and more difficult. Further, undisclosed technical details (s.a. data augmentations or training regularization) pose a challenge to the reproducibility of such methods, which is of great concern in the machine learning community in general (Bouthillier et al., 2019). One goal of this work is to counteract this worrying trend by providing a comprehensive comparison of important and current DML baselines under identical training conditions on standard benchmark datasets (Fig. 1). In addition, we thoroughly review common design choices of DML models which strongly influence generalization performance to allow for better comparability of current and future work.
On that basis, we extend our analysis to: *(i)* The process of data sampling which is well-known to impact the DML optimization (Schroff et al., 2015). While previous works only studied this process in the specific context of triplet mining strategies for ranking-based objectives (Wu et al., 2017; Harwood et al., 2017), we examine the model-agnostic case of sampling informative mini-batches. *(ii)* The generalization capabilities of DML models by analyzing the structure

[*] Equal contribution [1] Mila, Université de Montréal [2] HCI/IWR, Heidelberg University [3] University of Toronto [4] The Alan Turing Institute, University of Oxford. Correspondence to: Karsten Roth <karsten.rh1@gmail.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020. Copyright 2020 by the author(s).

of their learned embedding spaces. While we are not able to reliably link typically targeted concepts such as large inter-class margins (Liu et al., 2017; Deng et al., 2018) and intra-class variance (Lin et al., 2018) to generalization performance, we uncover a strong correlation to the compression of the learned representations. Lastly, based on this observation, we propose a simple, yet effective, regularization technique which effectively boosts the performance of ranking-based approaches on standard benchmark datasets as also demonstrated in Fig. 1. In summary, our most important contributions can be described as follows:

- We provide an exhaustive analysis of recent DML objective functions, their training strategies, the influence of data-sampling, and model design choices to set a standard benchmark. To this end, we will make our code publicly available.
- We provide new insights into DML generalization by analyzing its correlation to the embedding space compression (as measured by its spectral decay), inter-class margins and intra-class variance.
- Based on the result above, we propose a simple technique to regularize the embedding space compression which we find to boost generalization performance of ranking-based DML approaches.

This work is structured as follows: After reviewing related work in §2, we discuss and motivate our analyzed components of DML models and their training setup in §3. Finally in §4 we present the findings of our study, analyze DML generalization in §5 and close with a conclusion in §6.

2. Related Works

Deep Metric Learning: Deep Metric Learning (DML) has become increasingly important for applications ranging from image retrieval (Movshovitz-Attias et al., 2017; Roth et al., 2019; Wu et al., 2017; Lin et al., 2018) to zero-shot classification (Schroff et al., 2015; Sanakoyeu et al., 2019) and face verification (Hu et al., 2014; Liu et al., 2017). Many approaches use ranking-based objectives based on tuples of samples such as pairs (Hadsell et al., 2006), triplets (Wu et al., 2017; Yu et al., 2018), quadruplets(Chen et al., 2017) or higher-order variants like N-Pairs(Sohn, 2016), lifted structure losses (Oh Song et al., 2016; Yu et al., 2018) or NCA-based criteria(Movshovitz-Attias et al., 2017). Further, classification-based methods adjusted to DML (Deng et al., 2018; Zhai & Wu, 2018) have proven to be effective for learning distance preserving embedding spaces. To address the computational complexity of tuple-based methods[1], different sampling strategies have been introduced (Schroff et al., 2015; Wu et al., 2017; Ge, 2018; Roth et al., 2020). Moreover, proxy-based approaches address this issue by approximating class distributions using only few virtual representatives (Movshovitz-Attias et al., 2017; Qian et al., 2019).

Additionally, more involved research extending above objectives has been proposed: Sanakoyeu et al. (2019) follow a divide-and-conquer strategy by splitting and subsequently merging both the data and embedding space; Opitz et al. (2018); Xuan et al. (2018) employ an ensemble of specialized learners and Roth et al. (2019); Milbich et al. (2020a;b) combine DML with feature mining or self-supervised learning. Moreover, Lin et al. (2018) and Zheng et al. (2019) generate artificial samples to effectively augment the training data, thus learning more complex ranking relations. The majority of these methods are trained using the essential objective functions and, further, hinge on the training parameters discussed in our study, thus directly benefiting from our findings. Moreover, we propose an effective regularization technique to improve ranking-based objectives.

Mini-batch selection: The benefits of large mini-batches for training are well studied (Smith et al., 2017; Goyal et al., 2017; Keskar et al., 2016). However, there has been limited research examining effective strategies for the creation of mini-batches. Research into mini-batch creation has been done to improve convergence in optimization methods for classification tasks(Mirzasoleiman et al., 2020; Johnson & Guestrin, 2018) or to construct informative mini-batches using core-set selection to optimize generative models (Sinha et al., 2019). Similarly, we analyze mining strategies maximizing data diversity and compare their impact to standard heuristics employed in DML (Wu et al., 2017; Roth et al., 2019; Sanakoyeu et al., 2019)).

Generalization in DML: Generalization capabilities of representations (Achille & Soatto, 2016; Shwartz-Ziv & Tishby, 2017) and, in particular, of discriminative models has been well studied (Jiang* et al., 2020; Belghazi et al., 2018; Goyal et al., 2017), e.g. in the light of compression (Tishby & Zaslavsky, 2015; Shwartz-Ziv & Tishby, 2017) which is covered by strong experimental support (Goyal et al., 2019; Belghazi et al., 2018; Alemi et al., 2016). Verma et al. (2018) link compression to a 'flattening' of a representation in the context of classification. We apply this concept to analyze generalization in DML and find that strong compression actually hurts DML generalization. Existing works on generalization in metric learning focus on robustness of linear or kernel-based distance metrics (Bellet & Habrard, 2015; Bellet, 2013) and examine bounds on the generalization error (Huai et al., 2019). In contrast, we examine the correlation between generalization and structural characteristics of the learned embedding space.

[1] As an example, the number of triplets scales with $\mathcal{O}(N^3)$, where N is the dataset size.

3. Training a Deep Metric Learning Model

In this section, we briefly summarize key components for training a DML model and motivate the main aspects of our study. We first introduce the common categories of training objectives which we consider for comparison in Sec. 3.1. Next, in Sec. 3.2 we examine the data sampling process and present strategies for sampling informative mini-batches. Finally, in Sec. 3.3, we discuss components of a DML model which impact its performance and exhibit an increased divergence in the field, thus impairing objective comparisons.

3.1. The objective function

In Deep Metric Learning we learn an embedding function $\phi : \mathcal{X} \mapsto \Phi \subseteq \mathbb{R}^D$ mapping datapoints $x \in \mathcal{X}$ into an embedding space Φ, which allows to measure the similarity between x_i, x_j as $d_\phi(x_i, x_j) := d(\phi(x_i), \phi(x_j))$ with $d(.,.)$ being a predefined distance function. For that, let $\phi := \phi_\theta$ be a deep neural network parametrised by θ with its output typically normalized to the real hypersphere $\mathbb{S}^D$ for regularization purposes (Wu et al., 2017; Huai et al., 2019). In order to train ϕ_θ to reflect the semantic similarity defined by given labels $y \in \mathcal{Y}$, many objective functions have been proposed based on different concepts which we now briefly summarize.

Ranking-based: The most popular family are ranking-based loss functions operating on pairs (Hadsell et al., 2006), triplets (Schroff et al., 2015; Wu et al., 2017) or larger sets of datapoints (Sohn, 2016; Oh Song et al., 2016; Chen et al., 2017; Wang et al., 2019b). Learning ϕ_θ is defined as an ordering task, such that the distances $d_\phi(x_a, x_p)$ between an anchor x_a and positive x_p of the same class, $y_a = y_p$, is minimized and the distances $d_\phi(x_a, x_n)$ of to negative samples x_n with different class labels, $y_a \neq y_n$, is maximized. For example, triplet-based formulations typically optimize their relative distances as long as a margin γ is violated, i.e. as long as $d_\phi(x_a, x_n) - d_\phi(x_a, x_p) < \gamma$. Further, ranking-based objectives are also extended to histogram matching, as proposed in (Ustinova & Lempitsky, 2016).

Classification-based: As DML is essentially solving a discriminative task, some approaches (Zhai & Wu, 2018; Deng et al., 2018; Liu et al., 2017) can be derived from softmax-logits $l_i = W_j^T \phi(x_i) + b_j$. For example, Deng et al. (2018) exploit the regularization to the real hypersphere $\mathbb{S}^D$ and the equality $W_j^T x_i = \left\|W_j^T\right\| \|\phi(x_i)\| \cos \varphi_j$ to maximize the margin between classes by direct optimization over angles φ_j. Further, also standard cross-entropy optimization proves to be effective under normalization (Zhai & Wu, 2018).

Proxy-based: These methods approximate the distributions for the full class by one (Movshovitz-Attias et al., 2017) or more (Qian et al., 2019) learned representatives. By considering the class representatives for computing the training loss, individual samples are directly compared to an entire class. Additionally, proxy-based methods help to alleviate the issue of tuple mining which is encountered in ranking-based loss functions.

3.2. Data sampling

The synergy between tuple mining strategies and ranking losses has been widely studied (Wu et al., 2017; Schroff et al., 2015; Ge, 2018). To analyze the impact of data-sampling on performance in the scope of our study, we consider the process of mining informative mini-batches $\mathcal{B}$. This process is independent of the specific training objective and so far has been commonly neglected in DML research. Following we present batch mining strategies operating on both labels and the data itself: *label samplers*, which are sampling heuristics that follow selection rules based on label information only, and *embedded samplers*, which operate on data embeddings themselves to create batches $\mathcal{B}$ of diverse data statistics.

Label Samplers: To control the class distribution within $\mathcal{B}$, we examine two different heuristics based on the number, n, of 'Samples Per Class' (SPC-n) heuristic:

SPC-2/4/8: Given batch-size b, we randomly select b/n unique classes from which we select n samples randomly.

SPC-R: We randomly select $b-1$ samples from the dataset and choose the last sample to have the same label as one of the other $b-1$ samples to ensure that at least one triplet can be mined from $\mathcal{B}$. Thus, we effectively vary the number of unique classes within mini-batches.

Embedded Samplers: Increasing the batch-size b has proven to be beneficial for stabilizing optimization due to an effectively larger data diversity and richer training information (Mirzasoleiman et al., 2020; Brock et al., 2018; Sinha et al., 2019). As the DML training is commonly performed on a single GPU (limited especially due to tuple mining process on the mini-batch), the batch-size b is bounded by memory. Nevertheless, in order to 'virtually' maximize the data diversity, we distill the information content of a large set of samples $\mathcal{B}^*, b^* = |\mathcal{B}^*| > b$ into a mini-batch B by matching the statistics of $\mathcal{B}$ and $\mathcal{B}^*$ under the embedding ϕ. To avoid computational overhead, we sample $\mathcal{B}^*$ from a continuously updated memory bank $\mathcal{M}$ of embedded training samples. Similar to Misra & van der Maaten (2019), $\mathcal{M}$ is generated by iteratively updating its elements based on the steady stream of training batches $\mathcal{B}$. Using $\mathcal{M}$, we mine mini-batches by first randomly sampling $\mathcal{B}^*$ from $\mathcal{M}$ with $b^* = 1024$ and subsequently find a mini-batch $\mathcal{B}$ to match its data statistics by using one of the following criteria:

Greedy Coreset Distillation (GC): Greedy Coreset (Agarwal et al., 2005) finds a batch $\mathcal{B}$ by iteratively adding samples $x^* \in \mathcal{B}^*$ which maximize the distance from the samples that have already been selected $x \in \mathcal{B}$, thereby maximizing the covered space within Φ by solving $\min_{\mathcal{B}:|\mathcal{B}|=b} \max_{x*\in\mathcal{B}^*} \min_{x\in\mathcal{B}} d_\phi(x, x^*)$.

Matching of distance distributions (DDM): DDM aims to preserve the distance distribution of $\mathcal{B}^*$. We randomly select m candidate mini-batches and choose the batch $\mathcal{B}$ with smallest Wasserstein distance between normalized distance histograms of $\mathcal{B}$ and $\mathcal{B}^*$ (Rubner et al., 2000).

FRD-Score Matching (FRD): Similar to the recent GAN evaluation setting, we compute the frechet distance (Heusel et al., 2017)) between $\mathcal{B}$ and $\mathcal{B}^*$ to measure the similarity between their distributions using $FRD(\mathcal{B}, \mathcal{B}^*) = \|\mu_\mathcal{B} - \mu_{\mathcal{B}^*}\|_2^2 + \text{Tr}(\Sigma_\mathcal{B} + \Sigma_{\mathcal{B}^*} - 2(\Sigma_\mathcal{B}\Sigma_{\mathcal{B}^*})^{1/2})$, with $\mu_\bullet$, $\Sigma_\bullet$ being the mean and covariance of the embedded set of samples. Like in DDM, we select the closest batch $\mathcal{B}$ to $\mathcal{B}^*$ among m randomly sampled candidates.

3.3. Training parameters, regularization and architecture

Network	GN	IBN	R50
CUB200, R@1	45.41	48.78	43.77
CARS196, R@1	35.31	43.36	36.39
SOP, R@1	44.28	49.05	48.65

Table 1. *Recall performance of commonly used network architectures after ImageNet pretraining.* Final linear layer is randomly initialized and normalized.

Next to the objectives and data sampling process, successful learning hinges on a reasonable choice of the training environment. While there is a multitude of parameters to be set, we identify several factors which both influence performance and exhibit an divergence in lately proposed works.

Architectures: In recent DML literature predominantly three basis network architectures are used: GoogLeNet (Szegedy et al., 2015) (GN, typically with embedding dimensionality 512), Inception-BN (Ioffe & Szegedy, 2015) (IBN, 512) and ResNet50 (He et al., 2016) (R50, 128) (with optionally frozen Batch-Normalization layers for improved convergence and stability across varying batch sizes[2], see e.g. Roth et al. (2019); Cakir et al. (2019)). Due to the varying number of parameters and configuration of layers, each architecture exhibits a different starting point for learning, based on its initialization by ImageNet pretraining (Deng et al., 2009). Table 1 compares their initial DML performance measured in Recall@1 (R@1). The reference to differences in architecture is one of the main arguments used by individual works not compare themselves to competing approaches. Disconcertingly, even when reporting additional results using adjusted networks is feasible, typically only results using a single architecture are reported. Consequently, a fair comparison between approaches is heavily impaired.

Weight Decay: Commonly, network optimization is regularized using weight decay/L2-regularization (Krogh & Hertz, 1992). In DML, particularly on small datasets its careful adjustment is crucial to maximize generalization performance. Nevertheless, many works do not report this.

Embedding dimensionality: Choosing a dimensionality D of the embedding space Φ influences the learned manifold and consequently generalization performance. While each architecture typically uses an individual, standardized dimensionality D in DML, recent works differ without reporting proper baselines using an adjusted dimensionality. Again, comparison to existing works and the assessment of the actual contribution is impaired.

Data Preprocessing: Preprocessing training images typically significantly influences both the learned features and model regularization. Thus, as recent approaches vary in their applied augmentation protocols, results are not necessarily comparable. This includes the trend for increased training and test image sizes.

Batchsize: Batchsize determines the nature of the gradient updates to the network, e.g. datasets with many classes benefit from large batchsizes due to better approximations of the training distribution. However, it is commonly not taken into account as a influential factor of variation.

Advanced DML methodologies There are many extensions to objective functions, architectures and the training setup discussed so far. However, although extensions are highly individual, they still rely on these components and thus benefit from findings in the following experiments, evaluations and analysis.

4. Analyzing DML training strategies

Datasets As benchmarking datasets, we use:

CUB200-2011: Contains 11,788 images in 200 classes of birds. Train/Test sets are made up of the first/last 100 classes (5,864/5,924 images respectively) (Wah et al., 2011). Samples are distributed evenly across classes.

CARS196: Has 16,185 images/196 car classes with even sample distribution. Train/Test sets use the first/last 98 classes (8054/8131 images) (Krause et al., 2013).

Stanford Online Products (SOP): Contains 120,053 product images divided into 22,634 classes. Train/Test sets are provided, contain 11,318 classes/59,551 images in the Train and 11,316 classes/60,502 images in the Test set (Oh Song et al., 2016). In SOP, unlike the other benchmarks, most classes have few instances, leading to significantly different data distribution compared to CUB200-2011 and CARS196.

4.1. Experimental Protocol

Our training protocol follows parts of Wu et al. (2017), which utilize a ResNet50 architecture with frozen Batch-Normalization layers and embedding dim. 128 to be comparable with already proposed results with this architec-

[2] Note that Batch-Normalization is still performed, but no parameters are learned.

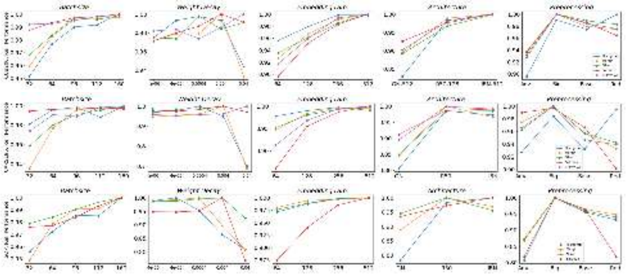


Figure 2. Evaluation of DML pipeline parameters and architectures on all benchmark datasets and their influence on relative improvement across different training criteria.

ture. While both GoogLeNet (Szegedy et al., 2015) and Inception-BN (Ioffe & Szegedy, 2015) are also often employed in DML literature, we choose ResNet50 due to its success in recent state-of-the-art approaches (Roth et al., 2019; Sanakoyeu et al., 2019). In line with standard practices we randomly resize and crop images to 224×224 for training and center crop to the same size for evaluation. During training, random horizontal flipping ($p = 0.5$) is used. Optimization is performed using Adam (Kingma & Ba, 2015) with learning rate fixed to 10^{-5} and *no* learning rate scheduling for unbiased comparison. Weight decay is set to a constant value of $4 \cdot 10^{-4}$, as motivated in section 4.2. We implemented all models in PyTorch (Paszke et al., 2017), and experiments are performed on individual Nvidia Titan X, V100 and T4 GPUs with memory usage limited to 12GB. Each training is run over 150 epochs for CUB200-2011/CARS196 and 100 epochs for Stanford Online Products, if not stated otherwise. For batch sampling we utilize the the SPC-2 strategy, as motivated in section 4.3. Finally, each result is averaged over multiple seeds to avoid seed-based performance fluctuations. All loss-specific hyperparameters are discussed in the supplementary material, along with their original implementation details. For our study, we examine the following evaluation metrics (described further in the supplementary): Recall at 1 and 2 (Jegou et al., 2011), Normalized Mutual Information (NMI) (Manning et al., 2010), F1 score (Sohn, 2016), mean average precision measured on recall of the number of samples per class (mAP@C) and mean average precision measured on the recall of 1000 samples (mAP@1000). Please see the supplementary (supp. A.3) for more information.

4.2. Studying DML parameters and architectures

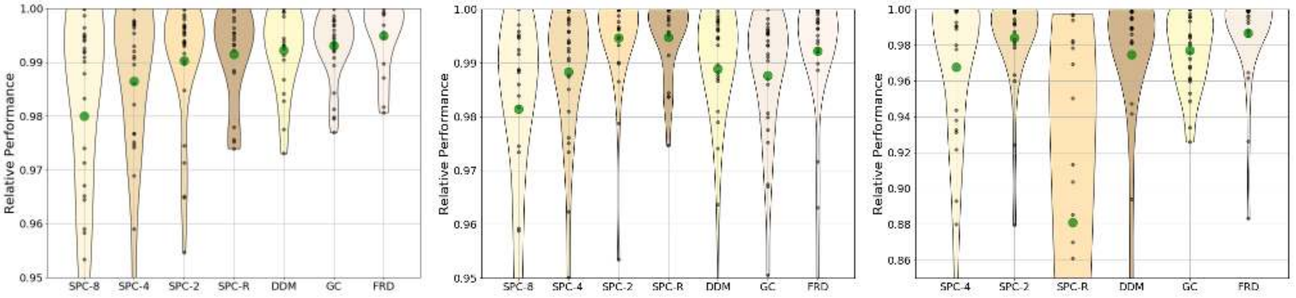
Now we study the influence of parameters & architectures discussed in Sec. 3.3 using five different objectives. For each experiment, all metrics noted in Sec. 4.1 are measured. For each loss, every metric is normalized by the maximum across the evaluated value range. This enables an aggregated summary of performance across all metrics, where differences correspond to relative improvement. Fig. 2 analyzes each factor by evaluating a range of potential setups with the other parameters fixed to values from Sec. 4.1: Increasing the batchsize generally improves results with gains varying among criteria, with particularly high relevance on the SOP dataset. For weight decay, we observe loss and dataset dependent behavior up to a relative performance change of 5%. Varying the data preprocessing protocol, e.g. augmentations and input image size, leads to large performance differences as well. *Base* follows our protocol described in Sec. 4.1. *Red.* refers to resizing of the smallest image side to 256 and cropping to 224x224 with horizontal flipping. *Big* uses *Base* but crops images to 256x256. Finally, we extend *Base* to *Adv.* with color jittering, changes in brightness and hue. We find that larger images provide better performance regardless of objective or dataset. Using the *Adv.* processing on the other hand is dependent on the dataset. Finally, we show that random resized cropping is a generally stronger operation than basic resizing and cropping.

All these factors underline the importance of a complete declaration of the training protocol to facilitate reproducibility and comparability. Similar results are observed for the choice of architecture and embedding dimensionality D. At the example of R50, our analysis shows that training objectives perform differently for a given D but seem to converge at $D = 512$. However, for R50 D is typically fixed to 128, thus disadvantaging some training objectives over others. Finally, comparing common DML architectures reveals their strong impact on performance with varying variance between loss functions. Highest consistencies seem to be achievable with R50 and IBN-based setups.

Implications: In order to warrant unbiased comparability, equal and transparent training protocols and model architectures are essential, as even small deviations can result in large deviations in performance.

4.3. Batch sampling impacts DML training

We now analyze how the data sampling process for mini-batches impacts the performance of DML models using the sampling strategies presented in Sec. 3.2. To conduct an unbiased study, we experiment with six conceptually different objective functions: Marginloss with Distance-Weighted Sampling, Triplet Loss with Random Sampling, ProxyNCA, Multi-Similarity Loss, Histogram loss and Normalized Softmax loss. To aggregate our evaluation metrics (cf. 4.1), we utilize the same normalization procedure discussed in Sec. 4.2. Fig. 3 summarizes the results for each sampling strategy by reporting the distributions of normalized scores of all pairwise combinations of training loss and evaluation metrics. Our analysis reveals that the batch sampling process indeed effects DML training with a difference in *mean* performance up to 1.5%. While there is no clear winner across all datasets, we observe that the SPC-2 and FRD samplers perform very well and, in particular, consistently



(a) CUB200-2011

(b) CARS196

(c) SOP

Figure 3. Comparison of mini-batch mining strategies on three different datasets. Performance measures Recall@1 and 2, NMI, mAP and F1 are normalized across metrics and loss function. We plot the distributions of relative performances for each strategy.

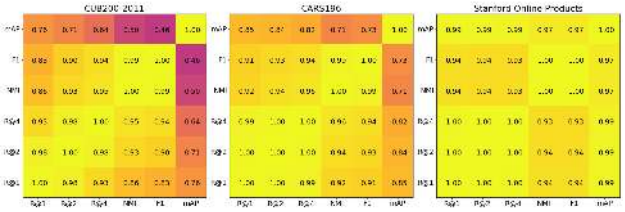


Figure 4. Metrics Correlation matrix for standard (Recall, NMI) and underreported retrieval metrics. mAP denotes mAP@C. Please refer to the supplementary for more information.

outperform the SPC-4 strategy which is commonly reported to be used in literature (Wu et al., 2017; Schroff et al., 2015). **Implications:** Our study indicates that DML benefits from data diversity in mini-batches, independent of the chosen training objective. This coincides with the general benefit of larger batchsizes as noted in section 4.2. While complex mining strategies may perform better, simple heuristics like SPC-2 are sufficient.

4.4. Comparing DML models

Based on our training parameter and batch-sampling evaluations we compare a large selection of 14 different DML objectives and 4 mining methods under fixed training conditions (see 4.1 & 4.2), most of which claim state-of-the-art by a notable margin. For ranking-based models, we employ distance-based tuple mining (D) (Wu et al., 2017) which proved most effective. We also include random, semihard sampling (Schroff et al., 2015) and a soft version of hard sampling (Roth & Brattoli, 2019) for our tuple mining study using the classic triplet loss. Loss-specific hyperparameters are determined via small cross-validation gridsearches around originally proposed values to adjust for our training setup. Exact parameters and method details are listed in supp. A.1. Table 2 summarizes our evaluation results on all benchmarks (**with other metric rankings** s.a. mAP@C or mAP@1000 **in the supplementary** (supp. I)), while Fig. 4 measures correlations between all evaluation metrics. Particularly on CUB200-2011 and CARS196 we find a higher performance saturation between methods as compared to SOP due to strong differences in data distribution. Generally, performance between criteria is much more similar than literature indicates, (see also concurrent work by Musgrave et al. (2020)). We find that representatives of ranking-based objectives outperform their classification/NCE-based counterparts, though not significantly. On average, margin loss (Wu et al., 2017) and multisimilarity loss (Wang et al., 2019a) offer the best performance across datasets, though not by a notable margin. Remarkably, under our carefully chosen training setting, a multitude of losses compete or even *outperform* more involved state-of-the-art DML approaches on the SOP dataset. For a detailed comparison to the state-of-the-art, we refer to the supplementary (supp. F). **Implications:** Under the same setup, performance saturates across methods, contrasting results reported in literature. Taking into account standard deviations, usually left unreported, improvements become even less significant. In addition, carefully trained baseline models are able to outperform state-of-the-art approaches which use considerable stronger architectures. Thus, to evaluate the true benefit of proposed contributions, baseline models need to be competitive and implemented under comparable settings.

5. Generalization in Deep Metric Learning

The previous section showed how different model and training parameter choices result in vastly different performances. However, how such differences can be explained best on basis of the learned embedding space is an open question and, for instance, studied under the concept of compression (Tishby & Zaslavsky, 2015). Recent work (Verma et al., 2018) links compression to class-conditioned flattening of representation, indicated by an increased decay of singular values obtained by Singular Value Decomposition (SVD) on the data representations. Thus, class representations occupy a more compact volume, thereby reducing the number of directions with significant variance. The subse-

Benchmarks→	CUB200-2011		CARS196		SOP	
Approaches ↓	R@1	NMI	R@1	NMI	R@1	NMI
Imagenet (Deng et al., 2009)	43.77	57.56	36.39	37.96	48.65	58.64
Angular (Wang et al., 2017)	62.10 ± 0.27	67.59 ± 0.26	78.00 ± 0.32	66.48 ± 0.44	73.22 ± 0.07	89.53 ± 0.01
ArcFace (Deng et al., 2018)	62.67 ± 0.67	67.66 ± 0.38	79.16 ± 0.97	66.99 ± 1.08	77.71 ± 0.15	90.09 ± 0.03
Contrastive (Hadsell et al., 2006) (D)	61.50 ± 0.17	66.45 ± 0.27	75.78 ± 0.39	64.04 ± 0.13	73.21 ± 0.04	89.78 ± 0.02
GenLifted (Hermans et al., 2017)	59.59 ± 0.60	65.63 ± 0.14	72.17 ± 0.38	63.75 ± 0.35	75.21 ± 0.12	89.84 ± 0.01
Hist. (Ustinova & Lempitsky, 2016)	60.55 ± 0.26	65.26 ± 0.23	76.47 ± 0.38	64.15 ± 0.36	71.30 ± 0.10	88.93 ± 0.02
Margin (D, $\beta = 0.6$) (Wu et al., 2017)	62.50 ± 0.24	67.02 ± 0.37	77.70 ± 0.32	65.29 ± 0.32	77.38 ± 0.11	**90.45 ± 0.03**
Margin (D, $\beta = 1.2$) (Wu et al., 2017)	**63.09 ± 0.46**	68.21 ± 0.33	79.86 ± 0.33	67.36 ± 0.34	**78.43 ± 0.07**	90.40 ± 0.03
Multisimilarity (Wang et al., 2019a)	62.80 ± 0.70	**68.55 ± 0.38**	**81.68 ± 0.19**	**69.43 ± 0.38**	77.99 ± 0.09	90.00 ± 0.02
Npair (Sohn, 2016)	61.63 ± 0.58	67.64 ± 0.37	77.48 ± 0.28	66.55 ± 0.19	75.86 ± 0.08	89.79 ± 0.03
Pnca (Movshovitz-Attias et al., 2017)	62.80 ± 0.48	66.93 ± 0.38	78.48 ± 0.58	65.76 ± 0.22	–	–
Quadruplet (D) (Chen et al., 2017)	61.71 ± 0.63	66.60 ± 0.41	76.34 ± 0.27	64.79 ± 0.50	76.95 ± 0.10	90.14 ± 0.02
SNR (D) (Yuan et al., 2019)	62.88 ± 0.18	67.16 ± 0.25	78.69 ± 0.19	65.84 ± 0.52	77.61 ± 0.34	90.10 ± 0.08
SoftTriple (Qian et al., 2019)	60.83 ± 0.47	64.27 ± 0.36	75.66 ± 0.46	62.66 ± 0.16	–	–
Softmax (Zhai & Wu, 2018)	61.66 ± 0.33	66.77 ± 0.36	78.91 ± 0.27	66.35 ± 0.30	76.92 ± 0.64	89.82 ± 0.15
Triplet (D) (Wu et al., 2017)	62.87 ± 0.35	67.53 ± 0.14	79.13 ± 0.27	65.90 ± 0.18	77.39 ± 0.15	90.06 ± 0.02
Triplet (H) (Roth & Brattoli, 2019)	61.61 ± 0.21	65.98 ± 0.41	77.60 ± 0.33	65.37 ± 0.26	73.50 ± 0.09	89.25 ± 0.03
Triplet (R) (Schroff et al., 2015)	58.48 ± 0.31	63.84 ± 0.30	70.63 ± 0.43	61.09 ± 0.27	67.86 ± 0.14	88.35 ± 0.04
Triplet (S) (Schroff et al., 2015)	60.09 ± 0.49	65.59 ± 0.29	72.51 ± 0.47	62.84 ± 0.41	73.61 ± 0.14	89.35 ± 0.02
R-Contrastive (D)	63.57 ± 0.66	67.63 ± 0.31	81.06 ± 0.41	67.27 ± 0.46	74.36 ± 0.11	89.94 ± 0.02
R-Margin (D, $\beta = 0.6$)	**64.93 ± 0.42**	68.36 ± 0.32	**82.37 ± 0.13**	68.66 ± 0.47	77.58 ± 0.11	90.42 ± 0.03
R-Margin (D, $\beta = 1.2$)	63.32 ± 0.33	67.91 ± 0.66	81.11 ± 0.49	67.72 ± 0.79	**78.52 ± 0.10**	90.33 ± 0.02
R-SNR (D)	62.97 ± 0.32	68.04 ± 0.34	80.38 ± 0.35	67.60 ± 0.20	77.69 ± 0.25	90.02 ± 0.06
R-Triplet (D)	63.28 ± 0.18	67.86 ± 0.51	81.17 ± 0.11	67.79 ± 0.23	77.33 ± 0.14	89.98 ± 0.04

Table 2. Comparison of Recall@1 and NMI performances for all objectives averaged over 5 runs. Each model is trained using the same training setting over 150 epochs for CUB/CARS and 100 epochs for SOP. 'R-' denotes model trained with ρ-regularization. **Bold** denotes best results excluding regularization. **Boldblue** marks overall best results. **Please note that a ranking on all other metrics (s.a. mAP@C, mAP@1000) as well as a visual summary can be found in the supplementary** (supp. I, supp. C)!

quent strong focus on the most discriminative directions is shown to be beneficial for classic classification scenarios with i.i.d. train and test distributions. However, this overly discards features which could capture data characteristics outside the training distribution. Hence, generalization in transfer problems like DML is hindered due to the shift in training and testing distribution (Bellet & Habrard, 2015). We thus hypothesize that actually retaining a considerable amount of directions of significant variance (DoV) is crucial to learn a well generalizing embedding function ϕ.

To verify this assumption, we analyze the spectral decay of the embedded training data $\Phi_\mathcal{X} := \{\phi(x)|x \in \mathcal{X}\}$ via SVD. We then normalize the sorted spectrum of singular values (SV) $\mathcal{S}_{\Phi_\mathcal{X}}$[3] and compute the KL-divergence to a D-dim. discrete uniform distribution $\mathcal{U}_D$, i.e. $\rho(\Phi) = \text{KL}(\mathcal{U}_D \mid\mid \mathcal{S}_{\Phi_\mathcal{X}})$[4]. We don't consider individual training class representations, as testing and training distribution are shifted[5]. Lower values of $\rho(\Phi)$ indicate more directions of significant variance. Using this measure, we analyze a large selection of DML objectives in Fig. 5 (rightmost) on CUB200-2011, CARS196 and SOP[6]. Comparing R@1 and $\rho(\Phi)$ reveals significant inverse correlation (≤ -0.63) between generalization and the spectral decay of embedding spaces Φ, which highlights the benefit of more directions of variance in the presence of train-test distribution shifts.

We now compare our finding to commonly exploited concepts for training such as *(i)* larger margins between classes (Deng et al., 2018; Liu et al., 2017), i.e. an increase in average inter-class distances $\pi_\text{inter}(\Phi) = \frac{1}{Z_\text{inter}} \sum_{y_l, y_k, l\neq k} d(\mu(\Phi_{y_l}), \mu(\Phi_{y_k}))$; *(ii)* explicitly introducing intra-class variance (Lin et al., 2018), which is indicated by an increase in average intra-class distance $\pi_\text{intra}(\Phi) = \frac{1}{Z_\text{intra}} \sum_{y_l \in \mathcal{Y}} \sum_{\phi_i, \phi_j \in \Phi_{y_l}, i\neq j} d(\phi_i, \phi_j)$. We also investigate *(iii)* their relation by using the ratio $\pi_\text{ratio}(\Phi) = \pi_\text{intra}(\Phi)/\pi_\text{inter}(\Phi)$, which can be regarded as an embedding space density. Here, $\Phi_{y_l} = \{\phi_i := \phi_\theta(x_i)|x_i \in \mathcal{X}, y_i = y_l\}$ denotes the set of embedded samples of a class y_l, $\mu(\Phi_{y_l})$ their mean embedding and Z_inter, Z_intra normalization constants. Fig. 5 compares these measures with $\rho(\Phi)$. It is evident that neither of the distance related measures $\pi_\bullet(\Phi)$ consistently exhibits significant correlation with generalization performance when taking all three datasets into

[3] Excluding highest SV which can obfuscate remaining DoVs.

[4] For simplicity we use the notation $\rho(\Phi)$ instead of $\rho(\Phi_\mathcal{X})$.

[5] For completeness, class-conditioned singular value spectra as Verma et al. (2018) are examined in supp. H.

[6] A detailed comparison can be found in supp. I.

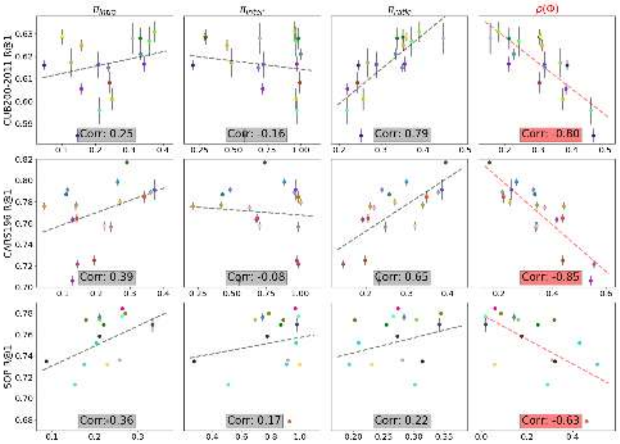


Figure 5. Correlation between generalization and structural properties derived from $\Phi_{\mathcal{X}}$ using different DML objectives on each dataset. *Left-to-Right*: Mean intra-class distances π_{intra} & inter-class distances π_{inter}, the ratio $\pi_{\text{intra}}/\pi_{\text{inter}}$ and spectral decay ρ.

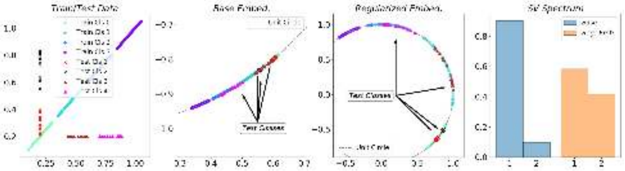


Figure 6. Toy example illustrating the effect of ρ-regularization. (Leftmost) training and test data. (Mid-left) A small, normalized two-layer fully-connected network trained with standard contrastive loss fails to separate all test classes due to excessive compression of the learned embedding. (Mid-right) The regularized embedding successfully separates the test classes by introducing a lower spectral decay. (Rightmost) Singular value spectra of training embeddings learned with and without regularization.

account. For CUB200-2011 and CARS196, we however find that an increased embedding space density (π_{ratio}) is linked to stronger generalisation. For SOP, its estimate is likely too noisy due to the strong imbalance between dataset size and amount of samples per class.

Implications: Generalization in DML exhibits strong inverse correlation to the SV spectrum decay of learned representations, as well as a weaker correlation to the embedding space density. This indicates that representation learning under considerable shifts between training and testing distribution is hurt by excessive feature compression, but may benefit from a more densely populated embedding space.

5.1. ρ-regularization for improved generalization

We now exploit our findings to propose a simple ρ-regularization for ranking-based approaches by counteracting the compression of representations. We randomly alter tuples by switching negative samples x_n with the positive

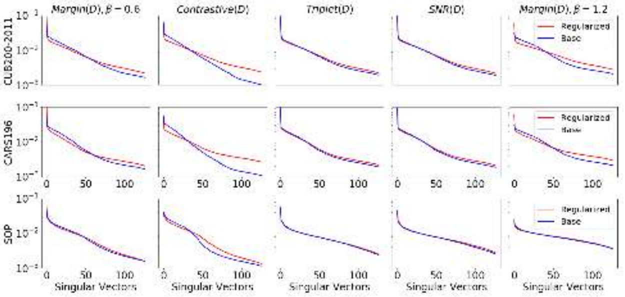


Figure 7. Sing. Value Spectrum for models trained with (red) and without (blue) ρ-regularization for various ranking-based criteria.

x_p in a given ranking-loss formulation (cf. Sec. 3.1) with probability p_{switch}. This pushes samples of the same class apart, enabling a DML model to capture extra non-label-discriminative features while dampening the compression induced by strong discriminative training signals.

Fig. 6 depicts a 2D toy example (details supp. G) illustrating the effect of our proposed regularization while highlighting the issue of overly compressed data representations. Even though the training distribution exhibits features needed to separate all test classes, these features are disregarded by the strong discriminative training signal. Regularizing the compression by attenuating the spectral decay $\rho(\Phi)$ enables the model to capture more information and exhibit stronger generalization to the unseen test classes. In addition, Fig. 7 verifies that the ρ-regularization also leads to a decreased spectral decay on DML benchmark datasets, resulting in improved recall performance (cf. Tab. 2 (bottom)), while being reasonably robust to changes in p_{switch} (see supp. B). In contrast, in the appendix we also see that encouraging higher compression seems to be detrimental to performance.

Implications: Implicitly regularizing the number of directions of significant variance can improve generalization.

6. Conclusion

In this work, we counteract the worrying trend of diverging training protocols in Deep Metric Learning (DML). We conduct a large, comprehensive study of important training components and objectives in DML to contribute to improved comparability of recent and future approaches. On this basis, we study generalization in DML and uncover a strong correlation to the level of compression and embedding density of learned data representation. Our findings reveal that highly compressed representations disregard helpful features for capturing data characteristics that transfer to unknown test distributions. To this end, we propose a simple technique for ranking-based methods to regularize the compression of the learned embedding space, which results in boosted performance across all benchmark datasets.

Acknowledgements

We would like to thank Alex Lamb for insightful discussions, and Sharan Vaswani & Dmitry Serdyuk for their valuable feedback (all Mila). We also thank Nvidia for donating NVIDIA DGX-1, and Compute Canada for providing resources for this research.

Reviewer Comments

Re: Data Augmentation/Different Spatial Resolution: We agree that data augmentation is an important factor to regularize training. We have made sure to analyze the impact of different image augmentations and resolutions on the performance.

Re: In-Shop dataset experiments: The data distribution of the In-Shop dataset is very similar to the SOP dataset, thus relative results are generally transferable, which is why we have decided not to include it in this work.

Re: Grammar and typos: We thank the reviewers for pointing out these errors and have made sure to correct them.

Re: Missing references: All mentioned references have been added.

References

Achille, A. and Soatto, S. Information dropout: Learning optimal representations through noisy computation, 2016.

Agarwal, P. K., Har-Peled, S., and Varadarajan, K. R. Geometric approximation via coresets. *Combinatorial and computational geometry*, 52:1–30, 2005.

Alemi, A. A., Fischer, I., Dillon, J. V., and Murphy, K. Deep variational information bottleneck, 2016.

Belghazi, M. I., Baratin, A., Rajeswar, S., Ozair, S., Bengio, Y., Courville, A., and Hjelm, R. D. Mine: Mutual information neural estimation, 2018.

Bellet, A. Supervised metric learning with generalization guarantees, 2013.

Bellet, A. and Habrard, A. Robustness and generalization for metric learning. *Neurocomputing*, 151:259–267, Mar 2015. ISSN 0925-2312. doi: 10.1016/j.neucom.2014.09.044. URL http://dx.doi.org/10.1016/j.neucom.2014.09.044.

Bouchacourt, D., Tomioka, R., and Nowozin, S. Multi-level variational autoencoder: Learning disentangled representations from grouped observations. In *AAAI 2018*, 2018.

Bouthillier, X., Laurent, C., and Vincent, P. Unreproducible research is reproducible. In *International Conference on Machine Learning*, pp. 725–734, 2019.

Brock, A., Donahue, J., and Simonyan, K. Large scale GAN training for high fidelity natural image synthesis. *CoRR*, abs/1809.11096, 2018. URL http://arxiv.org/abs/1809.11096.

Cakir, F., He, K., Xia, X., Kulis, B., and Sclaroff, S. Deep metric learning to rank. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Chen, W., Chen, X., Zhang, J., and Huang, K. Beyond triplet loss: a deep quadruplet network for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

Deng, J., Guo, J., Xue, N., and Zafeiriou, S. Arcface: Additive angular margin loss for deep face recognition, 2018.

Ge, W. Deep metric learning with hierarchical triplet loss. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 269–285, 2018.

Goyal, A., Islam, R., Strouse, D., Ahmed, Z., Botvinick, M., Larochelle, H., Bengio, Y., and Levine, S. Infobot: Transfer and exploration via the information bottleneck, 2019.

Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., and He, K. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

Hadsell, R., Chopra, S., and LeCun, Y. Dimensionality reduction by learning an invariant mapping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

Harwood, B., Kumar, B., Carneiro, G., Reid, I., Drummond, T., et al. Smart mining for deep metric learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2821–2829, 2017.

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

Hermans, A., Beyer, L., and Leibe, B. In defense of the triplet loss for person re-identification, 2017.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2017.

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

Huai, M., Xue, H., Miao, C., Yao, L., Su, L., Chen, C., and Zhang, A. Deep metric learning: The generalization analysis and an adaptive algorithm. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pp. 2535–2541. International Joint Conferences on Artificial Intelligence Organization, 7 2019. doi: 10.24963/ijcai.2019/352. URL https://doi.org/10.24963/ijcai.2019/352.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *International Conference on Machine Learning*, 2015.

Jacob, P., Picard, D., Histace, A., and Klein, E. Metric learning with horde: High-order regularizer for deep embeddings. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Jegou, H., Douze, M., and Schmid, C. Product quantization for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence*, 33(1):117–128, 2011.

Jiang*, Y., Neyshabur*, B., Krishnan, D., Mobahi, H., and Bengio, S. Fantastic generalization measures and where to find them. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SJgIPJBFvH.

Johnson, T. B. and Guestrin, C. Training deep models faster with robust, approximate importance sampling. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31*, pp. 7265–7275. Curran Associates, Inc., 2018.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

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

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. 2015.

Krause, J., Stark, M., Deng, J., and Fei-Fei, L. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 554–561, 2013.

Krogh, A. and Hertz, J. A. A simple weight decay can improve generalization. In *Advances in Neural Information Processing Systems*. 1992.

Lin, X., Duan, Y., Dong, Q., Lu, J., and Zhou, J. Deep variational metric learning. In *The European Conference on Computer Vision (ECCV)*, September 2018.

Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., and Song, L. Sphereface: Deep hypersphere embedding for face recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Lloyd, S. P. Least squares quantization in pcm. *IEEE Trans. Information Theory*, 28:129–136, 1982.

Manning, C., Raghavan, P., and Schütze, H. Introduction to information retrieval. *Natural Language Engineering*, 16 (1):100–103, 2010.

Milbich, T., Roth, K., Bharadhwaj, H., Sinha, S., Bengio, Y., Ommer, B., and Cohen, J. P. Diva: Diverse visual feature aggregation for deep metric learning. 2020a.

Milbich, T., Roth, K., Brattoli, B., and Ommer, B. Sharing matters for generalization in deep metric learning, 2020b.

Mirzasoleiman, B., Bilmes, J., and Leskovec, J. Coresets for accelerating incremental gradient methods, 2020. URL https://openreview.net/forum?id=SygRikHtvS.

Misra, I. and van der Maaten, L. Self-supervised learning of pretext-invariant representations, 2019.

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

Musgrave, K., Belongie, S., and Lim, S.-N. A metric learning reality check, 2020.

Oh Song, H., Xiang, Y., Jegelka, S., and Savarese, S. Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4004–4012, 2016.

Opitz, M., Waltner, G., Possegger, H., and Bischof, H. Deep metric learning with bier: Boosting independent embeddings robustly. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

Qian, Q., Shang, L., Sun, B., Hu, J., Li, H., and Jin, R. Softtriple loss: Deep metric learning without triplet sampling. 2019.

Roth, K. and Brattoli, B. Deep-metric-learning-baselines. https://github.com/Confusezius/Deep-Metric-Learning-Baselines, 2019.

Roth, K., Brattoli, B., and Ommer, B. Mic: Mining interclass characteristics for improved metric learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 8000–8009, 2019.

Roth, K., Milbich, T., and Ommer, B. Pads: Policy-adapted sampling for visual similarity learning, 2020.

Rubner, Y., Tomasi, C., and Guibas, L. J. The earth mover's distance as a metric for image retrieval. *Int. J. Comput. Vision*, 40(2):99–121, November 2000. ISSN 0920-5691. doi: 10.1023/A:1026543900054. URL https://doi.org/10.1023/A:1026543900054.

Sanakoyeu, A., Tschernezki, V., Buchler, U., and Ommer, B. Divide and conquer the embedding space for metric learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Schroff, F., Kalenichenko, D., and Philbin, J. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815–823, 2015.

Shwartz-Ziv, R. and Tishby, N. Opening the black box of deep neural networks via information, 2017.

Sinha, S., Zhang, H., Goyal, A., Bengio, Y., Larochelle, H., and Odena, A. Small-gan: Speeding up gan training using core-sets. *arXiv preprint arXiv:1910.13540*, 2019.

Smith, S. L., Kindermans, P.-J., Ying, C., and Le, Q. V. Don't decay the learning rate, increase the batch size. *arXiv preprint arXiv:1711.00489*, 2017.

Sohn, K. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in Neural Information Processing Systems*, pp. 1857–1865, 2016.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

Tishby, N. and Zaslavsky, N. Deep learning and the information bottleneck principle, 2015.

Ustinova, E. and Lempitsky, V. Learning deep embeddings with histogram loss. In *Advances in Neural Information Processing Systems*, 2016.

Verma, V., Lamb, A., Beckham, C., Najafi, A., Mitliagkas, I., Courville, A., Lopez-Paz, D., and Bengio, Y. Manifold mixup: Better representations by interpolating hidden states, 2018.

Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. The caltech-ucsd birds-200-2011 dataset. 2011.

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

Wang, X., Han, X., Huang, W., Dong, D., and Scott, M. R. Multi-similarity loss with general pair weighting for deep metric learning, 2019a.

Wang, X., Hua, Y., Kodirov, E., Hu, G., Garnier, R., and Robertson, N. M. Ranked list loss for deep metric learning. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019b.

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

Xuan, H., Souvenir, R., and Pless, R. Deep randomized ensembles for metric learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 723–734, 2018.

Yu, B., Liu, T., Gong, M., Ding, C., and Tao, D. Correcting the triplet selection bias for triplet loss. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 71–87, 2018.

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

Zhai, A. and Wu, H.-Y. Classification is a strong baseline for deep metric learning, 2018.

Zheng, W., Chen, Z., Lu, J., and Zhou, J. Hardness-aware deep metric learning. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.