Supplementary: Revisiting Training Strategies and Generalization Performance in Deep Metric Learning

A. Description of Methods

In this section, we briefly describe each DML training objective and triplet mining strategy used in our study, as well as the choice of their individual hyperparameters. General training parameters and details of the training protocol are already discussed in the main paper in Sec. 4.1. For notation, we refer to the embedding of an image x_i including output normalization as $\phi_i = \phi(x_i)$. The non-normalized version is denoted as ϕ_i^*. All methods operate on the mini-batch $\mathcal{B}$ containing image indices. If not mentioned otherwise, all embeddings operate in dimension $D = 128$.

A.1. Training Criteria

Contrastive (Hadsell et al., 2006) The contrastive training formalism is simple: Given embedding pairs $\mathcal{P}$ (sampled from a mini-batch of size b) containing an anchor ϕ_a from class y_a and either a positive ϕ_p with $y_p = y_a$ or a negative ϕ_n from a different class, $y_n \neq y_a$, the network ϕ is trained to minimize

$$\mathcal{L}_{contr} = \frac{1}{b} \sum_{(i,j)\in\mathcal{P}}^{b} \mathbb{I}_{y_i=y_j} d_e(\phi_i, \phi_j) + \mathbb{I}_{y_i\neq y_j} [\gamma - d_e(\phi_i, \phi_j)]_+ \tag{1}$$

with margin γ, which we set to 1. The margin ensures that embeddings are not projected arbitrarily far apart from each other. For our distance function we utilize the standard euclidean distance $d_e(x, y) = \|x - y\|_2$. We combine the contrastive loss with the distance-weighting negative sampling mentioned below.

Triplet (Hu et al., 2014) Triplets extend the contrastive formalism to provide a concurrent ranking surrogate for both negative and positive sample embeddings using triplets $\mathcal{T}$ sampled from a mini-batch:

$$\mathcal{L}_{tripl} = \frac{1}{b} \sum_{\substack{(a,p,n)\in\mathcal{T} \\ y_a=y_p\neq y_n}}^{b} [d_e(\phi_a, \phi_p) - d_e(\phi_a, \phi_n) + \gamma]_+ \tag{2}$$

with margin $\gamma = 0.2$, thus following recent implementations in e.g. Roth et al. (2019) or (Wu et al., 2017). Initial works (Schroff et al., 2015) using the triplet loss commonly utilized random or semihard triplet sampling and a GoogLeNet-based architecture. Recent methods typically employ the more effective distance-weighted sampling (Wu et al., 2017) and more powerful networks (Roth et al., 2019; Sanakoyeu et al., 2019). For completeness, we compare the triplet-loss performance combinde with random, semihard and distance-weighted sampling schemes introduced below.

Generalized Lifted Structure (Hermans et al., 2017) The Generalized Lifted Structure loss extends the standard lifted structure loss (Oh Song et al., 2016) to include all available anchor-positive and anchor-negative distance pairs within a mini-batch $\mathcal{B}$, instead of utilizing only a single anchor-positive combination:

$$\mathcal{L}_{genlift} = \sum_{a\in\mathcal{B}} \left[\log \sum_{p\in\mathcal{B}, y_a=y_p} \exp\left(d_e(\phi_a^*, \phi_p^*)\right) + \log \sum_{n\in\mathcal{B}, y_n\neq y_a} \exp\left(\gamma - d_e(\phi_a^*, \phi_n^*)\right) \right]_+ + \frac{\nu}{b} \cdot \sum_{a\in\mathcal{B}} \|\phi_a^*\|_2^2 \tag{3}$$

with mini-batch samples of a class c grouped into C, and sets of C contained in $\mathcal{C}$. ϕ^* denotes the non-normalized version of ϕ. The margin $\gamma = 1$ serves the standard purpose of avoiding over-distancing already correct image pairs. To account for increasing values, $\nu = 0.005$ regularizes the embeddings.

N-Pair (Sohn, 2016) N-Pair or N-Tuple losses extend the triplet formalism to incorporate all negatives in the mini-batch $\mathcal{B}$ by

$$\mathcal{L}_{npair} = \frac{1}{b} \sum_{\substack{(a,p)\in\mathcal{B}\\ y_a=y_p, a\neq p}} \log\left(1 + \sum_{\substack{n\in\mathcal{B}\\ y_a\neq y_n}} \exp\left(\phi_a^{*,T}\phi_n - \phi_a^{*,T}\phi_p^*\right)\right) + \frac{\nu}{b}\cdot\sum_{i\in\mathcal{B}}\|\phi_i^*\|_2^2 \tag{4}$$

with embedding regularization $\nu = 0.005$, as (Sohn, 2016) noted a slow convergence for normalized embeddings.

Angular (Wang et al., 2017) By introducing an angle-based penalty, the angular loss effectively introduces scale invariance and higher-order geometric constraints that are not explicitly introduced in normal contrastive losses:

$$\mathcal{L}_{ang} = \mathcal{L}_{npair}(\phi^*) + \frac{\lambda}{b} \sum_{\substack{(a,p)\in\mathcal{B}\\ y_a=y_p, a\neq p}} \left[\log(1 + \sum_{\substack{n\in\mathcal{B}\\ y_n\neq y_a}} \exp\left(4\tan^2\left(\alpha(\phi_a+\phi_p)^T\phi_n\right)\right) - 2\left(1+\tan^2(\alpha)\right)\phi_a^T\phi_n\right] \tag{5}$$

with angular margin α, which, as proposed in the original paper, is set to $\pi/4$. $\lambda = 2$ is the trade-off between standard ranking losses and the angular constraint. The N-Pair parameters are set as above.

Arcface (Deng et al., 2018) Arcface transforms the standard softmax formulation typically used in classification problem to retrieval-based problems by enforcing an angular margin between the embeddings ϕ and an approximate center $W \in \mathbb{R}^{c\times d}$ for each class, resulting in

$$\mathcal{L}_{arc} = -\frac{1}{b}\sum_{i\in\mathcal{B}} \log \frac{\exp(s\cdot\cos\left(W_{y_i}^T\phi_i + \gamma = 0.5\right))}{\exp\left(s\cdot\cos\left(W_{y_i}^T\phi_i + \gamma = 0.5\right)\right) + \sum_{\substack{j\in\mathcal{B}\\ y_i\neq y_j}} \exp\left(s\cdot\cos\left(W_{y_j}^T\phi_i\right)\right)} \tag{6}$$

Further, this training objective also introduces the additive angular margin penalty $\gamma = 0.5$ for increased inter-class discrepancy, while the scaling $s = 16$ denotes the radius of the effective utilized hypersphere $\mathbb{S}$. The class centers are optimized with learning rate 0.0005.

Histogram (Ustinova & Lempitsky, 2016) In contrast to many sample-based ranking objective functions, Histogram Loss learns to minimize the probability of a positive sample pair having a higher similarity score than a negative pair. Given a mini-batch $\mathcal{B}$, the sets of positive similarities $\mathcal{S}^+ = \{\phi_i^T\phi_j | (i,j) \in \mathcal{P}, y_i = y_j\}$ and negative similarities $\mathcal{S}^- = \{\phi_i^T\phi_j | (i,j) \in \mathcal{P}, y_i \neq y_j\}$, one optimises

$$\delta(s,r) = \frac{1}{\Delta}\left(\mathbb{I}_{s\in[t_{r-1},t_r]}\cdot(s - t_{r-1}) + \mathbb{I}_{s\in[t_{r-1},t_r]}\cdot(t_{r+1} - s)\right) \tag{7}$$

$$h^{+/-}(r) = \frac{1}{\|\mathcal{S}^{+/-}\|}\sum_{s\in\mathcal{S}^{+/-}}\delta(s,r) \tag{8}$$

$$\mathcal{L}_{hist} = \sum_{r\in R} h^-(r)\left(\sum_{q=1}^{r} h^+(q)\right) \tag{9}$$

resulting in soft, differentiable histogram assignments. The final objective $\mathcal{L}_{hist}$ then penalizes strong overlap between the probability of positive pairs having higher distance (i.e. its cumulative distribution to point r) than respective negative pairs. Such a histogram loss introduces a single hyperparameter, namely the degree of histogram discretisation R, which we set to 65 for CUB200-2011 and CARS196 and 11 for SOP in our study. In general, our implementation borrows from the original code base used in (Ustinova & Lempitsky, 2016).

Margin (Wu et al., 2017) Margin loss extends the standard triplet loss by introducing a dynamic, learnable boundary β between positive and negative pairs. This transfers the common triplet ranking problem to a relative ordering of pairs $\mathcal{P} = \{(i,j) | i,j \in \mathcal{B}, y_i \neq y_j\}$:

$$\mathcal{L}_{margin} = \sum_{(i,j)\in\mathcal{P}} \gamma + \mathbb{I}_{y_i=y_j}(d(\phi_i,\phi_j) - \beta) - \mathbb{I}_{y_i\neq y_j}(d_e(\phi_i,\phi_j) - \beta) \tag{10}$$

The learning rate of the boundary β is set to 0.0005, with initial value either 0.6 or 1.2 and triplet margin $\gamma = 0.2$. For our implementation, we utilise the distance-weighted triplet sampling method highlighted below.

MultiSimilarity (Wang et al., 2019a) Unlike contrastive and triplet based ranking methods, the MultiSimilarity loss concurrently evaluates similarities between anchor and negative, anchor and positive, as well as positive-positive and negative-negative pairs in relation to an anchor:

$$s_c^*(i,j) = \begin{cases} s_c(\phi_i,\phi_j) & s_c(\phi_i,\phi_j) > \min_{j\in\mathcal{P}_i} s_c(\phi_i,\phi_j) - \epsilon \\ s_c(\phi_i,\phi_j) & s_c(\phi_i,\phi_j) < \max_{k\in\mathcal{N}_i} s_c(\phi_i,\phi_k) + \epsilon \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

$$\mathcal{L}_{multisim} = \frac{1}{b}\sum_{i\in\mathcal{B}}\left[\frac{1}{\alpha}\log[1+\sum_{j\in\mathcal{P}_i}\exp(-\alpha(s_c^*(\phi_i,\phi_j)-\lambda))] + \frac{1}{\beta}\log[1+\sum_{k\in\mathcal{N}_i}\exp(\beta(s_c^*(\phi_i,\phi_k)-\lambda))]\right] \tag{12}$$

where $\mathcal{P}_x$ and $\mathcal{N}_x$ denote the set of positive and negative samples for a sample x, with cosine similarity $s_c(x,y) = x^T y$ for two normalized vectors $x, y \in \mathbb{R}^d$. For our hyperparameters, we use $\alpha = 2$, $\beta = 40$, $\lambda = 0.5$ and $\epsilon = 0.1$.

ProxyNCA (Movshovitz-Attias et al., 2017) The sampling complexity of tuples heavily affects the training convergence. ProxyNCA introduces a remedy by introducing class proxies, which act as approximations to entire classes. This way only an anchor is sampled and compared against the respective positive and negative class proxies. Utilizing one proxy $\psi_c \in \mathbb{R}^d$ per class $c \in \mathcal{C}$, ProxyNCA is then defined as

$$\mathcal{L}_{proxy} = -\frac{1}{b}\sum_{i\in\mathcal{B}}\log\left(\frac{\exp(-d_e(\phi_i,\psi_{y_i})}{\sum_{c\in\mathcal{C}\setminus\{y_i\}}\exp(-d(\phi_i,\psi_c)}\right) \tag{13}$$

Quadruplet (Chen et al., 2017) The quadruplet loss is an extension to the triplet loss, which introduces higher level ordering constraint on sample embeddings. By using an anchor, a positive and two exclusive negatives, the quadruplet criterion is defined as:

$$\mathcal{L}_{Quadr} = \sum_{\substack{i,j,k\in\mathcal{B}\\ y_i=y_j, y_j\neq y_k}} [d(\phi_i,\phi_j) - d(\phi_i,\phi_k) + \gamma_1]_+ + \sum_{\substack{i,j,k,l\in\mathcal{B}\\ y_i=y_j, y_j\neq y_k, y_l\neq y_k, y_l\neq y_j}} [d(\phi_i,\phi_k) - d(\phi_l,\phi_k) + \gamma_2]_+ \tag{14}$$

with margin parameters $\gamma_1 = 1$ and $\gamma_2 = 0.5$. We utilize distance-weighted sampling to propose the first negative sample k, which we found to work better than the quadruplet sampling scheme originally proposed in the paper.

SNR (Yuan et al., 2019) The Signal-to-Noise-Ratio loss (SNR) introduces a novel distance metric based on the ratio between anchor embedding variance and variance of noise, which is simply defined as the difference between anchor and compared embedding. This optimises the embedding space directly for informativeness. The complete loss can then be written as

$$\mathcal{L}_{SNR} = \sum_{i,j,k\in\mathcal{T}}\left[\frac{\sum_{m=1}^{D}(\phi_{i,m}-\phi_{j,m})}{\sum_{m=1}^{D}\phi_{i,m}^2} - \frac{\sum_{m=1}^{D}(\phi_{i,m}-\phi_{k,m})}{\sum_{m=1}^{D}\phi_{i,m}^2} + \gamma\right]_+ + \frac{\lambda}{b}\sum_{i\in\mathcal{B}}\left\|\sum_{m=1}^{D}\phi_{i,m}\right\| \tag{15}$$

with margin parameter $\gamma = 0.2$ and regularization $\lambda = 0.005$ to ensure zero-mean distributions. Note that $\phi_{i,m} = \phi(x_i)_m$.

SoftTriple (Qian et al., 2019) Similar to ProxyNCA, the SoftTriple objective function utilizes learnable data proxies to tackle the sampling problem. However, instead of class-discriminative proxies, a set of normalized intra-class proxies $\psi \in \Psi^c$ per class c are learned using the NCA-based similarity measure $\mathcal{S}_i^c$ of a sample i to all proxies of a class c. Denoting the set of available classes as $\mathcal{C}$ and the total set of proxies as Ψ, we get

$$\mathcal{S}_i^c = \sum_{\psi\in\Psi^c} \frac{\exp(\frac{1}{\gamma}\phi_i^T\psi)}{\sum_{\psi\in\Psi^c}\exp(\frac{1}{\gamma}\phi_i^T\psi)} \tag{16}$$

$$\mathcal{L}_{STBase} = -\frac{1}{b}\sum_{i\in\mathcal{B}}\log\frac{\exp(\lambda(\mathcal{S}_i^{y_i}-\delta))}{\exp(\lambda(\mathcal{S}_i^{y_i}-\delta))+\sum_{y\in\mathcal{Y}\setminus\{y_i\}}\exp(\lambda\mathcal{S}_i^y)} \tag{17}$$

$$\mathcal{L}_{SoftTriple} = \mathcal{L}_{STBase} + \tau\cdot\frac{\sum_{c\in\mathcal{C}}\sum_{\psi_1,\psi_2\in\Psi^c,\psi_1\neq\psi_2}\sqrt{2-2\psi_1^T\psi_2}}{|\mathcal{C}|\cdot|\Psi|\cdot(|\Psi|-1)} \tag{18}$$

The second term denotes a regularization on the learned proxies to ensure sparseness in the class set of proxies. For our tests, we utilised the following hyperparameter values (borrowing from the official implementation in (Qian et al., 2019)): $\tau = 0.2$, $\lambda = 8$, $\delta = 0.01$, $\gamma = 0.1$ and the number of proxies per class $|\Psi^c| = 2$ (higher values resulted in much worse performance). The proxy learning rate is set to 0.00001.

Normalized Softmax (Zhai & Wu, 2018) Similar to other classification-based losses in DML that are based on reformulations of the standard softmax function (such as $\mathcal{L}_{arc}$), the normalized softmax loss is optimized by comparing input embeddings ϕ_i to class proxies $\psi\in\mathbb{R}^D$ per class $c\in\mathcal{C}$:

$$\mathcal{L}_{NormSoft} = -\sum_{i\in\mathcal{B}}\log\left(\frac{\exp(\frac{\phi_i^T\psi_{y_i}}{T})}{\sum_{c\in\mathcal{C}\setminus\{y_i\}}\exp(\frac{\phi_i^T\psi_c}{T})}\right) \tag{19}$$

with temperature $T = 0.05$ for gradient boosting and class proxy learning rate set to 10^{-5}.

A.2. Tuple Mining

Basic contrastive, triplet or higher order ranking losses commonly need to mine their training tuples from the available mini-batch. In our study, we measure the influence of tuple sampling on the standard triplet loss, while utilising Distance-Weighted Mining for all ranking-based objective functions except N-Pair based methods.

Random Tuple Mining (Hu et al., 2014) The trivial way involves the random sampling of tuples. Simply put, per sample $\{x_i\}_{i\in\mathcal{B}}$ we select a respective positive $\{j|y_j = y_i, i\neq j, j\in\mathcal{B}\}$ or negative sample $\{k|y_j\neq y_i, i\neq k, k\in\mathcal{B}\}$.

Semihard Triplet Mining (Schroff et al., 2015) The potential number of triplets scales cubic in training set size. During learning, more and more of those triplets are correctly ordered and effectively provide no training signal (Schroff et al., 2015), thus impairing the remaining training process. To alleviate this, negative samples are carefully selected based on the anchor-positive sample distance (which are sampled at random). Given an anchor embedding ϕ_a and its positive ϕ_p, the negative is sampled randomly from the set

$$\phi_n \in \{\phi_n | n\in\mathcal{B}, y_n\neq y_a, \|\phi_a-\phi_p\|_2^2 < \|\phi_a-\phi_n\|_2^2\}. \tag{20}$$

This way, only negatives are considered which are reasonably hard to separate from an anchor. Moreover, this mining strategy avoids the sampling of overlay hard negatives, which often correspond to data noise and potentially lead to model collapses and bad local minima (Schroff et al., 2015).

Softhard Triplet Mining (Roth & Brattoli, 2019) While it was justifiably noted in (Schroff et al., 2015) that a selection of 'hard' samples hurts training, (Roth & Brattoli, 2019) show that a probabilistic (soft) selection of potentially hard candidates can actually benefit performance. Given an anchor embedding ϕ_a, positive ϕ_p and ϕ_n are randomly selected from

$$\phi_n \in \{\phi_n | n\in\mathcal{B}, y_n\neq y_a, \|\phi_a-\phi_n\|_2^2 < \underset{p\in\mathcal{B}, y_a=y_p}{\arg\max}\ \|\phi_a-\phi_p\|_2^2\} \tag{21}$$

and

$$\phi_a \in \{\phi_a | a\in\mathcal{B}, y_n = y_a, \|\phi_a-\phi_n\|_2^2 > \underset{n\in\mathcal{B}, y_a\neq y_p}{\arg\min}\ \|\phi_a-\phi_n\|_2^2\}. \tag{22}$$

Doing so provides a selection of 'hard' positivies and negatives. This reduces the risk of potential model collapses and bad local minima (as noted in Schroff et al. (2015)).

Distance-Weighted Tuple Mining (Wu et al., 2017) In DML, the embedding spaces are typically normalized to a D-dimensional (unit) hypersphere $\mathbb{S}^{D-1}$ for regularisation purposes (Wu et al., 2017). The analytical distribution of pairwise distances on a hypersphere follows

$$q(d_e(\phi_i, \phi_j)) \propto d_e(\phi_i, \phi_j)^{D-2}[1 - \frac{1}{4}d_e(\phi_i, \phi_j)]^{\frac{D-3}{2}} \tag{23}$$

for arbitrary embedding pairs $\phi_i, \phi_j \in \mathbb{S}^{D-1}$. In order to sample negatives from the whole range of possible distances to an anchor, Wu et al. (2017) propose to sample negatives based on a distance distribution inverse to q, i.e.

$$P(n|a) \propto \min(\lambda, q^{-1}(d_e(\phi_a, \phi_n)) \tag{24}$$

We set $\lambda = 0.5$ and limit the distances to 1.4.

A.3. Evaluation Metrics

In this section, we examine the evaluation metrics to measure the performance of the studied models on a the testset $\mathcal{X}_\text{test}$.

Recall@k (Jegou et al., 2011) Let

$$\mathcal{F}_q^k = \underset{\mathcal{F}\subset\mathcal{X}_\text{test}, |\mathcal{F}|=k}{\arg\min} \sum_{x_f\in\mathcal{F}} d_e(\phi(x_q), \phi(x_f)) \tag{25}$$

be the set of the first k nearest neighbours of a sample x_p, then we measure Recall@k as

$$R@k = \frac{1}{|\mathcal{X}_\text{test}|} \sum_{x_q\in\mathcal{X}_\text{test}} \begin{cases} 1 & \exists\, x_i \in \mathcal{F}_q^k \text{ s.t. } y_i = y_q \\ 0 & \text{otherwise} \end{cases} \tag{26}$$

which measures the average number of cases in which for a given query x_q there is at least one sample among its top k nearest neighbours x_i with the same class, i.e. $y_i = y_q$.

Normalized Mutual Information (NMI) (Manning et al., 2010) To measure the clustering quality using NMI, we embed all samples $x_i \in \mathcal{X}_\text{test}$ to obtain $\Phi_{\mathcal{X}_\text{test}}$ and perform a clustering (e.g. K-Means (Lloyd, 1982)). Following, we assign all samples x_i a cluster label w_i indicating the closest cluster center and define $\Omega = \{\omega_k\}_{k=1}^K$ with $\omega_k = \{i|w_i = k\}$ and $K = |\mathcal{C}|$ being the number of classes and clusters. Similarly for the true labels y_i we define $\Upsilon = \{\upsilon_c\}_{c=1}^K$ with $\upsilon_c = \{i|y_i = c\}$. The normalized mutual information is then computed as

$$NMI(\Omega, \Upsilon) = \frac{I(\Omega, \Upsilon)}{2(H(\Omega) + H(\Upsilon)} \tag{27}$$

with mutual Information $I(\cdot, \cdot)$ between cluster and labels, and entropy $H(\cdot, \cdot)$ on the clusters and labels respectively.

F1-Score (Sohn, 2016) The F1-score measures the harmonic mean between precision and recall and is a commonly used retrieval metric, placing equal importance to both precision and recall. It is defined as

$$F1 = \frac{2PR}{P + R} \tag{28}$$

with precision P and Recall R defined over nearest neighbour retrieval as done for Recall@k.

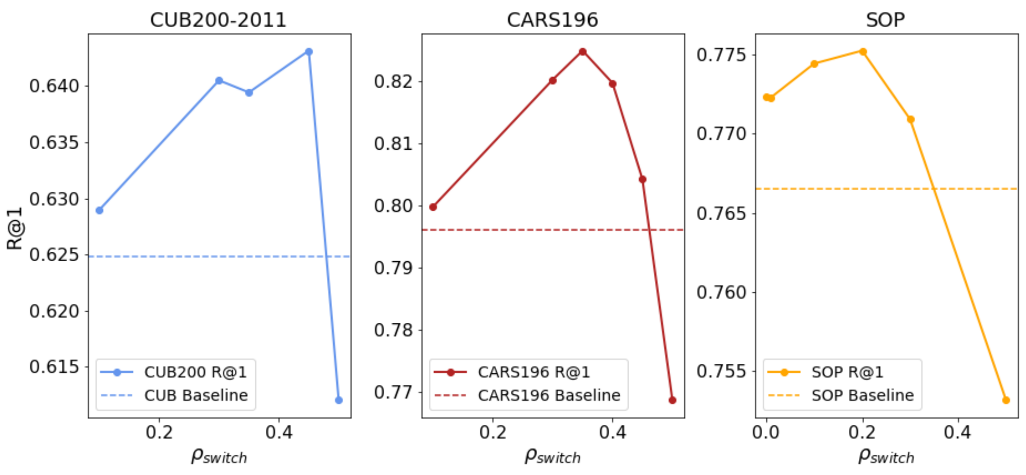


Figure 8. Analysis of the influence of p_{switch} on Recall@1 using margin loss with $\beta = 0.6$. Dashed lines denote performance without ρ-regularization.

Mean Average Precision measured on Recall (mAP) @C and @1000: The mAP-score measured on recall follows the same definition as standard mAP, however the recalled samples are determined by the nearest neighbour ranking. In our case, we propose to use mAP@C and mAP@C. mAP@C is equivalent to the mean over the class-wise average precision@k_c with k_c being the number of samples with label $c \in \mathcal{C}$, which only recalls as many samples as there are members in a class. For completeness, we also use mAP@1000, which recalls the 1000 nearest samples. With $\mathcal{F}_q^{k_c}$ defined as in eq. 25, this gives

$$\text{mAP@C} = \frac{1}{|\mathcal{X}_{\text{test}}|} \sum_{c\in\mathcal{C}} \sum_{x_q \in \mathcal{X}_{\text{test}} \wedge y_q = c} \frac{\left|\{x_i \in \mathcal{F}_q^{k_c} | y_i = y_q\}\right|}{k_c} \tag{29}$$

$$\text{mAP@1000} = \frac{1}{|\mathcal{X}_{\text{test}}|} \sum_{c\in\mathcal{C}} \sum_{x_q \in \mathcal{X}_{\text{test}} \wedge y_q = c} \frac{\left|\{x_i \in \mathcal{F}_q^{1000} | y_i = y_q\}\right|}{1000} \tag{30}$$

B. Evaluating the robustness of p_{switch} choices

For all benchmarks, we evaluate the change in performance for varying values of p_{switch}. As can be seen in figure 8, performance steadily increases up to a point of saturation, where the regulatory signal overshadows the discriminative one. In addition, its clear that performance increases are reasonably robust against changes in p_{switch}.

C. Visual comparison of DML objectives on benchmarks

This figure is the full version of the first page figure (Fig. 1). It qualitatively supports the saturation of performance noted in section 4 and table 2.

D. Correlation between performance and spectral decay ρ

Similar to Fig. 5 (rightmost) in the main paper, we now provide a more detailed illustration in Fig. 10 comparing the performance of the training objectives and their corresponding spectral decay $\rho(\Phi)$. For ranking losses, we further include the results using ρ-regularization while training, which further shows that in each case a gain in performance is related to a decrease of $\rho(\Phi)$. Especially the contrastive loss (Hadsell et al., 2006) greatly profits from our proposed regularization, as also indicated by the analysis of the singular value spectra (cf. Fig. 8 of main paper). Its large gains, more then 5% on the CARS196 dataset, is well explained by comparison of its training objective with those of triplet-based formulations. The latter optimizes over relative positive $(d_\phi(x_a, x_p)))$ and negative distances $(d_\phi(x_a, x_n))$ up to a fixed margin γ, which counteracts a compression of the embedding space to a certain extend. On the other hand, the constrastive loss, while controlling only the negative distances by γ, is able to perform an unconstrained contraction of entire classes, which

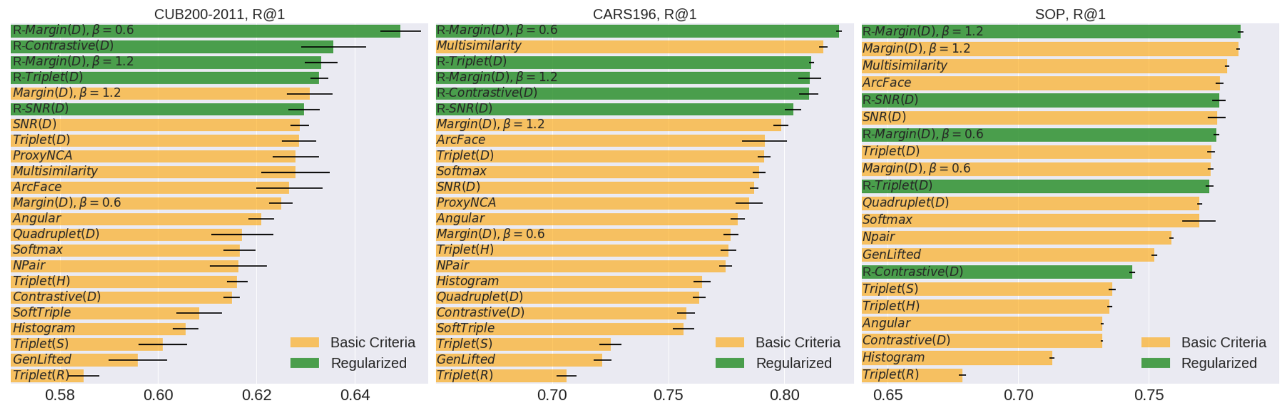


Figure 9. Mean recall performance and standard deviation of various DML objective functions trained with (green) and without (orange) our proposed regularization.

Approach	Architecture	Dim	R@1	R@10	R@100	NMI
DVML(Lin et al., 2018)	GoogLeNet	512	70.2	85.2	93.8	90.8
HTL(Ge, 2018)	Inception-BN	512	74.8	88.3	94.8	-
MIC(Roth et al., 2019)	ResNet50	128	77.2	89.4	95.6	90.0
D&C(Sanakoyeu et al., 2019)	ResNet50	128	75.9	88.4	94.9	90.2
Rank(Wang et al., 2019b)	Inception-BN	1536	**79.8**	**91.3**	**96.3**	90.4
ABE(Kim et al., 2018)	GoogLeNet	512	76.3	88.4	94.8	-
Margin (ours)(Wu et al., 2017)	ResNet50	128	78.4	-	-	90.4

Table 3. Comparison to the state-of-the-art DML methods on SOP(Oh Song et al., 2016). *Dim* denotes the dimensionality of ϕ_θ.

facilitates overly compressed embedding spaces Φ.

E. Analysis of per-class singular value spectra

In Sec. 5 of our main paper we analyze generalization in DML by considering the decay of the singular value spectrum over all embedded samples $\Phi_\mathcal{X}$. Thus, we analyze the general compression of the entire embedding space Φ as unseen test classes can be projected anywhere in Φ, in contrast to Verma et al. (2018) which conduct a class-conditioned analysis for i.i.d. classification problems. In order to show that the effect of ρ-regularization (as shown in Fig. 8 in main paper) is also reflected in the class-conditioned singular value spectrum, we perform SVD on Φ_{y_l} and subsequently average over all classes $y_l \in \mathcal{Y}$. Fig. 11 compares the sorted, first 35 singular values for both, models trained with and without ρ-regularization. We clearly see that the regularization decreases the average decay of singular values similar to the total singular value spectra shown in the main paper.

F. Comparison to state-of-the-art approaches on SOP dataset

In this section we provide a detailed comparison between current state-of-the-art DML approaches and our strongest baseline model, margin loss (D, $\beta = 1.2$) (Wu et al., 2017), on the SOP dataset in Tab. 6. The results for these approaches are taken from their public manuscripts. We observe that our baseline model outperforms each of the models using varying architectures, but especially other ResNet50-based implementations. While R50 proves to be a stronger base network (cf. Fig. 2 of main paper) than GoogLeNet based model, improvements over MIC and D&C using the same backbone by at least 0.9% and methods based on the similarly strong Inception-BN showcase the relevance of a well-defined baseline. Additionally, even though Rank and ABE employ considerable more powerful network ensembles, our carefully motivated baseline exhibits competitive performance.

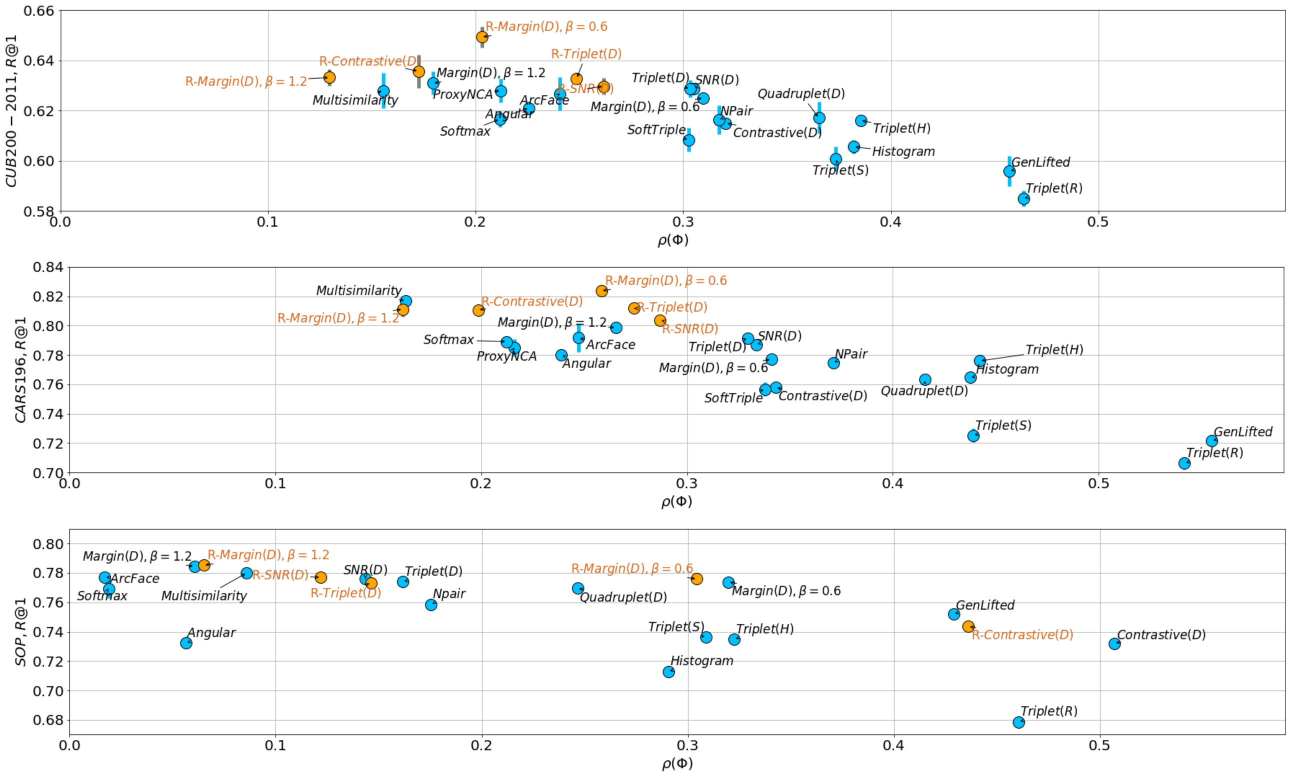


Figure 10. Relation between $\rho(\Phi)$ and generalization performance on Recall@1 for models trained with (orange) and without (blue) ρ-regularization. We report mean results and error-bars (gray). When error is small, bars are covered.

G. 2D Toy Examples

For our toy examples, we use a fully-connected network with two 30 neuron layers. Both input and embedding dimension are 2D, while the latter is normalized onto the unit circle. Each of the four training and test lines contain 15 samples taken from either the diagonal or vertical/horizontal line segments, respectively. We train the networks both with and without regularization for 200 iterations, a batchsize of 24 and learning rate of 0.03 using a standard contrastive loss (eq. 1) with margin $\gamma = 0.1$. For regularisation, we set $p_{\text{switch}} = 0.001$. Similar to Fig.6 in the main paper, Fig. 12 shows another 2D toy example based on vertical lines which again demonstrates the effect of compression and of our proposed ρ-regularization. The example consists of four training lines that are separable only by their x-coordinate and a test set of lines which are separable by their y-coordinate. As we observe, the test samples are collapsed onto a single point in the non-regularized embedding space, thus can not be distinguished. In contrast, the regularized representation allows us to separate the test classes and, further, exhibits a decreased decay in the singular value spectrum.

H. Influence of Manifold Mixup on DML

Now, we examine the effect of applying the regularization proposed in ManifoldMixup (Verma et al., 2018) on the DML transfer learning setting. As ManifoldMixup has been proposed to increase the compression of a learned representation in the context of standard supervised classification, it is expected to decrease the performance of DML models. For that, we train three different DML models on the CUB200-2011 dataset: (1) Normalized Softmax, (2) Triplet with Distance Sampling and (3) Margin loss with $\beta = 0.6$ and Distance Sampling. For (1), the implementation directly follows the standard implementation noted in Verma et al. (2018). For the ranking-based training objectives, we perform mixup in our ResNet50 and generate the mixed class labels, which consequently have either one (if image from the same class are mixed) or two entries (if images from different classes are mixed). Per (mixed) anchor embedding, this gives rise to up to two

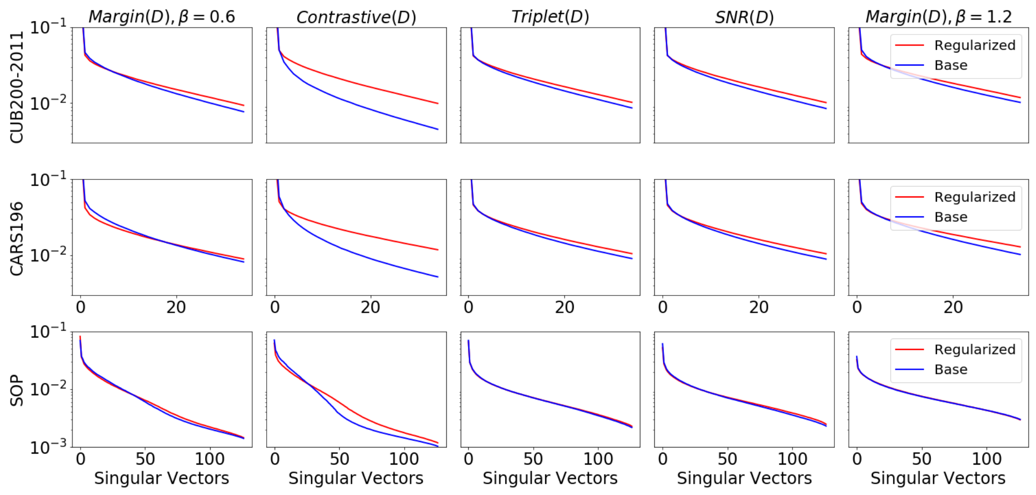


Figure 11. Averaged class-conditioned spectra of singular values for models trained with (red) and without (blue) ρ-regularization for various ranking-based loss functions.

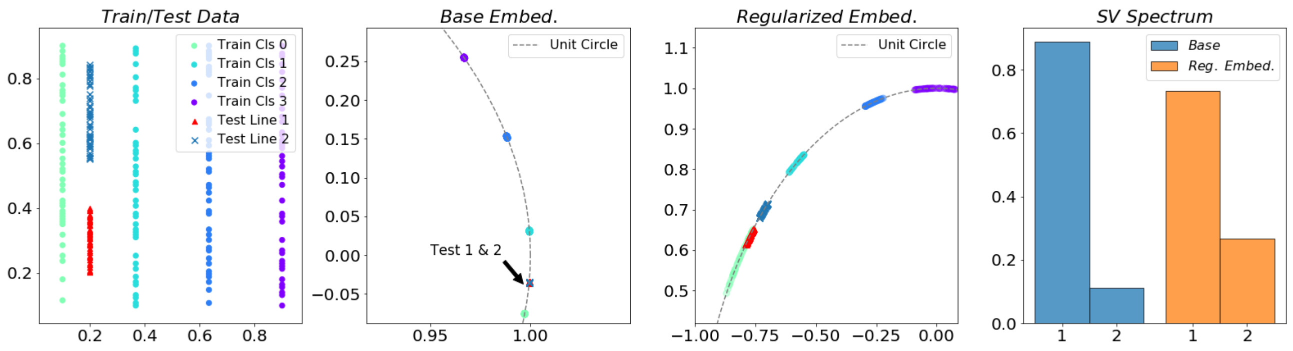


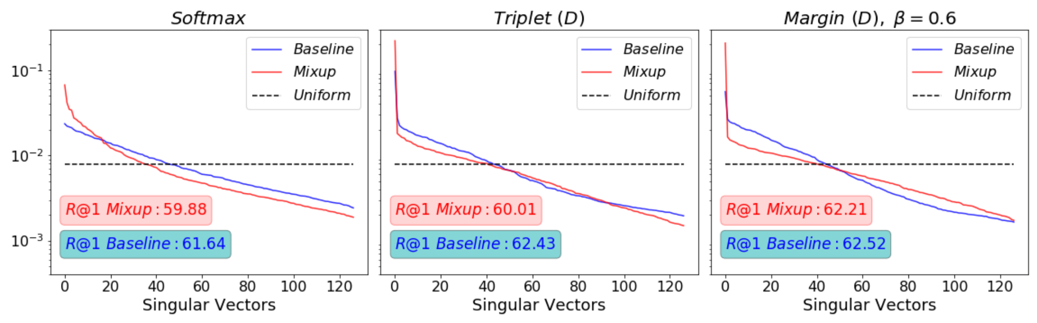
Figure 12. Toy example based on horizontally discriminative training data, where to goal is to generalize to vertically discriminative test data. (Leftmost) training and test data. (Mid-left) A small, normalized two-layer fully-connected network trained with standard contrastive loss fails to separate both test classes as it never has to utilize vertical discrimination. (Mid-right) The regularized embedding successfully separates the test classes by introducing additional features and decreasing the spectral decay. (Rightmost) Singular value spectra of training embeddings learned with and without regularization.

possible sets of triplets, for which we compute the loss and weigh it by the respective mixup coefficient λ_k:

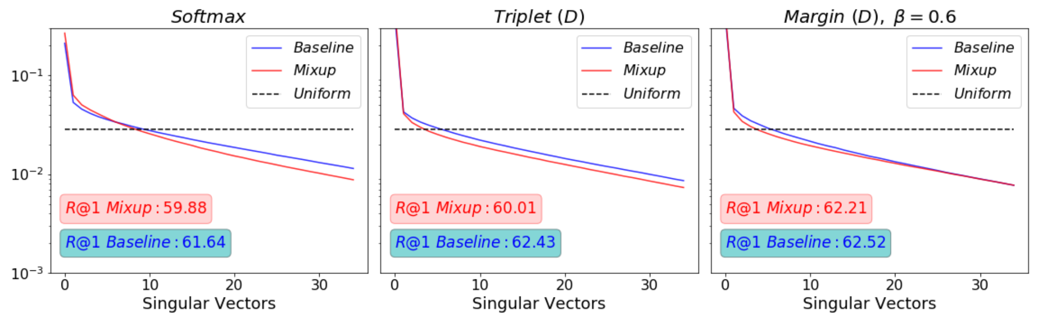
$$\mathcal{L}_{tripl}^{\text{Mix}} = \frac{1}{b}\sum_{k=1}^{2}\sum_{\substack{(a,p,n)\in\mathcal{T}^k \\ y_a=y_p\neq y_n}}^{b} \lambda_k \cdot \left[d_e(\phi_a^\lambda,\phi_p^\lambda) - d_e(\phi_a^\lambda,\phi_n^\lambda) + \gamma\right]_+ \tag{31}$$

where $\mathcal{T}^k$ denotes the set of triplets given the k-th mixup class-label entry and λ_k the respective interpolation value. We use the notation ϕ_x^λ to denote that we now operate on mixup embeddings. For training, we use the standard hyperparameters as described in Sec. 4.1. of the main paper and a mixup-α of 2 to sample the interpolation values $\lambda \sim \beta(\alpha,\alpha)$ (see Verma et al. (2018)).

The results after rerunning baselines and mixup-variants are shown in Fig. 13. As expected, applying ManifoldMixup leads to more compressed representations (indicated by a stronger spectral decay and lower $\rho(\Phi)$-scores) at the cost of reduced generalization performance. This holds both for the spectrum across the fully embedded dataset as well as on a class level.



(a) Singular Value Spectrum for all embeddings



(b) Singular Value Spectrum for per-class embeddings

Figure 13. Evaluation of Mixup Influence on zero-shot generalization under heavy distribution shift.

I. Detailed Results

This section contains detailed results per method and evaluation metric for the method comparisons (Tab. 2 in the main paper) and evaluation of batch-creation methods (Fig. 3 in main paper). The resp. tables for the method comparisons are Tab. 4 for CUB200-2011 (Wah et al., 2011), Tab. 5 for CARS196 (Krause et al., 2013) and Tab. 6 for Stanford Online Products (SOP) (Oh Song et al., 2016). In addition, the switch probability p_{switch} for each regularised method is noted as well. The batch-creation methods are evaluated in detail in Tab. 7 for CUB200-2011, Tab. 8 for CARS196 and Tab. 9 for SOP.

CUB200-2011(Wah et al., 2011)						
Approach	R@1	R@2	F1	mAP@C	mAP@1000	NMI
Imagenet	43.77	57.56	19.14	9.48	14.62	52.91
Angular	62.10 ± 0.27	73.68 ± 0.39	37.53 ± 0.13	22.06 ± 0.20	30.56 ± 0.27	67.59 ± 0.26
ArcFace	62.67 ± 0.67	74.38 ± 0.25	37.33 ± 0.51	23.05 ± 0.43	31.62 ± 0.68	67.66 ± 0.38
Contrastive (D)	61.50 ± 0.17	72.95 ± 0.32	35.40 ± 0.75	23.46 ± 0.18	32.38 ± 0.21	66.45 ± 0.27
GenLifted	59.59 ± 0.60	71.63 ± 0.38	34.86 ± 0.16	22.03 ± 0.14	30.30 ± 0.13	65.63 ± 0.14
Histogram	60.55 ± 0.26	72.08 ± 0.20	33.88 ± 0.56	22.65 ± 0.20	31.24 ± 0.27	65.26 ± 0.23
Multisimilarity	62.80 ± 0.70	74.37 ± 0.52	39.03 ± 0.63	22.58 ± 0.37	30.92 ± 0.49	$\mathbf{68.55 \pm 0.38}$
Margin (D, $\beta = 0.6$)	62.50 ± 0.24	74.15 ± 0.33	36.34 ± 0.61	23.83 ± 0.20	32.90 ± 0.29	67.02 ± 0.37
Margin (D, $\beta = 1.2$)	63.09 ± 0.46	74.41 ± 0.37	38.36 ± 0.66	23.61 ± 0.31	32.63 ± 0.40	68.21 ± 0.33
NPair	61.63 ± 0.58	73.33 ± 0.42	37.29 ± 0.42	22.16 ± 0.29	30.83 ± 0.30	67.64 ± 0.37
ProxyNCA	62.80 ± 0.48	74.03 ± 0.15	36.20 ± 0.73	23.94 ± 0.37	33.13 ± 0.59	66.93 ± 0.38
Quadruplet (D)	61.71 ± 0.63	73.26 ± 0.33	35.74 ± 0.62	23.20 ± 0.24	31.87 ± 0.29	66.60 ± 0.41
SNR (D)	62.88 ± 0.18	74.33 ± 0.26	36.91 ± 0.42	23.48 ± 0.14	32.24 ± 0.21	67.16 ± 0.25
SoftTriple	60.83 ± 0.47	71.61 ± 0.58	32.16 ± 0.50	22.43 ± 0.29	31.33 ± 0.29	64.27 ± 0.36
Softmax	61.66 ± 0.33	73.31 ± 0.39	35.94 ± 0.59	22.19 ± 0.20	30.67 ± 0.26	66.77 ± 0.36
Triplet (D)	62.87 ± 0.35	74.31 ± 0.28	37.30 ± 0.32	23.59 ± 0.12	32.64 ± 0.13	67.53 ± 0.14
Triplet (R)	58.48 ± 0.31	70.51 ± 0.24	31.95 ± 0.44	21.12 ± 0.10	29.08 ± 0.12	63.84 ± 0.30
Triplet (S)	60.09 ± 0.49	71.75 ± 0.27	34.46 ± 0.54	22.49 ± 0.26	31.40 ± 0.39	65.59 ± 0.29
Triplet (H)	61.61 ± 0.21	72.94 ± 0.34	35.10 ± 0.37	22.63 ± 0.23	31.22 ± 0.27	65.98 ± 0.41
R-Contrastive (D)	63.57 ± 0.66	74.57 ± 0.63	37.70 ± 0.53	23.54 ± 0.37	32.40 ± 0.66	67.63 ± 0.31
R-Margin (D, $\beta = 0.6$)	$\mathbf{64.93 \pm 0.42}$	$\mathbf{75.58 \pm 0.25}$	$\mathbf{38.93 \pm 0.54}$	$\mathbf{24.11 \pm 0.20}$	$\mathbf{33.38 \pm 0.27}$	68.36 ± 0.32
R-Margin (D, $\beta = 1.2$)	63.32 ± 0.33	74.80 ± 0.27	38.09 ± 0.97	22.80 ± 0.46	31.44 ± 0.67	67.91 ± 0.66
R-SNR (D)	62.97 ± 0.32	74.66 ± 0.06	38.25 ± 0.41	23.13 ± 0.23	32.23 ± 0.47	68.04 ± 0.34
R-Triplet (D)	63.28 ± 0.18	74.97 ± 0.28	38.03 ± 0.77	23.28 ± 0.28	32.36 ± 0.47	67.86 ± 0.51

Table 4. Comparison of DML setups for CUB200-2011. We report all relevant performance metrics. Training is done over 150 epochs.

CARS196(Krause et al., 2013)						
Approach	R@1	R@2	F1	mAP@C	mAP@1000	NMI
Imagenet	36.39	48.11	8.90	4.03	6.92	37.96
Angular	78.00 ± 0.32	85.97 ± 0.18	36.40 ± 0.75	22.18 ± 0.35	29.29 ± 0.37	66.48 ± 0.44
ArcFace	79.16 ± 0.97	87.02 ± 0.54	36.36 ± 1.97	23.44 ± 0.68	31.36 ± 0.99	66.99 ± 1.08
Contrastive (D)	75.78 ± 0.39	84.17 ± 0.27	33.10 ± 0.63	23.19 ± 0.33	30.43 ± 0.51	64.04 ± 0.13
GenLifted	72.17 ± 0.38	81.94 ± 0.30	32.46 ± 0.43	21.66 ± 0.24	28.66 ± 0.31	63.75 ± 0.35
Histogram	76.47 ± 0.38	84.50 ± 0.36	33.04 ± 0.67	23.21 ± 0.10	30.31 ± 0.14	64.15 ± 0.36
Multisimilarity	81.68 ± 0.19	88.86 ± 0.14	40.95 ± 0.72	24.22 ± 0.27	31.92 ± 0.44	$\mathbf{69.43 \pm 0.38}$
Margin (D, $\beta = 0.6$)	77.70 ± 0.32	85.67 ± 0.19	35.04 ± 0.49	24.08 ± 0.27	31.99 ± 0.29	65.29 ± 0.32
Margin (D, $\beta = 1.2$)	79.86 ± 0.33	87.46 ± 0.20	38.44 ± 0.64	24.72 ± 0.21	32.50 ± 0.28	67.36 ± 0.34
NPair	77.48 ± 0.28	85.73 ± 0.18	35.88 ± 0.40	23.15 ± 0.25	30.70 ± 0.14	66.55 ± 0.19
ProxyNCA	78.48 ± 0.58	86.20 ± 0.50	34.66 ± 0.48	23.82 ± 0.36	31.86 ± 0.34	65.76 ± 0.22
Quadruplet (D)	76.34 ± 0.27	84.67 ± 0.23	34.28 ± 0.88	23.49 ± 0.32	31.30 ± 0.48	64.79 ± 0.50
SNR (D)	78.69 ± 0.19	86.44 ± 0.22	35.88 ± 0.71	24.20 ± 0.43	32.17 ± 0.60	65.84 ± 0.52
SoftTriple	75.66 ± 0.46	83.72 ± 0.29	31.07 ± 0.56	22.90 ± 0.25	30.42 ± 0.31	62.66 ± 0.16
Softmax	78.91 ± 0.27	86.66 ± 0.23	35.51 ± 0.85	22.81 ± 0.14	29.84 ± 0.13	66.35 ± 0.30
Triplet (D)	79.13 ± 0.27	86.74 ± 0.17	35.89 ± 0.25	24.38 ± 0.18	32.26 ± 0.25	65.90 ± 0.18
Triplet (R)	70.63 ± 0.43	80.43 ± 0.26	29.02 ± 0.47	19.48 ± 0.20	25.97 ± 0.28	61.09 ± 0.27
Triplet (S)	72.51 ± 0.47	81.53 ± 0.29	31.61 ± 0.41	21.63 ± 0.26	28.32 ± 0.56	62.84 ± 0.41
Triplet (H)	77.60 ± 0.33	85.63 ± 0.27	34.71 ± 0.31	23.84 ± 0.13	31.31 ± 0.15	65.37 ± 0.26
R-Contrastive (D)	81.06 ± 0.41	88.06 ± 0.21	37.72 ± 0.84	24.55 ± 0.34	32.53 ± 0.43	67.27 ± 0.46
R-Margin (D, $\beta = 0.6$)	$\mathbf{82.37 \pm 0.13}$	$\mathbf{89.14 \pm 0.12}$	$\mathbf{39.28 \pm 0.41}$	$\mathbf{25.67 \pm 0.32}$	$\mathbf{34.57 \pm 0.30}$	68.66 ± 0.47
R-Margin (D, $\beta = 1.2$)	81.11 ± 0.49	88.20 ± 0.22	38.76 ± 0.94	24.17 ± 0.50	31.60 ± 0.79	67.72 ± 0.79
R-SNR (D)	80.38 ± 0.35	87.95 ± 0.37	38.62 ± 0.47	24.72 ± 0.15	33.02 ± 0.23	67.60 ± 0.20
R-Triplet (D)	81.17 ± 0.11	88.43 ± 0.18	38.72 ± 0.31	25.27 ± 0.22	33.18 ± 0.48	67.79 ± 0.23

Table 5. Comparison of DML setups for CARS196. We report all relevant performance metrics. Training is done over 150 epochs.

Stanford Online Products(Oh Song et al., 2016)						
Approach	R@1	R@2	F1	mAP@C	mAP@1000	NMI
Imagenet	48.65	53.82	11.97	17.47	18.58	58.64
Angular	73.22 ± 0.07	78.14 ± 0.06	34.20 ± 0.07	36.96 ± 0.07	40.76 ± 0.08	89.53 ± 0.01
ArcFace	77.71 ± 0.15	82.23 ± 0.09	37.15 ± 0.13	41.20 ± 0.11	45.76 ± 0.14	90.09 ± 0.03
Contrastive (D)	73.21 ± 0.04	77.87 ± 0.04	35.66 ± 0.14	37.43 ± 0.05	41.21 ± 0.06	89.78 ± 0.02
GenLifted	75.21 ± 0.12	80.25 ± 0.04	35.93 ± 0.09	39.03 ± 0.09	43.06 ± 0.14	89.84 ± 0.01
Histogram	71.30 ± 0.10	76.18 ± 0.08	31.58 ± 0.14	34.88 ± 0.10	38.55 ± 0.13	88.93 ± 0.02
Multisimilarity	77.99 ± 0.09	82.64 ± 0.08	36.75 ± 0.18	41.52 ± 0.07	46.23 ± 0.08	90.00 ± 0.02
Margin (D, $\beta = 0.6$)	77.38 ± 0.11	81.78 ± 0.12	$\mathbf{39.04 \pm 0.16}$	41.69 ± 0.14	45.92 ± 0.17	$\mathbf{90.45 \pm 0.03}$
Margin (D, $\beta = 1.2$)	78.43 ± 0.07	82.83 ± 0.09	38.63 ± 0.18	42.43 ± 0.12	46.90 ± 0.16	90.40 ± 0.03
Npair	75.86 ± 0.08	80.73 ± 0.06	35.40 ± 0.15	39.09 ± 0.10	43.22 ± 0.11	89.79 ± 0.03
Quadruplet (D)	76.95 ± 0.10	81.54 ± 0.05	37.43 ± 0.14	40.82 ± 0.13	45.04 ± 0.12	90.14 ± 0.02
SNR (D)	77.61 ± 0.34	82.34 ± 0.31	37.17 ± 0.37	41.47 ± 0.38	46.03 ± 0.47	90.10 ± 0.08
Softmax	76.92 ± 0.64	81.34 ± 0.63	36.01 ± 0.71	40.23 ± 0.78	44.29 ± 0.78	89.82 ± 0.15
Triplet (D)	77.39 ± 0.15	82.03 ± 0.08	36.98 ± 0.11	41.02 ± 0.12	45.64 ± 0.14	90.06 ± 0.02
Triplet (R)	67.86 ± 0.14	73.02 ± 0.11	28.98 ± 0.12	31.92 ± 0.17	35.46 ± 0.20	88.35 ± 0.04
Triplet (S)	73.61 ± 0.14	78.36 ± 0.14	33.65 ± 0.13	37.11 ± 0.06	41.24 ± 0.06	89.35 ± 0.02
Triplet (H)	73.50 ± 0.09	78.38 ± 0.06	33.01 ± 0.20	36.85 ± 0.05	40.64 ± 0.07	89.25 ± 0.03
R-Contrastive (D)	74.36 ± 0.11	78.85 ± 0.11	36.39 ± 0.07	38.45 ± 0.14	42.60 ± 0.14	89.94 ± 0.02
R-Margin (D, $\beta = 0.6$)	77.58 ± 0.11	81.93 ± 0.10	38.87 ± 0.19	41.74 ± 0.12	46.02 ± 0.12	90.42 ± 0.03
R-Margin (D, $\beta = 1.2$)	$\mathbf{78.52 \pm 0.10}$	$\mathbf{82.95 \pm 0.07}$	38.36 ± 0.16	$\mathbf{42.55 \pm 0.10}$	$\mathbf{46.77 \pm 0.11}$	90.33 ± 0.02
R-SNR (D)	77.69 ± 0.25	82.46 ± 0.17	36.78 ± 0.36	41.44 ± 0.31	45.74 ± 0.38	90.02 ± 0.06
R-Triplet (D)	77.33 ± 0.14	82.01 ± 0.12	36.63 ± 0.20	40.91 ± 0.12	45.57 ± 0.14	89.98 ± 0.04

Table 6. Comparison of DML setups for Stanford Online Products. We report all relevant performance metrics.. Training is done over 100 epochs.

CUB200-2011(Wah et al., 2011)					
Approach	R@1	R@2	F1	mAP@C	NMI
Histogram, SPC-2	57.92 ± 0.26	69.74 ± 0.03	31.21 ± 0.18	21.27 ± 0.28	63.26 ± 0.18
Histogram, SPC-4	57.88 ± 0.35	70.18 ± 0.28	31.15 ± 0.17	21.39 ± 0.26	63.37 ± 0.12
Histogram, SPC-8	57.87 ± 0.40	70.09 ± 0.41	31.72 ± 0.09	21.56 ± 0.07	63.51 ± 0.13
Histogram, DDM	58.22 ± 0.33	70.61 ± 0.26	32.08 ± 0.21	21.69 ± 0.26	63.55 ± 0.20
Histogram, GC	57.51 ± 0.40	69.64 ± 0.25	30.98 ± 0.43	21.16 ± 0.25	63.08 ± 0.18
Histogram, SPC-R	57.62 ± 0.11	69.37 ± 0.22	30.82 ± 0.29	21.03 ± 0.12	63.03 ± 0.28
Histogram, FRD	58.36 ± 0.19	70.79 ± 0.31	32.12 ± 0.29	21.67 ± 0.21	63.81 ± 0.24
Margin (D), SPC-2	62.66 ± 0.28	73.98 ± 0.08	37.77 ± 0.51	23.40 ± 0.22	67.76 ± 0.19
Margin (D), SPC-4	62.37 ± 0.25	74.05 ± 0.23	37.84 ± 0.30	23.34 ± 0.16	67.90 ± 0.15
Margin (D), SPC-8	62.04 ± 0.12	73.87 ± 0.22	37.03 ± 0.44	23.21 ± 0.29	67.36 ± 0.20
Margin (D), DDM	62.50 ± 0.23	74.31 ± 0.24	37.90 ± 0.41	23.32 ± 0.19	68.00 ± 0.25
Margin (D), GC	62.61 ± 0.26	74.29 ± 0.27	37.84 ± 0.83	23.43 ± 0.23	67.81 ± 0.46
Margin (D), SPC-R	62.36 ± 0.39	74.14 ± 0.36	37.62 ± 0.55	23.23 ± 0.28	67.47 ± 0.25
Margin (D), FRD	62.64 ± 0.34	74.08 ± 0.34	38.11 ± 0.69	23.37 ± 0.21	67.87 ± 0.33
MultiSimilarity, SPC-2	62.46 ± 0.25	74.13 ± 0.13	38.61 ± 0.42	22.26 ± 0.14	68.00 ± 0.20
MultiSimilarity, SPC-4	62.95 ± 0.12	74.61 ± 0.02	38.39 ± 0.37	22.66 ± 0.12	68.29 ± 0.22
MultiSimilarity, SPC-8	62.73 ± 0.19	74.22 ± 0.05	37.18 ± 0.07	22.82 ± 0.12	67.46 ± 0.07
MultiSimilarity, DDM	62.57 ± 0.31	74.49 ± 0.25	38.58 ± 0.70	22.31 ± 0.16	68.25 ± 0.32
MultiSimilarity, GC	62.65 ± 0.24	74.21 ± 0.28	38.79 ± 0.20	22.35 ± 0.11	68.08 ± 0.29
MultiSimilarity, SPC-R	62.36 ± 0.32	74.10 ± 0.33	37.99 ± 0.39	22.32 ± 0.22	68.01 ± 0.20
MultiSimilarity, FRD	63.19 ± 0.31	74.88 ± 0.27	38.70 ± 0.57	23.02 ± 0.22	68.24 ± 0.26
NPair, SPC-2	60.52 ± 0.88	73.12 ± 0.86	36.51 ± 0.55	22.12 ± 0.23	66.79 ± 0.55
NPair, SPC-4	59.80 ± 0.20	71.42 ± 0.29	34.23 ± 0.55	21.59 ± 0.17	65.31 ± 0.41
NPair, SPC-8	58.22 ± 0.07	70.29 ± 0.02	33.07 ± 0.38	20.84 ± 0.09	64.29 ± 0.11
NPair, DDM	60.13 ± 0.05	72.03 ± 0.15	35.24 ± 0.42	21.69 ± 0.01	66.05 ± 0.32
NPair, GC	60.85 ± 0.44	72.90 ± 0.52	36.13 ± 0.68	22.12 ± 0.16	66.71 ± 0.40
NPair, SPC-R	61.32 ± 0.07	73.08 ± 0.26	36.45 ± 0.19	22.28 ± 0.17	66.87 ± 0.25
NPair, FRD	61.23 ± 0.15	73.01 ± 0.17	36.26 ± 0.26	22.34 ± 0.17	67.04 ± 0.22
ProxyNCA, SPC-2	62.67 ± 0.43	73.96 ± 0.36	35.66 ± 0.26	23.64 ± 0.52	66.88 ± 0.29
ProxyNCA, SPC-4	62.50 ± 0.48	73.64 ± 0.47	35.46 ± 0.62	23.50 ± 0.44	66.59 ± 0.32
ProxyNCA, SPC-8	62.49 ± 0.39	74.07 ± 0.31	35.44 ± 0.60	23.90 ± 0.54	66.56 ± 0.32
ProxyNCA, DDM	62.63 ± 0.00	73.68 ± 0.00	36.35 ± 0.00	24.50 ± 0.00	67.08 ± 0.00
ProxyNCA, GC	62.97 ± 0.53	74.03 ± 0.50	36.67 ± 0.96	24.17 ± 0.43	67.15 ± 0.51
ProxyNCA, SPC-R	62.99 ± 0.84	74.07 ± 0.42	36.61 ± 0.77	23.96 ± 0.39	67.26 ± 0.78
ProxyNCA, FRD	63.12 ± 0.51	74.36 ± 0.31	37.37 ± 0.58	24.42 ± 0.30	67.54 ± 0.46
Softmax, SPC-2	61.51 ± 0.28	73.29 ± 0.23	35.36 ± 0.55	22.02 ± 0.07	66.43 ± 0.30
Softmax, SPC-4	61.55 ± 0.50	73.51 ± 0.22	35.72 ± 0.12	22.08 ± 0.15	66.63 ± 0.20
Softmax, SPC-8	61.55 ± 0.49	73.29 ± 0.24	35.35 ± 0.34	22.16 ± 0.03	66.22 ± 0.30
Softmax, DDM	61.72 ± 0.78	72.99 ± 0.30	36.65 ± 1.02	22.67 ± 0.35	66.79 ± 0.29
Softmax, GC	61.32 ± 0.43	72.84 ± 0.29	36.58 ± 0.46	22.51 ± 0.32	66.83 ± 0.11
Softmax, SPC-R	61.58 ± 0.23	73.30 ± 0.23	35.38 ± 0.49	21.89 ± 0.13	66.46 ± 0.32
Softmax, FRD	61.52 ± 0.69	72.75 ± 0.63	35.98 ± 0.89	22.23 ± 0.37	66.48 ± 0.36
Triplet (R), SPC-2	58.44 ± 0.89	70.42 ± 0.41	31.94 ± 0.57	20.72 ± 0.12	63.98 ± 0.22
Triplet (R), SPC-4	58.67 ± 0.43	70.79 ± 0.32	32.18 ± 0.45	20.86 ± 0.25	64.24 ± 0.35
Triplet (R), SPC-8	58.04 ± 0.23	70.22 ± 0.29	32.26 ± 0.18	20.67 ± 0.15	63.74 ± 0.09
Triplet (R), DDM	58.08 ± 0.45	70.00 ± 0.15	31.58 ± 0.33	20.65 ± 0.07	63.56 ± 0.08
Triplet (R), GC	58.42 ± 0.13	70.26 ± 0.13	31.78 ± 0.17	20.70 ± 0.07	63.96 ± 0.13
Triplet (R), SPC-R	58.33 ± 0.38	70.50 ± 0.28	31.32 ± 0.23	20.91 ± 0.10	63.50 ± 0.17
Triplet (R), FRD	58.00 ± 0.00	69.95 ± 0.15	31.42 ± 0.11	20.33 ± 0.19	63.46 ± 0.01

Table 7. CUB200-2011: Comparison of Batch-Sampling methods for various loss functions and sampling methods.

CARS(Krause et al., 2013)					
Approach	R@1	R@2	F1	mAP@C	NMI
Histogram, SPC-2	67.24 ± 1.04	77.37 ± 0.85	28.93 ± 0.65	19.89 ± 0.49	60.82 ± 0.59
Histogram, SPC-4	67.40 ± 0.53	77.53 ± 0.43	28.54 ± 0.74	19.85 ± 0.24	60.97 ± 0.58
Histogram, SPC-8	67.53 ± 0.77	77.69 ± 0.56	29.14 ± 0.76	20.04 ± 0.35	61.22 ± 0.38
Histogram, DDM	66.57 ± 0.49	76.93 ± 0.53	27.99 ± 0.41	19.55 ± 0.28	60.46 ± 0.38
Histogram, GC	66.36 ± 0.76	76.88 ± 0.57	27.70 ± 0.55	19.04 ± 0.47	60.40 ± 0.45
Histogram, SPC-R	64.34 ± 0.65	75.43 ± 0.49	27.07 ± 0.75	18.37 ± 0.26	59.90 ± 0.75
Histogram, FRD	67.06 ± 0.18	77.18 ± 0.22	28.19 ± 0.81	19.65 ± 0.22	60.31 ± 0.29
Margin (D), SPC-2	79.79 ± 0.40	87.27 ± 0.36	38.78 ± 0.48	24.84 ± 0.28	67.59 ± 0.24
Margin (D), SPC-4	79.73 ± 0.08	87.18 ± 0.11	38.29 ± 0.41	25.13 ± 0.21	67.48 ± 0.46
Margin (D), SPC-8	78.93 ± 0.23	86.71 ± 0.18	37.18 ± 0.29	24.51 ± 0.37	66.83 ± 0.24
Margin (D), DDM	80.13 ± 0.38	87.53 ± 0.12	38.26 ± 0.24	24.65 ± 0.12	67.32 ± 0.26
Margin (D), GC	80.22 ± 0.16	87.44 ± 0.03	37.91 ± 0.71	24.82 ± 0.34	67.14 ± 0.39
Margin (D), SPC-R	80.06 ± 0.48	87.37 ± 0.30	38.17 ± 1.01	24.54 ± 0.21	67.26 ± 0.37
Margin (D), FRD	80.23 ± 0.20	87.73 ± 0.10	38.59 ± 0.59	25.18 ± 0.12	67.56 ± 0.21
MultiSimilarity, SPC-2	81.59 ± 0.18	88.92 ± 0.07	40.79 ± 0.69	24.35 ± 0.25	69.63 ± 0.50
MultiSimilarity, SPC-4	81.78 ± 0.13	88.97 ± 0.13	41.02 ± 0.23	25.15 ± 0.16	69.47 ± 0.12
MultiSimilarity, SPC-8	81.32 ± 0.05	88.28 ± 0.10	39.09 ± 0.71	25.54 ± 0.23	68.36 ± 0.42
MultiSimilarity, DDM	81.77 ± 0.29	88.77 ± 0.24	40.94 ± 0.32	23.80 ± 0.06	69.26 ± 0.24
MultiSimilarity, GC	81.63 ± 0.11	88.72 ± 0.22	40.38 ± 0.21	24.27 ± 0.27	69.36 ± 0.22
MultiSimilarity, SPC-R	81.52 ± 0.16	88.74 ± 0.16	40.66 ± 0.31	24.67 ± 0.17	69.63 ± 0.22
MultiSimilarity, FRD	81.70 ± 0.18	88.96 ± 0.19	41.74 ± 0.39	24.25 ± 0.10	69.56 ± 0.17
NPair, SPC-2	76.35 ± 0.23	84.79 ± 0.17	35.72 ± 0.49	23.45 ± 0.10	66.22 ± 0.09
NPair, SPC-4	73.57 ± 0.15	82.76 ± 0.20	33.94 ± 0.21	22.83 ± 0.15	64.96 ± 0.18
NPair, SPC-8	71.97 ± 0.35	81.98 ± 0.30	32.69 ± 0.40	22.57 ± 0.15	63.99 ± 0.22
NPair, DDM	76.02 ± 0.32	84.47 ± 0.03	35.35 ± 0.07	23.40 ± 0.03	66.11 ± 0.11
NPair, GC	76.09 ± 0.11	84.64 ± 0.24	35.03 ± 0.26	23.23 ± 0.19	65.83 ± 0.15
NPair, SPC-R	75.79 ± 0.09	84.64 ± 0.12	35.14 ± 0.44	23.07 ± 0.13	65.91 ± 0.22
NPair, FRD	75.83 ± 0.49	84.49 ± 0.25	35.65 ± 0.70	23.32 ± 0.36	66.08 ± 0.53
ProxyNCA, SPC-2	78.48 ± 0.61	85.97 ± 0.39	34.82 ± 0.58	23.85 ± 0.38	65.74 ± 0.15
ProxyNCA, SPC-4	78.48 ± 0.61	85.94 ± 0.25	34.90 ± 0.57	23.77 ± 0.20	65.55 ± 0.44
ProxyNCA, SPC-8	78.08 ± 0.20	85.84 ± 0.28	33.35 ± 1.17	23.30 ± 0.25	65.26 ± 0.62
ProxyNCA, DDM	78.43 ± 0.30	86.30 ± 0.26	34.72 ± 0.65	23.62 ± 0.20	65.84 ± 0.32
ProxyNCA, GC	78.14 ± 0.55	85.92 ± 0.42	34.72 ± 0.34	23.43 ± 0.23	65.60 ± 0.28
ProxyNCA, SPC-R	78.45 ± 0.23	86.19 ± 0.21	35.18 ± 0.75	23.91 ± 0.19	66.19 ± 0.39
ProxyNCA, FRD	78.43 ± 0.06	87.09 ± 0.15	34.78 ± 0.43	23.72 ± 0.08	65.70 ± 0.15
Softmax, SPC-2	79.76 ± 0.26	87.70 ± 0.30	35.94 ± 0.33	24.04 ± 0.29	67.57 ± 0.27
Softmax, SPC-4	79.42 ± 0.39	87.47 ± 0.20	35.80 ± 0.59	23.91 ± 0.30	67.30 ± 0.37
Softmax, SPC-8	79.53 ± 0.38	87.30 ± 0.22	35.22 ± 0.64	24.03 ± 0.22	67.03 ± 0.29
Softmax, DDM	79.23 ± 0.32	87.40 ± 0.29	35.38 ± 0.59	23.52 ± 0.21	67.02 ± 0.27
Softmax, GC	79.22 ± 0.36	87.38 ± 0.36	35.50 ± 0.59	23.55 ± 0.25	67.24 ± 0.15
Softmax, SPC-R	79.53 ± 0.18	87.71 ± 0.09	36.71 ± 0.23	24.12 ± 0.26	67.79 ± 0.39
Softmax, FRD	79.25 ± 0.41	87.49 ± 0.48	35.36 ± 0.21	23.47 ± 0.26	67.12 ± 0.18
Triplet (R), SPC-2	69.73 ± 0.47	79.74 ± 0.28	28.58 ± 0.28	19.03 ± 0.19	60.64 ± 0.38
Triplet (R), SPC-4	69.86 ± 0.49	79.91 ± 0.51	28.84 ± 0.18	19.11 ± 0.08	60.97 ± 0.16
Triplet (R), SPC-8	69.32 ± 0.23	79.43 ± 0.64	28.38 ± 0.11	19.09 ± 0.03	60.63 ± 0.17
Triplet (R), DDM	69.78 ± 0.25	79.87 ± 0.35	28.07 ± 0.41	18.78 ± 0.32	60.38 ± 0.24
Triplet (R), GC	69.34 ± 0.29	79.41 ± 0.18	28.68 ± 0.78	18.89 ± 0.30	60.87 ± 0.36
Triplet (R), SPC-R	69.01 ± 0.38	79.33 ± 0.16	27.90 ± 0.28	18.43 ± 0.33	60.43 ± 0.26
Triplet (R), FRD	69.55 ± 0.58	79.52 ± 0.54	28.70 ± 0.75	18.77 ± 0.35	60.83 ± 0.51

Table 8. CARS196: Comparison of Batch-Sampling methods for various loss functions and sampling methods.

Stanford Online-Products(Oh Song et al., 2016)					
Approach	R@1	R@2	F1	mAP	NMI
Histogram, SPC-2	69.52 ± 0.10	74.57 ± 0.09	30.49 ± 0.05	33.20 ± 0.07	88.69 ± 0.01
Histogram, SPC-4	70.13 ± 0.24	75.17 ± 0.21	31.05 ± 0.00	33.85 ± 0.13	88.82 ± 0.01
Histogram, DDM	69.36 ± 0.06	74.39 ± 0.09	30.47 ± 0.13	33.20 ± 0.03	88.69 ± 0.01
Histogram, GC	68.42 ± 0.22	73.66 ± 0.15	30.03 ± 0.25	32.49 ± 0.24	88.58 ± 0.05
Histogram, SPC-R	59.06 ± 0.19	64.72 ± 0.06	22.74 ± 0.14	25.16 ± 0.07	86.90 ± 0.03
Histogram, FRD	69.71 ± 0.19	74.84 ± 0.17	30.65 ± 0.21	33.45 ± 0.07	88.71 ± 0.05
Margin (D), SPC-2	78.28 ± 0.08	82.69 ± 0.03	38.28 ± 0.11	42.36 ± 0.11	90.34 ± 0.03
Margin (D), SPC-4	77.51 ± 0.14	81.91 ± 0.13	37.52 ± 0.34	41.41 ± 0.22	90.19 ± 0.06
Margin (D), DDM	77.90 ± 0.13	82.28 ± 0.23	37.61 ± 0.68	41.82 ± 0.26	90.22 ± 0.11
Margin (D), GC	75.77 ± 0.40	80.40 ± 0.41	35.45 ± 0.63	39.55 ± 0.43	89.72 ± 0.13
Margin (D), SPC-R	68.28 ± 0.08	73.37 ± 0.09	25.64 ± 0.23	31.31 ± 0.13	87.55 ± 0.05
Margin (D), FRD	78.18 ± 0.18	82.60 ± 0.18	38.25 ± 0.53	42.20 ± 0.31	90.34 ± 0.20
MultiSimilarity, SPC-2	77.80 ± 0.07	82.47 ± 0.06	36.37 ± 0.08	41.31 ± 0.02	89.93 ± 0.03
MultiSimilarity, SPC-4	77.90 ± 0.13	82.53 ± 0.04	36.98 ± 0.10	41.51 ± 0.05	90.06 ± 0.06
MultiSimilarity, DDM	77.85 ± 0.03	82.60 ± 0.05	36.57 ± 0.22	41.35 ± 0.12	89.96 ± 0.06
MultiSimilarity, GC	76.51 ± 0.23	81.28 ± 0.17	35.24 ± 0.28	39.92 ± 0.27	89.67 ± 0.06
MultiSimilarity, SPC-R	72.16 ± 0.38	76.12 ± 0.23	31.77 ± 0.17	35.01 ± 0.17	88.25 ± 0.05
MultiSimilarity, FRD	77.97 ± 0.17	82.60 ± 0.22	36.44 ± 0.31	41.54 ± 0.27	89.95 ± 0.03
NPair, SPC-2	75.42 ± 0.16	80.36 ± 0.14	34.60 ± 0.29	38.49 ± 0.27	89.63 ± 0.06
NPair, SPC-4	70.42 ± 0.24	75.90 ± 0.25	32.60 ± 0.37	34.81 ± 0.21	89.11 ± 0.06
NPair, DDM	74.12 ± 0.32	79.20 ± 0.29	33.39 ± 0.54	36.99 ± 0.35	89.42 ± 0.09
NPair, GC	74.37 ± 0.31	79.27 ± 0.36	33.55 ± 0.67	37.47 ± 0.49	89.39 ± 0.14
NPair, SPC-R	68.23 ± 0.04	73.45 ± 0.04	28.26 ± 0.15	31.80 ± 0.02	88.19 ± 0.02
NPair, FRD	75.50 ± 0.41	80.43 ± 0.43	35.36 ± 0.52	38.98 ± 0.30	89.77 ± 0.13
Softmax, SPC-2	78.12 ± 0.21	82.39 ± 0.16	38.12 ± 0.21	42.17 ± 0.13	90.00 ± 0.09
Softmax, SPC-4	77.81 ± 0.28	82.21 ± 0.19	38.16 ± 0.17	42.12 ± 0.14	90.08 ± 0.02
Softmax, DDM	77.39 ± 0.33	81.67 ± 0.26	37.64 ± 0.26	41.97 ± 0.20	89.66 ± 0.10
Softmax, GC	78.50 ± 0.29	82.47 ± 0.24	38.68 ± 0.27	42.37 ± 0.31	90.22 ± 0.16
Softmax, SPC-R	78.38 ± 0.19	82.46 ± 0.10	38.64 ± 0.13	42.29 ± 0.30	90.31 ± 0.16
Softmax, FRD	77.58 ± 0.22	81.93 ± 0.23	38.01 ± 0.43	42.23 ± 0.19	90.08 ± 0.06
Triplet (R), SPC-2	66.86 ± 0.36	72.06 ± 0.28	27.38 ± 0.29	30.78 ± 0.23	88.01 ± 0.09
Triplet (R), SPC-4	67.13 ± 0.45	72.29 ± 0.39	27.46 ± 0.15	30.99 ± 0.18	88.02 ± 0.04
Triplet (R), DDM	66.96 ± 0.11	72.18 ± 0.10	27.47 ± 0.02	30.79 ± 0.01	88.01 ± 0.01
Triplet (R), GC	66.61 ± 0.14	71.75 ± 0.03	27.49 ± 0.22	30.44 ± 0.04	87.99 ± 0.03
Triplet (R), SPC-R	61.12 ± 0.02	66.39 ± 0.04	23.02 ± 0.09	26.07 ± 0.15	86.95 ± 0.01
Triplet (R), FRD	67.00 ± 0.22	72.04 ± 0.15	27.32 ± 0.16	30.79 ± 0.20	87.95 ± 0.10

Table 9. SOP: Comparison of Batch-Sampling methods for various loss functions and sampling methods.